upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

Electrical and Electronics Department
Public University of Navarre

Contributions to a Novel Remote Control and Configuration Extension for Interoperable Personal Health Devices (PHD) Based on ISO/IEEE11073 Standard

By

Héctor Gilberto Barrón González

A thesis submitted for the degree of

Philosophiae Doctor (PhD)

January 2016

upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

Electrical and Electronics Department
Public University of Navarre

Thesis presented by Hector Gilberto Barron Gonzalez in Partial fulfillment of the Requirements for the degree of Philosophiae Doctor (PhD) in the Public University of Navarre.

Hector Gilberto Barron Gonzalez

The present work has been carried out under our supervision in the Electrical and Electronics Department at Public University of Navarre and we authorize its presentation for the examination committee.

Luis Serrano Arriezu   Miguel Martínez de Espronceda Cámara

ABSTRACT

The ISO/IEEE11073 (X73) interoperability family of standards was initially conceived for clinical environments at the Point-of-Care (POC). The newest branch of X73, i.e. X73 for Personal Health Devices (X73PHD), allows the development of interoperable personal health ecosystems and brings benefits to both technology producers (design cost reduction, experience sharing, and marketing facilities) and their users (plug-and-play, accessibility, ease of integration, and prices). At the moment of this writing, more than 11 specializations have been successfully published by the PHD Working Group (PHD-WG). Recently, new use cases have been exposed in the PHD-WG, which show the need for a command and control procedure to allow configuring agent's parameters and settings from the manager. The need to define and model these new management functions in a general and standard way is needed from the PHD-WG standpoint.

In this context, this work presents a proposal to extend X73PHD by including a standardized definition and procedures for general remote command and control services. With this aim, an analysis of previous studies and related cases of use is developed and its results have been studied and discussed within the PHD-WG. The use cases studied include the Intelligent Holter (HOLTIN) service, the Sleep Apnea Breathing Therapy Equipment (SABTE), and the medication monitor. In addition, a classification of the findings is proposed for each use case. These findings are the base for the proposal of the new remote configuration extension package. Previous works such as the classic Domain Information Model (X73-10201) and the POC command and control draft (X73-20301) have also been taken into account. The final solution is defined following the PHD-WG guidelines and a draft version of the new exchange protocol (X73-20601) has been shared with the PHD-WG for discussion. Furthermore, this proposal has been implemented as proof of concept in a basic setup, as standard weigh scale, and an advanced setup, in the highly configurable HOLTIN electrocardiogram recorder. To conclude, the results of the proof-of-concept implementations demonstrate that the proposal is suitable to implement command and control in both use cases and possibly these conclusions can be extended to other PHDs.

RESUMEN

La norma de interoperabilidad ISO/IEEE11073 (X73) fue inicialmente concebida para entornos clínicos en el Punto de Cuidado (POC). La rama más reciente de esta norma, orientada a dispositivos de salud personal (X73PHD), permite el desarrollo de los ecosistemas de monitorización de salud interoperables y trae beneficios tanto a los fabricantes de tecnología (reducción de costos de diseño, intercambio de experiencias y conocimiento, facilidad de comercialización) como a sus usuarios (plug-and-play, accesibilidad, facilidad de integración, precios más competitivos). En el momento de esta escritura, más de 11 especializaciones se han publicado con éxito por el Grupo de Trabajo de X73PHD (PHD-WG). Más recientemente, han aparecido nuevos casos de uso en el PHD-WG que muestran la necesidad de un procedimiento de control remoto para permitir la configuración de parámetros y ajustes del agente desde el manager. La necesidad de definir y modelar estas nuevas funciones de gestión de una manera general y normalizada es clara desde el punto de vista PHD-WG.

En este contexto, este trabajo presenta una propuesta para ampliar X73PHD con la inclusión de una definición y procedimientos de configuración y control remoto generales. Con este objetivo, se han analizado estudios previos y los casos de uso relacionados, con control remoto y los resultados se han discutido en el PHD-WG. Los casos de uso incluyen el servicio Holter Inteligente (HOLTIN), el equipo de terapia de apnea del sueño (SABTE), y el monitor de medicación. Además, se propone una clasificación de los resultados para cada caso de uso, los cuales son la base para la propuesta de ampliación de X73PHD para incluir un paquete de control remoto. Trabajos previos tales como el modelo de información de dominio (DIM) clásico (X73-10201) y el paquete de control de POC en su versión borrador (X73-20301) también se han tenido en cuenta. La solución final se ha definido siguiendo las directrices del PHD-WG y la propuesta preliminar del nuevo protocolo de intercambio (X73-20601) se ha compartido con el PHD-WG para su discusión. Por otra parte, esta propuesta se ha implementado como prueba de concepto en una báscula, como configuración básica estándar, y en el dispositivo de grabación de electrocardiograma HOLTIN, el cual requiere de una configuración más avanzada. Como conclusión, los resultados de las implementaciones de prueba de concepto demuestran que la propuesta es adecuada para implementar control remoto en ambos casos de uso y posiblemente estas conclusiones puedan extenderse a otros PHDs.

Agradecimientos

Quiero expresar mis agradecimientos a quienes con su apoyo, me ayudaron a realizar mis estudios doctorales, lo cual no dejare de agradecerles nunca.

A mis directores de tesis Doctor Miguel Martínez de Espronceda y Luis Serrano Arriezu, por aceptarme en su grupo de investigación y asegurarse que mi camino estuviera siempre en la dirección correcta. Por brindarme sus conocimientos y motivación en la realización de esta investigación. Por haberme enseñado a trabajar con seriedad, responsabilidad y rigor académico, características que han fortalecido mi formación como investigador. Por su paciencia y sobre todo por su amistad, que no dejo que mi camino se truncara en ningún momento.

Al Doctor Jesús D. Trigo, por su colaboración, tiempo y esfuerzo dedicado a las investigaciones que soportan mi tesis doctoral.

Al MC. Esteban Báez Villarreal, por el apoyo que me brindo para la exposición y publicación de las investigaciones que soportan mi línea de investigación.

Al MC. Leopoldo Villarreal Jiménez quien me proporciono todas las herramientas necesarias para el desarrollo de mi línea de investigación y a la MA. Selene Guzmán por su gestión.

Y finalmente quiero agradecer a mis seres queridos por sus sacrificios, tiempo y amor. Agradezco en especial a mi Esposa Brenda Rubí de la Cruz quien además de mantenerse a mi lado estos años, me brindó su amor, apoyo y comprensión. A mi hijo Mikel Aitor por su motivación. A mis Padres Gilberto y Amalia quienes además de educarme para lograr mis metas me proporcionaron siempre su amor, su apoyo moral y económico. A mis hermanos Libia, Mariana y Ricardo y a mi compadre Sarmach Ulianov por su apoyo.

Table of Contents

List of Figures

List of Tables

Listings

Chapter 1 Introduction

This Chapter gives the background to the readers of this PhD dissertation. General concepts and definitions, such as agent, manager, personal health device, eHealth and mHealth, are revisited. Following this description of the background concepts, the scope and approach of the Thesis and the research context are described. Finally, the underlying hypotheses are outlined and the general and specific objectives are defined.

1.1 Preface

In the last decades, technological advances at the Point of Care (POC) have fostered the development of innovative medical devices. However, these advances have caused an initial heterogeneity, hindering their integration and interoperability with devices from different manufacturers [1-5]. To solve this inconvenience, the development of standards that ensure interoperability between medical devices and monitoring systems was a crucial issue and, thus, the ISO/IEEE11073 family of standards (also referred to as X73) was proposed [6]. This standard was initially developed and defined to enable the interoperable exchange of vital signs between medical devices (agents) and central equipment (managers) at the Point-of-Care (POC) [7-9].

As technology evolved, the integration of low-voltage low-power electronic components and high performance wireless technologies in medical devices, along with the growing number of people in adulthood who requests healthcare at home [10-13], prompted new use cases that encouraged the migration of health care from the POC to home environments [14-20]. However, such medical devices had been developed specifically for the POC, where there were limitations in electric power consumption and the communications were based on the wired interconnection of agent and manager devices. Therefore it was necessary to adapt the X73 family of standards to home health environments [21-28].

Analogously, different manufacturers had also developed proprietary solutions, which hampered an interoperable environment. This fact limits the range of devices on the market and, at the same time, it causes an increase in the cost of implementing a system for home telemonitoring. However, these were not the only causes that hindered interoperability. Factors such as the medical regulations and health policies can also be seen as a non-technical burden discourage the adoption of standards across all stakeholders of the Personal Health Device (PHD) industry [29].

A potential solution to this problem has been developed within the Personal Health Device Working Group (PHD-WG) [30], whose efforts have brought many improvements to the ISO/IEEE11073 family of standards. These efforts led to a new branch of the standard, namely, X73 for Personal Health Devices (also referred to as X73PHD) [30-32]. This new branch is mainly specified in the Optimized Exchange Protocol (11073-20601) and the related device specializations (11073-104xx). Simultaneously, Continua Health Alliance, a private organization composed of the public and private sector, worked along with the PHD-WG to provide a certification process to verify conformance of devices that implement any standard of the X73 family [33]. This collaborative research work on standardization into the PHD-WG including private companies, universities and research groups, provides benefits that may foster the deployment of interoperability of medical devices [34-35]. Besides, the PHD-WG is constantly developing new use cases (i.e. specializations) that enhance the integration of X73PHD in mobile Health (mHealth) scenarios [36]. These scenarios cover areas such as health telemonitoring and fitness, independent living, wellness and disease management [37-40].

These new scenarios demand wide-ranging configuration options in order to optimize the solutions and services patient-centered. Among these requirements, a key and challenging feature would be a service for remote control which would allow the modification of PHD parameters as well as the execution of commands for controlling the medical devices [41]. To develop such a service, it is really necessary to analyse the new requirements into each current use cases. These set of requirements should also work for those use case proposed in next future. Moreover, it is also necessary to develop configuration and remote control mechanisms in order to meet these use case requirements.

It is worth mentioning that old POC devices were designed without power consumption restrictions. Such devices are usually connected to the power line and therefore they could run heavy processing routines. Conversely PHDs are typically battery powered as it is mandatory to analyze the implementation of the remote control process on these devices to ensure that it does not demand an unacceptable extra processing load which would make an appreciable reduction of its autonomy.

Due to aforementioned features the command and control extension implementation must have a processing load as the same order of magnitude than current implementations of the X73PHD stack, which do not implement remote control. Obviously this implementation should not require extra hardware in order to make it attractive to manufacturers.

The deployment of remote control service could play a key role in numerous scenarios such as home health telemonitoring where the remote control and configuration could improve the performance and usefulness of medical devices adjusting their parameters based on the dynamics of the patient’s condition, such as chronic disease management, such as sports and wellness where PHDs could monitor the performance of athletes or even into research scenarios, where the PHDs can be used to collect data for further big data analysis.

1.2 Hypotheses

To date, the evolution of the X73 family of standards has included the so-called control package (X73-20301), which provides expertise for the remote control of medical devices at the POC. The main goal of this service is to provide the manager with the ability to remotely modify the attributes of the agent. By using this feature, the operation of the device could be accordingly changed into the desired form. However, X73-PHD does not consider any service of this type. As a result, PHDs cannot be configured from a manager in a remote location, and it is not possible to make any changes in its operating parameters.

For this reason, there is a need to update the X73-PHD standard with significant improvements in order to include a similar mechanism, such as the control package in the X73-POC. Moreover, this update must meet, at least, the following features:

1. To consume a minimal amount of hardware resources, such as memory and processing time.

2. To include software tools similar to real time operating systems that facilitates the management tasks used in data processing and communication.

3. To be a general remote control toolbox as new specializations are being developed at PHD-WG.

1.3 Objectives

The main objective of this research work is to propose and to implement a command and control extension for personal health devices with limited power consumption. This extension will be defined according to the rules set by X73PHD. This main objective will be based on the following sub-goals:

1. To review the state of art of X73 and current remote control proposals.

2. To study the existing and new use cases in the scope X73PHD to obtain the main features and requirements to be satisfied by a command and control extension proposal.

3. To develop a command and control proposal within the X73PHD so that all X73-104xx profiles could benefit from it.

4. To implement the command and control proposal into several X73PHD use cases as proof of concept.

1.4 Structure of this thesis

This thesis is organized as follows. Chapter 2 presents the state of art of interoperability in X73POC and X73PHD; and current remote control. Chapter 3 studies and analyses X73PHD use cases to obtain features and requirements needed to make a command and control proposal. Chapter 4 proposes a command and control extension to X73PHD. Chapter 5 implements our command and control proposal into two agents, i.e. the weighing scale and the electrocardiogram (ECG) event recorder, to demonstrate the feasibility of this research work. Finally, conclusions are drawn in Chapter 6.

Chapter 2 State of the art

This chapter presents the state of the art within the scope of the ISO/IEEE11073 family of standards. First, a review of the standards developed to the Point of Care is presented, laying the groundwork for the development of ISO/IEEE 11073 – Point-of-Care (X73POC). Second, a description of the evolution of the ISO/IEEE11073 standard to the new branch known as X73PHD is exposed. The parts of X73-20601 and its application in mHealth scenarios are described. Third, the latest research in the field of interoperability around the X73PHD and command and control is presented. Forth, given that security is a very important aspect in eHealth environment, its development status in the context of X73PHD is studied. Finally, the state of art around theoretical validation of X73PHD is reviewed.

2.1 ISO/IEEE11073 in the Point-of-Care (X73POC)

In order to allow plug-and-play intercommunication between the agent and the manager devices usually found in Intensive Unit Care (ICU) environments, the Institute of Electrical and Electronic Engineers (IEEE) Standards Association (IEEE-SA) defined a full-stack standard following the 7-layer Open Standard Interconnection (OSI) model. This standard, also known as X73 for the POC (X73POC), was based on other previous standards such as, firstly, the American Medical Information Bus (MIB), designed to provide automatic transmission of data from medical equipment to clinical data management computers; secondly the European Vital (ENV13734) standard, designed for the representation of vital signs; and, finally, the INTERMED (ENV13735) standard for medical device interoperability in the POC [43-45]. Later, several authors worked on to enable interoperability between medical devices at the POC and electronic health records [46-55]. In [46], tools such as XSchema and ICSGenerator provided the parsing capabilities necessary to assure that manufacturers are defining the information required by the X73. In [47], a two-way converter between HL7-aECG and SCP-ECG data formats using BioSig was developed. In [48], the MediCAN technology was used to provide an interoperable medical instrument interface between medical devices in a medical bus. In [49], a web application server that integrated an anesthesia workplace, an endoscopic surgery workplace, and a hospital information system into a service-oriented architecture (SOA) based network was developed. The web services implemented on it used an XML vocabulary taken from X73-10201 which was the base for the representation of the medical device and vital signs, reaching a level semantic interoperability. In [50], an interoperable medical instrument interface for networking using MediCAN technology suite as an open standard was developed. In [51], a healthcare platform based X73 with Power Line Communication (PLC) was made. This proposal tried to solve the problems caused by shaded areas on wireless communication and interferences between medical devices in the hospital. In [52], a comparison between the data model of X73 and MediCAN was presented to encourage interoperability demonstrations of medical devices at POC. In [53], the ISO11073/IEEE1073 POC Medical Device Communication (X73-POC-MDC) standard was selected to provide interoperability for ICU from an analysis made to different standards. In [54], a prototype of a wearable monitoring system based on interoperability standards was developed. This was used to demonstrate the plug-and-play wireless connectivity to system components. In [55], an integration profile based on CLSI POCT1-A, CEN

ISO/IEEE 11073 “VITAL” was used to provide a seamlessly information exchange for POC systems, vital function monitors and information systems.

2.2 Evolution of ISO/IEEE11073 to Personal Health Devices (X73PHD)

eHealth systems are the result of the integration of information and communication technologies into traditional health services [56-58]. These services are based on wireless technologies and wearable devices (although wired technologies such as USB are also needed). Quite frequently, eHealth services are focused on the prognostic, prevention and monitoring of diseases for specific patients by using biomarkers and their continuous recording in electronic medical records for later analysis. The market opportunities of eHealth are diverse. The predominant vision, as established by Continua Health Alliance, determines that the predominant applicable scenarios are disease management, health and fitness, and aging independently [64-68].

However, when it comes to eHealth service deployment, in order to achieve an expected level of operational cost-effectivity, it is necessary to overcome different challenges, such as interoperability and plug-and-play features that could ease device configuration and reduce management costs [59-63]. In this context, global organizations such as Continua Health Alliance and Integrating Healthcare Enterprise (IHE) define their integration profiles which will improve interoperability of medical devices to seamlessly integrate their data into data processing centers and electronic health records. In particular, Continua Health Alliance provides design guidelines and a product certification program.

There are several factors that propitiated the evolution of the classic X73 standard, i.e. X73POC, to the new version for personal health devices, i.e. X73PHD. These are, on one hand, the appearing of new wireless communications technologies such as Bluetooth, Ultra Wide Bans (UWB), Wi-Fi, and Global System Mobile (GSM) which removed the need of wires and improved patient’s comfort level; on the other hand, the use of low-voltage low-power hardware needed to implement low-cost wearable devices which reduced battery size and improved ergonomics. Therefore, this new branch extended the limits of healthcare applications from the POC to services based on mHealth and home telemonitoring focused on among others patient’s comfort. The new branch was developed by the PHD-WG and expected to be used in new wired and wireless scenarios such as Personal Area Network (PAN) and Body Area Network. The initial version of the X73PHD considered several transport technologies such as the following:

- **Universal Serial Bus (USB):** This industry adopted standard bus defines the cables, connectors and protocols used in a bus for connection, communication and power supply between computers, peripherals and electronic devices. Its development came from a group of companies in the sector that sought to unify the serial interface. In April 2007, USB published a first X73PHD compatible profile called the Personal Healthcare Device Class (PHDC). The USB PHDC profile provided a description of the full architecture that the personal health device and the host must support to interoperate. The PHDC is made up of data structures called descriptors that contain information about the device class and the commands used to exchange medical data [69-73].

- **ZigBee:** It is focused on low-power low-rate wireless communications and implements secure communications for lower data transfer rates, with a reduced power consumption profile [74]. The ZigBee Alliance, the organization that maintains the ZigBee standard, joined efforts with Continua Health Alliance to promote its use [75]. Several implementations have implemented proof of concepts aimed at aspects such as security, power consumption, wireless sensor networks, and compatibility with other standards such as HL7 [76-80]. A PAN based on the ZigBee HealthCare profile specification is shown in Figure 2.1, where two personal heath devices send their measurements using this transport technology.

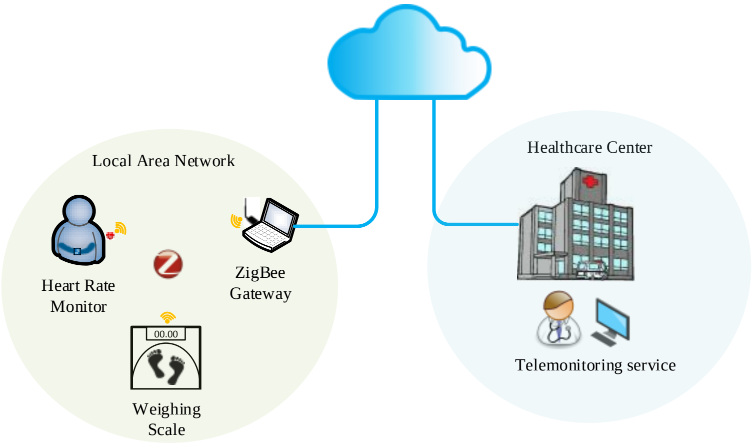


Figure 2.1: ZigBee HealthCare application.

- **Bluetooth:** It is a wireless technology standard for the interchange of data over a short distance, and it is based on radio waves in the ISM band from the 2.4 to 2.485 Ghz. It is commonly used for fixed or mobile devices using PANs and Body Area Networks (BANs) [81]. Bluetooth defines several application specific profiles. The Health Device Profile (HDP) is defined to cover personal health applications. It uses the Multi-Channel Adaptation Protocol (MCAP) as the underlying channel control protocol [82]. HDP defines roles, source and sink, which implement the agent and manager, respectively (Figure 2.2). Many developers have implemented HDP using smartphones and computers as gateways and using Wi-Fi and GPRS [83-85].

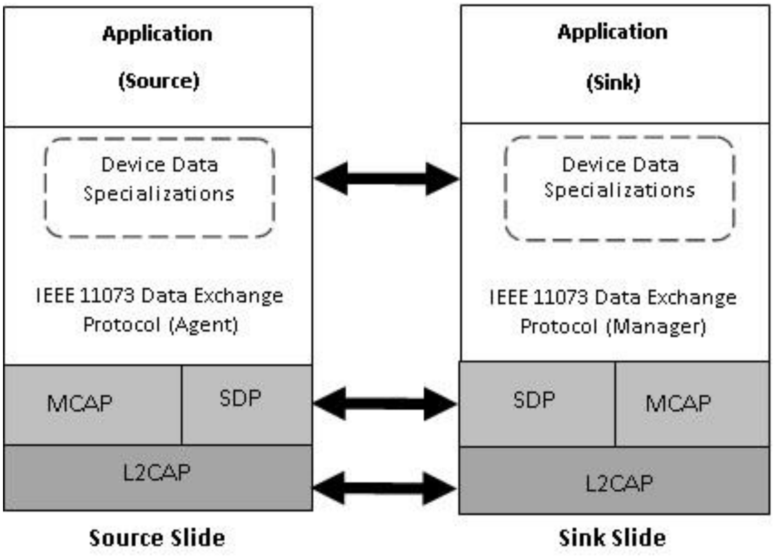

Application
(Source)
Device Data Specializations
IEEE 11073 Data Exchange Protocol (Agent)
MCAP
SDP
L2CAP
Source Slide
Application
(Sink)
Device Data Specializations
IEEE 11073 Data Exchange Protocol (Manager)
SDP
MCAP
L2CAP
Sink Slide

Figure 2.2: Bluetooth HDP.

In order to deal with wearable device requirements, the new branch is a simplified version of the X73POC standard. It maintains X73POC basic features and its modeling concepts but simplifies the layer stack and the services provided. The X73PHD framework is defined by the X73-20601 application profile, a self-contained document which is also known as the Optimized Exchange Protocol. To improve interoperability, the X73PHD defines also specializations of the X73-20601 document. These specializations further restrict the options allowed for a specific agent type. For example, in a thermometer the specialization determines that the types of measurements are restricted to patient’s temperature in several positions of the body. At the moment of this writing, more than 16 specializations have already been published and 3 other are in the draft state [86-104]. The different documents developed by the PHD-WG that have been published up to date which integrate the family of X73PHD are shown in Table 2.1. The first column shows the status. The second column shows the identifier or number of the document. The name of the document is in the third column, and the last column shows the reference number.

Table 2.1: X73PHD Protocol stack.

Status	Document Number	Document Name	Reference
Active	11073-10404-2008	Pulse Oximeter	[86]
Active	11073-10406-2011	Basic electrocardiograph (ECG) (1 to 3 lead ECG)	[87]
Active	11073-10407-2008	Blood pressure monitor	[88]
Active	11073-10408-2008	Thermometer	[89]
Active	11073-10415-2008	Weighing scale	[90]
Active	11073-10417-2015	Glucose meter	[91]
Active	11073-10418-2011	International Normalized Ratio (INR) Monitor	[92]
Active	11073-10419-2015	Insulin pump	[93]
Active	11073-10420-2010	Body composition analyser	[94]
Active	11073-10421-2010	Peak expiratory flow monitor (peak flow)	[95]
Active	11073-10424-2014	Sleep Apnea Breathing Therapy Equipment (SABTE)	[96]
Active	11073-10425-2014	Continuous Glucose Monitor (CGM)	[97]
Active	11073-10441-2013	Cardiovascular fitness and activity monitor	[98]
Active	11073-10442-2008	Strength fitness equipment	[99]
Active	11073-10471-2008	Independent living activity hub	[100]
Active	11073-10472-2010	Medication monitor	[101]
Draft	11073-10423	Sleep monitor	[102]
Draft	11073-10426	Home Healthcare Environment Ventilator	[103]
Draft	11073-10427	Monitoring Power Status of Critical Care Devices Equipment	[104]
Active	11073-20601	Optimized Exchange Protocol*	[105-106]

* Up to date, different versions of these standards have been published, from the first version in 2008 until the current version 2014.

2.2.1 X73PHD momentum within mobile health applications

The context of development of X73PHD is within the new paradigm of eHealth and personal health online services, wireless technologies, and personal managers based on smartphone platforms. Agents (such as weigh scales blood pressure and glucose monitors, and thermometers) are connected to the manager (typically a smartphone). Then, the manager, which is taking Third and Fourth Generation (3/4G) mobile networks to be online, is used to monitor patient’s status. This setup has many applications. It is common that manager applications analyse on real time the welfare of a person and generate data that could prevent health risks through a modification of lifestyle [107-109]. The cardiovascular scenario is an area that is becoming popular in mHealth appliances, mainly due to the difficulty to detect the diseases in hospitals and controlled environments, being necessary to monitoring the patient in its usual lifestyle. Several applications benefit from the advantage that smartphones may incorporate many peripherals and accessories [110]. Modern software applications for smartphones (apps) try to support decision taking about the condition of a patient by using contextual information. The support is given not only to the specialist but also directly to the patient [111,112]. There are also several online services that allow health professionals accessing to patient’s medical information through their mobile devices [113,114,115]. Given that these managers are battery-powered, energy efficiency is a key aspect [116-118]. Therefore, it is need to optimize the software implemented in those systems. Developing low-cost mHealth implementations is a difficult goal to meet for developers. An algorithm to analyse and synthesise APDU using the Patterns-based Methodology has been implemented for cost-effective development of standardized PHDs [119].

2.2.2 X73PHD technical framework

X73PHD defines interoperability between medical devices or and concentrator devices or, in terms of X73PHD, agents and managers, respectively. It is a transport agnostic protocol that implements the application layer protocol according to the 7-layer OSI model. This means that this protocol is independent of the transport technology used and the source code is written once and used everywhere (i.e. Bluetooth, ZigBee, etc. can use the same source code related to X73PHD). Some of the agents already defined in X73PHD are shown on the right part in Figure 2.3 (e.g., weighing scales, a temperature sensor, blood pressure monitor). These agents gather physiological information in several health scenarios. Then, the information is concentrated by the manager (e.g. the smartphone, personal computer or health appliance). An agent typically communicates with a single manager at the same time, but the manager can receive connections from multiple agents. To do that the manager uses a separate point-to-point connection and a per-agent session.

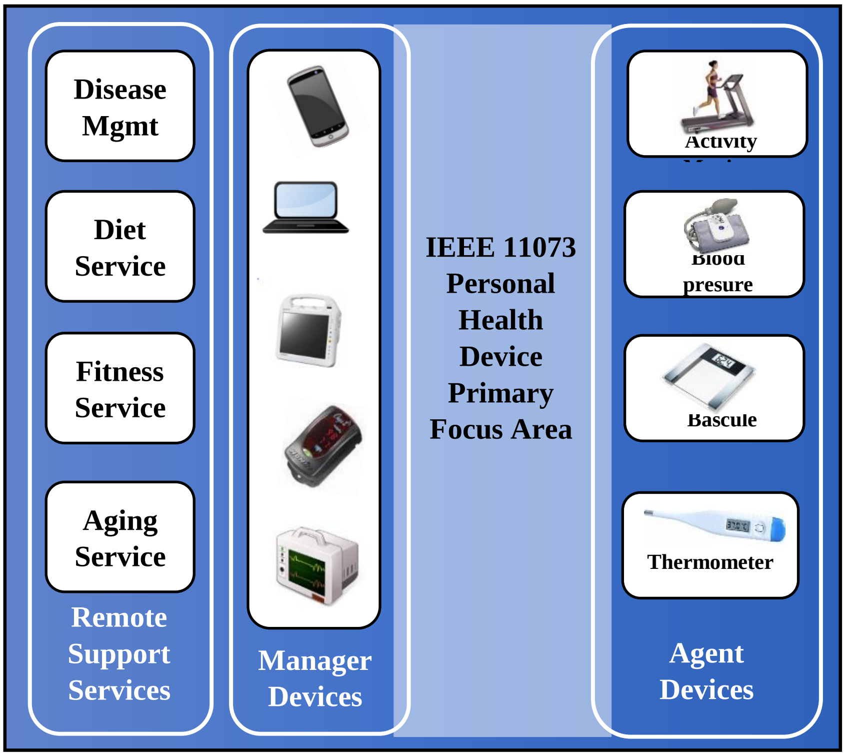


Figure 2.3: Primary focus area in the PHD-WG.

The full stacks of the agent and the manager are shown in Figure 2.4. The transport layer is at the bottom of the stack. As said before, it is outside of the scope of this standard. The Optimized Exchange Protocol is in the middle. It provides the generalized toolbox to support any agent profile. Nevertheless to specify the features to be implemented in an agent in detail, the Optimized Exchange Protocol needs to be further specialized. Therefore, agent specializations, which are at the top of this stack, provide the needed additional details. It is noteworthy that there are later versions of this standard, such as X73-20601-2010 and X73-20601-2014 which include additional procedures, data types, and corrections.

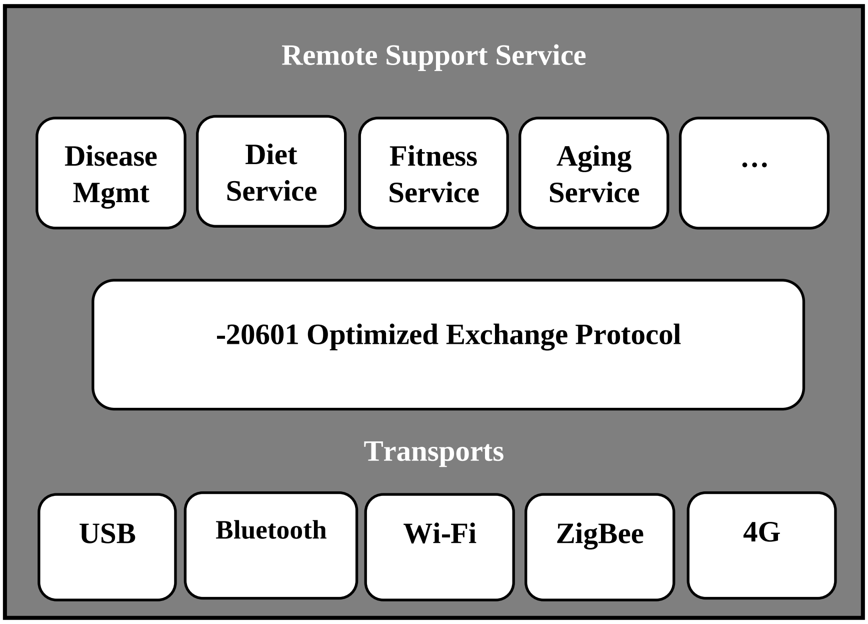


Figure 2.4: Scope of X73PHD.

The Optimized Exchange protocol document is internally divided into three parts:

A. Domain Information Model (DIM)

The personal health devices are defined by an object-oriented domain model which structures the data within the agent. This model is defined using objects which have attributes that represent information and status on the agent. The manager can poll these objects to obtain information periodically (manager initiated) and the agent can use them as a mechanism to notify status changes to the agent (agent initiated).

The DIM of a standard agent is shown in Figure 2.5. It has been modeled using Unified Modeling Language (UML) to represent the object classes of a personal health device as defined in X73PHD and the relationships among them. The main classes are the Medical Device System (MDS), scanner, and metrics such as Numeric, Real Time Sample Array (RT-SA), and Enumeration and Persistent Metrics (PM).

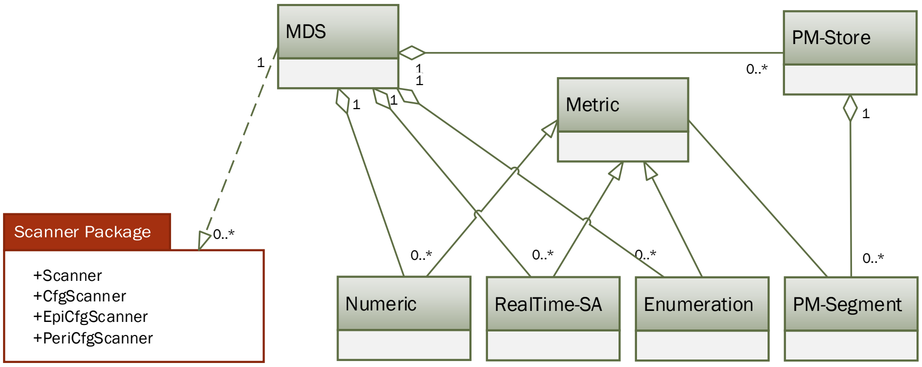


Figure 2.5: PHD Domain Information Model.

B. Service model

The service model integrates all object access services used to interchange of messages between the manager and the agent objects defined in the DIM. The available services in the current version of X73-20601 are:

- **Association Service:** Association request/response, release request/response and abort are defined, and these are used at the start or to end the association process between the agent and manager.

- **Event Report Services:** used to generate configuration and update reports.

- **Object Access Services:** the Services Get, Set and Actions (methods) are defined at the moment and are used to get the value the object attributes and the Set services is only implemented by the Scanner object to change the operational State of this. The Actions are present in the MDS Object and PM-Store objects for provide services to them.

C. Communication model

It provides the FSM of agents and managers, transport characteristics, and the over-the-wire allowed format to use in the point to point X73PHD connection. The communication process is divided into communication characteristics (point to point connections, manage for single or multiple connections, processing of APDUs, and selection of communication between reliable or best effort), the connection state machine, and the permitted interactions in each state.

2.3 Latest research in command and control over X73

There are several precedents related to remote control within X73POC and X73PHD. First, there was a standard development project for a command and control profile for the baseline profile (X73-20202) in the classic X73 (X73POC) which is in draft state and discontinued and is not compatible with X73PHD (Section 2.3.1). Second, there was a not published work, shared within the PHD, to do command and control in X73PHD using the SET service (Section 2.3.2). This approach did not get consensus in the PHD-WG as it presented several drawbacks. There were other approaches that may be considered to be related to remote control, in general, and are therefore mentioned in this Thesis (Section 2.3.1).

2.3.1 Remote control profile draft in X73POC (X73-10201 and X73-20301)

The X73-POC included command and control package in its DIM in X73-10201 [8]. It was supposed to be further specified in a new profile for remote control better known as X73-20301 [9]. However, this standardization work which was approved by IEEE (Project Approval Request) was neither continued nor, therefore, completed. At the moment of this writing it continues in draft state. Nevertheless the material developed is worthy as it gathers much knowledge and experience on command and control in POC.

The remote control package included in X73-10201 defines the Operating scanner, the Service and Control Object (SCO), and the Operation objects. The SCO is a proxy used by the manager to execute Remote Procedure Calls (RPCs) to modify the attributes of the instances of the Operation objects implemented by the agent. The information model of the control package contains a set of specialized Operation objects (Figure 2.6).

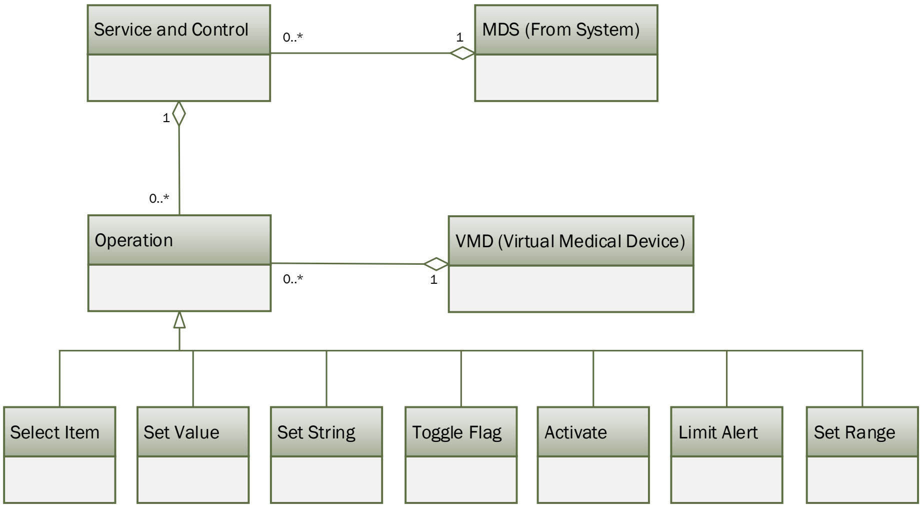


Figure 2.6: Control Package. Source: X73-10201 POC, rev 2004, Objects in the Control Package.

2.3.2 The SET service approach

The SET service allows modification of data attributes within objects in the agent. This service has two modes of operation which are the confirmed mode and unconfirmed mode. In the confirmed mode, the SET service requests to change an attribute. Then the agent modifies the attribute and confirms the change to the manager with a response message. The request/response message contain mainly the object identifier to modify attribute identifier, a modify operator, and the attribute values (see Figure 2.7). Gregor et al develop an implemented an X73-20601 compliant monitoring and control communication interface based on X73-20601 in which the SET Service is used to set attributes derived from numeric class [120].

This way to implement command and control functions had some drawback. Because of them, the approach was definitively rejected by the PHD-WG. The drawbacks are the following:

- There was no mean for the agent to indicate which attributes are accessible via the SET service.

- There is no mean for the agent to indicate additional information about the specification of the attributes themselves such as type of the attribute, possible or valid values, meaning, and contextual help text.

- Response errors are vague. The agent error response codes are limited to the ones defined in the error and reject messages (RORJ and ROER).

- The SET service is limited to attribute values. It is hard to specify a procedure such as start measurement, and reset memory to defaults.

There is no lock to implement transactional configurations (although there is some kind of operation atomicity when the SET contains several attributes).

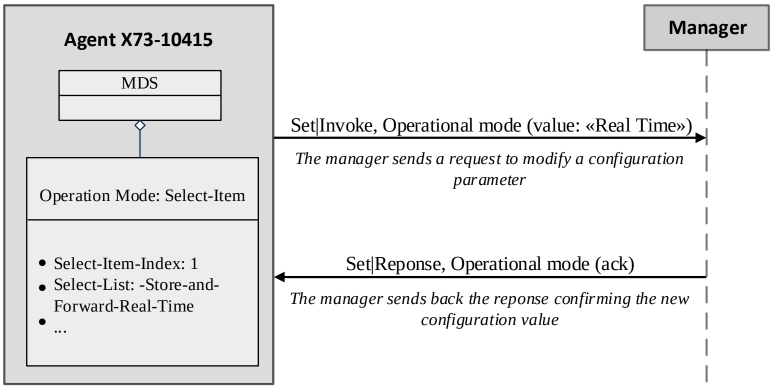

Agent X73-10415
MDS
Operation Mode: Select-Item
Select-Item-Index: 1
Select-List: -Store-and-Forward-Real-Time
...
Manager
Set|Invoke, Operational mode (value: «Real Time»)
The manager sends a request to modify a configuration parameter
Set|Reponse, Operational mode (ack)
The manager sends back the reponse confirming the new configuration value

Figure 2.7: SET service.

2.3.3 Other approaches

Besides the PHD-WG, some authors have developed several proposals to implement command and control in PHDs. Lasierra et al proposed the integration of Simple Network Management Protocol Version 3 (SNMPv3) within X73PHD [42]. Their setup addressed to manage technical information in home-based telemonitoring scenarios. The implementation of remote control in this proposal is limited to alert setup and modification and the selection of preconfigured events in response to specific alerts. Goga et al (2008) developed an analysis of X73PHD Finite State Machine (FSM). In order to develop this analysis, model-checking techniques such as PROMELA language and the Spin tool were used [121]. Martinez et al (2010) implemented an end-to-end standards-based solution integrating the X73PHD and ISO/EN13606 [23]. Their implementation consisted on a platform that used smartphones to adapt obtained vital signs from non-wireless devices into X73PHD-compliant medical devices. Then data gathered by smartphones was released to an ISO/EN 13606-compliant EHR server. They presented the concept of a configuration profile that allowed controlling the smartphone remotely. Lee et al (2015), proposed an X73PHD system that enables legacy healthcare devices to transmit vital sign data to an application-hosting device on a network [122,123]. Piniewski et al, in collaboration with Continua Health Alliance, developed interoperability guidelines for the emerging personal telehealth ecosystem [124,125]. Baig et al (2013) developed a review of smart health monitoring systems and an overview of their design and modeling [126].

2.4 Security and Privacy

Security is an important issue in data transmission. It must be seriously considered in the transmission of health data and vital signs. Nowadays, X73PHD framework does not consider any security mechanism. However, some issues are being discussed such as user identification, authentication and authorization. Although these characteristics may be used for selecting and preserving the patient information, a mechanism to ensure that the information sent will reach its destination without being altered or seen by undue people is necessary. Due to the aforementioned problematic, the PHD-WG has focused its efforts to add a mechanism for security and privacy within the X73PHD scope. As a result, two scenarios have been raised within of the PHD-WG. The first scenario focuses on the security between agent and manager. The solution is usually to use the features implemented in transportation technologies currently used. For example, authorization, authentication, and encryption may be guaranteed in Bluetooth using pairing and security services in L2CAP. The second scenario is focused on end-to-end communications. Several solutions propose to add a specific security mechanism within the X73PHD implemented for example as objects within the DIM. However, the solution remains open. Caranguian et al addressed their work to develop a method for the authentication of the telemedical appliance within of X73-PHD [127]. This method is based on encryption tools which include symmetric algorithms like RSA (Rivest, Shamir & Adleman). Egner et al (2012) focused their work on the management of secure authentication for mobile medical networks. In this work, the authentication occurs after the association and they propose the use of keys obtained from biometric measurements (fingerprint scans) using a novel algorithm [128].

Kliem et al. (2012) aimed at providing a security and communication architecture for eHealth scenarios. The security mechanism is based on a key exchange pattern, which allows agent authentication with the use of aggregators (smartphones, router or gateways) and the backend [129]. Kim et al. (2014) proposed data encryption through the use of block cipher algorithms (e.g., AES, Blowfish, etc.). To encrypt data, the agent concatenates the System-Id, the Date-and-Time, and the Dev-Config-Id MDS attributes, and encrypts them using a block cipher algorithm such as AES or Blowfish through a common secret key which the PHD and the manager allocate in advance. [130].

However, neither of methods aforementioned fully evaluates the security risks and are not considered to develop a security extension within X73PHD framework. Besides the complexity of this issue it raises serious ethical and legal considerations that must be taken into account. Daidi et al have made a study to analyze the effect of legal barriers in the standardization and interoperability of PHDs [29]. At this point a secure end-to-end configuration is technically possible, but some use cases such as setting the pacemaker, insulin infusion pumps and drug dispensers may discourage the use of the remote control due to the high-security risk that represents an unauthorized intervention in the data transmission.

2.5 Validation and verification

Validation of X73PHD-compatible agents is required to certify the correct implementation of X73-104zz profiles. In order to carry out this task, the Continua Health Alliance provided tools for so that their members can test their devices. Tools such as the Protocol Analyzer can be used to test the agent and manager of X73PHD and validate properly the implementation of their profile. In addition, there are open source tools for the validation of the protocol implementation. There is also a need to formally verify X73PHD. The Spin model checker is an open-source software verification tool used for the formal verification of software applications [131]. Some authors have used model checkers to validate their appliances. Goga et al. (2009) developed a model checker with Promela language to check the correctness of the FSM. With this tool they verified that the modeled system produced the desired behavior. The results of this research exposed that there were serious omissions in the standard design, but also they validated that most performance of the standard behaved correctly [132]. Another tool is the mCRL2. This is a formal specification language used for modeling, validation and verification of concurrent systems and protocols [133]. Keiren et al (2012) modeled the X73-20601 FSM for the agent and the manager and tested the session setup. Some errors in the process of association of the FSM were found which introduced deadlocks in the system. Later these were corrected by the PHD-WG in the 2010 version of the standard [134].

Chapter 3 Study of Requirements

In this chapter, the use cases that support this research are discussed. First, a methodology for the analysis of the use cases is proposed. Later, the operation of each use case and their requirements is described. The results obtained are presented and related requirements of each use case are classified in the section results, where also are shown the data types that could support these parameters. Finally, in the section discussion, some important considerations to be taken into account in the development of an extension of remote control and configuration are addressed.

3.1 Introduction

Use cases derived from X73-104zz specializations and others in draft state have been exposed within the working sessions of the PHD-WG. Those use cases show a need to control different types of settings. For example, medical devices may require specific personalizing parameters in order to obtain a correct visualization of the data sampled by the agent and other such as text strings that help the user with the operation of the device.

As a solution to these requirements, some ad-hoc methods have been developed in the X73-POC, but do not meet the requirements needed to implement a general solution that can be standardized. Therefore, the PHD-WG is aiming to standardize a general procedure that allows the addition of a remote control and configuration in any specialization (an extension package within the X73-20601 toolbox).

To develop and define this procedure, the first step is to gather and to analyze new use cases within the PHD-WG, including the use cases generated in existing specializations and also those derived from specializations in draft state. The second step is to develop a methodology to adapt the remote procedure in order to obtain the correct scope in X73-PHD.

3.2 Methodology

The PHD-WG within IEEE-SA works in the development of X73PHD. The PHD-WG members represent organizations and companies around the world. This group is mainly based on a mailing list. The group holds online meetings once a week and face-to-face sessions 2 or 4 times per year. At the moment of this Thesis, the PHD-WG has developed the optimized exchange protocol (2008, 2010, and 2014 versions) and more than 16 specializations. Currently it is working on the development of new specializations and improving existing ones.

Within PHD-WG’s constant work, a methodology has been followed to identify and analyze the use cases that claim for the control and configuration feature. The methodology includes several activities which are carried out within PHD-WG:

1. First, a call for proposals is issued within the PHD-WG.

2. Second, the members propose their use cases.

3 Third, the new use cases are deeply discussed within the PHD-WG.

4 Forth, the results obtained are then analysed to determine the requirements and parameters necessary of each use case.

5 Finally, taking these requirements and parameters into account, a classification of parameters is obtained, in order to lay the foundations for the development of the proposal for remote control and configuration.

3.3 HOLTIN

The first use case, HOLTIN, is an advanced version of the X73-10406 basic electrocardiograph (ECG) (1 to 3 lead led ECG) profile. The advanced features of this device are detailed below.

A. Automatic detection algorithm

In this use case, an automatic detection algorithm was developed and implemented to detect cardiovascular events. Its operation is based on the comparison of real time data acquisition, within the limits and ranges established in the detection algorithm.

A list of parameters for this algorithm is set out in the first column in Table 3.1. The second column shows a description of these parameters. These parameters reference are necessary for the implementation of the automatic detection algorithm in the new use cases and a remote control service is used for changes in those parameters.

Table 3.1: Automatic detection algorithm parameters.

Parameter Reference	Description
Asyst-Lim	Asystolia threshold
Brady-Lim	Bradycardia threshold
Taqui-Lim	Tachycardia threshold
QRS-Number	RR Intervals threshold

The use of each of these parameters within the automatic detection algorithm is the following:

- **Taqui-Lim and Qrs-Number:** This parameter stores the tachycardia limit in beats per minute and the number of consecutive QRS complexes with a heart rate greater than Taqui-Lim respectively. To start the recording of an event type, an automatic tachycardia heart rate must exceed the value Taqui-Lim beats which is a number greater than or equal to the value of a Qrs-Number.

- **Asyst-Lim:** This parameter determines the time interval (in seconds) without QRS complex, and once exceeded, it determines the existence of an arrhythmic episode of asystole. The typical value of this parameter is 2-4 seconds.

B. Storage parameters for record time of the ECG signal

The storage parameters set the duration of the recording time of the ECG signal, which are stored on the device as part of the information associated with cardiac events detected/reported. The first column of the Table 3.2 shows whether there exist parameters that can be modified to increase or decrease the recording time of the ECG signal for both automatic and manual detection in the X73-10406 specialization. In the second column, the parameters associated with current specializations are exposed. A description of each storage parameter is exposed in the third column.

Table 3.2: Storage parameters for recording time.

Parameter Reference	Supported by current specs	Description
ECG-Recorded-Span	Read only	Recording time events detected automatically
Automatic-Detection-Time	No	Recording time events detected automatically
Manual-Detection-Time	No	Recording time of events reported manually

Below the use of Storage parameters for recording time is described.

- **ECG-Recorded-Span:** The duration of the recording time interval for automatically detected events is set by this parameter and is equal to 120 seconds.

- **Automatic-Detection-Time:** This parameter is used to set the time interval (in seconds) of the ECG signal, around the detection instant, at which information associated with arrhythmic episode is stored.

The value of this record time is within the range of 30 seconds to 120 seconds.

- **Manual-Detection-Time:** This parameter is used to set the time interval for manual detection (in seconds) of the ECG signal, around the time of notification. The time interval value of this parameter is greater than the time required for storing the automatic events, and it may be within the range of 300 seconds to 600 seconds, or even higher.

C. Detection parameters

The HOLTIN platform has several operating parameters to optimize the sensitivity and specificity of the detection of the heart rate. Its operation mode is based on the morphological characteristics of the patient's ECG signal, which is analysed with a detection algorithm implemented in the agent. The configurable parameters necessary for the operation of the algorithm are shown in Table 3.3.

Table 3.3: Detection parameters of Heart Rate.

Parameter Reference	Description
Blanking-Window-Sensibility	Blanking window sensitivity
Blanking-Window-Time	Blanking window span

The following paragraph describes the use of parameters associated with the automatic detection algorithm.

- **Blanking-Window-Sensibility:** It set the sensitivity of the QRS complex detection algorithm and prevent sub-detecting them.

- **Blanking-Window-Time (in milliseconds):** It represents the blanking time for disqualification of the algorithm once the QRS complex in the ECG signal is detected.

Within the blanking window defined by these parameters, the algorithm suspends the process of searching for new QRS complexes. The blanking window is established after a valid QRS complex is detected. Within this window, a new QRS is considered valid only if its peak level is over a certain level of sensitivity determined by Blanking-Window-Sensibility parameter and the setting of this time interval avoids situations on - as produced when detecting the T wave amplitude and it has a morphology similar to the R-wave, that is, to increase the specificity of detection of the algorithm heartbeat.

D. Operational mode

The real time mode is the current operation mode of the X73-10406 specialization and it is limited to continuous data transmission from the agent to manager. HOLTIN needs two additional operating modes to the X73-10406 specialization which are shown in Table 3.4.

Table 3.4: New operational modes proposed.

Operational Mode	Span	Connected to manager	Real Time	Supported by current specs
Real time	Whole ECG	Always	Yes	Yes
Event driven	Cardiac events	When needed	No	No
Classic holter	Whole ECG	Periodically (by configuration)	No	No

The first column shows the event driven and classic holter operational modes proposed by the use case. The second column exposes the operational mode scope. In this column, it can be observed that the event driven uses an event detection algorithm. This algorithm enables the transmission of a data set stored in the agent. The span of automatic events is defined by Automatic-Detection-Time. In the classic holter operation mode, the ECG waveform is stored in a PM-store and sent to the manager every 5-10 minutes in bulk transfer. In the third column the modes about how the agent connects to the manager for each operating mode is shown. The fourth column exposes the operation in terms of the data acquisition and transmission of its data to the manager. Finally the support by currents specializations to these operational modes are shown in the fifth column.

3.4 Medication monitor

This use case is related to the X73-10472 Medication Monitor specialization. This is an advanced PHD used to provide medications to the patient. It consists of a device with a carousel loaded with medication. It is programmed to move forward at specific times to supply the medication is available to the patient. When the medication is ready, this device flashes and/or beeps to remind to the patient of the availability of medication. When the container of the device is ready to supply the medication, an event is generated. If the medication is not picked up after a predefined time interval, it also generates an event. A set of parameters are required to be able to program the times and intervals through X73-20601.

A. Medication dosage parameters

The PHD-WG determined that this use case requires parameters to modify the dosage to the user, using the remote control service. These parameters are shown in the table 3.5. In the first column can be observed the parameters required to modify the availability of the medication can be observed. In the second column and indication about if the parameter is supported or not by the currents specializations are indicated in the second column. A brief description about each parameter can be observed in the third column.

Table 3.5: Medication dosage parameters.

Parameter Reference	Supported by current specs	Description
Event-Dosage-Number	Read Only	Dosage Number
Time-Dosage-Available	No	Start and end of service time
Alarm-Duration	No	Start and end of alarm

- **Event-Dosage-Number:** This parameter allows to set the number of doses available for the user per day.

- **Alarm-Duration:** This parameter is used to set the time to alert the patient about the availability of the drug (start and end times).

- **Time-Dosage-Available:** This parameter is used to set the time duration in which the alarm is triggered.

3.5 Sleep Apnea Breathing Therapy Equipment (SABTE)

This PHD is used for the treatment of various types of sleep-related diseases. Its operational mode is based on the application of continuous positive airway pressure, inside the patient’s respiratory tract during the night. Moreover, this operation has four different types of settings for selecting the appropriate therapy – e.g. Continuous Positive Airway Pressure (CPAP), Auto-CPAP, and Bi-Level Positive Airway Pressure (Bi-Level PAP). These types of therapy require specific parameters for each configuration.

A. Therapy group parameter

Within the operation mode of this use case, four different types of therapy are available to the user. Also, each therapy requires a specific configuration of their operating parameters. Table 3.6. show all the parameters necessary for the application of all different therapies implemented in this use case. In the first column a group of parameters specific for each therapy can be observed.

Table 3.6: Therapy selection.

Therapy	Parameter Reference
CPAP	CPAP[1], PAP Waveform[2].
Auto-CPAP	Min APAP[1], Max APAP[1], PAP Waveform[2].
Bi-Level-PAP	IPAP[1], EPAP[1].
Auto-Bi-Level-PAP	Min IPAP[1], Max IPAP[1], Min EPAP[1], Max EPAP[1].

[1] Fixed or Self-Adjusting, measured in hectopascal or in centimetres of water (cmH_2O).

[2] Fixed or Self-Adjusting, measured in seconds.

[3] Fixed or Self-Adjusting, it can be an absolute flow value, or a given percentage of the peak inspiratory flow (usually 25%).

[4] Fixed or Self-Adjusting, it determines speed of rise of flow (volume control mode) or pressure (pressure control and pressure regulated volume control modes).

[5] Fixed or Self-Adjusting, It is determined by inspiratory time + inspiratory pause time in seconds.

[6] Fixed or Self-Adjusting, beats per minute (bpm).

- **CPAP:** The group of parameters corresponding to CPAP therapy defines the value for applied continuous positive air pressure, and it is the original type of breathing therapy which delivers air at a constant pressure if the patient is inhaling or exhaling.

- **Auto-CPAP:** The group corresponding to Auto-CPAP therapy, it is similar to CPAP therapy; however, its value is adjusted to the patient's requirements.

- **Bi-Level-PAP:** This group delivers preset parameters to inspiratory positive airway pressure (IPAP) and expiratory positive airway pressure (EPAP), and these are necessary for the application of Bi-Level-PAP therapy.

- **Auto-Bi-Level-PAP:** The group corresponding to Auto-Bi-Level-PAP therapy which is similar to CPAP therapy; however, its value is adjusted to the patient's requirements automatically.

B. Operational mode

This use case proposes the implementation of four different operational modes. The proposed operational modes and their parameters are shown in Table 3.7.

Table 3.7: Operational modes.

Operational mode	Connected to manager	Supported by current specs
Standby	When needed	Yes
Therapy	Always	Yes
Drying	Periodically (by configuration)	Yes
Mask Test	Periodically (by configuration)	Yes

In the first column the four operational modes are shown. The Standby parameter places the device in standby function. The therapy mode enables the therapeutic mode in the agent. The Drying mode enables the drying operation mode in the agent, and the Mask Test parameters configures the agent to operate in a test mode. In the second column the execution periodicity of these operational modes is shown and in the third column it is established if the parameters are supported or not in the standard.

3.6 Results

After analysing the use cases of proposed use within the PHD-WG, it has drawn up the results summarized in Table 3.8. Each use case has two types of requirements classified in two groups: first one specifies the operational modes; the second one concentrates the operating parameters for each use case. The use cases ECG Recorder, Medication Monitor, and Sleep Apnea Breathing Therapy Equipment use cases are exposed in the first column. In the second column are shown two operational modes for the ECG Recorder, these are Store-and-forward, and Real-time. The Medication monitor has no additional mode of operation, so this type of configuration data is discarded. Nevertheless the last use case has four operational modes: Standby, Therapy, Drying, and Mask-Test which show the importance of a setting parameter that would modify the operational mode in the agent. Finally, in the third column, the operating parameters of each use case are shown. These operating parameters are of interest to propose a command and control extension that allows to modifying their values.

Table 3.8: Use cases operational modes and setting parameters.

Use Case	Operational modes	Settings Parameters
(A)ECG Recorder	Store-and-forward Real-time	Operational-Mode, Taqui-Lim, Brady-Lim Asyst-Lim QRS-Number T-Wave-Time Search-Back
(B)Medication Monitor	-	Operational-Mode, Time-Dose-Available Alarm-duration Event-Dosage-Number
(C) Sleep Apnea Breathing Therapy Equipment	Standby, Therapy, Drying, Mask-Test	Therapy-Selector, CPAP, IPAP, EPAP, Inspiration-Trigger-Sensitivity, Expiration-Trigger-Sensitivity, Minimum-Respiratory-Frequency, Ramp-Start-Pressure, Ramp-Duration

In table 3.9 the necessary set of parameters have been added for each use case. In the first column, the Operational-mode, Therapy-selector and Configuration parameters are observed. The second column shows the use cases required for each parameter defined in the first column.

The type of data required to implement the parameters inside the remote control extension are specified in the third column. The type of value that represents each parameter is shown in the fourth column and in the last column shows an examples of data related to the three use cases.

Table 3.9: Setting classification.

Operational Characteristics	Use case	Type	Value-Type	Example
Operational-mode	(A), (B), (C)	Enumeration	MDC_DEV_PROFILE	Real Time, Store-and-forward
Therapy-Selector	(C)	Enumeration	MDC_<THERAPIE>_ENABLE	CPAP, Auto-CPAP, Bi-Level-PAP Auto-Bi-Level-PAP
Configuration Parameters	(A), (B), (C)	Enumeration, Compound Numeric Numeric Numeric Numeric Compound Numeric Compound Numeric	Taqui-Lim Asist-Lim Store-Time-Auto Dosage-Number Daily-Alarm-Times CPAP IPAP	100-200 beat-per-minute 2-4 seconds 30-120 seconds 30 dosages 1-24 times 4-20 cm3 H2O 4-30 cm3 H20

3.7 Discussion

After the three use cases were reviewed and analised, several discussion points were derived. The used methodology was based on the search for similarities present in the use cases proposed within the PHD-WG. The operational mode is a point of special interest due to the fact that this function is present in two of the three use cases analysed. This could lead to generate a proposal to enable different operational modes in the specializations that require it. The ECG recorder requires a setting that allows the user to select one of two operational modes available. The SABTE has four operational modes and also it has the ability to apply four types of therapy. However, at the time of writing this document, more use cases that share the need to implement this parameter could have been generated. The third use case, the medication monitor, has only one mode of operation, so it does not require to add new feature related to it.

The configuration parameters are another important requirement in all use cases due to the operations of the three devices are based on the settings of these parameters. At present, several data types needed to implement the required parameters have been previously defined in the current version of the standard; nevertheless, there is currently no mechanism to change their values.

Finally, it can be summarized that a mechanism to set the operational mode of the devices and their operating parameters are required. In addition, these requirements lay the foundations for development of a first version of an extension of remote control and configuration, which undoubtedly will need to be upgraded in use cases that may arise in the future.

Chapter 4 Proposal to the PHD-WG

This chapter presents the command and control extension proposed to the PHD-WG. A description of the methodology followed is shown in Section 4.2. Later, the proposal which is the main result of this Thesis is described in Section 4.3. This description is divided into four parts; the first part presents the additions (new scanner object classes) to the scanner package needed to support control package; the second part describes the control package DIM (mainly the SCO and the Operation objects); the third and fourth parts show additions made to the service model and the communication model, respectively (new invoke and notification services). Finally, some points of special interest generated from this proposal are discussed in Section 4.4.

4.1 Introduction

The development of this proposal was carried out within the PHD-WG. The help of its member has surely improved the quality of our proposal and which will ensure the maximum agreement. As it will be mentioned in the Discussion Section, there was an initial brain-storming moment where several other proposals leaded by the author were also discussed within the PHD-WG but finally rejected. This chapter centred in the success proposal which reuse elements of existing standards in X73POC subfamily such as the DIM published in X73-10201 and the POC command and control proposal, X73-20301, that, at the moment of this writing, continues to be in draft state [8,31]. However, these reused elements need to be adapted following the guidelines required by X73PHD and, then, integrated into X73PHD.

4.2 Methodology

The first step in the development of the command and control extension was to analyse the scope of the standard. This was necessary to determine how to add the proposal into X73-PHD with the aim of benefit all specializations existing and those generated in the future. To meet the above, the PHD-WG guided the development of this proposal defining the following guidelines:

1. To reuse as much as possible the elements taken from X73POC and extend it appropriately within the Optimized Exchange Protocol (X73-20601).

2. To develop the command and control extension keeping a balance between power consumption and performance.

3. To optimize the over-the-air transferred bytes with the aim of to accelerate the exchange of information.

In order to correctly integrate the proposal, it was necessary to analyze the modeling of the X73-20601 to know the constituent parts. This proposal adds features to the DIM, the service model, and the communication model:

- **Domain Information Model (DIM):** Class definitions and data types of their attributes are included in this part.

- **Service model:** New services for remote control, including GET, SET, action, and event reports, are included in this part.

- **Communication model:** FSM changes and new state machines goes here. State diagrams and communication details are also included.

4.3 The proposal

In the following sections, the different classes that make up this proposal will be presented as well as the attributes, methods, events and services that are incorporated.

4.3.1 Additions to the Scanner Package

As part of the command and control extension proposal, the UcfgScanner and Operating Scanner objects, taken from the Extended Services package in POC have been included. Below, the new resulting scanner package model with the two new classes is shown in Figure 4.1. The specification of each new class is presented in the next subsections.

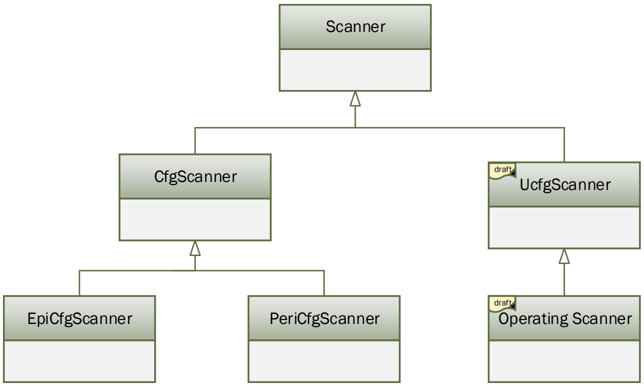


Figure 4.1: The new scanner package model.

4.3.1.1 UcfgScanner object

The UcfgScanner object scans a predefined set of managed. It has the following peculiarities:

- Event reports are typically used in confirmed mode.

- The list of scanned objects/attributes is fixed (i.e., cannot be configured).

- The UcfgScanner object is an abstract class; it cannot be instantiated.

The nomenclature code to identify the UcfgScanner class is MDC_MOC_SCAN_UCFG.

4.3.1.1.1 UcfgScanner attributes

Table 4.1 defines the set of UcfgScanner attributes that are supported for personal health device communication.

Table 4.1: UcfgScanner object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Confirm-Mode	MDC_ATTR_ CONFIRM_MODE	ConfirmMode	Optional
Confirm-Timeout	MDC_ATTR_ CONFIRM_TIMEOUT	RelativeTime	Optional
Transmit- Window	MDC_ATTR_TX_ WIND	INT-U16	Optional

Below the use of UcfgScanner attributes are described:

Confirm-Mode attribute: This attribute defines if confirmed event reports or unconfirmed event reports are used. In confirmed event reports, after receiving a message, the agent confirms the reception of a confirmed event report using a confirmation response. The data types used by the Confirm-Mode attribute are shown in Listing 4.1.

Listing 4.1: ConfirmMode definition.

```
ConfirmMode := INT-U16{
        unconfirmed(0),
        confirmed(1)
 }
```

Confirm-Timeout attribute: The Confirm-Timeout attribute determines the minimum time that the agent shall wait for a response message from the manager after sending a confirmed event report message before timing out and transitioning to the Unassociated state occurs. The relative time data type represents a time counter that is used to determine the relative time between events. This has a resolution of 125 μs (LSB), which is enough for sampling rates up to 8 kHz and span time periods up to 6.2 days. The data type used by the Confirm-Timeout attribute is the RelativeTime (Listing 4.2).

Listing 4.2: RelativeTime definition.

```
RelativeTime: = INT-U32
```

Transmit-Window attribute: The Transmit-Window attribute defines the maximum number of events not confirmed at a given time. The data type used by the Transmit- Window attribute is shown in Listing 4.3.

Listing 4.3: Transmit-Window attribute.

```
RelativeTime: = INT-U32
```

4.3.1.2 Operating Scanner object

The Operating Scanner object is responsible for providing all information about command and control configuration changes within the agent. This information mainly includes Operation objects. The operating scanner performs the following operations:

- Sends CREATE events for Operation object instances to inform about operations to the manager.
- Scans Operation object attributes together with attributes of the SCO.
- Provides a refresh mechanism for reloading the state of Operation object attributes.

The nomenclature code to identify the Operating scanner class is MDC_MOC_SCAN_UCFG_OP.

4.3.1.2.1 Operating Scanner object methods

The Operating Scanner object defines the methods that are shown in Table 4.2. These are used to obtain background information that allows the manager monitoring the current state of agent.

Table 4.2: Operating Scanner object methods.

Method/ Action	**Mode**	**Action-type**	**action-info-args**	**Resulting action-info-args**
Refresh-Operation-Context	Confirmed	MDC_ACT_REFR_OP_CTXT	RefreshObjList	OpCreateInfo (scan report no is 0)
Refresh-Operation-Attributes	Confirmed	MDC_ACT_REFR_OP_ ATTR	RefreshObjList	____

The specifications of Operating Scanner methods are the following:

Refresh-Operation-Context and Refresh-Operation-Attributes methods: These methods are used to refresh the current values of a list of objects and attributes, respectively. If the list in the request is empty, all objects in the scan list are refreshed. If a scanned-attribute is 0, all attributes of that object that are scanned are refreshed. If the object-glb-handle is 0 (in all components), the specified attribute ID is refreshed for all objects in the scan list. The data types used by the Refresh-Operation-Context method are shown in Listing 4.4.

Listing 4.4: Refresh-Operation-Context method definition.

```
RefreshObjList ::= SEQUENCE OF RefreshObjEntry

RefreshObjEntry ::= SEQUENCE {
        object-glb-handle          GLB-HANDLE,
        scanned-attribute          OID-Type
}
```

4.3.1.2.2 Operating Scanner object events

The Operating Scanner object defines the events in Table 4.3. These events are used to notify creation, deletion, or update event reports instances of SCO implemented by the agent.

Table 4.3: Operating Scanner object events.

Event	**Mode**	**Event-type**	**Event-info parameter**	**Event-reply-info**
Oper-Create-Notification	Confirmed/ Unconfirmed	MDC_NOTI_OP_CREAT	OpCreateInfo	—
Oper-Delete-Notification	Confirmed	MDC_NOTI_OP_DEL	OpDeleteInfo	—
Oper-Attribute-Update	Confirmed/ Unconfirmed	MDC_NOTI_OP_ATTR_UPDT	OpAttributeInfo	—

The following Event type definitions are defined for the Operating Scanner class.

Oper-Create-Notification Event: It generates a report through which the agent notifies the manager of all instances of classes SCO implemented in the agent. This report consists of the report number (scan-report-no), and a sequence of scan-info. The data types used by the Oper-Create-Notification event are shown in Listing 4.5.

Listing 4.5: Oper-Create-Notification event definition.

```
OpCreateInfo ::= SEQUENCE {
        scan-report-no              INT-U16,
        scan-info                   SEQUENCE OF OpCreateEntry
 }
```

Oper-Delete-Notification Event: It is used to delete scan reports. The data types used for Oper-Delete-Notification event is defined in Listing 4.6.

Listing 4.6: Oper-Delete-Notification event definition.

```
OpDeleteInfo ::= SEQUENCE {
        scan-report-no                    INT-U16,
deleted-op-list                    SEQUENCE OF OpDeleteEntry
}
```

Oper-Attribute-Update Event: It allows the agent to generate a report to visualize the changes in the attributes of the implemented instances through the OpAttributeInfo parameter. The data types used by the Oper-Attribute-Update event are shown in Listing 4.7.

Listing 4.7: Oper-Attribute-Update Event.

```
OpAttributeInfo ::= SEQUENCE {
           scan-report-no                    INT-U16,
           glb-scan-info                     SEQUENCE OF SingleCtxtOperScan
   }
```

4.3.2 The new Control Package

The classes defined in this package define the command and control operations supported by the agent. The model for remote control can be observed in Figure 4.2. The package is based on the concept of Operation. It represents a single configurable item in the agent. An operation can have virtual attributes. In contrast to a regular DIM attribute, a virtual attribute represents a concept within the agent that does not need to represent anything real at all.

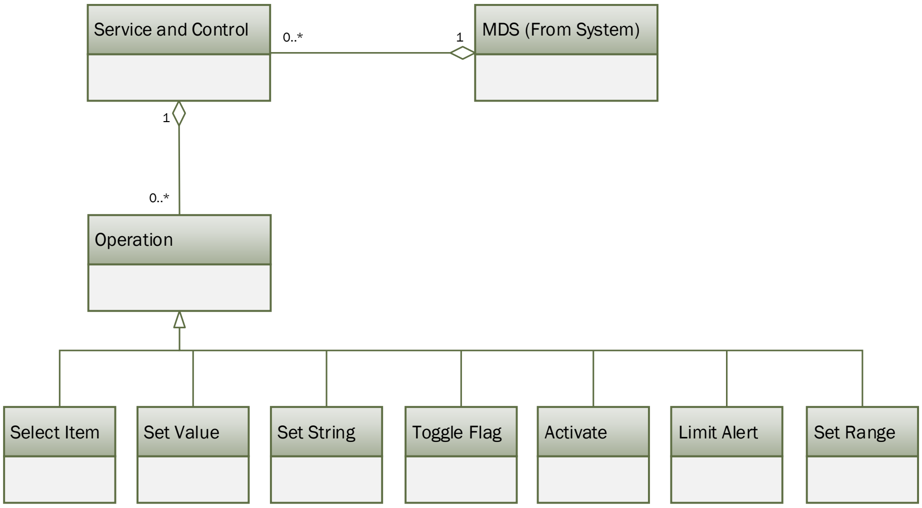


Figure 4.2: Remote Control Model.

The Operation object defines how a virtual attribute can be modified, providing all necessary information about legal attribute values. The Select-Item, Set-Value, Set-String, Toggle Flag, Activate, Limit Alert, and Set Range classes are operations (derived from Operation class) that specify several types of virtual attributes. For example, the Select-Item operation represents a selection control and allows selecting one item from a list of valid values. Set-Value operation allows setting the attribute value directly but additionally provides the limits and the step width (range and resolution) to the manager so that the manager can know a priori the valid values. Moreover, the operation object also defines various forms of human help texts that the manager can use to automatically build the user interface.

It is important to mention that the Operation objects cannot directly be accessed (for example, by using SET services defined in the service model in Clause A.10 of X73-20601 [31]). To access Operation objects the remote control operations must be directed to the SCO. This object supports a simple locking mechanism that prevents side effects caused by simultaneous calls (for example, if two managers are configuring the agent at the same time).

4.3.2.1 Service and Control object (SCO) class

The SCO is responsible for managing all remote control capabilities that are supported by an agent. The SCO is the primary access point for invoking remote control functions. It contains all Operation objects and provides a means for transaction processing. The SCO provides means for the following:

- Simple lock mechanism: it prevents inconsistencies when a device is controlled from multiple access points (e.g., local and remote.)

- State indications, this indication allows local and remote indication of ongoing controls.

The nomenclature code to identify the scanner class is MDC_MOC_CNTRL_SCO.

4.3.2.1.1 SCO class attributes

The set of SCO attributes that are supported for personal health device communication are defined in Table 4.4.

Below the use of SCO class attributes are described:

Handle attribute: It is an unsigned 16-bit number that identifies one of the object instances within an agent. The data type used by the Handle attribute is HANDLE (Listing 4.8).

Listing 4.8: HANDLE definition.

```
HANDLE ::= INT-U16
```

Table 4.4: SCO class attribute.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Handle	MDC_ATTR_HANDLE	HANDLE[1]	Mandatory Static
Sco-Capability	MDC_ATTR_SCO_CAPAB	ScoCapability	Mandatory Static
Sco-Help-Text-String	MDC_ATTR_SCO_HELP_TEXT _STRING	OCTET STRING[2]	Conditional Dynamic
Vmo-Reference	MDC_ATTR_VMO_ REF	HANDLE[1]	Optional Static
Activity-Indicator	MDC_ATTR_INDIC_ACTIV	ScoActivity Indicator	Mandatory Dynamic
Lock-State	MDC_ATTR_STAT_LOCK	Administrative State	Mandatory Dynamic
Invoke-Cookie	MDC_ATTR_ID_INVOK_ COOKIE	INT-U32[3]	Mandatory Dynamic

Sco-Capability attribute: It defines capabilities of the SCO such as the activity indicator, the SCO lock flag and contextual help messages. The data types used by the Sco-Capability attribute are shown in Listing 4.9.

Listing 4.9: ScoCapability definition.

```
ScoCapability ::= BITS-16 {
        act-indicator(0),
        sco-locks(1),
        sco-ctxt-help(8)
}
```

Sco-Help-Text-String attribute: It is used to store text that can help to describe the aim of this SCO. The ASN.1 data type definitions for OCTET STRING is shown in the Table F.1 within the Annex F of X73-20601 [31].

Vmo-Reference attribute: It is used to reference a controlled item. Its type is HANDLER (the same as the Handle attribute).

Activity-Indicator attribute: It indicates to the manager that the SCO is in use. Depending on its value act-ind-off, to off, bit act-ind-on (1) for on and bit act-ind-blinking (2) to activate a blink in the Activity Indicator. The data type used by the Activity-Indicator attribute is the ScoActivityIndicator, defined in listing 4.10.

Listing 4.10: ScoActivityIndicator attribute definition.

```
ScoActivityIndicator ::= INT-U16 {
        act-ind-off(0),
        act-ind-on(1),
        act-ind-blinking(2)
}
```

Lock-State attribute: It is used to lock or release the SCO so that only one process can access the SCO at the same time. This attribute's data type is the AdministrativeState defined in listing 4.11.

Listing 4.11: AdministrativeState definition.

```
AdministrativeState ::= INT-U16 {
          locked(0),
          unlocked(1),
          shuttingDown(2)
   }
```

Invoke-Cookie attribute: It is a transaction identifier that is used to map calls invoking an operation with event messages, as the attribute update operation and error messages that may result after the execution of any operation. The attribute value uniquely identifies a transaction. Therefore, as a result of different invoke operations, different values will have Invoke-Cookie. The data type used by this attribute is shown in Listing 4.12.

Listing 4.12: Invoke-Cookie attribute definition.

```
RelativeTime ::= INT-U32
```

4.3.2.1.2 SCO object methods

The methods implemented in the SCO object are exposed in Table 4.5.

Table 4.5: SCO methods.

Method/Action	**Mode**	**Action-type**	**action-info-args**	**Resulting action-info-args**
Operation-Invoke	Confirmed	MDC_ACT_SCO_OP_INVOKE	OperationInvoke	OperationInvokeResult
Get-Ctxt-Help	Confirmed	MDC_ACT_GET_CTXT_HELP	CtxtHelpRequest	CtxtHelpResult

The specifications of SCO methods are the following:

Operation-Invoke method: It is the main procedure to configure the agent and modify its Operation objects. The request arguments includes a checksum for error checking, an invoke-cookie containing the information context of the operation and the argument op-elem-list, which contains information such as the operation instance number, the type of operation and the list of attributes affected by the operation. The ASN.1 specification is shown in Listing 4.13.

Listing 4.13: Operation-Invoke method definition.

```
OperationInvoke ::= SEQUENCE {
        checksum                INT-I16,
        invoke-cookie           INT-U32,
        op-elem-list            OpInvokeList
}
```

Get-Ctxt-Help method: It provides the summary of Operation objects. The request contains a list with an item for each Operation object. The data type for the Get-Ctxt-Help request argument is the CtxtHelpRequest, which is shown in Listing 4.14.

Listing 4.14: CtxtHelpRequest ASN.1 definition.

```
CtxtHelpRequest ::= SEQUENCE {
        type                  OID-Type,
        op-instance-no     InstNumber
}
```

4.3.2.1.3 SCO object events

The events implemented in the SCO object are shown in Table 4.6.

Table 4.6: SCO object events.

Event	**Mode**	**Event-type**	**Event-info parameter**	**Event-reply-info**
SCO-Operating-Request	Confirmed	MDC_NOTI_SCO_OP_REQ	ScoOperReqSpec(optional)	—
SCO-Operation-Invoke-Error	Confirmed/ Unconfirmed	MDC_NOTI_SCO_OP_INVOK_ERR	ScoOperInvokeError	—

The details of these events are the following:

SCO-Operating-Request event: It can be used to request additional information about the operation context. The first element op-req-id defines a number identifying the request. The op-req-info contains additional information about the operation request. The data types used by the SCO-Operating-Request event are shown in Listing 4.15.

Listing 4.15: SCO-Operating-Request event definition.

```
ScoOperReqSpec ::= SEQUENCE {
        op-req-id                    PrivateOid,
        op-req-info                   ANY DEFINED BY op-req-id
}
```

SCO-Operation-Invoke-Error event: It gathers information about an error related to an operation invoke request. The data types used by the SCO-Operating-Request event are shown in Listing 4.16.

Listing 4.16: SCO-Operation-Invoke-Error attribute definition.

```
ScoOperInvokeError ::= SEQUENCE {
        invoke-cookie           INT-U32,
        operror                 INT-U16 {
                    op-err-unspec(0),
                    checksum-error(1),
                    sco-lock-violation(2),
                    unknown-operation(3),
                    invalid-value(4),
                    invalid-mod-type(5)
        },
        failed-operation-list     SEQUENCE OF InstNumber
}
```

4.3.2.2 Operation class

It is the abstract base class which represents remote controllable items. Each Operation object allows the system to modify some specific item (i.e., a virtual attribute) in a specific way defined by the Operation object. Operation objects are not directly accessible by GET and SET services as they do not have a Handle attribute. The Operation objects instantiated by an agent completely defines its remote control interface. The nomenclature code to identify the operation class is MDC_MOC_CNTRL_OP.

4.3.2.2.1 Operation class attributes

Table 4.7 defines the set of Operation attributes that are supported for personal health device communication.

Table 4.7: Operation object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Instance-Number	MDC_ATTR_ID_INSTNO	InstNumber	Mandatory
Operation-Spec	MDC_ATTR_OP_SPEC	OperSpec	Mandatory
Operation-Text-Strings	MDC_ATTR_OP_TEXT_STRING	OperTextStrings	Optional
Operation-Text-Strings-Dyn	MDC_ATTR_OP_TEXT_STRING_DYN	OperTextStrings	Optional
Vmo-Reference	MDC_ATTR_VMO_REF	HANDLE[1]	Optional
Operational-State	MDC_ATTR_OP_STAT	OperationalState	Mandatory

Below the use of Operation class attributes are described:

Instance-Number attribute: It is used within the context of an SCO to define a unique operation identifier. The data type used by Instance-Number attribute is shown in Listing 4.17.

Listing 4.17: Instance-Number attribute type definition.

```
InstNumber :: = INT-U16
```

Operation-Spec attribute: It specifies and describes the Operation function and grouping. Its first element, vattr-id, defines the virtual attribute identifier. The op-target defines the metric or object-oriented nomenclature partition. The options define special elements. The range of importance of the operation is defined by level and the relations between different operations are defined by grouping. The data types used by the SCO-Operating-Request attribute are shown in Listing 4.18.

Listing 4.18: Operation-Spec attribute definition.

```
OperSpec ::= SEQUENCE {
          vattr-id          OID-Type,
          op-target         OID-Type,
          options           OpOptions,
          level             OpLevel,
          grouping          OpGrouping
}
```

Operation-Text-Strings attribute: It provides an additional static help text to the user. It contains text with the meaning of the operation, help guidance for the user, and text to show when the operation is confirmed. The label string indicates the meaning of the operation. The data type implemented by the Operation-Text-Strings attribute is the OperTextStrings, shown in Listing 4.19.

Listing 4.19: OperTextStrings attribute definition.

```
  OperTextStrings ::= SEQUENCE {
        label                           OCTET STRING,
        help                           OCTET STRING,
        confirm                         OCTET STRING
  }
```

Operational-State attribute: It allows enable and disable the Operation object. The data type used by the Operational-State attribute is the OperationState, shown in Listing 4.20.

Listing 4.20: OperationalState definition.

```
OperationalState ::= INT-U16 {
        disable(0),
        Enable(1),
}
```

4.3.2.3 Select Item class

It allows selecting one item out of a given list. The list can have different types such as OID-Type, a number or a human readable UTF-8 string. This class is derived from the operation base class. The nomenclature code to identify the Select Item class is MDC_MOC_CNTRL_OP_SEL_IT.

As an example, there could be a list with different units of measurement for weight, such as pounds or kilograms. Using this option one of these units can be selected.

4.3.2.3.1 Select Item class attributes

Table 4.8 defines the set of Select Item class attributes that are supported for personal health device communication.

Table 4.8: Select Item class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Selected-Item -Index	MDC_ATTR_INDEX_SEL	INT-U16	Mandatory
Nom-Partition	MDC_ATTR_ID_NOM_ PARTITION	NomPartition	Conditional
Select-List	MDC_ATTR_LIST_SEL	SelectList	Mandatory

Below the use of Select Item class attributes are described:

Selected-Item-Index attribute: It indexes a list of possible parameters and valid values for an attribute. The data type used by the Selected-Item-Index attribute is shown in Listing 4.21.

Listing 4.21: Select-Item-Index attribute definition.

```
INT-U16 :: = INTEGER (0..65535)
```

Nom-Partition attribute: It specifies the nomenclature partition that is used. Its values is 0 in case of unspecified partition, 1 for the object-oriented partition, 2 for metric partition, 3 for alerts and event partition, 4 for dimension units partition, and 5 for virtual attribute partition. The data type implemented by the Nom-Partition attribute is the NomPartition as shown in listing 4.22.

Listing 4.22: NomPartition definition.

```
NomPartition ::= INT-U16 {
                    nom-part-unspec(0),
                    nom-part-obj(1),
                    nom-part-metric(2),
                    nom-part-alert(3),
                    nom-part-dim(4),
                    nom-part-vattr(5),
                    nom-part-pgrp(6),
                    nom-part-sites(7),
                    nom-part-infrastruct(8),
                    nom-part-fef(9),
                    nom-part-ecg-extn(10),
                    nom-part-idco-extn(11),
                    nom-part-phd-dm(128),
                    nom-part-phd-hf(129),
                    nom-part-phd-ai(130),
                    nom-part-ret-code(255),
                    nom-part-ext-nom(256),
                    nom-part-priv(1024)
  }
```

Select-List: It contains a list of possible choices of values. The data type used by the Select-List attribute is defined in listing 4.23.

Listing 4.23: Select-List attribute definition.

```
    SelectList ::= CHOICE {
            oid-list                [1] SEQUENCE OF OID-Type,
            value-list              [3] SEQUENCE OF FLOAT-Type,
            value-u-list            [4] SEQUENCE OF SelectUValueEntry,
            string-list             [5] SEQUENCE OF OCTET STRING
 }
```

4.3.2.4 Set Value class

It allows the system adjusting a value within a given range. It allows defining resolutions, step width and units. As an example, this operation might be used to set the limit used for the detection of tachycardia in an agent which implements cardiovascular events detection algorithms. The nomenclature code to identify the Set Value class is MDC_MOC_CNTRL_OP_SEL_VAL.

4.3.2.4.1 Set Value class attributes

Table 4.9 defines the set of Set Value class attributes that are supported for personal health device communication.

Table 4.9: Set Value Operation object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Current-Value	MDC_ATTR_VAL_CURR	FLOAT-Type	Mandatory
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Mandatory
Step-Width	MDC_ATTR_VAL_STEP_ WIDTH	OpValStepWidth	Optional
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	Optional

Below the use of Set Value class attributes are described:

Current-Value attribute: This attribute sets a value that might be defined within a range of values. The data type used by the Current-Value attribute is shown in Listing 4.24.

Listing 4.24: Current-Value attribute definition.

```
FLOAT-Type:: = INT-U32
```

Set-Value-Range attribute: It defines the maximum and minimum limits of the valid range for the current-value attribute. It also defines the resolution of current-value. The data type used by the Set-Value-Range attribute is shown in Listing 4.25.

Listing 4.25: Set-Value-Range attribute definition.

```
OpSetValueRange ::= SEQUENCE {
        minimum                    FLOAT-Type,
        maximum                    FLOAT-Type,
        resolution                 FLOAT-Type
}
```

Step-Width attribute: It is an array of ranges which correspond to minimum step widths. The data type used by the Step-Width attribute is shown in Listing 4.26.

Listing 4.26: Step-Width attribute definition.

```
OpValStepWidth ::= SEQUENCE OF StepWidthEntry

StepWidthEntry ::= SEQUENCE {
        upper-edge                 FLOAT-Type,
        step-width                 FLOAT-Type
}
```

Unit-Code attribute: It defines the units of Current-Value attribute using the nomenclature code. The data type used by the Unit-Code attribute is shown in Listing 4.27.

Listing 4.27: Unit-Code attribute definition.

```
OID-Type:: = INT-U16 – 16-bit integer type.
```

4.3.2.5 Set String class

It contains a configurable string. This class is derived from the operation base class. For example, using this operation, information relative to personal data (name, EHR number, an URL, etc.) could be remotely set. The nomenclature code to identify the Set String class is MDC_MOC_CNTRL_OP_SET_STRING.

4.3.2.5.1 Set String class attributes

Table 4.10 defines the set of String Operation class attributes that are supported for personal health device communication.

Table 4.10: Set String Operation object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Current-String	MDC_ATTR_STRING_CURR	OCTET STRING	C[a]
Set-String-Spec	MDC_ATTR_SET_STRING_ SPEC	SetStringSpec	Mandatory Static

Below the use of Set String class attributes are described:

Current-String attribute: It hold the current text value. The data type used by the Current-String attribute is shown in on Annex F within X73-20601.

Set-String-Spec attribute: It describes the properties of the string such as terminator, character, string length, displayable or not. The max-str-len defines the maximum supported string length, and the character size in bits (e.g. 7.8 or 16) is defined by char-size and set-str-opt groups is a set of bits that defines the following: the bit 0 indicates that this string is terminated with NULL character, the bit 1 indicates that the string is displayable, the bit 2 indicates the variable length of the string and the bit 3 indicates if the content of the string is hidden. The data type used by the Set-String-Spec attribute is shown in Listing 4.28.

Listing 4.28: Set-String-Spec attribute definition.

```
SetStringSpec ::= SEQUENCE {
          max-str-len       INT-U16,
          char-size         INT-U16,
          set-str-opt        SetStrOpt
}

-- Options for the string

SetStrOpt ::= BITS-16 {
          setstr-null-terminated(0),
           setstr-displayable(1),
           setstr-var-length(2),
           setstr-hidden-val(3)
}
```

4.3.2.6 Toggle Flag class

It allows a switch to be toggled (with two states, e.g., on/off). For example, with this operation, a visual indicator might be activated in an agent to alert the user about any pending operation or one in execution (supply some replacement device, remote control operation execution, measurement failure).The nomenclature code to identify the Toggle Flag class is MDC_MOC_CNTRL_OP_TOG.

4.3.2.6.1 Toggle Flag class attributes

Table 4.11 defines the set of Toggle Flag Operation class attributes that are supported for personal health device communication.

Table 4.11: Toggle Flag Operation object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Toggle-State	MDC_ATTR_STAT_OP_ TOG	ToggleState	Mandatory
Toggle-Label-Strings	MDC_ATTR_STAT_OP__ TOG	ToggleLabelStrings	Mandatory

Below the use of Toggle Flag class attributes are described:

Toggle-State attribute: It allows alternating an operational state or a component. The bit 0 places the state at 0 and the bit 1 changes it to 1. The data type that defines the Toggle-State attribute is defined in Listing 4.29.

Listing 4.29: Toggle-State attribute definition.

```
ToggleState ::= INT-U16 {
        tog-state0(0),
        tog-state1(1)
}
```

Toggle-Label-Strings attribute: It allows enabling text label string, and if the lbl-state0 is selected, the label string is disabled, and with lbl-state1, it is enabled. The data type used by the Toggle-Label-Strings attribute is shown in Listing 4.30.

Listing 4.30: Toggle-Label-Strings attribute definition.

```
    ToggleLabelStrings ::= SEQUENCE {
lbl-state0                          OCTET STRING,
lbl-state1                          OCTET STRING
                        }
```

4.3.2.7 Activate class

It allows starting a defined procedure (e.g., a zero pressure). For example, through this operation, additional functions can be activated in the PHDs. In the case of an electrocardiogram event recorder, an application example could be the generation of manual events from the manager (a smartphone).The nomenclature code to identify the Activate class is MDC_MOC_CNTRL_OP_ACTIV. The Activate Operation object class does not define any attributes, method, event or services.

4.3.2.8 Limit Alert class

It allows adjusting the limits of an alarm detector and enabling or disabling the limits of the alarm. As an example, through this operation, the voltage limits for the operation and replacement of a battery of PHDs can be set. The nomenclature code to identify the Limit Alert class is MDC_MOC_CNTRL_OP_LIM.

4.3.2.8.1 Limit Alert class attributes

Table 4.12 defines the set of Limit Alert Operation class attributes that are supported for personal health device communication.

Table 4.12: Limit Alert Operation object class attributes.

Attribute name	**Attribute ID**	**Attribute type**	**Qualifiers**
Alert-Op-Capability	MDC_ATTR_AL_OP_CAPAB	AlOpCapab	Mandatory
Alert-Op-State	MDC_ATTR_AL_OP_STAT	CurnLimAlStat	Mandatory
Current-Limits	MDC_ATTR_LIMIT_CURR	CurLimAlVal	Mandatory
Alert-Op-Text-String	MDC_ATTR_AL_OP_TEXT_STRING	AlOpTextStrings	Optional
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Mandatory
Unid-Code	MDC_ATTR_UNIT_CODE	OID-Type	Mandatory.
Metric-Id	MDC_ATTR_ID_PHYSIO	OID-Type	Mandatory.

Below the use of Limit Alert class attributes are described:

Alert-Op-Capability attribute: It indicates that component can be switched on or off. Several flags are supported: the bits 1 and 2 allow selecting low and high limit, and the bit 5 allows setting the limit automatically. The bits 8 and 9 enable the capability to switch on/off to the low and high limit, and finally the bit 10 allows enabling the complete alert system. The data type used by the Alert-Op-Capability attribute is shown in Listing 4.31.

Listing 4.31: Alert-Op-Capability attribute definition.

```
AlOpCapab ::= BITS-16 {
        low-limit-sup(1),
        high-limit-sup(2),
        auto-limit-sup(5),
        low-lim-on-off-sup(8),
        high-lim-on-off-sup(9),
        lim-on-off-sup(10)
}
```

Alert-Op-State attribute: It defines the current limit alert state. The bit 0 indicates that all the alerts are disabled. If the bit 1 is set the low-limit violation detection is off and if the bit 2 is set, the high-limit violation detection is off. The data type used by the Alert-Op-State attribute is shown in Listing 4.32.

Listing 4.32: Alert-Op-State attribute definition.

```
CurLimAlStat ::= BITS-16 {
        lim-alert-off(0),
         lim-low-off(1),
        lim-high-off(2)
}
```

Current-Limits attribute: It defines the current alarm limits and its compound by two limits values of FLOAT-type for the lower and upper limits. These can be set by Operation-Invoke method. The data type used by the Current-Limits attribute is shown in Listing 4.33.

Listing 4.33: Current-Limits attribute definition.

```
CurLimAlVal ::= SEQUENCE {
        lower FLOAT-Type,
        upper FLOAT-Type
}
```

Alert-Op-Text-String attribute: It allows defining individual text for upper and lower limit. It is composed by lower and upper text octet strings. The data type used by the Alert-Op-Text-String attribute is shown in Listing 4.34.

Listing 4.34: Alert-Op-Text-String attribute definition.

```
AlertOpTextString ::= SEQUENCE {
        lower-text              OCTET STRING,
        upper-text              OCTET STRING,
}
```

Set-Value-Range attribute: It allows defining the ranges maximum, minimum and the resolution. The data type used by the Set-Value-Range attribute is shown in Listing 4.35.

Listing 4.35: Set-Value-Range attribute definition.

```
This attribute is within the Set Value definitions.

  OpSetValueRange ::= SEQUENCE {
        minimum                 FLOAT-Type,
        maximum                 FLOAT-Type,
         resolution             FLOAT-Type
  }
```

Unit-Code attribute: It defines the nomenclature code for the units of measure from the nom-part-dim partition (e.g., MDC_DIM_KILO_G). The data type used by the Unit-Code attribute is an OID-Type, shown in Listing 4.36.

Listing 4.36: Unit-Code attribute.

```
OID-Type:: = INT-U16 – 16-bit integer type.
```

Metric-Id attribute: It contains dynamic identification about the metric (e.g., label about metric). Usually this attribute is part of an observed value. The data type used by the Metric-Id attribute is an OID-Type, shown in Listing 4.37.

Listing 4.37: Metric-Id attribute definition.

```
OID-Type:: = INT-U16 – 16-bit integer type.
```

4.3.2.9 Set Range class

It permits the system to adjust low and high values (i.e., a value range) within defined boundaries. This class is derived from the operation base class. For example, using this operation, a valid range of values for a parameter can be set. The nomenclature code to identify the Set Range class is MDC_MOC_CNTRL_OP_SET_RANGE.

4.3.2.9.1 Set Range class attributes

Table 4.13 defines the set of Set Range Operation Operation class attributes that are supported for personal health device communication.

The following attributes are defined for the Set Range class.

Range-Op-Text attribute: It defines individual texts labels for upper and lower value limits. The data type that defines the Range-Op-Text attribute is defined in listing 4.38.

Listing 4.38: Range-Op-Text attribute definition.

```
RangeOpText ::= SEQUENCE {
        low-text OCTET STRING,
        high-text OCTET STRING
}
```

The attributes Current-Range, Set-Value-Range, Step-Width and Unit-Code has been previously defined in the section 4.3.2.4.1.

Table 4.13: Set Range Operation object class attributes.

Attribute name	Attribute ID	Attribute type	Remark	Qualifiers
Current-Range	MDC_ATTR_RANGE_CURR	CurrentRange	Reporting period of the event reports.	Mandatory
Range-Op-Text	MDC_ATTR_RANGE_OP_TEXT_STRING	RangeOpText	Static attribute to define individual texts for upper and lower boundaries	Optional
Set-Value-Range	MDC_ATTR_VAL_RANGE	OpSetValueRange	Range of legal values.	Mandatory
Step-Width	MDC_ATTR_VAL_STEP_WIDTH	OpValStepWidth	Allowed step width	Optional
Unit-Code	MDC_ATTR_UNIT_CODE	OID-Type	From dimensions nomenclature partition.	Optional

4.3.4 Additions to the Service Model.

The service model defines the interactions between the agent and manager. Table 4.14 summarizes all the services implemented in the proposed command and control extension.

The Scanner implements the following service:

- SET (from the Scanner class): It is used by the manager to enable or disable the Operational-State attribute. When it is enabled, the scanner starts sending notifications.

The Operating Scanner implements the following services:

- Refresh-Operation-Context and Refresh-Operation-Attributes methods: They allow the manager to get the configuration context for a certain list of Operation objects (the whole Operation object attributes or specific ones, respectively).

- Oper-Create-Notification, Oper-Delete-Notification, and Oper-Attribute-Update: They are used by the agent to transmit changes of configuration context (creation of new Operations, changes on them and their attributes, and its deletion, respectively).

The SCO implements the following services:

- Operation-Invoke: It allows the manager to modify the values of Operation objects.

- Get-Ctxt-Help: It allows the manager to gather Operation object context information.

- SCO-Operating-Request: It is used by the manager to get information about the type of Operation objects.

- SCO-Operation-Invoke-Error: It allows the manager to detect errors during a remote control operation.

Table 4.14: New services defined in the proposed command and control extension.

Object	**Service**	**Type**	**Parameter**	**Results**	**Remarks**
MDS	GET	-	GetArgument Simple	GetResult Simple	Allows the manager to retrieve the value of attributes of the MDS object in the agent.
	EVENT	MDS-Configuration-Event	ConfigReport	ConfigReport Rsp	This event creates a configuration report to inform the DIM of the agent.
		MDS-Dynamic-Data-Update-Var	ScanReportInfoVar	-	This event provides dynamic data (related with measurements) from the agent for some or all of the objects that the agent supports. The event is triggered by an MDS-Data-Request from the manager system, or it is sent as an unsolicited message by the agent.
		MDS-Dynamic-Data-Update-Fixed	ScanReportInfoFixed	-	This event creates a data report to provide dynamic data to manager for some or all of the agents' objects in fixed format.
		MDS-Dynamic-Data-Update-MP-Var	ScanReportInfoMPVar	-	This event provides dynamic data (related to measurements) from the agent for some or all of the objects that the agent supports, but also allows inclusion of data from multiple persons.
		MDS-Dynamic-Data-Update-MP-Fixed	ScanReportInfoMPFixed	-	This event provides dynamic data (typically measurements) from the agent for some or all of the metric objects or the MDS object that the agent supports.
	ACTION	MDS-Data-Request	DataRequest	DataResponse	This method is used to enable or disable

					measurement data transmission from the agent.
		Set-Time	SetTimeInvoke		This operation allows setting the date and time of the agent.
		Set-Base-Offset-Time	SetBOTimeInvoke		This action is used to select the date and time to be set in base offset time format.
PM-Store	GET		GetArgument Simple	GetResult Simple	The manager uses the GET service to retrieve the values of all PM-store object attributes
	EVENT	Segment-Data-Event	SegmentDataEvent	SegmentDataResult	This event sends data stored in the Fixed-Segment-Data of a PM-segment from the agent to the manager. The event is triggered by the manager by the Trig-Segment-Data-Xfer method.
	ACTION	Clear-Segments	SegmSelection	-	This method allows the manager to delete the data stored in one or more selected PM-segments.
		Get-Segment-Info	SegmSelection	SegmentInfoList	This method is used particularly to retrieve the attributes and their data contents from the PM-segment object instances identified by the SegmSelection parameter.
		Trig-Segment-Data-Xfer	TrigSegmDataXfer-Req	TrigSegmDataXferRsp	This method allows the manager to start the transfer of the Fixed-Segment-Data attribute of a specified PM-segment.
Scanner	SET	-	OperationalState	SetArgument Simple	This attribute is used to enables the capability to generate event reports. Through the set service, the state of the attribute is changed from disable

					to enable.
PeriCfgScanner	EVENT	Buf-Scan-Report-Var	ScanReportInfoVar		These events are used to send reports containing Periodic data. These shall be sent with a time interval equal to the Reporting-Interval attribute value. When a periodic configurable scanner is enabled by a manager, scan reports should be sent within a reasonable time and synchronized to the reporting interval of the scanner.
		Buf-Scan-Report-Fixed	ScanReportInfoFixed		
		Buf-Scan-Report-Grouped	ScanReportInfoGrouped		
		Buf-Scan-Report-MP-Var	ScanReportInfoMPVar		
		Buf-Scan-Report-MP-Fixed	ScanReportInfoMPFixed		
		Buf-Scan-Report-MP-Grouped	ScanReportInfoMPGrouped		
Operating Scanner	EVENT	Oper-Create-Notification	OpCreateInfo	ConfigReport Rsp	This event creates a Configuration Report to inform to the manager about the configuration of the agent
		Oper-Delete-Notification	OpDeleteInfo	-	This event is used to delete event report notifications.
		Oper-Attribute-Update	OpAttributeInfo	-	This event creates a Confirmed event report to updates the value of the attributes of an object.
	ACTION	Refresh-Operation-Context	RefreshObjList	OpCreateInfo	This action method is used to indicate a list of operation objects in which scanned operations are refreshed.

		Refresh-Operation-Attributes	RefreshObjList	-	This action method is used to indicate a list of attributes which scanned attributes are refreshed.
SCO	ACTION	Operation-Invoke	Operation Invoke	Operation InvokeResult	This action is used to execute an operation in a specific attribute of an object.
		Get-Ctxt-Help	CtxtHelp Request	CtxtHelp Result	This action is used to request about context information of an operation.
	EVENT	SCO-Operating-Request	ScoOperReqSpec (optional)	-	This event creates a confirmed event report to request an operation in the agent.
		SCO-Operation-Invoke-Error	ScoOperInvokeError	-	These events create a confirmed event report to inform the manager about an error generated by an incorrect operational request.

4.3.5. Additions to the Communication Model

In the current version of the standard, the communication model has a Finite State Machine (FSM) which describes the process of association between an agent and a manager. The command and control proposal complements the FSM by incorporating additional states that are available within the Operating state. These have been defined to maximize backward compatibility with legacy managers, and, therefore, flow diagrams found in X73-20601 (version 1 and 2) are still valid.

4.3.5.1. Finite State Machine additions

The new version of the communication model includes the command and control extension package which defines the common communication features in a bidirectional way between agents and managers when the remote control and configuration occurs. This is governed by a Finite State Machine (FSM) that defines the states an agent goes through during the process of remote control (Figure 4.3). The manager implements a complementary FSM.

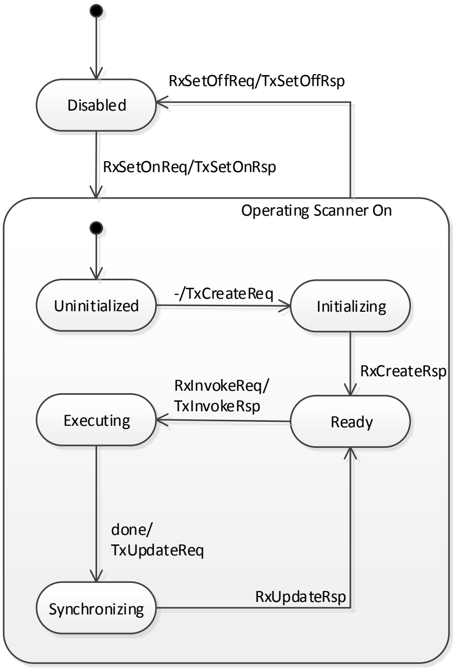


Figure 4.3: Agent's FSM.

This process is allowed within the Operating state (as defined in the X73-20601 communication model). Within the Operating state, agent's command and control FSM initially begins in the Disabled state (which is reflected in the Operational-State attribute in the Operating scanner). Then, the manager sends the Operating Scanner a SET request to change the Operational-State attribute value to enabled (RxSetOnReq). The Operating scanner responds acknowledging the state change (TxSetOnRsp). From the Uninitialized state, the FSM goes, automatically, to the Initialized state. During this transition, the Operating scanner sends a report to the manager (TxCreateReq) which gathers information about every Operation instance that the agent implements. When the agent receives the response from the manager (RxCreateResponse), the FSM goes to the Ready state. In this state, the manager can invoke a method to modify the operations in the SCO, by sending the corresponding command to the agent (RxInvokeReq), and then the SCO acknowledges the invocation (TxInvokeRsp). The agent then goes to the Executing state. At some time, the SCO executes the invocation requested and, when it is done, the Operating Scanner sends the manager a report with the attribute changes (TxUpdateReq) to synchronize the manager. When the manager acknowledges the reception of attribute changes (RxUpdateRsp), the FSM goes from the Synchronizing state to the Ready state.

4.3.5.2. Sequence diagram in detail

Within this section the typical command and control sequence diagrams are analised in detail, indicating the regular operation of the command and control FSM and the Application Protocol Data Units (APDUs) interchanged.

The first APDU to start a sequence of remote control is a SET to the Operating-State attribute in the Operating scanner. It is sent by the manager to enable it (Figure 4.4). This process enables the generation of event reports. On this interaction, the manager modifies the Operational-State using the replace option on the modify operator.

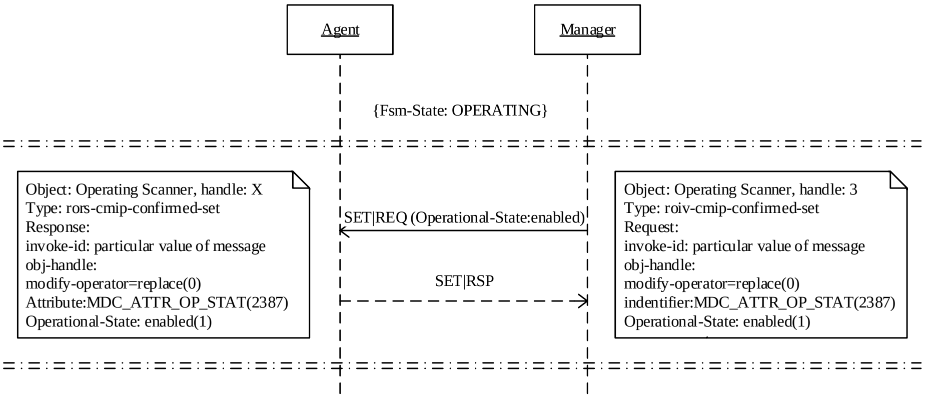

Agent
Manager
{Fsm-State: OPERATING}
Object: Operating Scanner, handle: X
Type: rors-cmip-confirmed-set
Response:
invoke-id: particular value of message
obj-handle:
modify-operator=replace(0)
Attribute:MDC_ATTR_OP_STAT(2387)
Operational-State: enabled(1)
SET|REQ (Operational-State:enabled)
SET|RSP
Object: Operating Scanner, handle: 3
Type: roiv-cmip-confirmed-set
Request:
invoke-id: particular value of message
obj-handle:
modify-operator=replace(0)
indentifier:MDC_ATTR_OP_STAT(2387)
Operational-State: enabled(1)

Figure 4.4: Enable Operating scanner by SET'ing its Operational-State to enable.

In the next interaction, an APDU is sent to the manager to create proxies for Operation objects in the agent (Figure 4.5). This is done by the Operating scanner using an Oper-Create-Notification report. The report contains remote control capabilities of the agent. The manager interprets this information and, for example, shows a User Interface (UI) form to the technician.

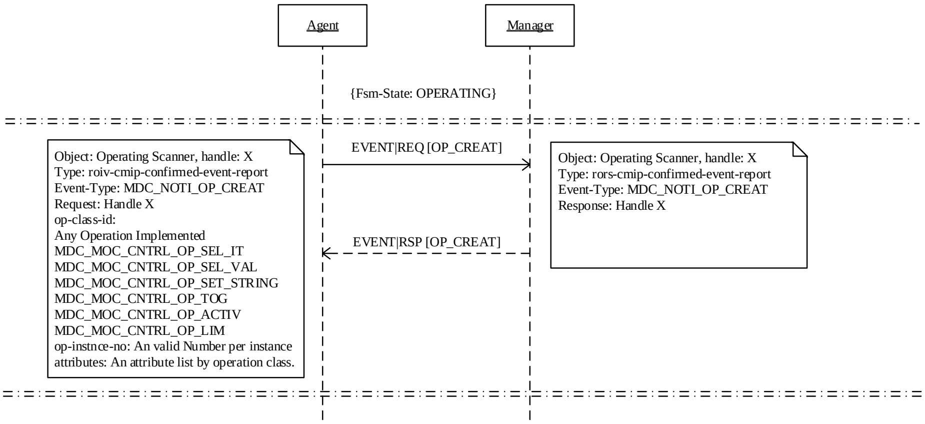

Agent
Manager
{Fsm-State: OPERATING}
Object: Operating Scanner, handle: X
Type: roiv-cmip-confirmed-event-report
Event-Type: MDC_NOTI_OP_CREAT
Request: Handle X
op-class-id:
Any Operation Implemented
MDC_MOC_CNTRL_OP_SEL_IT
MDC_MOC_CNTRL_OP_SEL_VAL
MDC_MOC_CNTRL_OP_SET_STRING
MDC_MOC_CNTRL_OP_TOG
MDC_MOC_CNTRL_OP_ACTIV
MDC_MOC_CNTRL_OP_LIM
op-instnce-no: An valid Number per instance
attributes: An attribute list by operation class.
EVENT|REQ [OP_CREAT]
EVENT|RSP [OP_CREAT]
Object: Operating Scanner, handle: X
Type: rors-cmip-confirmed-event-report
Event-Type: MDC_NOTI_OP_CREAT
Response: Handle X

Figure 4.5: Create Event Report.

Once the manager has a feedback from the user about configuration changes, it can perform an update of Operation objects implemented in the agent (Figure 4.6). To do so, the manager sends the Operation-Invoke to the SCO object. The contents of this request include the OperationInvoke argument and, within this argument, the op-class-id and action-type defines operation will take place. In response, the argument OperationInvokeResult is added, and its value describes whether the operation was successful or was there was any error in its development.

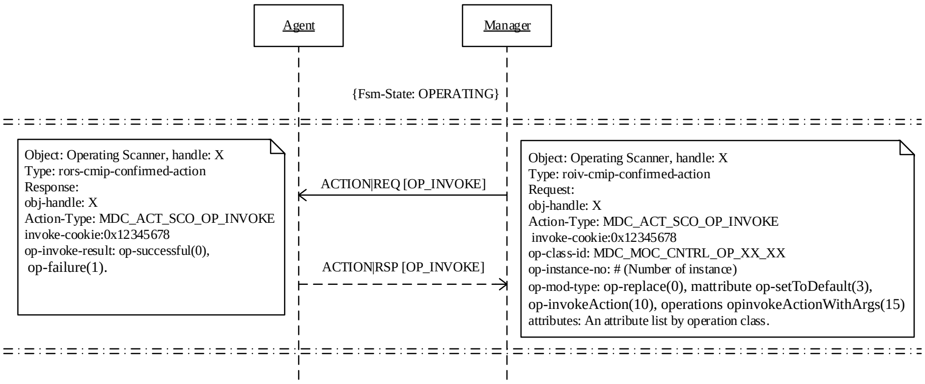

Agent
Manager
{Fsm-State: OPERATING}
Object: Operating Scanner, handle: X
Type: rors-cmip-confirmed-action
Response:
obj-handle: X
Action-Type: MDC_ACT_SCO_OP_INVOKE
invoke-cookie:0x12345678
op-invoke-result: op-successful(0),
op-failure(1).
ACTION|REQ [OP_INVOKE]
ACTION|RSP [OP_INVOKE]
Object: Operating Scanner, handle: X
Type: roiv-cmip-confirmed-action
Request:
obj-handle: X
Action-Type: MDC_ACT_SCO_OP_INVOKE
invoke-cookie:0x12345678
op-class-id: MDC_MOC_CNTRL_OP_XX_XX
op-instance-no: # (Number of instance)
op-mod-type: op-replace(0), mattribute op-setToDefault(3),
op-invokeAction(10), operations opinvokeActionWithArgs(15)
attributes: An attribute list by operation class.

Figure 4.6: Operation Invoke.

Finally, in order to inform the manager about the configuration changes in the agent, the agent reports the changes in an Oper-Attribute-Update notification which is sent by the Operating Scanner object. The argument contains the op-class-id, its instance number, the attribute-id, and the result of the operation (Figure 4.7).

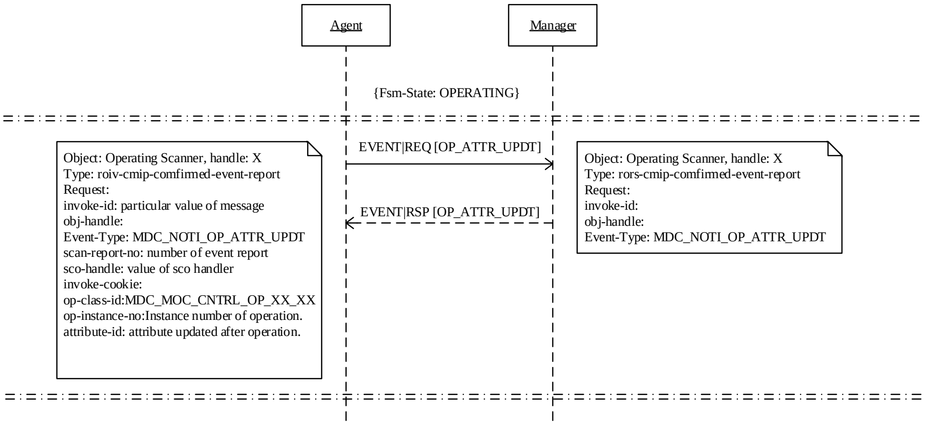

Agent
Manager
{Fsm-State: OPERATING}
Object: Operating Scanner, handle: X
Type: roiv-cmip-comfirmed-event-report
Request:
invoke-id: particular value of message
obj-handle:
Event-Type: MDC_NOTI_OP_ATTR_UPDT
scan-report-no: number of event report
sco-handle: value of sco handler
invoke-cookie:
op-class-id:MDC_MOC_CNTRL_OP_XX_XX
op-instance-no:Instance number of operation.
attribute-id: attribute updated after operation.
EVENT|REQ [OP_ATTR_UPDT]
EVENT|RSP [OP_ATTR_UPDT]
Object: Operating Scanner, handle: X
Type: rors-cmip-comfirmed-event-report
Request:
invoke-id:
obj-handle:
Event-Type: MDC_NOTI_OP_ATTR_UPDT

Figure 4.7: Operation attribute update.

4.4 Discussion

Initially, several proposals were made to the PHD-WG in an open brain-storming round. There were several suggestions both from members of the PHD-WG, our research group and the author of this Thesis. They included the use of SET service, to do some kind of light-weigh version of X73-20301 using a few methods implemented within the MDS, and to define a new System Control Object that embraced all the requirements needed. Nevertheless, the consensus of the PHD-WG was clear (as defined in Section 4.2) and the main interest of the group was to reuse the experience gained in the development of classic remote control extension package (X73-20301).

For the end user, it is expected that the implementation of command and control features will increase the quality of patient-centred services given that it could enable implementing telecontrol and tele-configuration of medical devices though long distances using standards. As it was seen in Chapter 2, telecontrol will integrate mHealth technologies such as smartphones or PDAs in home telemonitoring services and BAN networks. In these scenarios, health professionals will have the ability to change the configuration of PHDs in real time, enabling an improvement in data acquisition tasks from physiological variables, or adjusting the configuration to implement a therapy based on the dynamics of patient's condition.

Regarding data types defined in the proposal, the obj-global-handle with GLB-HANDLE type used in the Operating scanner was taken from the classic X73POC. At the moment of this writing, the discussion within the PHD-WG about its utility in X73PHD is still open and the decision has to be taken. The alternatives are to assign a global identifier code to X73PHD agents, or to redefine the type to be of HANDLE type.

Moreover, to ensure that the remote control package can be used in any specialization of the X73PHD family, its inclusion into X73-20601 has been suggested to the PHD-WG. An important aspect in terms of compatibility is that the addition of command and control extension requires increasing standard's version of the X73-20601. This limits the compatibility with devices that implement earlier versions of the standard. To solve this problem, agents implementing command and control must include a legacy configuration mode without remote control as mandated by current versions of X73-20601. Therefore, compatibility could be guaranteed.

At the moment of this writing, the proposal counts on a degree of consensus within the PHD-WG and meets the requirements found in use cases described in this work (Chapter 3):

- It provides the attributes that can be accessed or modified by the manager.

- It promotes validation rules (e.g. a list of possible legal attribute values).

- It allows an agent defining virtual attributes. These attributes usually correspond to technical or functional parameters of special interest to the administration of the device not needed in health reports.

- It provides a lock mechanism that prevents concurrency problems (two or more manager trying to configure the agent at the same time).

Given that it has been discussed within the PHD-WG, it will possibly meet the requirements of other use cases and specializations in the future. However, other use cases might arise which could define additional requirements. For this reason, the command and control extension is limited to be the first milestone.

Chapter 5 Proof of concept

This Chapter is organized as follows. First, an introduction and a general context of the proof of concept are described. Second, the methodology implemented for the development of this proof-of-concept is exposed. This describes the hardware platform used as well as low level primitives that handle APDU management and its implementation using the Real Time Operating System (RTOS) namely FreeRTOS. Then the implementation of the weigh scale and the ECG event recorder are carried out in Section 5.3 and 5.4, respectively. Finally, a discussion about lessons learnt is given in Section 5.5.

5.1 Introduction

The next step is to test the proposal of the command and control extension on several use cases. The purpose of this test is to analyse the performance of the software in hardware with limited features like low-power, low-voltage microcontrollers and wireless transmitters used in home telemonitoring and mHealth scenarios. For this reason, the proposal of integrating the command and control extension within the X73-20601 was applied into two use cases. First, on a weight scale, the command and control extension was used to modify the units of measurement (kilograms or pounds). Second, on a more complex implementation based on the ECG event recorder was developed. In this device, the command and control extension is used to adjust the values of its operational parameters by requests from the manager. The higher complexity of this implementation is due to the diversity and quantity of operations that can be executed.

The reasons to implement a remote control operation in these agents are diverse. In the case of the weighing scale, the main objective is to show the minimum hardware requirements that an agent require to implement this extension. Since this task seems to be easy to implement, it requires of the greater part of the command and control extension implementation. However, in the case of the ECG event recorder, this implementation has a high number of configuration parameters that have to be modeled using 3 different Operation classes.

A test platform which allows the implementation of these two use cases has been developed to test this proof-of-concept. This platform can be divided into two parts, hardware and software. The test plataform is designed with an LPC1769 microcontroller which integrates the ARM Cortex M3 architecture and is used to develop applications that require low power consumption and high performance in terms of data processing. This microcontroller is plugged to an LPCXpresso Base Board, in order to include the terminals necessary to access to the input and output of data of the microcontroller and to use its LCD display required by the weighing scale to show its measurement. Likewise a WT12-A Bluetooth module which implements HDP is used. This module enables data communication via multiple logical channels carried out by a single master radio channel between agent and manager compatible devices and the needed software was developed over the Free Real Time Operating System (FreeRTOS). The software footprint is small compared to other RTOS and does not require significant hardware resources for its implementation. FreeRTOS provides the task for the management of the APDUs and allows the automatic data transmission through of interrupts generated in the agent when an APDU is received. Also, it allows the development of optimized applications for ARM architecture without generating an overload of processing data that may lead to high power consumption.

The last step in implementing the proposal is to adapt the command and control extension to each of the use cases. Therefore, this adaptation adds the features of each use case to its corresponding X73 specialization. Then the command and control extension is added within the DIM, service model and communication model on each use case [135].

Finally, an analysis of the processing load required for the implementation of this proposal is made in both cases. The purpose of this analysis is to determine whether the processing load does not represent an overload for agents.

5.2 Methodology

To develop this proof of concept, the following methodological order to obtain desired results was defined.

1. First, testing of the platform that allows the implementation of an agent or manager is developed. This platform must meet the operational characteristics required for eHealth and mHealth scenarios. The details of this platform are described in section 5.2.1.

2. Second, the Bluetooth module used provides the transport technology that the test platform requires. However, this is not fully compatible with the APDUs messages. Therefore, it is required to add a header to the APDUs of the agent and manager in order to make the Bluetooth module to be compatible with the APDUs messages.

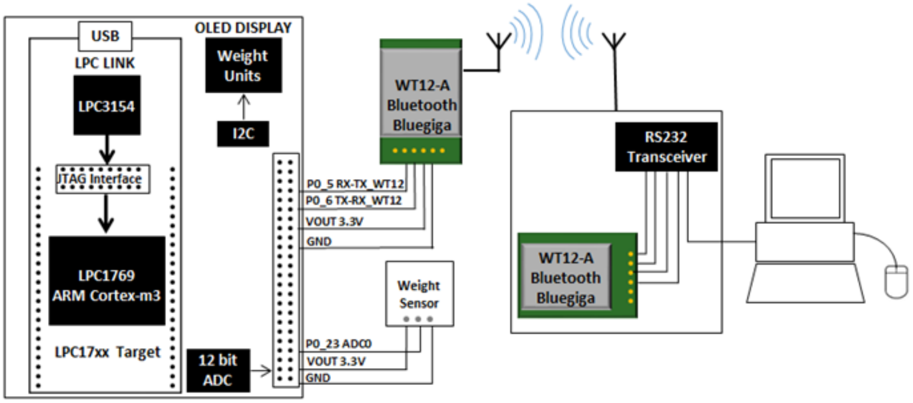
3. Third, the next step is to model each use case using the corresponding specialization for each implementation (the X73-10415 for the Weighing scale and the X73-10406 for the ECG Event Recorder). This modeling includes the features of each case as well as the command and control extension within the DIM, service model and communication model sections.

4. Finally, within the communication model section it is define a basic sequence diagram to validate the development of a remote control operation. Also, an example of the MDER encoded binary messages used in the development of APDUs is shown.

The following points describe how were integrated the different components of the test platform and its conditioning with the operating system for the implementation of the agent and manager.

5.2.1 Testing Platform

The platform configuration is depicted in Figure 5.1. Agents have been programmed into an ARM Cortex-M3 microcontroller which uses a Bluetooth transceiver for sending and receiving information, see Figure 5.2. Likewise managers have been implemented into a PC and a Bluetooth transceiver which has been connected using a RS-232, see Figure 5.3. The WT12 Bluetooth modules implement HDP, where agents have been configured as source and managers as sinks.



Agent Manager

Figure 5.1: Hardware architecture set-up for the testing platform.

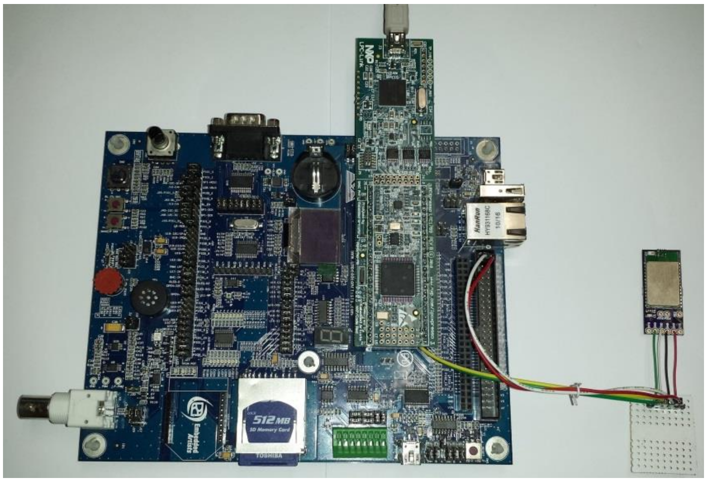


Figure 5.2: Agent: LPC1769 development board and Breaking Board.

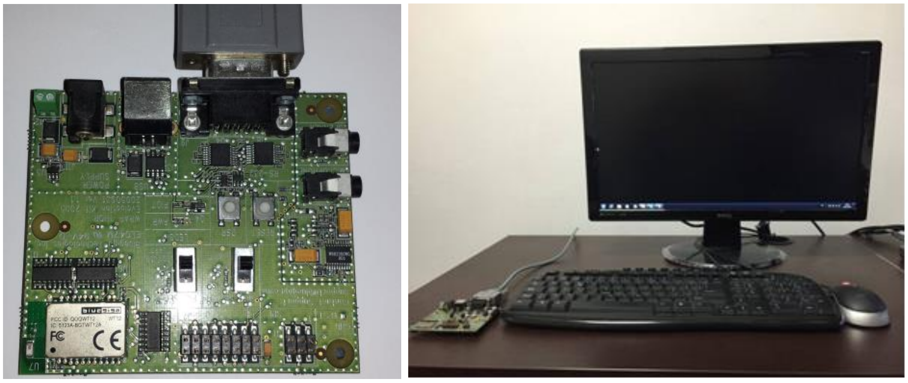


Figure 5.3: BlueGiga Development Board and personal computer used to implement the managers.

5.2.2 Bluetooth Management

The profile Bluetooth HDP is implemented by the WT12 module. It provides a protocol to handle and multiplex the two types of channels needed to implement X73PHD which are reliable and the best-effort data channels. The reliable data channels are used for the reliable transmission of information. This type of channel detects the loss or corruption of transmitted packets, and allows their retransmission by starting a retransmission protocol. The best-effort channels are used to transmit the packets as good as possible although packet loss is permitted and packet order is not guaranteed.

Regarding the APDUs, their specification is defined in Abstract Syntax Notation One (ASN.1) and encoded in Medical Device Encoding Rules (MDER). To correctly encode each APDU, these are first encapsulated by PrstApdu which is interpreted according to the association procedure in an OCTET STRING (Figure 5.4).

DataApdu Type	Size	Octet Length	Invoke-Id	Choice	Choice Length	Data

Figure 5.4: PrstApdu ASN.1 example structure encoded with MDER.

The agent is configured as source and the manager as sink. Given that WT12 does not allow specifying random profiles (or source data types in terms of HDP), we use the most closely profile to the use cases, in case of HOLTIN the X73-10406 was used and X73-10415 for weighing scale. The X73PHD application layer was implemented in the microcontroller. It receives data frames from WT12 via UART. In the personal computer, it works similarly (over RS-232). On this setup, any device can initiate the communication. The first channel to be established is the control channel. Once this channel has been created, one or more data channels between PHDS are negotiated. In order to handle multiple channel communication over a single UART the WT12 provides a multiplexing mode. Each protocol frame is formatted following the diagram in Figure 5.5. An algorithm to analyze and synthesize APDU has been implemented using the Patterns-based Methodology [119].

SOF	LINK	FLAGS	LENGTH	APDU	nLINK

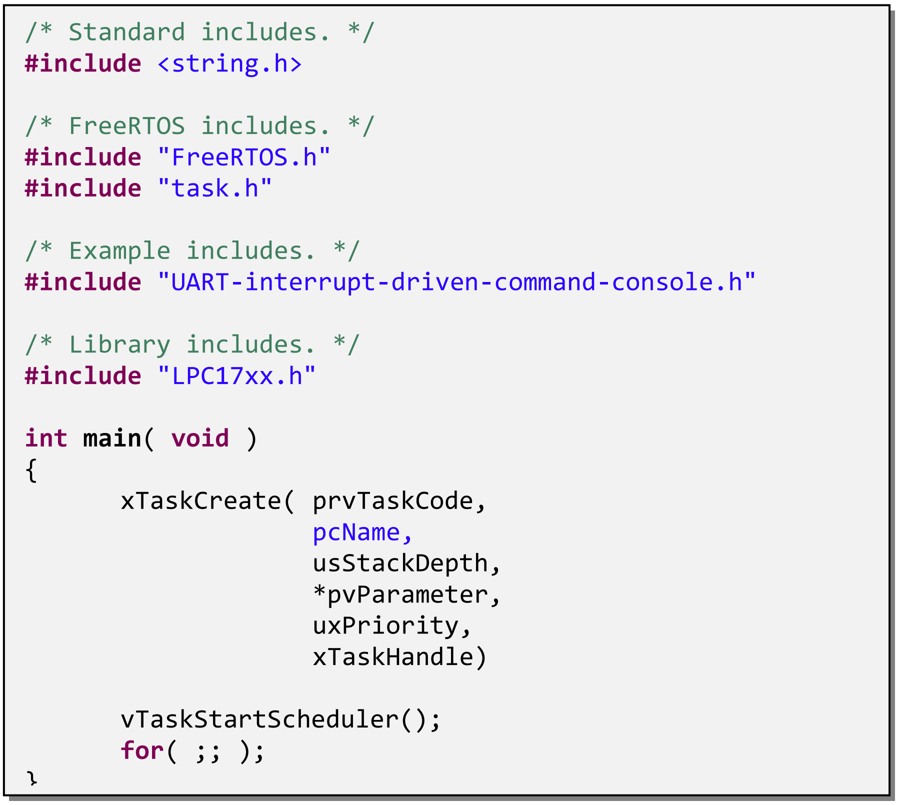
Figure 5.5: Frame format used to communicate with Bluegiga module.

SOF	Start of Frame
LINK	LINK ID
FLAGS	Frame Flags
LENGTH	Size of data fields in bytes
APDU	Presentation APDU
nLINK	{LINK}XOR 0xFF

5.2.3 Development of source code over FreeRTOS

FreeRTOS is a Real Time Operating System. It is developed for use with microcontrollers, but its scope can be extended to other devices such as microprocessors and digital signal processors (DSP). It requires a minimal ROM, RAM and processing overhead. Its footprint will be in the region of 4K to 9K bytes due to the core of the RTOS kernel is contained in only 3 C files and one more for the microcontroller specific source [136]. Furthermore, the use of their API is extremely simple. Listing 5.1 shows the main.c structure. In this source, the order of the libraries used can be seen classified according to their application. Later within the main function, the API function to create a task is shown. The parameters of the function xTaskCreate govern the form in which a task is executed. The prvTaskCode is a pointer to the function that implements the task. The pcName is a name used for debugging purposes. The usStackDepth is a parameter that determines the size of the stack reserved for the task.

Listing 5.1: FreeRTOS Source code.



```
/* Standard includes. */
#include <string.h>

/* FreeRTOS includes. */
#include "FreeRTOS.h"
#include "task.h"

/* Example includes. */
#include "UART-interrupt-driven-command-console.h"

/* Library includes. */
#include "LPC17xx.h"

int main( void )
{
       xTaskCreate( prvTaskCode,
                    pcName,
                    usStackDepth,
                    *pvParameter,
                    uxPriority,
                    xTaskHandle)

       vTaskStartScheduler();
       for( ;; );
}
```

The pvParameter is used to pass values to the function. The uxPriority sets the execution priority and the xTaskHandle is a Handler used to interact between tasks.

As in the xTaskCreate API function, there are more functions that concentrate most of the FreeRTOS features [137]. At the end of the source code the line vTaskStartScheduler executes the task manager, which is responsible for managing the tasks implemented within the RTOS.

Although the RTOS manages most of the hardware resources available, the FreeRTOS allows the execution of interrupts, such as the interruption of the Universal Asynchronous Receiver-Transmitter (UART) peripheral. This interrupt is used to transmit and receive data to the Bluegiga Module. Listing 5.2 shows the function necessary to send and receive data from the interrupt subroutine. The full code for this interrupt can be found in the lpc17xx_uart.c library on the lpc17xx.cmsis.driver.library.

Listing 5.2: UART Send and Receive function.

```
uint32_t UART_Receive(LPC_UART_TypeDef *UARTx, uint8_t
*rxbuf, uint32_t buflen, TRANSFER_BLOCK_Type flag)

uint32_t UART_Send(LPC_UART_TypeDef *UARTx, uint8_t
*txbuf,uint32_t buflen, TRANSFER_BLOCK_Type flag)
```

In the message reception, it is necessary to pass each character received to a data queue, which stores the message to be read for the UART task. The Listing 5.3 shows the task that will handle the APDUs received and transmitted. With this implementation taking advantage of the characteristics of FreeRTOS, it is expected to maximize the efficiency of the used hardware resources and provide tools that allows the feedback device behaviour and above all, to ensure the operational stability.

Listing 5.3: Task for Management of APDUs.

```
void vUARTCommandConsoleStart( void )
{
   xTaskCreate( prvUARTCommandConsoleTask,
                (const int8_t * const ) "UARTCmd",
                 configUART_COMMAND_CONSOLE_STACK_SIZE,
                 NULL,
                  configUART_COMMAND_CONSOLE_TASK_PRIORITY,
                   &xCommandConsoleTask );
}
```

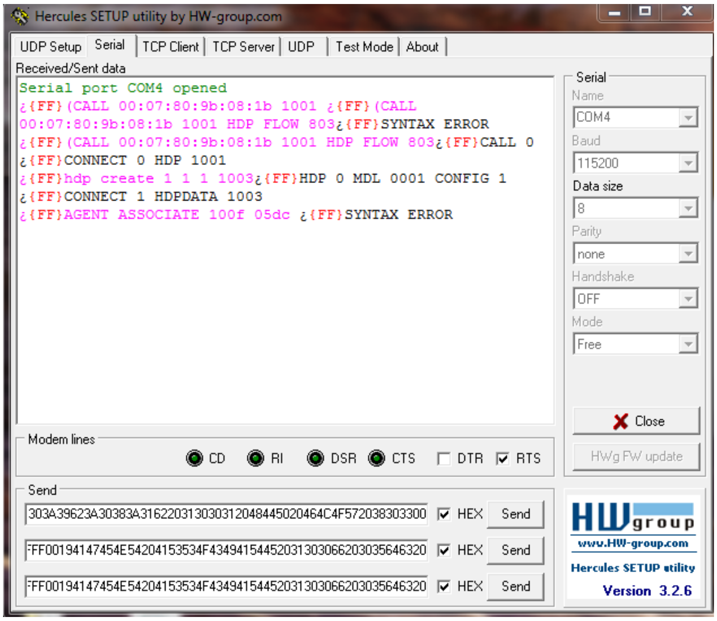
5.2.4 Manager

The managers currently used do not implement the command and control extension. Therefore, the manager has been implemented in a microcontroller in the same way as the agent. In Appendix B the full MDER codes for Weighing scale and Basic Electrocardiograph can be found.

The new APDUs which integrate the command and control extension have been implemented in the corresponding device. If the message exchange has been carried out correctly, the visual indicator is activated.

Moreover as an additional tool to validate the exchange of messages manually, the Hercules Hyperterminal (Figure 5.6) has been selected [138]. This tool has been configured to receive data serially. The messages interchanged between the agent and the manager were checked using this tool. The messages were sent manually to the corresponding APDUs for each message received from the agent.

Although this tool may seem rudimentary, this method allows immediate verification of the interaction between both devices.



```
Serial port COM4 opened
¿{FF}(CALL 00:07:80:9b:08:1b 1001 ¿{FF}(CALL
00:07:80:9b:08:1b 1001 HDP FLOW 803¿{FF}SYNTAX ERROR
¿{FF}(CALL 00:07:80:9b:08:1b 1001 HDP FLOW 803¿{FF}CALL 0
¿{FF}CONNECT 0 HDP 1001
¿{FF}hdp create 1 1 1 1003¿{FF}HDP 0 MDL 0001 CONFIG 1
¿{FF}CONNECT 1 HDPDATA 1003
¿{FF}AGENT ASSOCIATE 100f 05dc ¿{FF}SYNTAX ERROR
```

Serial
Name: COM4
Baud: 115200
Data size: 8
Parity: none
Handshake: OFF
Mode: Free

□ DTR ☑ RTS

303A39623A30383A316220313030312048445020464C4F572038303300 ☑ HEX
FF00194147454E54204153534F43494154452031303066203035646320 ☑ HEX
FF00194147454E54204153534F43494154452031303066203035646320 ☑ HEX

Figure 5.6: Hercules hyperterminal.

5.3 Implementation in weight scale

5.3.1 X73 Model

A. Domain information model

For agent deployment with the X73-10415 profile, it has been proposed that a new domain information model to be used so it includes some classes of the current version of X73-20601 and some others that are part of remote control proposal.

In Figure 5.7, the new classes are added to dim marked with draft label, while existing classes are exposed using standard model classes. The following sections describe each of the classes in this implementation.

The classes that are present in the current version of X73-20601 are the following:

- Medical Device System (MDS) class: This class groups general information in attributes such as Handle, System model, System Id, Device configuration Id, and System-Type-Spec-List of the mandatory type, and some other conditional attributes that help to represent correctly the MDS. It is only possible to instantiate an object of this class.

- Body weight: It represents the nomenclature code to identify the numeric class is MDC_MOC_VMO_METRIC_NU. The attributes that this class defines as mandatory type are Handle, Type, Metric-Spec-Small, Unit-Code, and Attribute-Value-Map.

- Scanner: It is an abstract base class for the unconfigurable scanner class. It provides a flow control mechanism that allows the manager to start and stop agent reporting functions.

The new classes defined for this implementation are the following:

- Operating scanner: It reports the information specifically about the operations implemented in the PHDs. Only a single object of this class is needed for this implementation.

- Unconfigurable scanner (UcfgScanner): It is an abstract base class for the operating scanner class. It represents an observer that monitors the DIM and reports its changes.

- Service and Control Object (SCO) class: It provides mechanisms as Simple transaction processing to modify and operate with operation objects. With this class, the inconsistencies that may occur when a device is controlled from multiple access points (e.g., local and remote) and during the processing of control commands are prevented. The mandatory attributes for this class are Handle, Sco-capability, and Activity-Indicator to give feedback that the system is under remote control. Lock-State is used to prevent unwanted actions, and if locked, no operation can be invoked. Invoke-Cookie to Transaction ID is assigned by invoke. It is only possible to instantiate an object of this class.

- Operation object class: It is an abstract class. This class represents a single remote controllable item. Its attributes contain information that allows the manager to correctly interpret its purpose. The mandatory attributes for this class are Instance-Number, and it is used for operation identification, Operation-spec, which provides a structure defining operation types and properties; and the Operational-State, which is used to specify whether an operation is accessible. This class is not directly accessible but can be controlled by the SCO and observed using the Operating scanner.

- Select Item object: It represents a select item control. The selectable values are given as a list in the Select-Item-List attribute, whose implementation is a mandatory type, as well as the Index attribute. In this implementation, there is a single object of this class, which enables changing the unit of measurement (kilograms or pounds).

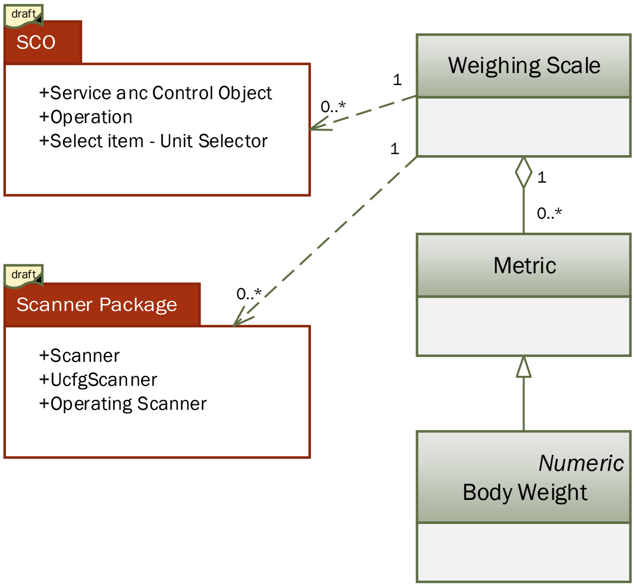


Figure 5.7: The proposed DIM of the Weighing Scale. The packages which include new classes are marked using a tag with the text “draft”.

It is noteworthy that all classes that implement the command and control extension package use a number of attributes which are implemented in the PHDs according to necessary requirements. The attributes required for each class are summarized in Table 5.1.

Table 5.1: Attributes required for the classes of the command and control extension package.

Object	Class	Attributes	Types, respectively ({} = possible values, [] = members)
Service and Control	Service and Control Object	Handle, Sco-Capability, Sco-Help-Text- String, Vmo-Reference, Activity- Indicator, Lock-State, Invoke-Cookie.	INT-U16, BITS-16 {act-indicator(0), sco-locks(1), sco-ctxt-help(8)}, OCTET STRING, INT-U16, INT-U16 {act-ind-off(0), act-ind-on(1), act-ind-blinking(2)}, INT-U16 {locked(0), unlocked(1), shuttingDown(2)}, INT-U32
Unit Selector	Operation	Instance-Number, Operation-Spec, Operation-Text-Strings.	INT-U16, SEQUENCE[vattr-id, op-target, options, level, grouping], SEQUENCE[label, help, confirm-message]
	Select Item	Selected-Item-Index, Nom-Partition, Select-List.	INT-U16, NomPartition, CHOICE{oid-list, value-list, value-u-list,string-list}
Operating Scanner	UcfgScanner	Confirm-Mode,Confirm-Timeout,Transmit-Window.	INT-U16 {unconfirmed(0) or confirmed(1)} , INT-U32, INT-U16.
	Operating Scanner	-	-

B. Service Model

The service model defines the interactions between the agent and manager in the X73-20601 architecture. Table 5.2 shows the object access services implemented in the agent X73-10415.

Table 5.2: Weighing Scale object access services. Highlighted rows correspond to the implementation of the remote control extension.

Object	Service	Type	Parameter	Results
MDS	GET	-	GetArgumentSimple	GetResultSimple
	EVENT	MDS-Configuration-Event	ConfigReport	ConfigReportRsp
		MDS-Dynamic-Data-Update-Fixed	ScanReportInfoFixed	-
	ACTION	Set-Time	SetTimeInvoke	
Operating Scanner	SET	-	OperationalState	SetArgumentSimple
	EVENT	Oper-Create-Notification	OpCreateInfo	ConfigReportRsp
		Oper-Attribute-Update	OpAttributeInfo	-
	ACTION	Oper-Create-Notification	OperationInvoke	-
SCO	ACTION	Operation-Invoke	OperationInvoke	OperationInvoke Result
		Get-Ctxt-Help	CtxtHelpRequest	CtxtHelpResult

Below, the services implemented on the MDS of the Agent X73-10415 are described:

- GET: This service allows the manager to retrieve the value of attributes of the MDS object in the agent X73-10415.

- Set-Time method: This method allows the manager to set the date and time in the agent.

- MDS-Configuration-Event: This notification is based in an Event Report used by the agent to report DIM configuration.

- MDS-Dynamic-Data-Update-Fixed: This notification is used by the agent to send attribute changes to the manager device (i.e., weight measurements and battery warnings).

The Operating Scanner implements the following services:

- SET (from the Scanner class): This service is used by the manager to enable or disable the Operational-State attribute. When it is enabled, the scanner starts sending notifications (MDS-Configuration-Event and MDS-Dynamic-Data-Update-Fixed). When it is disabled, it pauses and saves what it reported up to that moment.

- Refresh-Operation-Context and Refresh-Operation-Attributes: These methods allow the manager to obtain the context of the configuration for a certain list of Operation objects (the whole Operation object attributes or specific ones, respectively).

- Oper-Create-Notification, Oper-Delete-Notification, and Oper-Attribute-Update: These notifications are used by the agent to transmit changes in context (creation of new Operations, changes on them, and deletions, respectively).

The SCO implements the following services:

- Operation-Invoke: This method allows the manager to develop an operation on the agent, for example, to modify the values of Operation objects.

- Get-Ctxt-Help: This method allows the manager to gather Operation object context information in the agent.

- SCO-Operating-Request: This method is used by the manager to get information about the type of Operation objects implemented on the agent.

- SCO-Operation-Invoke-Error: This method allows the manager to detect errors during a remote control operation.

C. Communication Model

1. Finite State Machine

Previously a standard model of the FSM of remote control was presented; this model defines the guidelines to develop a remote control operation. In general terms, this FSM is equal in any implementation, and the changes are made only when different operations are invoked, and this is due to each operation having its own arguments. In the Weigh Scale, the implementation of the RC FSM can be observed at figure 5.8.

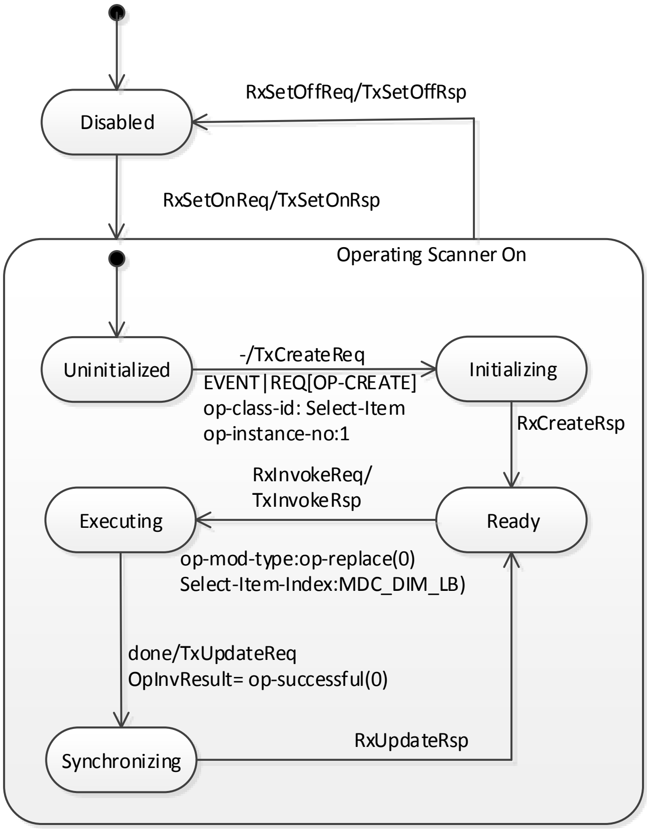


Figure 5.8: Remote Control Weighing Scale Finite State Machine.

On this device, the remote control process starts when the manager device requests to enable the Operational State because the RC FSM (Remote control Finite State Machine) initially begins in the Disabled state, which can be observed through the operational-state attribute of Operational Scanner. This operation takes place when the manager sends to the Operating Scanner a request to set the Operational-State attribute to enabled (RxSetOnReq). Later, the Operating Scanner responds, acknowledging the state change (TxSetOnRsp). At this point, the FSM goes from the not initialized state to the initialized state automatically. Then the FSM goes to the Operational On State which the Operating Scanner sends the manager a report (TxCreateReq) which gathers information about every Operation instance that the agent implements. Specifically for the X73-10415 implementation, this report contains the Select-Item operation and its attributes Selected-Item-Index and Select-List implemented within SCO. For this implementation only one operation (Select-Item) is available. Inside of the Selected-Item-Index can be found the unit of measurement (kilogram and pounds), and that unit could be selected from the Select-List attribute. Later, when the agent receives the response from the manager (RxCreateResponse), the FSM goes to the Ready state. In this state, the manager can invoke a method to modify the operations in the SCO, and for this proof of concept, the manager invokes the Select Item operation, in order to modify the measurement unit in the agent using the replace operator. Subsequently, the SCO acknowledges the invocation (TxInvokeRsp) and the agent goes to the Executing state. The next step starts when the SCO executes the invocation requested and, when it is done, the Operating Scanner sends the manager a report with the attribute changes (TxUpdateReq) to synchronize the manager. This report reflects the changes at the unit code attribute which contains MDC_DIM_LB replacing the MDC_DIM_KILO_G. When the manager acknowledges the reception of attribute changes (RxUpdateRsp), the FSM goes from the Synchronizing state to the Ready state.

2. Sequence Diagram

For this implementation, the aim of the remote control operation is to change the unit of measurement in which the patient's weight is acquired.

Next, a series of figures which integrate the sequence diagram used by this implementation are shown. The objective of the first interaction between the agent and manager is to activate the Operational State attribute. Figure 5.9 shows that this sequence is unchanged and regardless of the specialization implemented in the agent, the content of this message remains constant.

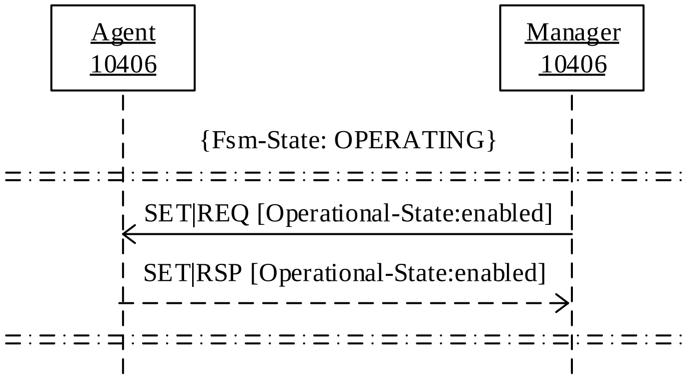

Agent
10406
Manager
10406
{Fsm-State: OPERATING}
SET|REQ [Operational-State:enabled]
SET|RSP [Operational-State:enabled]

Figure 5.9: Enable Operating Scanner by setting the operational-state.

In the second interaction in Figure 5.10, the agent sends an event report with the remote control features implemented. In this case the agent includes in the event report that Select Item operation is available within the remote control features of SCO object class.

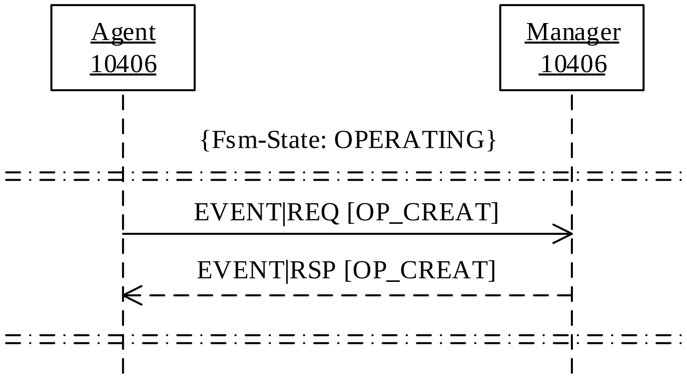


Figure 5.10: Event Report, Operating Scanner Create.

Once the manager has received the event report from the agent, the manager can implement any operation included in the event report. For this particular case, the manager invokes a Select-Item operation to modify an attribute for the agent, as shown in Figure 5.11.

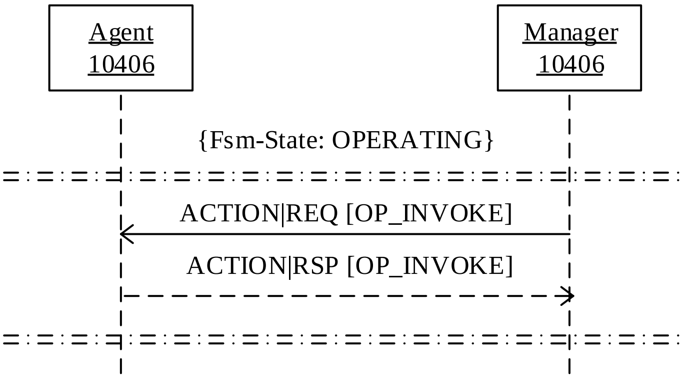


Figure 5.11: Action Request, Operation Invoke.

To verify that the remote control operation is performed correctly, the Oper-Attribute-Update event of Operating Scanner can be used as is exposed in Figure 5.12. Through this event, the agent sends an event report with the changes made by the operation of the remote control. This report shows that the unit of measurement of the X73-10415 has been modified from kilograms to pounds.

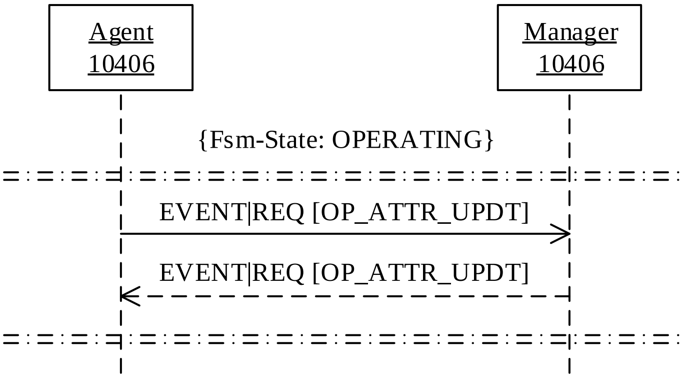


Figure 5.12: Event request, Operation attribute update.

3. MDER Encoded Binary Messages

From the sequence diagram an MDER-encoded binary message was derived. This exemplifies a remote control sequence between one PHD implementing an agent of the X73-10415, the Weighing Scale, and the other implementing a manager of the X73PHD. An example of the presentation APDU is shown in Listing 5.4. Particularly this example corresponds to the event config report, which should include all instances deployed on the agent and its configuration.

Listing 5.4: Event config report.

	EVENT CONFIG REPORT	
A	0xE7 0x00	APDU CHOICE Type (PrstApdu)
	0x00 0x5A	CHOICE.length = 90
	0x00 0x58	OCTET STRING.length = 88
	0x00 0x01	invoke-id = 0x0001 (start of DataApdu. MDER encoded.)
	0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
	0x00 0x52	CHOICE.length = 82
	0x00 0x00	obj-handle = 0 (MDS object)
	0xFF 0xFF 0xFF 0xFF	event-time = agent does not have a RelativeTime clock
	0x0D 0x1C	event-type = MDC_NOTI_CONFIG
	0x00 0x48	event-info.length = 72 (start of ConfigReport)
	0x40 0x00	dev-config-id = extended-config-start (16384)
	0x00 0x03	config-obj-list.count = 3 Measurement objects will be “announced”
	0x00 0x42	config-obj-list.length = 66
	0x00 0x06	obj-class = MDC_MOC_VMO_METRIC_NU
B	0x00 0x01	obj-handle = 1 (1st Measurement is body weight)
	0x00 0x04	attributes.count =4
	0x00 0x24	attributes.length = 36
	0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
	0x00 0x04	attribute-value.length = 4
	0x00 0x02 0xE1 0x40	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL
	0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
	0x00 0x02	attribute-value.length = 2
	0xD0 0x40	intermittent, stored data, msmt aperiodic, agent init, measured
	0x09 0x96	attribute-id = MDC_ATTR_UNIT_CODE
	0x00 0x02	attribute-value.length = 2
	0x06 0xC3	MDC_DIM_KILO_G
	0x0A 0x55	attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
	0x00 0x0C	attribute-value.length = 12
	0x00 0x02	AttrValMap.count = 2
	0x00 0x08	AttrValMap.length = 8
	0x0A 0x56 0x00 0x04	MDC_ATTR_NU_VAL_OBS_SIMP | value length = 4
	0x09 0x90 0x00 0x08	MDC_ATTR_TIME_STAMP_ABS | value length = 8
	0x00 0x2B	obj-class = MDC_MOC_CNTRL_SCO(43)
C	0x00 0x02	obj-handle = 2 (SCO object)
	0x00 0x01	attributes.count = 1
	0x00 0x06	attributes.length = 6
	0x09 0x76	attribute-id = MDC_ATTR_SCO_CAPAB(2422)
	0x00 0x02	attribute-value.length = 2
	0x00 0x00	sco-capability none
	0x00 0x18	obj-class = MDC_MOC_SCAN_UCFG_OP(24)
D	0x00 0x03	obj-handle = 3 (Operating scanner object)
	0x00 0x00	attributes.count = 0
	0x00 0x00	attributes.length = 0

The header which includes the APDU type, size, invoke-id, object handler and the main event features is shown in Section A. The sections B, C and D show the rest of the objects implemented in this agent and their configurations. The object representing the measurement is shown in Section B. The Service and Control object is described at section C. The UcfgScanner object responsible for reporting the remote control features implemented in the agent and its settings is shown in section D.

Listing 5.5 shows MDER encoded binary messages for the Action Request Operation Invoke. The section at the header includes information about the APDU type, size, invoke-id and message type. The SCO object capabilities are shown in section B and the section C shows the operation implemented. In this case, the op-class-id defines the Select Item operation and its attribute Select-item-Index, which contains a list which will be selected for the unit of measurement.

Listing 5.5: Operation Invoke.

```
A   ACTION|REQ: OPERATION INVOKE
    ---------------------------
    0xE7 0x00                  APDU CHOICE Type (PrstApdu)
    0x00 0x26                  CHOICE.length = 38
    0x00 0x24                  OCTET STRING.length = 36
    0x00 0x05                  invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
    0x01 0x07                  CHOICE(Remote Operation Invoke | Confirmed Action)
    0x00 0x1E                  CHOICE.length = 30
--------------------------------------------------------------------------------------
    0x00 0x03                  obj-handle = 3 (SCO  object)
B   0x0C 0x0B                  action-type = MDC_ACT_SCO_OP_INVOKE
    0x00 0x18                  action-info-args.length = 24
    0x00 0x00                  checksum = not-used(0)
    0xFF 0x33 0x44 0x55        invoke-cookie(should be random)
    0x00 0x01                  op-invoke-list.count
    0x00 0x0C                  op-invoke-list.size = 12
--------------------------------------------------------------------------------------
    0x00 0x2d                  op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
C   0x00 0x01                  op-instance-no = 1
    0x00 0x00                  op-mod-type->op-replace(0)
    0x09 0x32                  attribute-id = MDC_ATTR_INDEX_SEL(2354)
    0x00 0x02                  attribute-value.length = 2
    0x00 0x01                  Second element is selected (MDC_DIM_LB)
```

5.3.2 Results

1. Implementation conformance statements (ICSs)

In Table 5.3, a set of implementation conformance statements (ICSs) used for the implementation of this proposal are summarized. The first column shows an index that represents a specific feature as the specific codes in GEN-x (which specifies the versions or revisions supported by the implementation and some high-level system definitions), REQ-y, SRV-x (it defines which services that are defined in the service model are implemented), ATTR-n-a (the attributes implemented), NOTI-n-s (notification reports implemented), and ACT-n-t (action service implemented), POC-x (PHD object class implemented). In the second column are listed the implemented features and the reference to it is listed in the third column. Finally the last column indicates the feature status and qualifier.

Table 5.3: ICSs showing implementation conformance.

Index	Feature	Reference*	Status, Support
GEN-1	Implementation/Description	-	Extended configuration Weighing Scale. Configuration 1500
GEN-2	Std Doc Revision	-	IEEE 11073-20601-2010a
GEN-3	Conformance Adherence - Level 1	-	Yes
GEN-4	Conformance Adherence - Level 1	-	IEEE 11073-10415-2008
GEN-5	Communication Profile and hardware	-	Bluetooth/HDP
REQ-1	State Machine	-	Yes (20601)
REQ-2	Protocol Messages	-	Yes (20601)
REQ-3	Objects	-	Yes (20601)
REQ-4	Encoding	-	MDER
REQ-5	Nomenclature	-	Yes (10101)
REQ-6	Transport	-	Type-I compliance
SRV-1	GET Service	6.3.2.6.1[1]	Sends or accepts command
SRV-2	SET Service	8.4.2[2]	Sends or accepts command
SRV-3	Confirmed SET Service	8.4.2[2]	Sends or accepts command
SRV-4	EVENT REPORT Service	8.4.1[2]	Sends or accepts command
SRV-5	Confirmed EVENT REPORT Service	8.4.1[2]	Sends or accepts command
SRV-6	ACTION Service	8.4.4[2]	Sends or accepts command
SRV-7	Confirmed ACTION service	8.4.4[2]	Sends or accepts command
SRV-8	CREATE Service	8.4.5[1]	Sends or accepts command
POC-1	Simple_MDS: Weighing Scale	6.5.1[3]	Implemented

POC-2	Numeric: Body Weight	6.6.2[3]	Implemented
POC-3	Scanner	7.7.1[2]	Implemented
POC-4	UcfgScanner	7.7.6[2]	Implemented
POC-5	Operating scanner	7.7.9[2]	Implemented
POC-6	System Control Object (SCO)	7.6.1[2]	Implemented
POC-7	Operation	7.6.2[2]	Implemented
POC-OP-1[3]	Unit-Selector	7.6.3.1[1]	Implemented
ATTR-1-1	Handle	7.1.2.3[1]	Implemented
ATTR-1-2	System-Model	6.3.2.3[1]	Implemented/M, Static (1500)
ATTR-1-3	System-Id	6.3.2.3[1]	Implemented/M, Static (11:22:33:44:55:66:77:88)
ATTR-1-4	Dev-Configuration-Id	6.3.2.3[1]	Implemented/M, Static (U-WSBT-001)
ATTR-1-5	Date-and-Time	6.3.2.3[1]	Implemented/C, Observational
ATTR-1-6	System-Type-Spec-List	6.3.2.3[1]	Implemented/M, Static (MDC_DEV_SPEC_PROFILE_SCALE v1)
ATTR-2-1	Handle	7.1.2.3[1]	Implemented/M, Static (1)
ATTR-2-2	Type	6.3.3.3[1]	Implemented/M, Static (MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL)
ATTR-2-3	Metric-Spec-Small	6.3.3.3[1]	Implemented/M, Static (0xF040)
ATTR-2-4	Unit-Code	6.3.3.3[1]	Implemented/M, Static (MDC_DIM_KILO_G)
ATTR-2-5	Attribute-Value- Map	6.3.3.3[1]	Implemented/M, Static (MDC_ATTR_NU_VAL_OBS_SIMP 4, MDS_ATTR_TIME_STAMP_ABS 8).
ATTR-2-6	Simple-Nu-Observed-Value	6.3.4.3[1]	Implemented/C, Observational

ATTR-3-1	Handle	7.1.2.3[1]	Implemented/M
ATTR-3-2	Operational-State	6.3.9.3.3[1]	Implemented/M
ATTR-6-1	Handle	7.1.2.3[1]	Implemented/M, Static
ATTR-6-2	Sco-Capability	7.6.1.1[1]	Implemented/M, Static
ATTR-6-3	Activity- Indicator	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-6-4	Lock-State	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-6-5	Invoke-Cookie	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-OP-1-1[3]	Instance-Number	7.6.2.1[1]	Implemented/M
ATTR-OP-1-2[3]	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-1-3[3]	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-1-4[3]	Selected-Item-Index	7.6.3.1[1]	Implemented/M
ATTR-OP-1-5[3]	Select-List	7.6.3.1[1]	Implemented/M
ATC-0-1	Set-Time	6.3.2.4[1]	Confirmed
NOTI-0-1	MDS-Configuration-Event	6.3.2.5[1]	Confirmed
NOTI-0-2	MDS-Dynamic-Data-Update-Fixed	6.3.2.5[1]	Confirmed

(1) Section in X73-20601-2008 [19]

(2) Section in X73-10201-2004 [20]

(3) Section in X73-10415-2008 [90]

2. Footprint

Table 5.4, shows the size of each APDU. The middle of table shows the entire size in bytes using the command and control extension package, and the right side shows the byte size of the binary code of the current X73-10415. At the end of the table, you can see the full size of each sequence, and it is clear that the number of resulting bytes to implement the remote control to modify a parameter is higher.

Table 5.4: Comparison of the total size of sequence diagrams using remote control and the current version. The highlighted cells correspond to the new features.

APDU	Without remote control (bytes)	With remote control (bytes)
ASSOC|REQ	54	54
ASSOC|RESP	48	48
EVENT|REQ CONFIG	72	94
EVENT|REQ RESPONSE	26	26
EVENT|REQ: MDS DATA REPORT FIXED FORMAT (AGENT)	46	46
EVENT|RSP: MDS DATA REPORT FIXED FORMAT	22	22
SET|REQ OPERATIONAL STATE ON		26
SET|RESP OPERATIONAL STATE ON		24
EVENT|REQ OPERATING SCANNER CREATE		90
EVENT|RESP OPERATING SCANNER CREATE		22

ACTION|REQ OPERATION INVOKE		40
ACTION|RESP OPERATION INVOKE		24
EVENT|REQ OPERATING ATTRIBUTE UPDATE		60
EVENT|RESP OPERATING ATTRIBUTE UPDATE		22
GET REQ MDS ALL ATTRIBUTES	18	18
GET RSP MDS ALL ATTRIBUTES	92	92
SET|REQ: MDS TIME		32
SET|RSP: MDS TIME		30
SET|REQ: OPERATIONAL STATE OFF		26
SET|RSP: OPERATIONAL STATE OFF		22
ASSOC|REQ: RELEASE	6	6
ASSOC|RSP: RELEASE	6	6
TOTAL	**390**	**830**

5.4 Implementation ECG event recorder

5.4.1 X73 Model

A second test platform was developed for testing the command and control extension package. This platform consists of an agent that implements the X73-10406, and the remote control extension included in the DIM of the X73-20601.

The following subsections describe the parts that allow the implementation of this proof of concept.

A. Domain Information Model

For agent deployment with the X73-10406 profile, it has been proposed that a new domain information model be used that includes some classes of the current version of X73-20601 and some others that are part of the remote control proposal.

In Figure 5.13, the new classes are added to dim marked with draft label, while existing classes are exposed using a standard model classes. The following sections describe each of the classes in this implementation.

The classes that are present in the current version of X73-20601 are the following:

- Medical Device System (MDS) class: This class groups general information into attributes such as Handle, System model, System Id, Device configuration of mandatory type, and some other conditional attributes that help to represent correctly the MDS. It is only possible to instantiate an object of this class.

- Heart rate: This object is used to represent the heart rate measurement, and it defines the attributes Handle, Type, Metric-Spec-Small, Unit-Code and Attribute-Value-map as mandatory types. In the case of represent the instantaneous heart rate, the Type attribute shall be set to MDC_PART_PHD_DIM | MDC_ECG_HEART_RATE_INSTANT, but, if this represents an average value and the Type attribute shall be set to MDC_PART_SCADA|MDC_ECG_HEART_RATE_INSTANT.
  The Metric-Spec-Small attribute is used to define the modality of a

heart rate measurement. If the heart rate represents a beat-to-beat measurement, the mss-msmt-btb-metric as well as the mss-msmtaperiodic bits shall be set for the Metric-Spec-Small attribute. The Unite-Code attribute defines the unit of measurement in beats per minute with the nomenclature MDC_DIM_BEAT_PER_MIN.

- ECG waveform: The representation of ECG waveforms is made by this attribute. ECG wave forms are transmitted as a series of samples where each waveform is represented as a separate object. This object defines the attributes Handle, Type, Metric-Spec-Small, Unit-Code, Sample-Period, Simple-Sa-Observed-Value, Scale-and-Range-Specification, and Sa-Specification as mandatory type. The Type attribute for this implementation is defined as lead III with the nomenclature MDC_ECG_ELEC_POTL_III.

- Device status: It is a derivative of the enumeration class, this object is used to indicate the loss of the lead wire or the lead signal. Among the attributes that are required to be implemented is the Handle attribute, which represents a reference ID. This handler is also used to identify objects in the event reports sent to the manager. The Type attribute which mainly contains the nomenclature partition in this case, is the MDC_PART_PHD_DM, MDC_ECG_DEV_STAT, and the Metric-Spec-Small, which describes the structure of the measurement and the Enum-Observed-Value-Basic-Bit-Str, which defines a set of 8 possible strings that could indicate the type of failure that causes the loss of signal.

- Context data trigger: The Context Data Trigger enumeration object informs the user of the reason for transmitted segments of ECG waveform, heart rate, or R-R interval measurement data, such as a user pressing a button or an automatic event due to a data analysis mechanism in the agent (e.g., in case of detection of a cardiac event).

- Scanner: It is an abstract base class for the unconfigurable scanner class. It provides a flow control mechanism that allows the manager start and stop agent reporting functions.

The new classes defined for this implementation are the following;

- Operating scanner: It reports the information specifically about the operations implemented in the PHDs. Only a single object of this class is needed for this implementation.

- Unconfigurable scanner (UcfgScanner): It is an abstract base class for the operating scanner class. It represents an observer that monitors the DIM and reports its changes.

- Service and Control Object (SCO) class: It provides mechanisms as Simple transaction processing to modify and operate with operation objects. With this class, the inconsistencies when a device is controlled from multiple access points (e.g., local and remote) and during the processing of control commands are prevented. The mandatory attributes for this class are Handle, Sco-capability, and Activity-Indicator to give feedback that the system is under remote control; Lock-State, used to prevent unwanted actions, and if locked, no operation can be invoked; and Invoke-Cookie to Transaction ID, assigned by invoke. It is only possible to instantiate an object of this class.

- Operation: It is an abstract class. This class represents a single remote controllable item. Its attributes contain information that allows the manager to correctly interpret its purpose. The mandatory attributes for this class are Instance-Number, and it is used for operation identification, Operation-spec, which provides a structure defining operation types and properties, and Operational-State, used to specify whether an operation is accessible. This class is not directly accessible, but it can be controlled by the SCO and observed using the Operating scanner.

- Select item: This operation can be used to select any element of a list. The selectable values are given as a list in the Select-Item-List attribute, whose implementation is the mandatory type, as well as the Index attribute. In this implementation, there is a single object of this class, which enables changing the unit of measurement (kilograms or pounds).

- Set Value: This operation is used to change or modify any attribute value of float type which needs to be within the predefined range.

- Activate: This operation can be used to start a manual event.

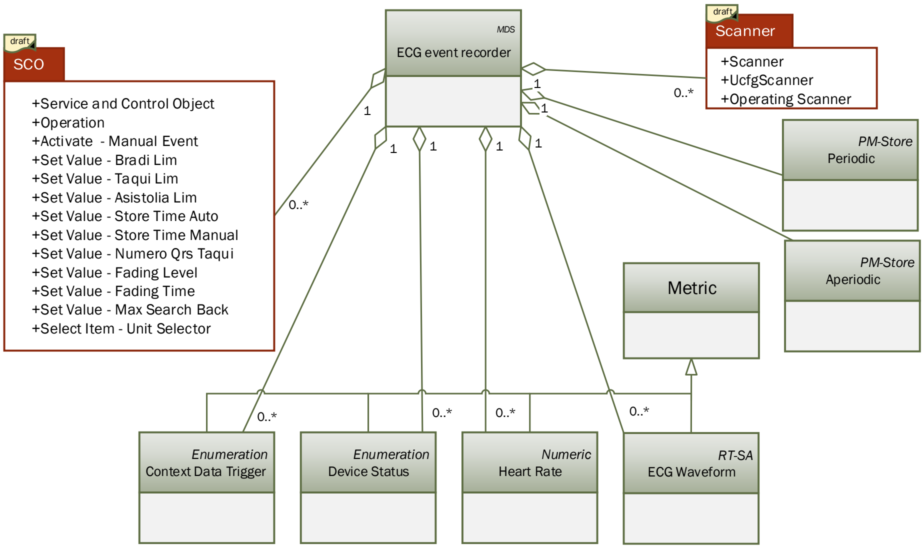


Figure 5.13: ECG Domain Information model.

All classes that are implemented by the command and control extension package, use a number of attributes which are implemented in the PHDs according to the necessary requirements. The attributes required for the ECG event recorder based on the profile X73-10406 are exposed on the Table 5.5.

Table 5.5: DIM Class.

Object	**Class**	**Attributes**	**Types, respectively ({} = possible values, [] = members)**
MDS	MDS	Handle, System-Type-Spec-List, System-Model, System-Id, Dev-Configuration-Id.	INT-U16, SEQUENCE[type, version], SEQUENCE[manufacturer, model-number], OCTET STRING, INT-U16 { manager-config-response(0),standard-config-start(1),standard-config-end(16383), extended-config-start(16384),extended-config-end(32767),reserved-start(32768), reserved-end(65535)},
Context Data Trigger	Enumeration	Handle, Type, Metric-Spec-Small, Enum-Observed-Value-Simple-OID.	INT-U16, SEQUENCE[partition, code], BITS-16{ mss-avail-intermittent(0), mss-avail-stored-data(1), mss-upd-aperiodic(2), mss-msmt-aperiodic(3), mss-msmt-phys-ev-id(4),mss-msmt-btb-metric(5), mss-acc-manager-initiated (8), mss-acc-agent-initiated(9), mss-cat-manual(12), mss-cat-setting(13), mss-cat-calculation(14), mss-cat-configurable(15)}, INT-U16.
Device Status	Enumeration	Handle, Type, Metric-Spec-Small, Enum-Observed-Value-Basic-Bit-Str.	INT-U16, SEQUENCE [partition, code], BITS-16{ mss-avail-intermittent(0), mss-avail-stored-data(1), mss-upd-aperiodic(2), mss-msmt-aperiodic(3), mss-msmt-phys-ev-id(4),mss-msmt-btb-metric(5), mss-acc-manager-initiated (8), mss-acc-agent-initiated(9), mss-cat-manual(12), mss-cat-setting(13), mss-cat-calculation(14), mss-cat-configurable(15)}, BITS-16.

Heart Rate	Numeric	Handle, Metric-Spec-Small, Unit-Code, Instance-Number, Operation-Spec, Operational-State.	INT-U16, BITS-16{ mss-avail-intermittent(0), mss-avail-stored-data(1), mss-upd-aperiodic(2), mss-msmt-aperiodic(3), mss-msmt-phys-ev-id(4),mss-msmt-btb-metric(5), mss-acc-manager-initiated (8), mss-acc-agent-initiated(9), mss-cat-manual(12), mss-cat-setting(13), mss-cat-calculation(14), mss-cat-configurable(15)}, INT-U16, INT-U16, SEQUENCE [vattr-id, op-target, options, level, grouping], INT-U16 {disabled(0), enabled(1), notAvailable(2)}
ECG Waveform	RT-SA	Handle, Type, Metric-Spec-Small, Unit-Code,	INT-U16, BITS-16{ mss-avail-intermittent(0), mss-avail-stored-data(1), mss-upd-aperiodic(2), mss-msmt-aperiodic(3), mss-msmt-phys-ev-id(4),mss-msmt-btb-metric(5), mss-acc-manager-initiated (8), mss-acc-agent-initiated(9), mss-cat-manual(12), mss-cat-setting(13), mss-cat-calculation(14), mss-cat-configurable(15)},
Service and Control	Service and Control Object	Handle, Sco-Capability, Sco-Help-Text- String, Vmo-Reference, Activity- Indicator, Lock-State, Invoke-Cookie.	INT-U16, BITS-16 {act-indicator(0), sco-locks(1), sco-ctxt-help(8)}, OCTET STRING, INT-U16, INT-U16 {act-ind-off(0), act-ind-on(1), act-ind-blinking(2)}, INT-U16 {locked(0), unlocked(1), shuttingDown(2)}, INT-U32
Unit Selector	Operation	Instance-Number, Operation-Spec, Operation-Text-Strings.	INT-U16, SEQUENCE[vattr-id, op-target, options, level, grouping], SEQUENCE[label, help, confirm-message]
	Select Item	Selected-Item-Index, Nom-Partition, Select-List.	INT-U16, INT-U16{ nom-part-unspec(0),nom-part-obj(1), nom-part-metric(2), nom-part-alert(3),nom-part-dim(4), nom-part-vattr(5), nom-part-pgrp(6),nom-part-sites(7), nom-part-infrastruct(8), nom-part-fef(9),nom-part-ecg-extn(10), nom-part-idco-extn(11), nom-part-phd-dm(128), nom-part-phd-hf(129), nom-part-phd-ai(130), nom-part-ret-code(255), nom-part-ext-nom(256), nom-part-priv(1024)}, CHOICE{ oid-list, value-list, value-u-list, string-list}.
Bradi Lim	Operation	Instance-Number, Operation-	INT-U16, SEQUENCE[vattr-id, op-target, options, level, grouping], INT-

Taqui Lim Asistolia Lim Store-Time-Auto Store-Time-Manual Numero-Qrs-Taqui Max-Search-Back Fading Time Fading Level		Spec, Operational-State, Vmo-Reference.	U16 {disabled(0), enabled(1), notAvailable(2)}, INT-U16.
	Set Value	Current-Value, Set-Value-Range, Unit Code.	INT-U32, SEQUENCE[minimum, maximum, resolution], INT-U16.
Manual Event	Operation	Instance-Number, Operation-Spec, Operational-State, Vmo-Reference.	INT-U16, SEQUENCE[vattr-id, op-target, options, level, grouping], INT-U16 {disabled(0), enabled(1), notAvailable(2)}, INT-U16.
	Activate		
Operating Scanner	UcfgScanner	Confirm-Mode,Confirm-Timeout,Transmit-Window.	INT-U16 {unconfirmed(0) or confirmed(1)} , INT-U32, INT-U16.
	Operating Scanner	-	-
Periodic	PM-store	Handle, PM-Store-Capab, Store-Sample-Algorithm, Operational-State, Number-Of-Segments, Clear-Timeout.	INT-U16, BITS-16 {pmsc-var-no-of-segm(0),pmsc-segm-id-list-select(3), pmsc-epi-seg-entries(4),pmsc-peri-seg-entries(5),pmsc-abs-time-select(6), pmsc-clear-segm-by-list-sup(7), pmsc-clear-segm-by-time-sup(8), pmsc-clear-segm-remove(9), pmsc-clear-segm-all-sup(10), pmsc-multi-person(12)}, INT-U16 {st-alg-nos(0), st-alg-moving-average(1),st-alg-recursive(2),st-alg-min-pick(3), st-alg-max-pick(4),st-alg-median(5),st-alg-trended(512),st-alg-no-downsampling(1024), st-alg-manuf-specific-start(61440),st-alg-manuf-specific-end(65535)}, INT-U16 {disabled(0), enabled(1), INT-U16, INT-U32.

Aperiodic	PM-store	Handle, PM-Store-Capab, Store-Sample-Algorithm, Operational-State, Number-Of-Segments, Clear-Timeout.	INT-U16, BITS-16 {pmsc-var-no-of-segm(0),pmsc-segm-id-list-select(3), pmsc-epi-seg-entries(4),pmsc-peri-seg-entries(5),pmsc-abs-time-select(6), pmsc-clear-segm-by-list-sup(7), pmsc-clear-segm-by-time-sup(8), pmsc-clear-segm-remove(9), pmsc-clear-segm-all-sup(10), pmsc-multi-person(12)}, INT-U16 {st-alg-nos(0), st-alg-moving-average(1),st-alg-recursive(2),st-alg-min-pick(3), st-alg-max-pick(4),st-alg-median(5),st-alg-trended(512),st-alg-no-downsampling(1024), st-alg-manuf-specific-start(61440),st-alg-manuf-specific-end(65535)}, INT-U16 {disabled(0), enabled(1), INT-U16, INT-U32.

B. Basic Electrocardiograph Service Model

The services implemented in the Basic Electrocardiograph are exposed on the table 5.6 and described below:

MDS

- GET: This service allows the manager to retrieve the value of attributes of the MDS object in the agent X73-10415.

- Set-Time method: This method allows the manager to set the date and time in the agent.

- MDS-Configuration-Event: This event is used to create a Configuration report to inform the DIM of the agent.

- MDS-Dynamic-Data-Update-Var: This event provides dynamic data (typically measurements) from the agent for some or all of the objects that the agent supports. The event can be triggered by an MDS-Data-Request from the manager system, or it is sent as an unsolicited message by the agent.

- MDS-Dynamic-Data-Update-Fixed: This event is used to create a Data report to provide dynamic data to manager for some or all of the agents' objects in fixed format.

- MDS-Dynamic-Data-Update-MP-Var: This event provides dynamic data (related to measurements) from the agent for some or all of the objects that the agent supports, but also allows inclusion of data from multiple persons.

- MDS-Dynamic-Data-Update-MP-Fixed: This event provides dynamic data (typically measurements) from the agent for some or all of the metric objects or the MDS object that the agent supports.

- Set-Time: This operation allows to setting the date and time of the agent.

- Set-Base-Offset-Time: This action is used to select the date and time to be set in base offset time format.

PM-Store:

- Get: The manager uses the GET service to retrieve the values of all PM-store object attributes.

- Segment-Data-Event: The Manager uses this event to send data stored in the Fixed-Segment-Data of a PM-segment to the agent.

- Clear-Segments: This method allows the manager to delete the data currently stored in one or more selected PM-segments.

- Get-Segment-Info: This method is used particularly to retrieve the attributes and their data contents from the PM-segment object instances identified by the SegmSelection parameter.

- Trig-Segment-Data-Xfer: This method allows the manager to start the transfer of the Fixed-Segment-Data attribute of a specified PM-segment.

Scanner:

- Set: Enables the capability to generate event reports.

Operating:

- Oper-Create-Notification: Configuration Report to inform the manager of the configuration of the agent

- Oper-Attribute-Update: Confirmed event report with the updates on the value of the attributes of an object.

- Oper-Create-Notification: This action allows creating a confirmed event report with the remote control features enabled in the agent.

SCO:

- Operation-Invoke: Executes an operation in a specific attribute of an object.

- Get-Ctxt-Help: Executes a request about context information of an operation.

- SCO-Operating-Request: Confirmed event report to request an operation in the agent.

- SCO-Operation-Invoke-Error: Confirmed event report to inform the manager about an error generated by an incorrect operational request.

Table 5.6: ECG event recorder services.

Object	**Service**	**Type**	**Parameter**	**Results**
MDS	GET	-	GetArgument Simple	GetResult Simple
	EVENT	MDS-Configuration-Event	ConfigReport	ConfigReport Rsp
		MDS-Dynamic-Data-Update-Var	ScanReportInfoVar	
		MDS-Dynamic-Data-Update-Fixed	ScanReportInfoFixed	-
		MDS-Dynamic-Data-Update-MP-Var	ScanReportInfoMPVar	
		MDS-Dynamic-Data-Update-MP-Fixed	ScanReportInfoMPFixed	
	ACTION	Set-Time	SetTimeInvoke	
		Set-Base-Offset-Time	SetBOTimeInvoke	
PM-Store	GET		GetArgumentSimple	GetResultSimple
	EVENT	Segment-Data-Event	SegmentDataEvent	SegmentDataResult
	ACTION	Clear-Segments	SegmSelection	-

		Get-Segment-Info	SegmSelection	SegmentInfoList
		Trig-Segment-Data-Xfer	TrigSegmDataXfer-Req	TrigSegmDataXferRsp
Scanner	SET	-	OperationalState	SetArgumentSimple
Operating Scanner	EVENT	Oper-Create-Notification	OpCreateInfo	ConfigReportRsp
		Oper-Attribute-Update	OpAttributeInfo	-
	ACTION	Oper-Create-Notification	OperationInvoke	-
SCO	ACTION	Operation-Invoke	OperationInvoke	OperationInvokeResult
		Get-Ctxt-Help	CtxtHelpRequest	CtxtHelpResult
	EVENT	SCO-Operating-Request	ScoOperReqSpec(optional)	-
		SCO-Operation-Invoke-Error	ScoOperInvokeError	-

upna
Universidad
Pública de Navarra
Nafarroako
Unibertsitate Publikoa

C. Basic Electrocardiograph Communication Model

1. Finite State Machine

Previously, a FSM of remote control standard model was presented; this model defines the guidelines to develop a remote control operation. In general terms this FSM is equal in any implementation, and the changes are presented only when different operations are invoked, this is due to each operation having its own arguments. In the Basic Electrocardiograph the implementation the RC FSM can be observed at Figure 5.14.

In this implementation, the manager will try to modify an attribute group of the ECG agent, using a remote control operation. The remote control process starts when the manager device requests to enable the Operational State due the RC FSM (Remote control Finite State Machine) initially begins in the Disabled state. It can be observed through the operational-state attribute of Operational Scanner. This operation takes place when the manager sends to the Operating Scanner a request to set the Operational-State attribute to enabled (RxSetOnReq). Later, the Operating Scanner responds, acknowledging the state change (TxSetOnRsp). At this point the FSM goes from the not initialized state to the initialized state automatically. Then the FSM goes to the Operational On State and the Operating Scanner sends the manager a report (TxCreateReq) which gathers information about every Operation instance that the agent implements. Specifically for the X73-10406 implementation, this report shows that SCO implements 9 Set value and 1 Select Item operation object within SCO. Initially, these objects would contain attributes with an initial value that allows to the agent go to operative state when turned on.

Later, when the agent receives the response from the manager (RxCreateResponse), the FSM goes to the Ready state. In this state, the manager can invoke a method to execute operations in the SCO, and for this proof of concept, the manager invokes 10 operations, 9 for Set value instances and 1 for Select item instance. This is performed in order to modify the operational attributes in the agent. Subsequently, the SCO acknowledges the invocation (TxInvokeRsp) and the agent goes to the Executing state. The next step starts when the SCO executes the invocation requested and, when it is done, the Operating Scanner sends the manager a report with the attribute changes (TxUpdateReq) to synchronize the manager. This report reflects the changes at all the attributes. When the manager acknowledges the reception of attribute changes (RxUpdateRsp), the FSM goes from the Synchronizing state to the Ready state.

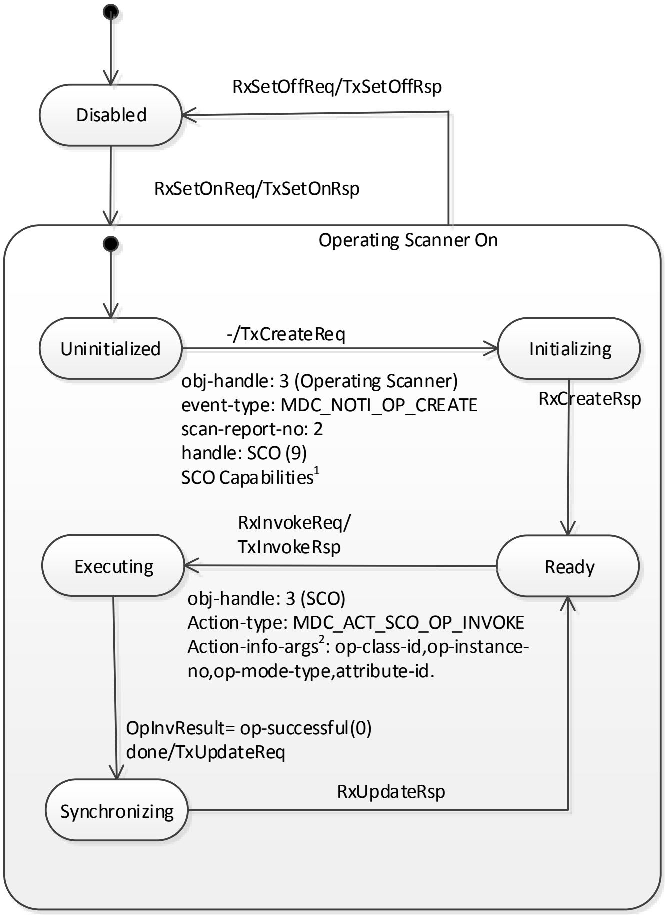


Figure 5.14: Finite state machine.

2. Sequence Diagram

As has been exposed above, the start of an operation of remote control begins to enable the operational-state attribute. In the following sequence diagram exposed at Figure 5.15, it can be observed that the manager invokes the set service to enable the operational-state attribute from Operating Scanner on the agent X73-10406.

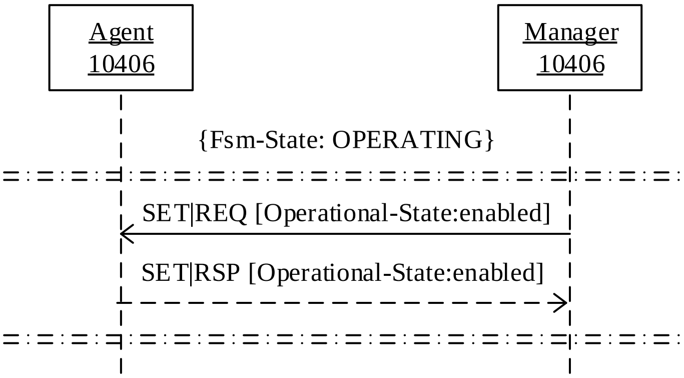

Agent 10406
Manager 10406
{Fsm-State: OPERATING}
SET|REQ [Operational-State:enabled]
SET|RSP [Operational-State:enabled]

Figure 5.15: Enable Operating Scanner by setting as enabled.

After the previous step, the agent is capable of generating event reports which may include the features and configuration of the remote control features implemented in the agent. In the following Figure 5.16, the agent sends an event report which includes all the capabilities implemented at the Agent X73-10406. In the text box which symbolizes the message that the agent sends to the manager, it can be seen that the agent has implemented two types of operations, Select Value and Select Item. For the first operation, there are nine instances and for the second only one instance is necessary.

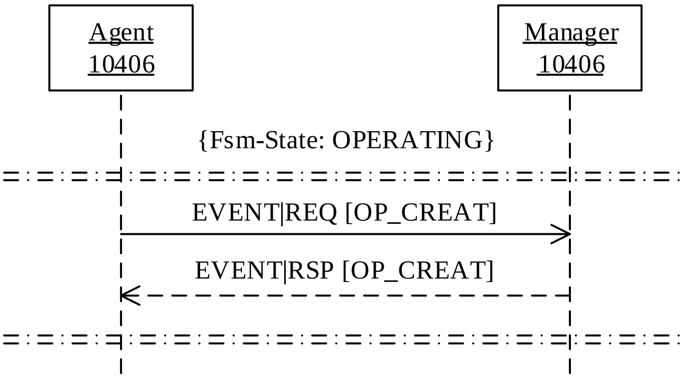


Figure 5.16: Event Report, Operating Scanner Create.

The next Figure 5.17 shows the sequence diagram corresponding to operation invoke from the manager. In this case the manager requests to modify 10 instances of operation class implemented at the agent. Nine of these instances are Set Value operation and 1 is of Select Item type. As a result, the response message includes the argument op-successful (0) if the operation is developing successfully.

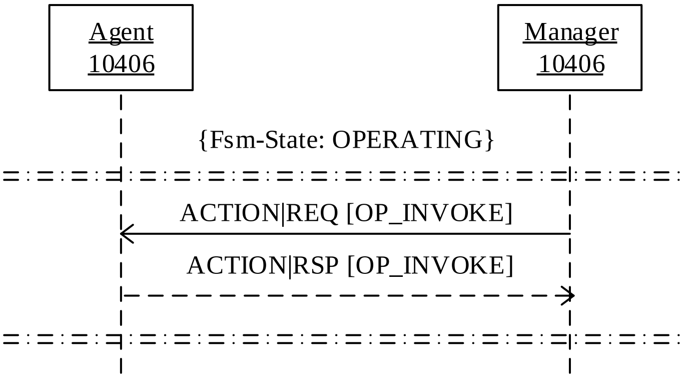


Figure 5.17: Action Request, Operation Invoke.

To conclude the remote operation, an operation attribute update event may be used. The last sequence diagram exposed at Figure 5.18 shows this message. The aim of the operation attribute update event is to visualize if the changes made by the operation of the remote control in the agent are correct. At the message can be seen the new values of the attributes modified by the remote control operation.

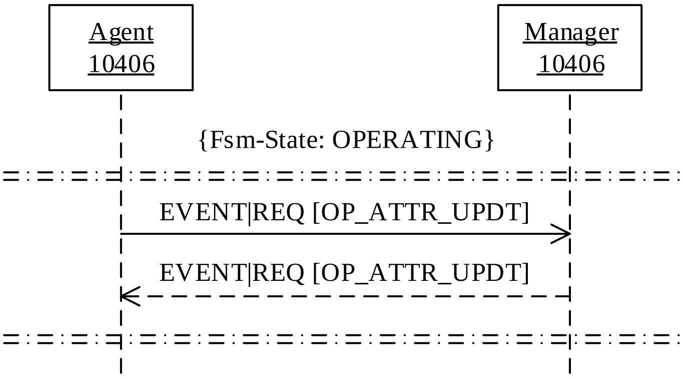


Figure 5.18: Event request, Operation attribute update.

As discussed above, the sequence diagram of this implementation shows the minimum requirements necessary to perform a remote control operation. With the exception of the update event attributes, the first three more messages are always needed to successfully perform a remote control operation.

3. Basic Electrocardiograph MDER Encoded Binary Messages.

Listing 5.6 shows the coding MDER segment which corresponds to the event that generates a report of the configuration implemented in the agent. In section A, it can be observed that it is the MDS instance that generates the MDC_NOTI_CONFIG event. This report indicates that it contains an extended configuration and also indicates that it contains one object associated with measurement features implemented in the agent. This object is an instance of the Metric class and it corresponds to Heart Rate object as can be observed in section B. The next sections of the MDER code as the segment C, D and E show the attributes implemented within the Heart Rate object. The segment B shows the handle of the Heart Rate object. It contains four attributes, the MDC_ATTR_ID_TYPE attribute in the section C, and is used to indicate de partition of this type of object. The segment in the section D contains the attribute MDC_ATTR_METRIC_SPEC_SMALL, which describes the characteristics of the measurement. The segment in the section E defines MDC_ATTR_UNIT_CODE, which defines the unit of the measurement, and the last segment on the section F defines the attributes that are reported in the fixed format data update messages.

Listing 5.6: Event request config.

```
     EVENT|REQ: CONFIG (Agente a Manager)
     ------------------------------
A    0xE7 0x00                  APDU CHOICE Type (PrstApdu)
     0x01 0x2A                  CHOICE.length = 298
     0x01 0x28                  OCTET STRING.length = 296
     0x00 0x01                  invoke-id = 0x0001 (start of DataApdu. MDER encoded.)
     0x01 0x01                  CHOICE(Remote Operation Invoke | Confirmed Event Report)
     0x01 0x20                  CHOICE.length = 288
     0x00 0x00                  obj-handle = 0 (MDS object)
     0xFF 0xFF 0xFF 0xFF        event-time = agent does not have a RelativeTime clock
     0x0D 0x1C                  event-type = MDC_NOTI_CONFIG
     0x01 0x16                  event-info.length = 278 (start of ConfigReport)
     0x40 0x00                  dev-config-id = extended-config-start (16384)
     0x00 0x00                  obj-handle = 0 (MDS object)
     0xFF 0xFF 0xFF 0xFF        event-time = agent does not have a RelativeTime clock
     0x0D 0x1C                  event-type = MDC_NOTI_CONFIG
     0x01 0x0A                  event-info.length = 266 (start of ConfigReport)
     0x40 0x00                  dev-config-id = extended-config-start (16384)
     0x00 0x08                  config-obj-list.count = 8 objects will be “announced”
     0x01 0x04                  config-obj-list.length = 260
-------------------------------------------------------------------------------------------
     0x00 0x06                  obj-class = MDC_MOC_VMO_METRIC_NU
B    0x00 0x01                  obj-handle = 1 ( 1st Measurement is heart rate)
     0x00 0x04                  attributes.count = 4
     0x00 0x24                  attributes.length = 36
-------------------------------------------------------------------------------------------
     0x09 0x2F                  attribute-id = MDC_ATTR_ID_TYPE
C    0x00 0x04                  attribute-value.length = 4
     0x00 0x02 0x4B 0x5C        MDC_PART_SCADA | MDC_ ECG_HEART_RATE
-------------------------------------------------------------------------------------------
     0x0A 0x46                   attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
D    0x00 0x02                  attribute-value.length = 2
     0x02 0x02                  stored data, agent init
-------------------------------------------------------------------------------------------
     0x09 0x96                  attribute-id = MDC_ATTR_UNIT_CODE
E    0x00 0x02                  attribute-value.length = 2
     0x0A 0xA0                  MDC_DIM_BEAT_PER_MIN
-------------------------------------------------------------------------------------------
     0x0A 0x55                  attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
F    0x00 0x08                  attribute-value.length = 12
     0x00 0x02                  AttrValMap.count = 2
     0x00 0x08                  AttrValMap.length = 8
     0x0A 0x4C 0x00 0x02        MDC_ATTR_NU_VAL_OBS_BASIC, 2
     0x09 0x91 0x00 0x04        MDC_ATTR_TIME_STAMP_REL, 4
```

Listing 5.7 shows an APDU of a remote control operation. As can be seen the overall size of the APDU does not correspond to the coding of the figure, and this is due to that only an operation has been selected to illustrate the development of a remote control operation in a specific instance. The segment 6.5 of the figure contains the coding where the operation invoke request develops. And finally on the segment 6.5c can be seen the code of Select-Value operation. This operation is integrated mainly by operation instance, the type of modifier of attribute and the attribute-id which will be modified.

Listing 5.7: Action Request Operation Invoke.

```
A    ACTION|REQ: OPERATION INVOKE
     ---------------------------
     0xE7 0x00                  APDU CHOICE Type (PrstApdu)
     0x00 0x90                  CHOICE.length = 144
     0x00 0x8E                  OCTET STRING.length = 142
     0x00 0x05                  invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
     0x01 0x07                  CHOICE(Remote Operation Invoke | Confirmed Action)
     0x00 0x88                  CHOICE.length = 136
     ---------------------------------------------------------------------------------
     0x00 0x03                  obj-handle = 3 (SCO  object)
B    0x0C 0x0B                  action-type = MDC_ACT_SCO_OP_INVOKE
     0x00 0x82                  action-info-args.length = 130
     0x00 0x00                  checksum = not-used(0)
     0xFF 0x33 0x44 0x55        invoke-cookie(should be random)
     0x00 0x01                  op-invoke-list.count
     0x00 0x78                  op-invoke-list.size = 120
     ---------------------------------------------------------------------------------
     0x00 0x2F                  op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
C    0x00 0x01                  op-instance-no = 1
     0x00 0x00                  op-mod-type->op-replace(0)
     0x00 0x01                  attributes.count
     0x00 0x08                  attributes.length =8
     0x09 0x9D                  attribute-id = MDC_ATTR_VAL_CURR(2461)
     0x00 0x04                  attribute-value.length = 4
     0x00 0x00 0x00 0x28        new Bradycardia-Limit value= 40
```

5.4.2 Results

1. Implementation conformance statements (ICSs)

In Table 5.7, a set of implementation conformance statements (ICSs) used for the implementation of this proposal are summarized. The first column shows an index that represents a specific feature as the specific codes GEN-x (specifies the versions or revisions supported by the implementation and some high-level system definitions), REQ-y, SRV-x (It defines which services that are defined in the service model are implemented), ATTR-n-a (The attributes implemented), NOTI-n-s (Notification reports implemented), and ACT-n-t (Action service implemented),POC-x (PHD object class implemented). In the second column are listed the implemented features and the reference to it is listed in the third column. Finally the last column indicates the feature status and qualifier.

Table 5.7: ICSs showing implementation conformance.

Index	**Feature**	**Reference***	**Status, Support**
GEN-1	Implementation/Description	-	Extended configuration
GEN-2	Std Doc Revision	-	IEEE 11073-20601-2010a
GEN-3	Conformance Adherence - Level 1	-	Yes
GEN-4	Conformance Adherence - Level 1	-	IEEE 11073-10406-2011
GEN-5	Communication Profile and hardware	-	Bluetooth/HDP
REQ-1	State Machine	-	Yes (20601)
REQ-2	Protocol Messages	-	Yes (20601)
REQ-3	Objects	-	Yes (20601)
REQ-4	Encoding	-	MDER
REQ-5	Nomenclature	-	Yes (10101)
REQ-6	Transport	-	Type-I compliance
SRV-1	GET Service	6.3.2.6.1[1]	Sends or accepts command
SRV-2	SET Service	8.4.2[2]	Sends or accepts command
SRV-3	Confirmed SET Service	8.4.2[2]	Sends or accepts command
SRV-4	EVENT REPORT Service	8.4.1[2]	Sends or accepts command
SRV-5	Confirmed EVENT REPORT Service	8.4.1[2]	Sends or accepts command
SRV-6	ACTION Service	8.4.4[2]	Sends or accepts command
SRV-7	Confirmed ACTION service	8.4.4[2]	Sends or accepts command
SRV-8	CREATE Service	8.4.5[1]	Sends or accepts command
POC-1	Simple_MDS: Basic Electrocardiograph (1 to 3 leads)	6.5.1[3]	Implemented
POC-2	Enumeration: Device Status	6.9.2[3]	Implemented
POC-3	Enumeration: Context Data Trigger	6.9.3[3]	Implemented

POC-4	Numeric: Heart Rate	6.7.2[3]	Implemented
POC-5	RT-SA: Ecg Waveform	6.8.2[3]	Implemented
POC-6	PM-Store: Periodic	6.10[3]	Implemented
POC-7	PM-Store: APeriodic	6.10[3]	Implemented
POC-8	Scanner	7.7.1[2]	Implemented
POC-9	UcfgScanner	7.7.6[2]	Implemented
POC-10	Operating scanner	7.7.9[2]	Implemented
POC-11	System Control Object (SCO)	7.6.1[2]	Implemented
POC-12	Operation	7.6.2	Implemented
POC-OP-1[4]	Manual-Event	7.6.7[1]	Implemented
POC-OP-2[4]	Bradi-Lim	7.6.4[1]	Implemented
POC-OP-3[4]	Taqui-Lim	7.6.4[1]	Implemented
POC-OP-4[4]	Asistolia-Lim	7.6.4[1]	Implemented
POC-OP-5[4]	Store-Time-Auto	7.6.4[1]	Implemented
POC-OP-6[4]	Store-Time-Manual	7.6.4[1]	Implemented
POC-OP-7[4]	Qrs-Taqui-Number	7.6.4[1]	Implemented
POC-OP-8[4]	Max-Search-Back	7.6.4[1]	Implemented
POC-OP-9[4]	Fading-Time	7.6.4[1]	Implemented
POC-OP-10[4]	Fading-Level	7.6.4[1]	Implemented
POC-OP-11[4]	Digital-Gaing	7.6.3[1]	Implemented
ATTR-1-1	Handle	7.1.2.3[1]	Implemented
ATTR-1-2	System-Model	6.3.2.3[1]	Implemented/M, Static
ATTR-1-3	System-Id	6.3.2.3[1]	Implemented/M, static (11:22:33:44:55:66:77:88)
ATTR-1-4	Dev-Configuration-Id	6.3.2.3[1]	Implemented/M, Static
ATTR-1-5	Date-and-Time	6.3.2.3[1]	Implemented/C, Observational
ATTR-1-6	System-Type-Spec-List	6.3.2.3[1]	Implemented/M, static (MDC_DEV_SPEC_PROFILE_ECG, 1)

ATTR-2-1	Handle	7.1.2.3[1]	Implemented/M, Dynamic
ATTR-2-2	Type	6.3.3.3[1]	Implemented/M, Static
ATTR-2-3	Metric-Spec-Small	6.3.3.3[1]	Implemented/M, Dynamic
ATTR-2-4	Enum-Observed-Value-Basic-Bit-Str	6.3.6.3[1]	Implemented/C, Observational
ATTR-3-1	Handle	7.1.2.3[1]	Implemented/M, Dynamic
ATTR-3-2	Type	6.3.3.3[1]	Implemented/M, Static
ATTR-3-3	Metric-Spec-Small	6.3.3.3[1]	Implemented/M, Dynamic
ATTR-3-4	Enum-Observed-Value-Simple-OID	6.3.6.3[1]	Implemented/C, Observational
ATTR-4-1	Handle	7.1.2.3[1]	Implemented/M, Dynamic
ATTR-4-2	Type	6.3.3.3[1]	Implemented/M, Static (MDC_PART_SCADA |, MDC_ECG_HEART_RATE)
ATTR-4-3	Metric-Spec-Small	6.3.3.3[1]	Implemented/M, Dynamic
ATTR-4-4	Unit-Code	6.3.3.3[1]	Implemented/M, Static (MDC_DIM_BEAT_PER_MIN)
ATTR-4-5	Instance-Number	7.6.2.1[1]	Implemented/M
ATTR-4-6	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-4-7	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-5-1	Handle	7.1.2.3[1]	Implemented/M, Dynamic
ATTR-5-2	Type	6.3.3.3[1]	Implemented/M, Static (MDC_ECG_ELEC_POTL_III)
ATTR-5-3	Metric-Spec-Small	6.3.3.3[1]	Implemented/M, Dynamic
ATTR-5-4	Unit-Code	6.3.3.3[1]	Implemented/M, Static
ATTR-5-5	Sample-Period	6.3.5.3[1]	Implemented/M, Static
ATTR-5-6	Simple-Sa-Observed-Value	6.3.5.3[1]	Implemented/M, Observational
ATTR-5-7	Scale-and-Range-Specification	6.3.5.3[1]	Implemented/M, Static

ATTR-5-8	Sa-Specification	6.3.5.3[1]	Implemented/M, Static
ATTR-6-1	Handle	7.1.2.3[1]	Implemented/M
ATTR-6-2	PM-Store-Capab	6.3.7.3[1]	Implemented/M
ATTR-6-3	Store-Sample-Algorithm	6.3.7.3[1]	Implemented/M
ATTR-6-4	Operational-State	6.3.7.3[1]	Implemented/M, Dynamic
ATTR-6-5	Number-of-Segments	6.3.7.3[1]	Implemented/M, Dynamic
ATTR-7-1	Handle	7.1.2.3[1]	Implemented/M
ATTR-7-2	PM-Store-Capab	6.3.7.3[1]	Implemented/M
ATTR-7-3	Store-Sample-Algorithm	6.3.7.3[1]	Implemented/M
ATTR-7-4	Operational-State	6.3.7.3[1]	Implemented/M, Dynamic
ATTR-7-5	Number-of-Segments	6.3.7.3[1]	Implemented/M, Dynamic
ATTR-8-1	Handle	7.1.2.3[1]	Implemented/M
ATTR-8-2	Operational-State	6.3.9.3.3[1]	Implemented/M
ATTR-11-1	Handle	7.1.2.3[1]	Implemented/M, Static
ATTR-11-2	Sco-Capability	7.6.1.1[1]	Implemented/M, Static
ATTR-11-3	Activity- Indicator	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-11-4	Lock-State	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-11-5	Invoke-Cookie	7.6.1.1[1]	Implemented/M, Dynamic
ATTR-OP-1-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =11)
ATTR-OP-1-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-1-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-2-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =1)
ATTR-OP-2-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-2-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-2-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-2-5	Current-Value	7.6.4.1[1]	Implemented/M

			(Bradycardia-Limit)
ATTR-OP-2-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-3-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =2)
ATTR-OP-3-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-3-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-3-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-3-5	Current-Value	7.6.4.1[1]	Implemented/M (Taquicardia-Limit)
ATTR-OP-3-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-4-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =3)
ATTR-OP-4-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-4-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-4-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-4-5	Current-Value	7.6.4.1[1]	Implemented/M (Asistolia-Lim)
ATTR-OP-4-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-5-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =4)
ATTR-OP-5-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-5-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-5-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-5-5	Current-Value	7.6.4.1[1]	Implemented/M (Store-Time-Auto)
ATTR-OP-5-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-6-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =5)
ATTR-OP-6-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-6-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-6-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-6-5	Current-Value	7.6.4.1[1]	Implemented/M

			(Store-Time-Manual)
ATTR-OP-6-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-7-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =6)
ATTR-OP-7-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-7-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-7-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-7-5	Current-Value	7.6.4.1[1]	Implemented/M (Qrs-Taqui-Number)
ATTR-OP-7-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-8-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =7)
ATTR-OP-8-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-8-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-8-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-8-5	Current-Value	7.6.4.1[1]	Implemented/M (Max-Search-Back)
ATTR-OP-8-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-9-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =8)
ATTR-OP-9-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-9-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-9-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-9-5	Current-Value	7.6.4.1[1]	Implemented/M (Fading-Time)
ATTR-OP-9-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-10-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =9)
ATTR-OP-10-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-10-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-10-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-10-5	Current-Value	7.6.4.1[1]	Implemented/M

			(Fading-Level)
ATTR-OP-10-6	Set-Value-Range	7.6.4.1[1]	Implemented/M
ATTR-OP-11-1	Instance-Number	7.6.2.1[1]	Implemented/M (op-instance-no =10)
ATTR-OP-11-2	Operation-Spec	7.6.2.1[1]	Implemented/M
ATTR-OP-11-3	Operational-State	7.6.2.1[1]	Implemented/M
ATTR-OP-11-4	Vmo-Reference	7.6.2.1[1]	Implemented/M
ATTR-OP-11-5	Selected-Item-Index	7.6.3.1[1]	Implemented/M (new gain:x4)
ATTR-OP-11-6	Select-List	7.6.3.1[1]	Implemented/M
ATC-0-1	Set-Time	6.3.2.4[1]	Confirmed
NOTI-0-1	MDS-Configuration-Event	6.3.2.5[1]	Confirmed
NOTI-0-2	MDS-Dynamic-Data-Update-Fixed	6.3.2.5[1]	Confirmed

(1) Section in X73-20601-2008 [19]
(2) Section in X73-10201-2004 [20]
(3) Section in X73-10406-2011 [87]

In addition, the ICSs that determine the scope of the implementation can be observed. The ICSs corresponding to the attributes Sco-Capability, Operational-State, Operation-Spec, Selected-Item-Index , and Select-List, Current-Value, Set-Value-Range, Step-Width, Unit-Code are currently part of this proposal.

2. Footprint

Table 5.8, shows the size of each APDU. The middle of table shows the entire size in bytes using the command and control extension package, and the right side shows the byte size of the binary code of the current X73-10406. At the end of the table, you can see the full size of each sequence, and it is clear that the number of resulting bytes to implement the remote control to modify a parameter is higher.

Table 5.8: Comparison of the total size of sequence diagrams using remote control and the current version. The highlighted cells correspond to the new features.

APDU	Without remote control (bytes)	With remote control (bytes)
Association Request	54	54
Association Response	48	48
Event Request Configuration	116	298
Event Response Configuration	26	26
Set Request Operational State ON		26
Set Response Operational State ON		24
Event Request Operating Scanner Create		682
Event Response Operating Scanner Create		22
Action Request Operation Invoke		204
Action Response Operation Invoke		24
Event Request Operating Attribute Update		204
Event Response Operating Attribute Update		22
Get Request MDS All Attributes	18	18
Get Response MDS All Attributes	104	104
Set Request MDS Time		28
Set Response MDS Time		30
Set Request Operational State Off		26
Set Response Operational State Off		22
Association Release Request	6	6
Association release Response	6	6
Full Size	378	1874

5.5 Discussion

A. Command and control extension package

In this work two use cases have been implemented with different requirements and these implementations could go from a minimum to a maximum, in terms of the requirements for remote control operations. Therefore, it could be considered whether it is possible to define the implementation Scanner Package and SCO class with at least one instance of the operation object corresponding to the minimum requirements of implementing the command and control extension, so that developers are aware of the extra requirements for add this extension to its agents.

However, both implementations are still far from using the full potential of the remote control extension, which models the implementation of 7 different types of operations through the SCO. Thus, in the future, it is expected to be able to support to new use cases with a greater number of requirements.

B. Weight Scale

The aim of this proof of concept based on the weighing scale is to design a basic implementation which adds the command and control extension to allow the remote modification of a single parameter, in this case, the measuring units parameter. It is important to note that though the specialization of the scale is the least complex the level of complexity that requires adding the extension of remote control is significantly high. Also, it provides all the capabilities to test the remote control proposal.

On the other hand, due to new objects that have been added to implement the command and control extension, a manager -- that does not implement this extension -- will be unable to recognize an event report containing the agent configuration with a command and control extension enabled, mainly due it could not interpret the information classes that are not present in the current version of the standard. Therefore, to get the agent associated with the manager, the agent could provide a configuration that does not include objects which are present within the command and control extension. In order to facilitate the coexistence of both versions, it is necessary to increase the version of the standard. In this way, if the manager is not compatible with the version that includes the command and control extension, it can reject this setting and use a supported configuration of X73PHD during the association procedure.

C. ECG event recorder

The implementation of the ECG event recorder requires a higher level of complexity than the implementation of the X73-10415 agent, and it performs 3 types of operations compared to the weigh scale that implements only one type of operation. It should not be forgotten that one implementation of the command and control extension package regardless of the number of operations that the device performs is significantly higher. In addition, within its functionality, it has the the capability of data storage by deploying instances of the class PM-store, which allows implementing this feature.

According to the results in the bottommost row in Table 5.8, implementing the remote control process would imply that the relative weight of the all size implementation in bytes of the agent with remote control procedure represents approximately 395% compared to the specialization without a remote control.This percentage might suggest that the overhead of adding the extension is considerably high. Nevertheless, APDUs related to ECG recording transmission have been omitted in this study given that their variability is dependant on the patient. In practice these APDUs are dozens or hundreds of types more frequent which depends on several factor (such us the operating mode, patient waveform features, communication pepriodicity). Thus if they were considered the percentage would be reduced considerably. However, due to several factors –assuming, if the electrocardiogram recorder could require configuration every time it associates with the manager, user preferences (some users may want to reconfigure the agent more frequently), medical prescriptions (a formal caregiver may require continuous reconfiguration of the agents to better support clinical decisions making), among others–, providing quantitative measures on how deeply the incorporation of remote control would affect X73-compliant agents is an intricate task. Given that this increase in transmitted bytes could increase energy consumption, it could be needed to optimize the protocol further. A possible improvement is to include a new Configuration-Version attribute within the Operating scanner which identifies a static configuration and to define a new procedure within the Operating scanner which allows the manager initiating the Operating scanner to start monitoring changes from that static configuration, thus, skipping or avoiding Operation creation APDUs. Therefore, the implementer chooses a static configuration for its agent and assigns to this configuration a Configuration-Version value (e.g. “1”). When the manager connects, if it has previously connected to that agent, it can remember its static configuration assigned to the Configuration-Version value and avoid the procedure of Operating scanner configuration retranmission (Oper-Create).

Furthermore, the characteristics of the hardware used are closely linked to energy consumption and therefore with its autonomy. Just mention the increased computing power of 32-bit microcontrollers (Cortex-M3 processor, MIPS to 1250 100MHz) with their 16-bit predecessors (The MSP430x5xx Series, MIPS 25 to 25 MHz) [139,140], which could indicate that the overload of add the command and control extension could be even insignificant. However, in future research could be a deeper analysis to obtain a specific result.

As in the proof of concept of the weighing scale, a manager that does not implement the command and control extension will be unable to recognize an event report containing the agent configuration with command and control extension enabled, so it is necessary to follow the same mechanism which consist in to increases the version of the standard. In this way, if the manager is not compatible with the version that includes the command and control extension, it can reject this setting and use a supported configuration of the X73PHD during the association procedure.

Chapter 6 Conclusion and future lines

In this last chapter, conclusions of the command and control extension proposal and the results of its implementation on two proof of concept agents are collected and linked to initial hypotheses and objectives of this research work. Likewise a short list of potential future research lines is pointed out.

6.1 General conclusion

This research work has addressed a new command and control extension proposal for agents and managers of the X73PHD family of standards. This proposal has been implemented from manager to agent Personal Health Device taking into account the features of each agent, since not all of these features are always used. Then the agent could make, using Operating Scanner class, a configuration report including its remote control functions to be sent back to the manager, which could directly interact with any Service and Control Object (SCO) configurations.

To implement our proposal, a methodological procedure to ensure a proper integration on each use case has been developed. This methodology provides the guidelines for the application of the command and control extension in any use cases.

To ensure this proposal may be applied on any X73PHD specialization, it has been embedded into X73-20601, where it has been integrated into the Domain Information Model (DIM), communication model and service model. Therefore, experienced developers working with the current version of X73PHD could easily adopt our proposal as it is based on the established format of this standard.

Currently there is no a manager that can be used to perform remote control operations based on X73PHD family of standards. For this reason, it has been necessary to implement a test platform that meet the right balance between capacity and hardware resources allowing us to test our command and control extension proposal. Due to the results obtained using this test bench, it was concluded that these tools can be used to test other proof of concept.

6.2 Particular conclusions

Regarding the current state of the art of X73 and remote control proposal:

1) In previous work of remote control service development, a similar service in a specialization, called X73-20301, has been proposed. Nevertheless, there is not mention about the mechanism that lets to know the remote control capabilities implemented in the agent and is not included in the Control Package.

2) Another research work propose to use a service that can be used to trigger predefined agent features, taking into account the Set service but it need another tools that aid in the execution of remote control operations, which is the main limiting factor in the scope of this service.

3) The ability to enable report generation, The information concerning the characteristics of remote control and the remote control operations necessary to support the requirements exposed by previous works are considered together to the development of a command and control extension proposal.

4) The main tasks in the PHD-WG was to develop a proposal for a remote control that could be included into the family of the standard X73PHD. Later it was requested information about the use cases needed to normalize the remote control process, which lead us to develop a command and control extension proposal.

Regarding the requirements and reviewed use cases:

1) The use cases SABTE and HOLTIN shares a parameter that would modify their modes of operation. HOLTIN has the ability to operate in Real-Time, Event-Driven and Classic holder . Likewise SABTE has the modes Standby, Therapy, Drying and Mask. Based on the above it can be concluded that in order to implement these features to add a new parameter that offers supports to this service is needed.

2) A common requirements in all use cases analyzed are the settings parameters(see Table 3.8). Those parameters can be implemented using the Numeric, Enumeration and Real-Time classes available in the current version of X73-20601. However, to modify these parameters, the implementation of Operation objects from SCO class in the Control package is required

3) There is a clear tendency to have more than one mode of operation on agents based on high performance hardware. Therefore, to implement an attribute that allows to choose the desired mode of operation is essential.

4) Although the Weight Scale use case is not really included in the use cases gathered into the PHD-WG, it has been selected due to the small number of parameters that might require into remote control mode. In this way the remote control realization on this use case could arise the minimum requirements to implement the command and control extension into the rest of agents.

5) The use case Basic Electrocardiograph, has been selected due to its complexity (modes of operation, recorded data, configuration data, etc.). Likewise this use case has almost the same requirements in terms of data types than required by t SABTE , so this implementation could be an example of the development of an agent with capabilities of configuration and remote control of medium complexity.

Regarding the command and control extension proposal:

1) The family of standards X73PHD provides an application framework to include our command and control extension proposal where all actors; users, stakeholders and manufacturers, could be benefited.

2) The command and control extension was implemented using modeling through Object-oriented programming in order not to increase the difficulty of implementation. In this way any developer working with the current version of X73PHD could implement it without requiring additional training.

3) In scenarios where a manager interacts with MDS of different PHDs, such as home telemonitoring, it is necessary to add an identifier for each MDS. In the classical model of X73, particularly in X73-10201 there is a GLB-HANDLE identifier, whose inclusion in X73PHD is kept as an open-discussion-point.

4) There are several limitations that could make more difficult the integration of the command and control extension within X73PHD. For example, to integrate or not the extension of remote control may be a decision that manufacturers and developers should take, however, this requires adding additional features in X73PHD to

enable interoperability between a manager and an agent. Likewise the update of the standard version with additional configurations, as discover the capabilities of the remote control in the agent before to execute a remote control operation, are limitations which can make more difficult the adoption of our proposal.

Regarding of the implementations in the proof of concept:

1) The weigh scale and HOLTIN use cases were implemented as a proof of concept. Both use cases were fully covered using our command and control proposal.

2) The implementation of both proof of concepts was carried out considering power consumption and performance. They were implemented using the Patterns-based Methodology.

3) Several conclusions were obtained from the weigh scale implementation:
   a) The minimum basic requirements to implement the command and control extension in a simple agent were shown.
   b) It illustrated a simple case that can be transferred to both similar use cases and more complex ones, in term of requirements.

4) Several conclusions were obtained from the HOLTIN use case:
   a) It analyzed a more complex command and control application which included a higher variety and complexity of operations.
   b) It proved the use of the Activate operation to manage the control of manual events. Although this feature is covered by the extension, it is necessary to specify these functions further within the corresponding specialization.

5) The LPC1769 microcontroller with ARM Cortex-M3 architecture and FreeRTOS were integrated to provide a test platform for the proof-of-concept implementations. FreeRTOS, which is optimized for this kind of microcontroller-based applications, provided features to manage the processing time and full multitasking control. In addition, the Cortex-M3 architecture is optimized for development of applications that require a RTOS.

6) In the specializations of the standard, it is included an example of binary codes that show the messages exchanged between the agent and the manager. Likewise, in Annex B, the example APDUs of the two use cases that have been developed and implemented in the proof of concept are attached. They can be used to validate the extension of remote control by third parties or to another type validation in a software or hardware level.

6.3 Future Lines

This research works has opened several future lines as following.

Regarding the requirements and reviewed use cases:

1) It has emerged a proposal within the PHD-WG to add geo-location to PHDs. Therefore, it may be required of a package containing classes and services necessary for the user's location and monitor user mobility.

Regarding the proposal of command and control extension proposal:

1) Until now, the command and control extension presupposed that the manager is authorized to perform any operation that the SCO report has available. Nevertheless in the PHD-WG has emerged a discussion about whether it is necessary to introduce a mechanism to check whether the manager has the authority to perform an operation. Also it was discussed if it can even set a mechanism with different levels of authorizations. This could be done through an attribute as authorization-descriptor, right through which the requirements are established to implement an operation. However, the discussion is just beginning and could emerge more alternatives.

2) The validation tools like model checkers or protocol analysers that can be certified if agents and managers were correctly implemented have been developed for the current version of the standard.

Therefore, to develop tools to validate the implementation of the extension of control is of special interest.

3) The process in which the agent sends a standard or extended configuration to the manager could be simplified by storing the first configuration of the agent using the attribute Config-Id already defined in the current version of X73-20601. Later, this attribute will be used in future reconnections reducing the duration of the exchange of messages and simplifying the process of association.

Regarding of the implementations in the proof of concept:

1) The command and control extension was validated for two use cases. However, further work is needed in the development and integration of our proposal within more use cases. The goal of this is to achieve the validation of this extension and include it within new versions of X73PHD.

2) In recent months new tools for transport technologies have come to market, such as Bluetooth Low Energy and low power Wi-Fi, as well as tools that integrate both technologies in a single transceiver. This allows for more efficient use of energy during the transmission of data and integrates multiple transport technologies in one device. For this reason, it is interesting to do an analysis to assess the energy consumption that represents the transmission of information between agent and manager, for messages that are already covered by the standard and for messages of remote control operations.

Apendix A

List of Publications

A.1 Publications in internacional Journals

[1] **Héctor Gilberto Barrón-González,** Miguel Martínez-Espronceda, Jesús Daniel Trigo, Santiago Led, Luis Serrano, “Lessons learned from the implementation of remote control for the interoperability standard ISO/IEEE11073-20601 in a standard weighing scale”, Computer Methods and Programs in Biomedicine, 2015, DOI: 10.1016/j.cmpb.2015.09.015.

[2] Miguel Martínez-Espronceda, Jesús D. Trigo, Santiago Led, **H. Gilberto Barrón-González**, Javier Redondo, Alfonso Baquero, Luis Serrano, “Event-Driven, Pattern-Based Methodology for Cost-Effective Development of Standardized Personal Health Devices”, Computer Methods and Programs in Biomedicine, 2014, vol 117(2), pp. 168-178, DOI: 10.1016/j.cmpb.2014.07.008.

A.2 Book chapters

[1] **Hector Gilberto Barrón-González ,** Miguel Martínez-Espronceda, J. D. Trigo, Santiago Led, Luis Serrano, "Recent Advances in mHealth: An Update to Personal Health Device Interoperability Based on ISO/IEEE11073", Mobile Health,2015,vol 3, pp 375-397, DOI:10.1007/978-3-319-12817-7_17, ISBN: 978-3-319-12816-0.

A.3 Publications in international conference

[1] **Barrón-González H.G,** Martínez-Espronceda M, Led S, Serrano L, Fischer C, Clarke M,"New use cases for remote control and configuration of interoperable medical devices", In:Proceedings of the Annual International Conference of the IEEE Engineering in

Medicine and Biology Society, 2013, pp. 4787-4790, DOI: 10.1109/EMBC.2013.6610618.

[2] **Barrón-González, H.G.,** Martínez-Espronceda, M., Led, S., Serrano, L,"Research benefits of using interoperability standards in remote command and control to implement Personal Health Devices", In: Proceedings of the Mediterranean Conference on Medical and Biological Engineering and Computing, (MEDICON 2013), vol. 41, pp. 1459-1463. DOI: 10.1007/978-3-319-00846-2_361.

[3] **Barron-Gonzalez, H.G.,** Martinez-Espronceda, M., Trigo, J.D., Led, S., Serrano, L, Proposal of a novel remote command and control configuration extension for interoperable Personal Health Devices (PHD) based on ISO/IEEE11073 standard,In: proceedings of the 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014, art. no. 6945072, pp. 6312-6315, DOI: 10.1109/EMBC.2014.6945072.

[4] Martinez-Espronceda, M; Martinez, I; Led, S; Serrano, L; Trigo, JD; Marzo, A; Escayola, J; **Barron, HG;** Garcia, J, "Implementación del Estándar ISO/IEEE11073 en Dispositivos de Salud Personal Inalámbricos: Lecciones Aprendidas", In: Proceedings of XXVIII Congreso Anual de la Sociedad Española de Ingeniería Biomédica (CASEIB 2010).

Apendix B

Weighing Scale

ASSOC|REQ: ASSOCIATE

0xE2 0x00	APDU CHOICE Type (AarqApdu)
0x00 0x32	CHOICE.length = 50
0x80 0x00 0x00 0x00	assoc-version
0x00 0x01 0x00 0x2A	data-proto-list.count = 1 | length = 42
0x50 0x79	data-proto-id = 20601
0x00 0x26	data-proto-info length = 38
0x80 0x00 0x00 0x00	protocol-version, This bit shall be set if IEEE Std 11073-20601a is supported
0xA0 0x00	encoding-rules = MDER or PER
0x80 0x00 0x00 0x00	nomenclature-version
0x00 0x00 0x00 0x00	functional-units – e.g., flag ability to enter test association,at this case it is reserver for future use.
0x00 0x80 0x00 0x00	system-type = sys-type-agent
0x00 0x08	system-id length = 8 and value
0x88 0x77 0x66 0x55 0x44 0x33 0x22 0x11	
0x40 0x00	dev-config-id = extended-config-start (16384)
0x00 0x81 0x01 0x01	data-req-mode-capab
0x00 0x00 0x00 0x00	option-list.count = 0 | option-list.length = 0

ASSOC|RSP: ASSOCIATE

0xE3 0x00	APDU CHOICE Type (AareApdu)
0x00 0x2C	CHOICE.length = L
0x00 0x03	result = accepted-unknown-config
0x50 0x79	data-proto-id = 20601
0x00 0x26	data-proto-info length = L
0x80 0x00 0x00 0x00	protocol-version, This bit shall be set if IEEE Std 11073-20601a is supported
0x80 0x00	encoding-rules = MDER
0x80 0x00 0x00 0x00	nomenclature-version
0x00 0x00 0x00 0x00	functional-units,at this case it is reserver for future
0x80 0x00 0x00 0x00	system-type = sys-type-manager
0x00 0x08	system-id length = 8 and value
0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	
0x00 0x00	manager's response in dev-config-id is always 0
0x00 0x00 0x00 0x00	manager's response in data-req-mode-capab is always 0
0x00 0x00 0x00 0x00	option-list.count = 0 | option-list.length = 0

EVENT|REQ: CONFIG

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x5A	CHOICE.length = 90
0x00 0x58	OCTET STRING.length = 88
0x00 0x01	invoke-id = 0x0001 (start of DataApdu. MDER encoded).
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x52	CHOICE.length = 82
0x00 0x00	obj-handle = 0 (MDS object)

0xFF 0xFF 0xFF 0xFF	event-time = agent does not have a RelativeTime clock
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x48	event-info.length = 72 (start of ConfigReport)
0x40 0x00	dev-config-id = extended-config-start (16384)
0x00 0x03	config-obj-list.count = 3 Measurement objects will be “announced”
0x00 0x42	config-obj-list.length = 66
0x00 0x06	obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01	obj-handle = 1 (? 1st Measurement is body
0x00 0x04	attributes.count =4
0x00 0x24	attributes.length = 36
0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04	attribute-value.length = 4
0x00 0x02 0xE1 0x40	MDC_PART_SCADA | MDC_MASS_BODY_ACTUAL
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0xD0 0x40	intermittent, stored data, msmt aperiodic, agent init, measured
0x09 0x96	attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02	attribute-value.length = 2
0x06 0xC3	MDC_DIM_KILO_G
0x0A 0x55	attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x0C	attribute-value.length = 12
0x00 0x02	AttrValMap.count = 2
0x00 0x08	AttrValMap.length = 8
0x0A 0x56 0x00 0x04	MDC_ATTR_NU_VAL_OBS_SIMP | value length = 4
0x09 0x90 0x00 0x08	MDC_ATTR_TIME_STAMP_ABS | value length = 8
0x00 0x2B	obj-class = MDC_MOC_CNTRL_SCO(43)
0x00 0x02	obj-handle = 2 (SCO object)
0x00 0x01	attributes.count = 1
0x00 0x06	attributes.length = 6
0x09 0x76	attribute-id = MDC_ATTR_SCO_CAPAB(2422)
0x00 0x02	attribute-value.length = 2
0x00 0x00	sco-capability none
0x00 0x18	obj-class = MDC_MOC_SCAN_UCFG_OP(24)
0x00 0x03	obj-handle = 3 (Operating scanner object)
0x00 0x00	attributes.count = 0
0x00 0x00	attributes.length = 0

EVENT|RSP: CONFIG

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x16	CHOICE.length = 22
0x00 0x14	OCTET STRING.length = 20
0x00 0x01	invoke-id = 0x0001 (start of DataApdu. MDER encoded.)
0x02 0x01	CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E	CHOICE.length = 14
0x00 0x00	obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF	currentTime = manager doesn't support (i.e. not have a local counter)
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x04	event-reply-info.length = 4
0x40 0x00	ConfigReportRsp.config-report-id = 0x4000

0x00 0x00	ConfigReportRsp.config-result = accepted-config

EVENT|REQ: MDS DATA REPORT FIXED FORMAT

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x2A	CHOICE.length = 42
0x00 0x28	OCTET STRING.length = 40
0x00 0x02	invoke-id = 0x0002
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x22	CHOICE.length = 34
0x00 0x00	obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF	event-time = mpt supported
0x0D 0x1D	event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x18	event-info.length = 24//Secuencia de EventReportArgumentSimple
0xF0 0x00	ScanReportInfoFixed.data-req-id = data-req-id-agent-initiated(61440)
0x00 0x00	ScanReportInfoFixed.scan-report-no = 0
0x00 0x01	ScanReportInfoFixed.obs-scan-fixed.count = 1
0x00 0x10	ScanReportInfoFixed.obs-scan-fixed.length = 16
0x00 0x01	ScanReportInfoFixed.obs-scan-fixed.value[0].obj-
0x00 0x0C	ScanReportInfoFixed.obs-scan-fixed.value[0]. obs-val-data.length = 12
0xFF 0x00 0x02 0xFA	Simple-Nu-Observed-Value = 76.2 (kg)
0x20 0x07 0x12 0x06	Absolute-Time-Stamp = 2007-12-06T12:10:0000
0x12 0x10 0x00 0x00	

EVENT|RSP: MDS DATA REPORT FIXED FORMAT

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x00 0x02	invoke-id = 0x0002 (mirrored from invocation)
0x02 0x01	CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x0A	CHOICE.length = 10
0x00 0x00	obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF	currentTime = manager doesn't support it
0x0D 0x1D	event-type = MDC_NOTI_SCAN_REPORT_FIXED
0x00 0x00	event-reply-info.length = 0

SET|REQ: OPERATING SCANNER (OPERATIONAL STATE=ON)

0xE7 0x00	Type (PrstApdu)
0x00 0x16	CHOICE.length = 22
0x00 0x14	OCTET STRING.length = 20
0x00 0x03	invoke-id = 0x0001 // invoke-id at manager
0x01 0x05	CHOICE(Remote Operation Invoke |SET)
0x00 0x0E	CHOICE.length = 14
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x01	modification-list.count=1
0x00 0x08	modification-list.length=8

0x00 0x00	modification-list[0].modify-operator=replace(0)
0x09 0x53	modification-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	modification-list[0].ANY_DEFINED_BY.size=2
0x00 0x01	modification-list[0].ANY_DEFINED_BY.Operational-State=enabled(1)

<u>SET|RSP: OPERATING SCANNER</u>

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x00 0x03	invoke-id = 0x0001
0x02 0x05	CHOICE(Remote Operation Response |SET)
0x00 0x0C	CHOICE.length = 12
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x01	attribute-list.count=1
0x00 0x06	attribute-list.length=6
0x09 0x53	attribute-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	attribute-list[0].ANY_DEFINED_BY.size= MDC_ATTR_OP_STAT(2387)
0x00 0x01	attribute-list[0].ANY_DEFINED_BY.Operational-State=enabled(1)

<u>EVENT|REQ: OPERATING SCANNER CREATE</u>

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x56	CHOICE.length = 86
0x00 0x54	OCTET STRING.length = 84
0x00 0x04	invoke-id = 0x0003
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x4E	CHOICE.length = 78
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x1D	event-type = MDC_NOTI_OP_CREAT
0x00 0x44	event-info.length = 68
0X00 0x02	scan-report-no= 2
0X00 0X01	ScanReportInfo.count=1 //scan-info SEQUENCE OF OpCreateEntry
0X00 0x3E	ScanReportInfo.length=62 //scan-info SEQUENCE OF OpCreateEntry
0x00 0x00	context-id = 0
0x00 0x03	Handle SCO =03
0x00 0x01	count = 1
0x00 0x36	length = 54
0x00 0x2d	op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
0x00 0x01	op-instance-no = 1
0x00 0x2C	obj-class = MDC_MOC_CNTRL_OP
0x00 0x03	attribute-list.count = 4
0x00 0x2C	attribute-list.length = 44
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC

0x00 0x0C	attribute-value.length = 12
0x00 0x00	vattr-id. Check 10101 -- B.6 -- Currently no attr
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0x32	attribute-id = MDC_ATTR_INDEX_SEL(2354)
0x00 0x02	attribute-value.length = 2
0x00 0x00	First element is selected
0x09 0x36	attribute-id = MDC_ATTR_LIST_SEL(2358)
0x00 0x0C	attribute-value.length = 12
0x00 0x01	CHOICE = oid-list[1]
0x00 0x08	CHOICE.length
0x00 0x02	oid-list.count
0x00 0x04	oid-list.size
0x06 0xC3	MDC_DIM_KILO_G(1731)
0x06 0xE0	MDC_DIM_LB(1760)

EVENT|RSP: OPER CREATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x00 0x04	invoke-id = 0x0003 (mirrored from invocation)
0x02 0x01	CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x0A	CHOICE.length = 10
0x00 0x03	obj-handle = 3 (Operating Scanner object)
	0xFF 0xFF 0xFF 0xFF
	event-time not supported
0x0D 0x1D	event-type = MDC_NOTI_OP_CREAT
0x00 0x00	event-reply-info.length = 0

ACTION|REQ: OPERATION INVOKE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x26	CHOICE.length = 38
0x00 0x24	OCTET STRING.length = 36
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x01 0x07	CHOICE(Remote Operation Invoke | Confirmed Action)
0x00 0x1E	CHOICE.length = 30
0x00 0x03	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x18	action-info-args.length = 24
0x00 0x00	checksum = not-used(0)
0xFF 0x33 0x44 0x55	invoke-cookie(should be random)
0x00 0x01	op-invoke-list.count
0x00 0x0C	op-invoke-list.size = 12
0x00 0x2d	op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
0x00 0x01	op-instance-no = 1
0x00 0x00	op-mod-type->op-replace(0)
0x09 0x32	attribute-id = MDC_ATTR_INDEX_SEL(2354)
0x00 0x02	attribute-value.length = 2
0x00 0x01	Second element is selected (MDC_DIM_LB)

ACTION|RSP: OPERATION INVOKE RESULT

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x14	CHOICE.length = 20
0x00 0x12	OCTET STRING.length = 18
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x02 0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00 0x0C	CHOICE.length = 12
0x00 0x00	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x06	action-info-args.length = 6
0xFF 0x33 0x44 0x55	invoke-cookie
0x00 0x00	OpInvResult= op-successful(0),

EVENT|REQ: OPERATION ATTRIBUTE UPDATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x38	CHOICE.length = 56
0x00 0x36	OCTET STRING.length = 54
0x00 0x04	invoke-id = 0x0004
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x30	CHOICE.length = 48
0x00 0x03	obj-handle = 3 (Operating Scanner)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x0B	event-type = MDC_NOTI_OP_ATTR_UPDT(3339) (ASN.1 data type = OpAttributeInfo)
0x00 0x26	event-info.length = 38
0x00 0x03	scan-report-no = 3
0x00 0x01	list.count = 1
0x00 0x20	list.length = 32
0x00 0x00	context-id ->MdsContext=0 (The same as above)
0x00 0x01	list.count = 1
0x00 0x1A	list.length = 26
0x00 0x03	sco-handle=3
0xFF 0x33 0x44 0x55	invoke-cookie->INT-U32,
0x00 0x01	lock-state->AdministrativeState,
0x00 0x01	op-elem-attr-up-list.count = 1
0x00 0x0E	op-elem-attr-up-list.length = 14
0x00 0x2D	op-class-id= MDC_MOC_CNTRL_OP_SEL_IT(45)
0x00 0x01	op-instance-no->InstNumber,
0x00 0x04	attributes.count = 1
0x00 0x0C	attributes.length = 6
0X09 0X32	attribute-id = MDC_ATTR_INDEX_SEL
0x00 0x02	attribute-value.length = 2
0x00 0x01	Selected-Item-Index=1

EVENT|RSP: OPERATION ATTRIBUTE UPDATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x10	CHOICE.length = 16
0x00 0x0E	OCTET STRING.length = 14
0x00 0x04	invoke-id = 0x0004
0x01 0x01	CHOICE(Remote Operation Response | Confirmed Event Report)

0x00 0x08	CHOICE.length = 8
0x00 0x03	obj-handle = 3 (Operating Scanner)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x0B	event-type = MDC_NOTI_OP_ATTR_UPDT
0x00 0x00	event-info.length = 0

GET|REQ: MDS ALL ATTRIBUTES

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x0E	CHOICE.length = 14
0x00 0x0C	OCTET STRING.length = 12
0x00 0x07	invoke-id = 0x0003
0x00 0x03	CHOICE(Remote Operation Invoke|Get)
0x00 0x06	CHOICE.length = 6
0x00 0x00	handle = 0 (MDS Object)
0x00 0x00	attribute-id-list.count = 0 (all atributes)
0x00 0x00	attribute-id-list.length = 0

GET|RSP: MDS ALL ATTRIBUTES

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x4C	CHOICE.length = 76
0x00 0x4A	OCTET STRING.length = 74
0x00 0x07	invoke-id = 0x0003
0x02 0x03	CHOICE(Remote Operation Invoke|Get)
0x00 0x66	CHOICE.length = 110
0x00 0x00	handle = 0 (MDS object)
0x00 0x01	attribute-list.count = 1
0x00 0x04	attribute-list.length = 4
0X0A 0X5A	attribute id = MDC_ATTR_SYS_TYPE_SPEC_LIST
0X00 0X08	attribute-value.length = 8
0x00 0x01	TypeVerList count = 1
0x00 0x04	TypeVerList length = 4
0x10 0x0F	attribute-id = MDC_DEV_SPEC_PROFILE_SCALE
0x00 0x01	version = version 1 of the specialization
0X09 0X28	attribute-id = MDC_ATTR_ID_MODEL
0x00 0x1A	attribute-value.length = 26
0x00 0x0A 0x54 0x68 0x65 0x43 0x6F 0x6D 0x70 0x61 0x6E 0x79	string length = 10 | "TheCompany"
0x00 0x0C 0x54 0x68 0x65 0x53 0x63 0x61 0x6C 0x65 0x41 0x42 0x43 0x00	string length = 12 | "TheScaleABC\0"
0x09 0x84	attribute-id = MDC_ATTR_SYS_ID
0x00 0x0A	attribute-value.length = 10
0x00 0x08 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88	octet string length = 8 | EUI-64
0x0a 0x44	attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00 0x02	attribute-value.length = 2
0x40 0x00	dev-config-id = 16384 (extended-config-start)
0x09 0x87	attribute-id =MDC_ATTR_TIME_ABS
0x00 0x08	attribute-value.length = 8
0x20 0x13 0x08 0x01 0x10 0x30 0x00 0x00	Date&Time=2013-08-01T10:30:0000

SET|REQ: MDS TIME

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x1C	CHOICE.length = 28

```
0x00 0x1A                     OCTET STRING.length = 26
0x00 0x08                     invoke-id = 0x0004
0x01 0x05                     CHOICE(Remote Operation Invoke |SET)
0x00 0x14                     CHOICE.length = 20
0x00 0x00                     obj-handle = 0 (MDS object)
0x00 0x01                     modification-list.count=1

0x00 0x0E                     modification-list.length=14
0x00 0x00                     modify-operator=replace(0)

0x09 0x87                     attribute.identifier=
                              MDC_ATTR_TIME_ABS(2439)
0x00 0x08                     attribute.size=8
0x20 0x13 0x10 0x23           AbsoluteTime
0x16 0x30 0x00 0x00
```

SET|RSP: MDS TIME

```
0xE7 0x00                     APDU CHOICE Type (PrstApdu)
0x00 0x1A                     CHOICE.length = 26
0x00 0x18                     OCTET STRING.length = 24
0x00 0x08                     invoke-id = 0x0004
0x02 0x05                     CHOICE(Remote Operation Invoke |SET)
0x00 0x12                     CHOICE.length = 18
0x00 0x03                     obj-handle = 0 (MDS object)
0x00 0x01                     attribute-list.count=1

0x00 0x0C                     attribute-list.length=12
0x09 0x87                     attribute.identifier=
                              MDC_ATTR_TIME_ABS(2439)
0x00 0x08                     attribute.size=8
0x20 0x13 0x10 0x23           AbsoluteTime
0x16 0x30 0x00 0x00
```

SET|REQ: OPERATING SCANNER (OPERATIONAL STATE=OFF)

```
0xE7 0x00                     Type (PrstApdu)
0x00 0x16                     CHOICE.length = 22
0x00 0x14                     OCTET STRING.length = 20
0x00 0x05                     invoke-id = 0x0005
0x01 0x05                     CHOICE(Remote Operation Invoke |SET)
0x00 0x0E                     CHOICE.length = 14
0x00 0x03                     obj-handle = 3 (Operating Scanner object)
0x00 0x01                     modification-list.count=1

0x00 0x08                     modification-list.length=8

0x00 0x00                     modify-operator=replace(0)

0x09 0x53                     identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02                     size=MDC_ATTR_OP_STAT(2387)
0x00 0x00                     Operational-State=disabled(0)
```

SET|RSP: OPERATING SCANNER

```
0xE7 0x00                     APDU CHOICE Type (PrstApdu)
0x00 0x12                     CHOICE.length = 18
0x00 0x10                     OCTET STRING.length = 16
0x00 0x05                     invoke-id = 0x0005
```

0x02 0x05	CHOICE(Remote Operation Response |SET)
0x00 0x03	obj-handle = 3 (Operating Scanner object)--Esto
0x00 0x01	attribute-list.count=1
0x00 0x06	attribute-list.length=6
0x09 0x53	attribute-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	attribute-list[0].ANY_DEFINED_BY.size=MDC_ATTR_OP_STAT(2387)
0x00 0x01	attribute-list[0].ANY_DEFINED_BY.Operational-State=disabled(0)

ASSOC|REQ: RELEASE

0xE4 0x00	APDE CHOICE Type (RlreApdu)
0x00 0x02	CHOICE.length =2
0x00 0x00	reason = normal

ASSOC|RSP: RELEASE

0xE5 0X00	APDU CHOICE Type(RleApdu)
0x00 0x02	CHOICE.length = 2
0x00 0x00	reason = normal

Basic Electrocardiograph

ASSOC|REQ: ASSOCIATE (Agente a Manager)

```
0xE2    0x00                          APDU CHOICE Type (AarqApdu)
0x00    0x32                          CHOICE.length        = 50
0x80    0x00 0x00 0x00                assoc-version
0x80    0x00 0x00 0x2A                data-proto-list.count = 42
0x50    0x79                          data-proto-id                  = 20601
0x00    0x26                          data-proto-info length = 38
0x40    0x00      0x00      0x00      protocolVersion
0x80    0x00                          encoding rules           = MDER
0x80    0x00      0x00      0x00      nomenclatureVersion
0x00    0x00      0x00      0x00      funcionalUnits - no Test Association capabilities
0x00    0x80      0x00      0x00      systemType= sys-type-agent
0x00    0x08                          system-id length       = manufacturer and device-
0x00    0x00      0x00      0x00
0x00    0x00      0x00      0x00
0x40    0x00                          dev-config-id = extended configuration
0x00    0x00                          data-req-mode-flags
0x01    0x00                          data-req-init-agent-count = 1
0x00    0x00      0x00      0x00      optionList.count = 0
```

ASSOC|RSP: ASSOCIATE (Manager a Agente)

```
0xE3    0x00                          APDU CHOICE Type (AareApdu)
0x00    0x1A                          CHOICE.length = 26
0x00    0x03                          result = accepted-unknown-config
0x50    0x79                          data-proto-id = 20601
0x00    0x26                          data-proto-info length = 38
0x40    0x00      0x00      0x00      protocolVersion
0x80    0x00                          encodign rules = MDER
0x80    0x00      0x00      0x00      nomenclatureVersion
0x00    0x00      0x00      0x00      functionalUnits = normal Association
0x80    0x00      0x00      0x00      systemType = sys-type-manager
0x00    0x08                          system-id  length = manufacturer and device-
```

EVENT|REQ: CONFIG (Agente a Manager)

```
0xE7 0x00                             APDU CHOICE Type (PrstApdu)
0x01 0x2A                             CHOICE.length = 298
0x01 0x28                             OCTET STRING.length = 296
0x00 0x01                             invoke-id = 0x0001 (start of DataApdu. MDER
                                      encoded.)
0x01 0x01                             CHOICE(Remote Operation Invoke | Confirmed
                                      Event Report)
0x01 0x20                             CHOICE.length = 288
0x00 0x00                             obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF                   event-time = agent does not have a RelativeTime
                                      clock
0x0D 0x1C                             event-type = MDC_NOTI_CONFIG
0x01 0x16                             event-info.length = 278 (start of ConfigReport)
0x40 0x00                             dev-config-id = extended-config-start (16384)
0x00 0x00                             obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF                   event-time = agent does not have a RelativeTime
                                      clock
0x0D 0x1C                             event-type = MDC_NOTI_CONFIG
```

0x01 0x0A	event-info.length = 266 (start of ConfigReport)
0x40 0x00	dev-config-id = extended-config-start (16384)
0x00 0x08	config-obj-list.count = 8 objects will be “announced”
0x01 0x04	config-obj-list.length = 260
0x00 0x06	obj-class = MDC_MOC_VMO_METRIC_NU
0x00 0x01	obj-handle = 1 (1st Measurement is heart rate)
0x00 0x04	attributes.count = 4
0x00 0x24	attributes.length = 36
0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04	attribute-value.length = 4
0x00 0x02 0x4B 0x5C	MDC_PART_SCADA | MDC_ ECG_HEART_RATE
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0x02 0x02	stored data, agent init
0x09 0x96	attribute-id = MDC_ATTR_UNIT_CODE
0x00 0x02	attribute-value.length = 2
0x0A 0xA0	MDC_DIM_BEAT_PER_MIN
0x0A 0x55	attribute-id = MDC_ATTR_ATTRIBUTE_VAL_MAP
0x00 0x08	attribute-value.length = 12
0x00 0x02	AttrValMap.count = 2
0x00 0x08	AttrValMap.length = 8
0x0A 0x4C 0x00 0x02	MDC_ATTR_NU_VAL_OBS_BASIC, 2
0x09 0x91 0x00 0x04	MDC_ATTR_TIME_STAMP_REL, 4
0x00 0x2B	obj-class = MDC_MOC_CNTRL_SCO(43)
0x00 0x02	obj-handle = 9 (SCO object)
0x00 0x01	attributes.count = 1
0x00 0x06	attributes.length = 6
0x09 0x76	attribute-id = MDC_ATTR_SCO_CAPAB(2422)
0x00 0x02	attribute-value.length = 2
0x00 0x00	sco-capability none
0x00 0x18	obj-class = MDC_MOC_SCAN_UCFG_OP(24)
0x00 0x03	obj-handle =10 (Operating scanner object)
0x00 0x00	attributes.count = 0
0x00 0x00	attributes.length = 0
0x00 0x09	obj-class=MDC_MOC_VMO_METRIC_SA_RT
0x00 0x03	obj-handle =3 (ECG Waveform object)
0x00 0x07	attributes.count = 7
0x00 0x36	attributes.length = 54
0x09 0x2F	attribute-id= MDC_ATTR_ID_TYPE
0x00 0x02	attributes.length = 4
0x00 0x02 0x01 0x3D	MDC_PART_SCADA | MDC_ECG_ELEC_POTL_III
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0x02 0x02	mss-avail-stored-data(1),mss-acc-agent-initiated(9),00000010 00000010
0x09 0x96	attribute-id=Unit Code
0x00 0x02	attribute-value.length = 2
0x10 0xB2	MDC_DIM_MILLI_VOLT
0x09 0x8D	attribute-id=MDC_ATTR_TIME_PD_SAMP –>Sample-Period
0x00 0x04	attribute-value.length =4
0x00 0x00 0x1F 0x40	1 seg=8000*125 ms
0x09 0x6F	attribute-id=MDC_ATTR_SCALE_SPECN_I16 1uV/div, ADC 16bits, 250 Sps (ECG ~ 2mVpp min -2mV a max 2mV)
0x00 0x0C	attribute-value.length = 12

0x00 0x00 0x07 0xD0	lower-absolute-value
0x00 0x00 0x07 0xD0	upper-absolute-value
0x00 0x01	lower-scaled-value
0x00 0x01	upper-scaled-value
0x09 0x6D	attribute-id=MDC_ATTR_SA_SPECN
0x00 0x06	attribute-value.length =6
0x00 0x32	array-size = 50?
0x10 0xFF	sample-size = 16, significant-bits = 255
0x20 0x00	SaFlags = 0010 0000 0000 0000
0x00 0x05	obj-class=MDC_MOC_VMO_METRIC_ENUM
0x00 0x04	obj-handle =4 (Device Status object)
0x00 0x02	attributes.count = 2
0x00 0x0E	attributes.length = 14
0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
0x00 0x04	attribute-value.length =4
0x00 0x02 0x55 0xD8	MDC_PART_SCADA | MDC_ECG_DEV_STAT
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0x00 0x00	
0x00 0x05	obj-class=MDC_MOC_VMO_METRIC_ENUM
0x00 0x05	obj-handle =5 (Context Data Trigger)
0x00 0x02	attributes.count = 2
0x00 0x0E	attributes.length = 14
0x09 0x2F	attribute-id = MDC_ATTR_ID_TYPE
0x00 0x0A	attribute-value.length =10
0x00 0x02 0x55 0xD9	MDC_PART_SCADA | MDC_ECG_EVT_CTXT_GEN
0x0A 0x46	attribute-id = MDC_ATTR_METRIC_SPEC_SMALL
0x00 0x02	attribute-value.length = 2
0x00 0x00	pagina 163 remote control proposal
0x00 0x3D	obj-class = MDC_MOC_VMO_PMSTORE
0x00 0x06	obj-handle =6 (Pm Stores Periodic)
0x00 0x06	attributes.count = 6
0x00 0x28	attributes.length = 40 bytes
0x0A 0x4D	attribute-id = MDC_ATTR_PM_STORE_CAPAB
0x00 0x02	attribute-value.length = 2
0x84 0x60	pmsc-var-no-of-segm | pmsc-peri-seg-entries | pmsc-clear-segm-remove_| pmsc-clear-segm-all-sup
0x09 0x43	attribute-id = MDC_ATTR_STORE_SAMPLE_ALG
0x00 0x02	attribute-value.length = 2
0x04 0x00	st-alg-no-downsampling
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enabled
0x09 0x8D	attribute-id = MDC_ATTR_TIME_PD_SAMP
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x06 0x40	200 miliseconds = 200 * 8000
0x09 0x51	attribute-id = MDC_ATTR_NUM_SEG
0x00 0x02	attribute-value.length = 2
0x00 0x01	1 PM-segment is currently stored
0x0A 0x63	attribute-id = MDC_ATTR_CLEAR_TIMEOUT
0x00 0x04	attribute-value.length = 4
0x00 0x02 0x71 0x00	20 seconds = 20* 8000
0x00 0x3D	obj-class = MDC_MOC_VMO_PMSTORE
0x00 0x07	obj-handle =7 (Pm Stores Aperiodic)
0x00 0x07	attributes.count = 5
0x00 0x1C	attributes.length = 32 bytes
0x0A 0x4D	attribute-id = MDC_ATTR_PM_STORE_CAPAB

0x00 0x02	attribute-value.length = 2
0x84 0x60	pmsc-var-no-of-segm | pmsc-peri-seg-entries | pmsc-clear-segm-remove_| pmsc-clear-segm-all-sup
0x09 0x43	attribute-id = MDC_ATTR_STORE_SAMPLE_ALG
0x00 0x02	attribute-value.length = 2
0x04 0x00	st-alg-no-downsampling
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x00	enabled
0x09 0x51	attribute-id = MDC_ATTR_NUM_SEG
0x00 0x02	attribute-value.length = 2
0x00 0x01	1 PM-segment is currently stored
0x0A 0x63	attribute-id = MDC_ATTR_CLEAR_TIMEOUT
0x00 0x04	attribute-value.length = 4
0x00 0x02 0x71 0x00	20 seconds = 20x8000

EVENT|RSP: CONFIG (Manager a Agente)

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x16	CHOICE.length = 22
0x00 0x14	OCTET STRING.length = 20
0x00 0x01	invoke-id = 0x0001 (start of DataApdu. MDER encoded.)
0x02 0x01	CHOICE (Remote Operation Response | Confirmed Event Report)
0x00 0x0E	CHOICE.length = 14
0x00 0x00	obj-handle = 0 (MDS object)
0xFF 0xFF 0xFF 0xFF	currentTime = manager doesn't support (i.e. not have a local counter)
0x0D 0x1C	event-type = MDC_NOTI_CONFIG
0x00 0x04	event-reply-info.length = 4
0x40 0x00	ConfigReportRsp.config-report-id = 0x4000
0x00 0x00	ConfigReportRsp.config-result = accepted-config

SET|REQ: OPERATING SCANNER (OPERATIONAL STATE=ON) (Manager a Agente)

0xE7 0x00	Type (PrstApdu)
0x00 0x16	CHOICE.length = 22
0x00 0x14	OCTET STRING.length = 20
0x00 0x03	invoke-id = 0x0001 // invoke-id at manager
0x01 0x05	CHOICE(Remote Operation Invoke |SET)
0x00 0x0E	CHOICE.length = 14
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x01	modification-list.count=1
0x00 0x08	modification-list.length=8
0x00 0x00	modification-list[0].modify-operator=replace(0)
0x09 0x53	modification-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	modification-list[0].ANY_DEFINED_BY.size=2
0x00 0x01	modification-list[0].ANY_DEFINED_BY.Operational-State=enabled(1)

SET|RSP: OPERATING SCANNER

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x14	CHOICE.length = 20
0x00 0x12	OCTET STRING.length = 18
0x00 0x03	invoke-id = 0x0001
0x02 0x05	CHOICE(Remote Operation Response |SET)
0x00 0x0C	CHOICE.length = 12
0x00 0x03	obj-handle = 3 (Operating Scanner object)--Esto
0x00 0x01	attribute-list.count=1
0x00 0x06	attribute-list.length=6
0x09 0x53	attribute-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	attribute-list[0].ANY_DEFINED_BY.size=MDC_ATTR_OP_STAT(2387)
0x00 0x01	attribute-list[0].ANY_DEFINED_BY.Operational-State=enabled(1)

EVENT|REQ: OPERATING SCANNER CREATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x02 0xA6	CHOICE.length = 678
0x02 0xA4	OCTET STRING.length = 676
0x00 0x03	invoke-id = 0x0003
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x02 0x9E	CHOICE.length = 670
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x1D	event-type = MDC_NOTI_OP_CREAT
0x02 0x94	event-info.length = 660
0X00 0x02	scan-report-no= 2
0X00 0X01	ScanReportInfo.count=1
0X02 0x8E	ScanReportInfo.length=654
0x00 0x00	context-id = 0
0x00 0x09	Handle SCO =09
0x00 0x0A	count = 10 // TODO: Total de operaciones
0x02 0x86	length = 646
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x01	op-instance-no = 1
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x01	vattr-id = MDC_OP_LIM_EV_BRADY
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2

```
0x00 0x01                      HANDLE = 1 (HR)
0x09 0x9D                      attribute-id =MDC_ATTR_VAL_CURR
0x00 0x04                      attribute-value.length = 4
0x00 0x00 0x00 0x32            Bradycardia-Limit =50x10E0 bpm
0x09 0xA0                      attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C                      attribute-value.length = 12
0x00 0x00 0x00 0x1E            minimum 30 (FLOAT-Type)
0x00 0x00 0x00 0x32            maximum 50 (FLOAT-Type)
0x00 0x00 0x00 0x01            resolution 1 (FLOAT-Type)//Poner 1 en todos.
0x00 0x2F                      op-class-id =
                               MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x02                      op-instance-no = 2
0x00 0x05                      attribute-list.count = 5
0x00 0x34                      attribute-list.length = 52
0x09 0x52                      attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C                      attribute-value.length = 12

0x00 0x02                      vattr-id= MDC_ECG_TACHY
0x00 0x00                      op-target
0x40 0x00                      options = supports-default(1),
0x20 0x00                      level        = op-level-professional(2),
0x00 0x00                      group  = 0
0x00 0x00                      priority =0
0x09 0x53                      attribute-id = MDC_ATTR_OP_STAT
0x00 0x02                      attribute-value.length = 2
0x00 0x01                      enable
0x09 0xA4                      attribute-id = MDC_ATTR_VMO_REF
0x00 0x02                      attribute-value.length = 2
0x00 0x01                      HANDLE = 1 (HR)
0x09 0x9D                      attribute-id =MDC_ATTR_VAL_CURR
0x00 0x04                      attribute-value.length = 4
0x00 0x00 0x00 0x96            Taquicardia-Limit  =150 bpm
0x09 0xA0                      attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C                      attribute-value.length = 12
0x00 0x00 0x00 0x64            minimum
0x00 0x00 0x00 0xC8            maximum
0x00 0x00 0x00 0x01            resolution=1
0x00 0x2F                      op-class-id =
                               MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x03                      op-instance-no = 3
0x00 0x05                      attribute-list.count = 5
0x00 0x34                      attribute-list.length = 52
0x09 0x52                      attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C                      attribute-value.length = 12

0x00 0x03                      vattr-id = MDC_ECG_ASYSTOLE
0x00 0x00                      op-target
0x40 0x00                      options = supports-default(1),
0x20 0x00                      level        = op-level-professional(2),
0x00 0x00                      group  = 0
0x00 0x00                      priority =0
0x09 0x53                      attribute-id = MDC_ATTR_OP_STAT
0x00 0x02                      attribute-value.length = 2
0x00 0x01                      enable
0x09 0xA4                      attribute-id = MDC_ATTR_VMO_REF
0x00 0x02                      attribute-value.length = 2
0x00 0x01                      HANDLE = 3 (ECG RT-SA)
0x09 0x9D                      attribute-id =MDC_ATTR_VAL_CURR
0x00 0x04                      attribute-value.length = 4
0x00 0x00 0x3E 0x80            Asistolia-Limit value=2 seg
0x09 0xA0                      attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C                      attribute-value.length = 12
```

0x00 0x00 0x3E 0x80	minimum = 2 seg (16000*125ms)
0x00 0x00 0x7D 0x00	maximum = 4 seg (3200*125ms)
0x00 0x00 0x1F 0x40	resolution=1 seg (8000*125 ms)
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x04	op-instance-no = 4
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x04	vattr-id = MDC_ECG_STORE_TIME_AUTO
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x03	HANDLE = 3 (ECG RT-SA)
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x02	attribute-value.length = 2
0x00 0x0C 0x35 0x00	Store-Time-Auto value=100 seg
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x00 0x1F 0x40	minimum = 1
0x00 0x18 0x6A 0x00	maximum = 200 seg
0x00 0x00 0x1F 0x40	resolution= 1 seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x05	op-instance-no = 5
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x05	vattr-id = MDC_ECG_STORE_TIME_MANUAL
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x03	HANDLE = 3 (ECG RT-SA)
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x04	attribute-value.length =4
0x00 0x36 0xEE 0x80	Store-Time-Manual value=450 seg
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x24 0x9F 0x00	minimum =300 seg
0x00 0x49 0x3E 0x00	maximum = 600 seg
0x00 0x00 0x1F 0x40	resolution = 1 Seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x06	op-instance-no = 6

0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x06	vattr-id = MDC_ECG_QRS_TAQUI_NUMBER
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x01	HANDLE = 1 (HR)
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x03	Qrs-Taqui-Number value=3
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x00 0x00 0x02	minimum =2
0x00 0x00 0x00 0x04	maximum =4
0x00 0x00 0x00 0x01	resolution= 1
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no = 7
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x07	vattr-id =MDC_ECG_MAX_SEARCH_BACK_VALUE
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x03	HANDLE = 3
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x02	attribute-value.length =8
0x00 0x00 0x00 0x01	Max-Search-Back value=1
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x00 0x00 0x01	minimum =1
0x00 0x00 0x00 0x04	maximum =4
0x00 0x00 0x00 0x01	resolution=1 Sin Magnitud
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x08	op-instance-no = 8
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12

0x00 0x08	vattr-id = MDC_ECG_FADING_TIME
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x03	HANDLE = RT-SA
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x02	attribute-value.length = 2
0x00 0x00 0x1F 0x40	Fading-Time value=1 seg
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x00 0x00 0x00	minimum =0 seg
0x00 0x00 0x3E 0x80	maximum =2 seg (2x8000)
0x00 0x00 0x00 0x32	resolution= 4 ms
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x09	op-instance-no = 9
0x00 0x05	attribute-list.count = 5
0x00 0x34	attribute-list.length = 52
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x09	vattr-id = MDC_ECG_FADING_PERCENT
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x03	HANDLE = 3
0x09 0x9D	attribute-id =MDC_ATTR_VAL_CURR
0x00 0x06	attribute-value.length = 6
0x00 0x00 0x00 0x32	Fading-Level (Percent) =%50
0x09 0xA0	attribute-id =MDC_ATTR_VAL_RANGE
0x00 0x0C	attribute-value.length = 12
0x00 0x00 0x00 0x00	minimum =0%
0x00 0x00 0x00 0x64	maximum =100% = 100E0 = 10000E-2 = 0xFE
0x00 0x27 0x10	
0x00 0x00 0x00 0x01	resolution= 1%
0x00 0x2d	op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
0x00 0x0A	op-instance-no = 10
0x00 0x05	attribute-list.count = 5
0x00 0x3E	attribute-list.length = 62
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12
0x00 0x0A	vattr-id = MDC_ECG_DIGITAL_GAIN
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0

0x00 0x00	priority =0

0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x01	HANDLE = ??
0x09 0x32	attribute-id = MDC_ATTR_INDEX_SEL(2354)
0x00 0x02	attribute-value.length = 2
0x00 0x00	First element is selected
0x09 0x36	attribute-id = MDC_ATTR_LIST_SEL(2358)
0x00 0x18	attribute-value.length = 24
0x00 0x03	CHOICE = value-list[3]
0x00 0x14	CHOICE.length =24
0x00 0x04	value-list.count=4
0x00 0x10	value-list.size=16
0x00 0x00 0x00 0x01	Gain = 1
0x00 0x00 0x00 0x02	Gain = 2
0x00 0x00 0x00 0x05	Gain = 5
0x00 0x00 0x00 0x0A	Gain = 10
0x00 0x32	op-class-id = MDC_MOC_CNTRL_OP_ACTIV.
0x00 0x0B	op-instance-no = 11
0x00 0x03	attribute-list.count = 3
0x00 0x1C	attribute-list.length = 28
0x09 0x52	attribute-id = MDC_ATTR_OP_SPEC
0x00 0x0C	attribute-value.length = 12

0x00 0x0B	vattr-id = MDC_ECG_MANUAL_EVENT
0x00 0x00	op-target
0x40 0x00	options = supports-default(1),
0x20 0x00	level = op-level-professional(2),
0x00 0x00	group = 0
0x00 0x00	priority =0
0x09 0x53	attribute-id = MDC_ATTR_OP_STAT
0x00 0x02	attribute-value.length = 2
0x00 0x01	enable
0x09 0xA4	attribute-id = MDC_ATTR_VMO_REF
0x00 0x02	attribute-value.length = 2
0x00 0x05	HANDLE = 4

EVENT|RSP: OPER CREATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x00 0x04	invoke-id = 0x0003 (mirrored from invocation)
0x02 0x01	CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x0A	CHOICE.length = 10
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0xFF 0xFF 0xFF 0xFF	event-time not supported
0x0D 0x1D	event-type = MDC_NOTI_OP_CREAT
0x00 0x00	event-reply-info.length = 0

ACTION|REQ: OPERATION INVOKE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x90	CHOICE.length = 144
0x00 0x8E	OCTET STRING.length = 142

0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x01 0x07	CHOICE(Remote Operation Invoke | Confirmed Action)
0x00 0x88	CHOICE.length = 136
0x00 0x03	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x82	action-info-args.length = 130
0x00 0x00	checksum = not-used(0)
0xFF 0x33 0x44 0x55	invoke-cookie(should be random)
0x00 0x01	op-invoke-list.count
0x00 0x78	op-invoke-list.size = 120
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x01	op-instance-no = 1
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x28	new Bradycardia-Limit value= 40
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x02	op-instance-no = 2
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0xB4	new Taquicardia-Limit value= 180
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x03	op-instance-no = 3
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x5D 0xC0	new Asistolia-Limit value= 3 seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x04	op-instance-no = 4
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length =4
0x00 0x0E 0xa6 0x00	new Store-Time-Auto value= 120 seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x05	op-instance-no = 5
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x3D 0x09 0x00	new Store-Time-Manual value= 500 seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x06	op-instance-no = 6
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count

0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x02	new Qrs-Taqui-Number value= 2
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no = 7
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x02	attribute-value.length = 2
0x00 0x00 0x00 0x02	new Max-Search-Back value= 2
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no =8
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x02	attribute-value.length = 2
0x00 0x00 0x00 0x??	new Fading-Time value= 2 seg
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no =9
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x08	attributes.length =8
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x02	attribute-value.length = 2
0x17 0x3C	new Fading-Time Level= 60%
0x00 0x2d	op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
0x00 0x01	op-instance-no = 10
0x00 0x00	op-mod-type->op-replace(0)
0x00 0x01	attributes.count
0x00 0x06	attributes.length =6
0x09 0x32	attribute-id = MDC_ATTR_INDEX_SEL(2354)
0x00 0x02	attribute-value.length = 2
0x00 0x03	new gain :x4

ACTION|RSP: OPERATION INVOKE RESULT

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x14	CHOICE.length = 20
0x00 0x12	OCTET STRING.length = 18
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x02 0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00 0x0C	CHOICE.length = 12
0x00 0x00	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x06	action-info-args.length = 6
0xFF 0x33 0x44 0x55	invoke-cookie
0x00 0x00	OpInvResult= op-successful(0),

EVENT|REQ: OPERATION ATTRIBUTE UPDATE

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0xC8	CHOICE.length = 200
0x00 0xC6	OCTET STRING.length = 198

0x00 0x04	invoke-id = 0x0004
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0xC0	CHOICE.length = 192
0x00 0x03	obj-handle = 3 (Operating Scanner)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x0B	event-type = MDC_NOTI_OP_ATTR_UPDT(3339) (ASN.1 data type = OpAttributeInfo)
0x00 0xB2	event-info.length = 178
0x00 0x03	scan-report-no = 3
0x00 0x01	list.count = 1
0x00 0xAC	list.length = 172
0x00 0x00	context-id ->MdsContext=0 (The same as above)
0x00 0x01	list.count = 1
0x00 0xA6	list.length = 166
0x00 0x03	sco-handle=3
0xFF 0x33 0x44 0x55	invoke-cookie->INT-U32,
0x00 0x01	lock-state->AdministrativeState,
0x00 0x0B	op-elem-attr-up-list.count = 11
0x00 0x9E	op-elem-attr-up-list.length = 158
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x01	op-instance-no = 1
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x28	new Bradycardia-Limit value= 40
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x02	op-instance-no = 2
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0xB4	new Taquicardia-Limit value= 180
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x03	op-instance-no = 3
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x5D 0xC0	new Asistolia-Limit value=3 seg (3*8000)
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x04	op-instance-no = 4
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x0E 0xA6 0x00	new Store-Time-Auto value= 120 seg (120*8000)
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x05	op-instance-no = 5
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length =4
0x00 0x06 0x1A 0x80	new Store-Time-Auto value= 500 seg
0x00 0x2F	op-class-id =

	MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x06	op-instance-no = 6
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x03	new Numero-Qrs-Taqui value= 3
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no = 7
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x02	new Max-Search-Back value= 2
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no =8
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x04	attribute-value.length = 4
0x00 0x00 0x00 0x??	new Fading-Time value= 100 ms (.1 seg * 8000)
0x00 0x2F	op-class-id = MDC_MOC_CNTRL_OP_SEL_VAL
0x00 0x07	op-instance-no =9
0x00 0x04	attributes.count = 1
0x00 0x0A	attributes.length = 10
0x09 0x9D	attribute-id = MDC_ATTR_VAL_CURR(2461)
0x00 0x02	attribute-value.length = 4
0x00 0x00 0x17 0x3C	new Fading-Time value= 60%
0x00 0x2d	op-class-id = MDC_MOC_CNTRL_OP_SEL_IT
0x00 0x01	op-instance-no = 10
0x00 0x04	attributes.count = 1
0x00 0x08	attributes.length = 8
0x09 0x32	attribute-id = MDC_ATTR_INDEX_SEL(2354)
0x00 0x02	attribute-value.length = 2
0x00 0x03	Selected-Item-Index =3 (ganancia :x4)

<u>EVENT|RSP: OPERATION ATTRIBUTE UPDATE</u>

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x10	CHOICE.length = 16
0x00 0x0E	OCTET STRING.length = 14
0x00 0x04	invoke-id = 0x0004
0x01 0x01	CHOICE(Remote Operation Response | Confirmed Event Report)
0x00 0x08	CHOICE.length = 8
0x00 0x03	obj-handle = 3 (Operating Scanner)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x0B	event-type = MDC_NOTI_OP_ATTR_UPDT
0x00 0x00	event-info.length = 0

<u>ACTION|REQ: OPERATION INVOKE (Activate)</u>

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x22	CHOICE.length = 34
0x00 0x20	OCTET STRING.length = 32
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER

	encoded.)
0x01 0x07	CHOICE(Remote Operation Invoke | Confirmed Action)
0x00 0x1A	CHOICE.length = 26
0x00 0x03	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x14	action-info-args.length = 20
0x00 0x00	checksum = not-used(0)
0xFF 0x33 0x44 0x55	invoke-cookie(should be random)
0x00 0x01	op-invoke-list.count
0x00 0x0A	op-invoke-list.zise = 10
0x00 0x32	op-class-id = MDC_MOC_CNTRL_OP_ACTIV
0x00 0x0B	op-instance-no = 11
0x00 0x0A	op-mod-type->op-invokeAction(10)
0x00 0x00	attributes.count
0x00 0x00	attributes.length =0

ACTION|RSP: OPERATION INVOKE RESULT (Activate)

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x14	CHOICE.length = 20
0x00 0x12	OCTET STRING.length = 18
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x02 0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00 0x0C	CHOICE.length = 12
0x00 0x00	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x06	action-info-args.length = 6
0xFF 0x33 0x44 0x55	invoke-cookie
0x00 0x00	OpInvResult= op-successful(0),

EVENT|REQ: OPERATION ATTRIBUTE UPDATE(Activate)

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x32	CHOICE.length = 50
0x00 0x30	OCTET STRING.length = 48
0x00 0x04	invoke-id = 0x0004
0x01 0x01	CHOICE(Remote Operation Invoke | Confirmed Event Report)
0x00 0x2A	CHOICE.length = 42
0x00 0x03	obj-handle = 3 (Operating Scanner)
0x00 0x00 0x00 0x00	event-time = 0
0x0D 0x0B	event-type = MDC_NOTI_OP_ATTR_UPDT(3339) (ASN.1 data type =OpAttributeInfo)
0x00 0x20	event-info.length = 32
0x00 0x03	scan-report-no = 3
0x00 0x01	list.count = 1
0x00 0x1A	list.length = 26
0x00 0x00	context-id ->MdsContext=0 (The same as above)
0x00 0x01	list.count = 1
0x00 0x14	list.length = 20
0x00 0x03	sco-handle=3
0xFF 0x33 0x44 0x55	invoke-cookie->INT-U32,
0x00 0x01	lock-state->AdministrativeState,
0x00 0x01	op-elem-attr-up-list.count = 1
0x00 0x08	op-elem-attr-up-list.length = 8
0x00 0x2F	op-class-id = op-class-id = MDC_MOC_CNTRL_OP_ACTIV
0x00 0x01	op-instance-no = 1

0x00 0x00	attributes.count = 0
0x00 0x00	attributes.length = 0

EVENT|RSP: OPERATION ATTRIBUTE UPDATE(Activate)

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x14	CHOICE.length = 20
0x00 0x12	OCTET STRING.length = 18
0x00 0x05	invoke-id = 0x0002 (start of DataApdu. MDER encoded.)
0x02 0x07	CHOICE(Remote Operation Response | Confirmed Action)
0x00 0x0C	CHOICE.length = 12
0x00 0x00	obj-handle = 3 (SCO object)
0x0C 0x0B	action-type = MDC_ACT_SCO_OP_INVOKE
0x00 0x06	action-info-args.length = 6
0xFF 0x33 0x44 0x55	invoke-cookie
0x00 0x00	OpInvResult= op-successful(0),

GET|REQ: MDS ALL ATTRIBUTES

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x0E	CHOICE.length = 14
0x00 0x0C	OCTET STRING.length = 12
0x00 0x07	invoke-id = 0x0003
0x00 0x03	CHOICE(Remote Operation Invoke|Get)
0x00 0x06	CHOICE.length = 6
0x00 0x00	handle = 0 (MDS Object)
0x00 0x00	attribute-id-list.count = 0 (all atributes)
0x00 0x00	attribute-id-list.length = 0

GET|RSP: MDS ALL ATTRIBUTES

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x4C	CHOICE.length = 76
0x00 0x4A	OCTET STRING.length = 74
0x00 0x07	invoke-id = 0x0003
0x02 0x03	CHOICE(Remote Operation Invoke|Get)
0x00 0x66	CHOICE.length = 102
0x00 0x00	handle = 0 (MDS object)
0x00 0x01	attribute-list.count = 1
0x00 0x04	attribute-list.length = 4
0X0A 0X5A	attribute id = MDC_ATTR_SYS_TYPE_SPEC_LIST
0X00 0X08	attribute-value.length = 8
0x00 0x01	TypeVerList count = 1
0x00 0x04	TypeVerList length = 4
0x10 0x06	attribute-id = MDC_DEV_SPEC_PROFILE_ECG
0x00 0x01	version = version 1 of the specialization
0X09 0X28	attribute-id = MDC_ATTR_ID_MODEL
0x00 0x1A	attribute-value.length = 26
0x00 0x0A 0x54 0x68	string length = 10 | "TheCompany"
0x65 0x43 0x6F 0x6D	
0x70 0x61 0x6E 0x79	
0x00 0x0C 0x54 0x68	string length = 12 | "____________\0"
0x00 0x00 0x00 0x00	
0x00 0x00 0x00 0x00 0x00 0x00	
0x09 0x84	attribute-id = MDC_ATTR_SYS_ID
0x00 0x0A	attribute-value.length = 10

```
0x00 0x08 0x11 0x22 0x33 0x44 0x55 0x66 0x77 0x88 octet string length = 8 | EUI-64
0x0a 0x44                                         attribute-id = MDC_ATTR_DEV_CONFIG_ID
0x00 0x02                                         attribute-value.length = 2
0x40 0x00                                         dev-config-id = 16384 (extended-config-start)
```

SET|REQ: MDS TIME

```
0xE7 0x00                                         APDU CHOICE Type (PrstApdu)
0x00 0x1C                                         CHOICE.length = 28
0x00 0x1A                                         OCTET STRING.length = 26
0x00 0x08                                         invoke-id = 0x0004
0x01 0x05                                         CHOICE(Remote Operation Invoke |SET)
0x00 0x14                                         CHOICE.length = 20
0x00 0x00                                         obj-handle = 0 (MDS object)
0x00 0x01                                         modification-list.count=1

0x00 0x0E                                         modification-list.length=14
0x00 0x00                                         modify-operator=replace(0)

0x09 0x87                                         attribute.identifier=
                                                  MDC_ATTR_TIME_ABS(2439)
0x00 0x08                                         attribute.size=8
0x20 0x15 0x01 0x06                               AbsoluteTime
```

SET|RSP: MDS TIME

```
0xE7 0x00                                         APDU CHOICE Type (PrstApdu)
0x00 0x1A                                         CHOICE.length = 26
0x00 0x18                                         OCTET STRING.length = 24
0x00 0x08                                         invoke-id = 0x0004
0x02 0x05                                         CHOICE(Remote Operation Invoke |SET)
0x00 0x12                                         CHOICE.length = 18
0x00 0x03                                         obj-handle = 0 (MDS object)
0x00 0x01                                         attribute-list.count=1

0x00 0x0C                                         attribute-list.length=12
0x09 0x87                                         attribute.identifier=
                                                  MDC_ATTR_TIME_ABS(2439)
0x00 0x08                                         attribute.size=8
0x20 0x13 0x10 0x23                               AbsoluteTime
0x16 0x30 0x00 0x00
```

SET|REQ: OPERATING SCANNER (OPERATIONAL STATE=OFF)

```
0xE7 0x00                                         Type (PrstApdu)
0x00 0x16                                         CHOICE.length = 22
0x00 0x14                                         OCTET STRING.length = 20
0x00 0x05                                         invoke-id = 0x0005
0x01 0x05                                         CHOICE(Remote Operation Invoke |SET)
0x00 0x0E                                         CHOICE.length = 14
0x00 0x03                                         obj-handle = 3 (Operating Scanner object)
0x00 0x01                                         modification-list.count=1

0x00 0x08                                         modification-list.length=8

0x00 0x00                                         modify-operator=replace(0)

0x09 0x53                                         identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02                                         size=MDC_ATTR_OP_STAT(2387)
```

0x00 0x00	Operational-State=disabled(0)

SET|RSP: OPERATING SCANNER

0xE7 0x00	APDU CHOICE Type (PrstApdu)
0x00 0x12	CHOICE.length = 18
0x00 0x10	OCTET STRING.length = 16
0x00 0x05	invoke-id = 0x0005
0x02 0x05	CHOICE(Remote Operation Response |SET)
0x00 0x03	obj-handle = 3 (Operating Scanner object)
0x00 0x01	attribute-list.count=1
0x00 0x06	attribute-list.length=6
0x09 0x53	attribute-list[0].identifier=MDC_ATTR_OP_STAT(2387)
0x00 0x02	attribute-list[0].ANY_DEFINED_BY.size=MDC_ATTR_OP_STAT(2387)
0x00 0x01	attribute-list[0].ANY_DEFINED_BY.Operational-State=disabled(0)

ASSOC|REQ: RELEASE

0xE4 0x00	APDE CHOICE Type (RlreApdu)
0x00 0x02	CHOICE.length =2
0x00 0x00	reason = normal

ASSOC|RSP: RELEASE

0xE5 0X00	APDU CHOICE Type(RleApdu)
0x00 0x02	CHOICE.length = 2
0x00 0x00	reason = normal

References

[1] R. Schrenker and T. Cooper, "Building the Foundation for Medical Device Plug-and-Play Interoperability", MD&DI's Medical Electronics Manufacturing, 2001, pp 10, web: http://www.medicalelectronicsdesign.com/article/building-foundation-medical-device-plug-and-play-interoperability last accesed: 27/09/2015.

[2] Susan F. Whitehead, Julian M. Goldman, Getting Connected for Patient Safety: How Medical Device "Plug-and-Play" Interoperability Can Make a Difference, Patient Safety & Quality Healthcare, 2008, web:http://psqh.com/getting-connected-for-patient-safety-how-medical-device-plug-and-play-interoperability-can-make-a-difference

[3] Goldman, J. M., Schrenker, R. A., Jackson, J. L. & Whitehead, S. F. (2005). "Plug-and-play in the operating room of the future", Biomedical Instrumentation and Technology, 2005, vol. 39(3), pp 194-199.

[4] M. Galarraga, M. Martinez-Espronceda, L. Serrano, I. Martinez, and M. deToledo, Necesidad de Estandares para Interoperabilidad de Dispositivos Medicos: Donde y Por Que," in XXIV Congreso Anual de la Sociedad Española de Ingenieria Biomedica (CASEIB), 2006, pp. 415-418, ISBN: 84-9769-159-8.

[5] Miguel Galarraga, Luis Serrano, Ignacio Martínez, Paula de Toledo, Standards for medical device communication: X73 POC-MDC, Studies in health technology and informatics, 2006,vol 121, pp 242.

[6] ISO/TC 215: Health informatics, last visit: September 2015. [Online]. Available: http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=54960

[7] "Health Informatics-Point of care, Part 10101: Nomenclature," IEEE Std 11073-10101, 2004.

[8] "Health Informatics-Point of care, Part 10201: Domain Information Model" IEEE Std 11073-10201, 2004.

[9] "Health Informatics-Point of care, Part 20301: MDAP optional package, remote control" IEEE Draft 11073-20301.

[10] "World population ageing: 1950-2050 - executive summary," Tech. Rep., 2004, last visit: September 2015. [Online]. Available: http://www.un.org/esa/population/publications/worldageing19502050/

[11] W. Lutz, W. Sanderson, and S. Scherbov, "The end of world population growth," Nature, 2001, vol. 412, no. 6846, pp. 543-545, DOI: 10.1038/35087589.

[12] European population Committee of the Council of Europe, "Recent demographic developments in Europe 2005, (2006). ISBN 978-92-871-5973-1.

[13] J. Alho, M. Alders, H. Cruijsen, N. Keilman, T. Nikander, and D. Pham, "New forecast: Population decline postponed in Europe," Statistical Journal of the United Nations Economic Commission for Europe,2006, vol. 23, no. 1, pp. 1-10, 2006.

[14] J. Corchado, J. Bajo, and A. Abraham, “GerAmi: Improving healthcare delivery in geriatric residences",2008, IEEE Intelligent Systems, vol. 23, no. 2, pp. 19-25.

[15] G. Paré, M. Jaana, and C. Sicotte, “Systematic review of home telemonitoring for chronic diseases: The evidence base," Journal of the American Medical Informatics Association,2007, vol. 14, no. 3, pp. 269-277,

[16] J. Parkka, M. Van Gils, T. Tuomisto, R. Lappalainen, and I. Korhonen, “A wireless wellness monitor for personal weight management," in Proceedings of the IEEE/EMBS Region 8 International Conference on Information Technology Applications in Biomedicine, ITAB,2000, Arlington, VA, pp. 83-88.

[17] M. Rasid and B. Woodward, “Bluetooth telemedicine processor for multichannel biomedical signal transmission via mobile cellular networks," IEEE Transactions on Information Technology in Biomedicine, vol. 9, no. 1, pp. 35-43, 2005.

[18] Paula De Toledo, Silvia Jiménez, Francisco Del Pozo, A telemedicine system to support a new model for care of chronically ill patients,Journal of Telemedicine and Telecare,2002,vol 8, pp 17-19.

[19] Paré G. Jaana M, Sicotte C, “Systematic review of home telemonitoring for chronic diseases: the evidence base”, J Am Med Inform Assoc, 2007, vol. 14, pp 269-77.

[20] Corchado, J.M. Bajo, J. Abraham, A.,GerAmi: “Improving Healthcare Delivery in Geriatric Residences,Intelligent systems”, IEEE,2008, vol. 23, pp. 19-25, DOI:10.1109/MIS.2008.27.

[21] Miguel Galarraga, Luis Serrano, I Martinez, Paula de Toledo, Melvin Reynolds, “Telemonitoring systems interoperability challenge: an updated review of the applicability of ISO/IEEE 11073 standards for interoperability in telemonitoring,” In; Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2007, pp. 6161–6165, DOI: 10.1109/IEMBS.2007.4353761.

[22] Miguel Galarraga, Ignacio Martinez, Luis Serrano, Paula de Toledo, J Escayolan, Julián Fernández, Silvia Jimenez-Fernandez, Santiago Led, Miguel Martinez-Espronceda, E Viruete, José García, “Proposal of an ISO/IEEE11073 platform for healthcare telemonitoring: plug-and-play solution with new use cases”, in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, 2007,pp. 6709 – 6712, DOI: 10.1109/IEMBS.2007.4353900.

[23] I Martínez, J Fernández, M Galarraga, L Serrano, P de Toledo, J Escayola, S Jiménez-Fernández, S Led, M Martínez-Espronceda, J García, Implementation experience of a patient monitoring solution based on end-to-end standards, in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society,EMBS, 2007,pp. 6425 – 6428, DOI:10.1109/IEMBS.2007.4353827.

[24] M. Martinez-Espronceda, I. Martinez, L. Serrano, S. Led, J. D. Trigo, A. Marzo, J. Escayola, H.G Barron-González and J. Garcia, Lessons Learned Implementing the ISO/IEEEl1073 Standard into Wearable Personal Devices, in Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine, ITAB, 2010, pp. 1 – 4, DOI: 10.1109/ITAB.2010.5687618 .

[25] F Wartena, J. Muskens and L. Schmitt, “Continua: The Impact of a Personal Telehealth Ecosystem”, in Proceedings of the International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED, 2009, pp. 13 – 18, DOI:10.1109/eTELEMED.2009.8.

[26] S. Led, M. Martínez-Espronceda, J. Redondo,A. Baquero,L. Serrano, L. Cabezas, M . Niegowski, Wearable electrocardiogram (ECG) recorder for a Mobile Point-of-Care based on recent interoperability standards, in Proceedings of the IEEE International Conference of Point-of-Care Healthcare Technologies (PHT), 2010,pp. 287-290, DOI: 10.1109/PHT.2013.6461341.

[27] J.D. Trigo, F. Chiarugi, A. Alesanco, M. Martinez-Espronceda, L. Serrano, C. E. Chronaki, J. Escayola, I. Martinez, and J. Garcia, Interoperability in digital electrocardiography: harmonization of ISO/IEEE x73-PHD and SCP-ECG, IEEE Transactions on Information Technology in Biomedicine, 2010, vol.14, no. 3, pp. 1303, DOI: 10.1109/TITB.2010.2064330.

[28] N. Jae-Choong, S. Won-Kyeong, B. Jae-Seoung,Cho You-Ze, Design and Development of a u-Health System Based on the ISO/IEEE11073 PHD Standards, in Proceedings of the IEEE 17th Asia-Pacific Conference on Communications, 2011, pp. 789-793, DOI:10.1109/APCC.2011.6152915.

[29] Z. Daidi, M. Kirwan, S. Duan, Regulatory Barriers Blocking Standardization of Interoperability, JMIR Mhealth Uhealth, 2013, DOI: 10.2196/mhealth.2654.

[30] Personal Health Device Working Group (PHD-WG) Home Page. [Online] Available from: URL: http://standards.ieee.org/develop/wg/PHD.html Last visit: 9/15.

[31] IEEE Std 11073-20601-2008. Application profile-Optimized Exchange Protocol. IEEE Std 11073-104xx. Part 104xx: Device specializations.

[32] IEEE Std 11073-20601-2010, Application profile - Optimized Exchange Protocol. IEEE Std 11073-104xx. Part 104xx: Device specializations.

[33] Continua Health Alliance. Available from: URL: http://continuaalliance.org, Last visit: 9/15.

[34] H. G. Barrón-González, M. Martínez-Espronceda, S. Led, L. Serrano, "Research Benefits of Using Interoperability Standards in Remote Command and Control to Implement Personal Health Devices", in proceedings of XIII Mediterranean Conference on Medical and Biological Engineering and Computing, 2013,vol. 41, pp 1459-1463.

[35] Hector Gilberto Barrón-González, Miguel Martínez-Espronceda, J. D. Trigo, Santiago Led, Luis Serrano,"Recent Advances in mHealth: An Update to Personal Health Device Interoperability Based on ISO/IEEE11073", Mobile Health: A technological Road Map,2015, vol. 3, pp 375-397, DOI:10.1007/978-3-319-12817-7_17.

[36] PHD-WG, Active Projects Managed by this Working Group, Available from: URL: https://standards.ieee.org/develop/wg/PHD.html, Last visit: 9/15.

[37] Blount M, "Remote health-care monitoring using Personal Care Connect", in Proceedings of the IBM Systems Journal , 2007, vol 46, pp 95 - 113, DOI: 10.1147/sj.461.0095.

[38] Tae-Soo Lee, Joo-Hyun Hong and Myeong-Chan Cho, "Biomedical Digital Assistant for Ubiquitous Healthcare", in Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007, pp 1790 - 1793, DOI: 10.1109/IEMBS.2007.4352659.

[39] De Capua, C, Meduri, A. and Morello, R, "A remote doctor for homecare and medical diagnoses on cardiac patients by an adaptive ECG analysis", IEEE International Workshop on Medical Measurements and Applications, 2009, pp 31 - 36, DOI: 10.1109/MEMEA.2009.5167949.

[40] Jones, V, Gay, V. and Leijdekkers, P, "Body Sensor Networks for Mobile Health Monitoring: Experience in Europe and Australia", in Proceedings of the Fourth International Conference on Digital Society, 2010, pp 204 - 209, DOI: 10.1109/ICDS.2010.41.

[41] H.G. Barrón-González, , M. Martínez-Espronceda, S. Led, L. Serrano, ,Christoph Fischer, M. Clarke, "New Use Cases for Remote Control and Configuration of Interoperable Medical Devices",in Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013, pp 4787 - 4790, DOI 10.1109/EMBC.2013.6610618.

[42] Lasierra, N., Alesanco, Á., García, J. “An SNMP-based solution to enable remote ISO/IEEE 11073 technical management”, IEEE Transactions on Information Technology in Biomedicine, 16 (4), art.no. 6178799, pp. 709-719. 2012.

[43] Aenor, UNE-ENV 13734:2002, . [Online] Available from: URL: http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0026411#.Vgs0ivmqqko Last visit: 9/15.

[44] Aenor, UNE-ENV 13735:2002, [Online] Available from: URL: http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0026410#.Vgs4-_mqqko Last visit: 9/15.

[45] David F. Franklin and David V. Ostler, "The P1073 Medical Information Bus", IEEE Micro, 2002, pp 52 - 60, DOI: 10.1109/40.45827.

[46] J.J. Garguilo, S. Martinez, R. Rivello, M. Cherkaoui, "Moving Toward Semantic Interoperability of Medical Devices", In; Proceedings of High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007, pp 13-19, DOI:10.1109/HCMDSS-MDPnP.2007.16.

[47] A. Schloegl, F Chiarugi, E. Cervesato, E Apostolopoulos,"Two-way converter between the HL7 aECG and SCP-ECG data formats using BioSig", Computers in Cardiology, 2007, pp 253-256, DOI:10.1109/CIC.2007.4745469.

[48] Mujumdar, S. ; Thongpithoonrat, P. ; Gurkan, D. ; McKneely, P.K.; "Performance analysis of a proposed tightly-coupled medical instrument network based on can protocol", In; Proceedings of Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2010,pp 360-364, DOI:10.1109/IEMBS.2010.5627692.

[49] Strahle, M. ; Drugerwerk AG, Lubeck ; Ehlbeck, M. ; Prapavat, V. ; Kuck, K.;"Towards a Service-Oriented Architecture for Interconnecting Medical Devices and Applications", In; Proceedings of Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, 2007. HCMDSS-MDPnP, pp 25-27, DOI:10.1109/HCMDSS-MDPnP.2007.27.

[50] Thongpithoonrat, P. ; McKneely, P.K. ; Gumudavelli, S. ; Gurkan, D.; Chapman, F.M.; "Networking and plug-and-play of bedside medical instruments", In; Proceedings of 30th Annual International Conference of the

IEEE Engineering in Medicine and Biology Society, 2008. EMBS 2008, pp 1514-1517, DOI:10.1109/IEMBS.2008.4649456.

[51] Ahn, J., Heo, J., Lim, S., Kim, W.,"A study of ISO/IEEE 11073 platform based on power line communication", In; Proceedings of 14th Asia-Pacific Conference on Communications, APCC 2008, art. no. 4773813.

[52] Gumudavelli, S., McKneely, P.K., Thongpithoonrat, P., Gurkan, D., Chapman, F.M.,"Medical instrument data exchange", In; Proceedings of Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology", art. no. 4649530, pp. 1809-1812.

[53] Galarraga, M., Serrano, L., Martínez, I., De Toledo, P.,"Standards for medical device communication: X73 POC-MDC", Studies in Health Technology and Informatics, 2006, Volume 121, 2006, Pages 242-256.

[54] Warren, S., Lebak, J., Yao, J.,"Lessons learned from applying interoperability and information exchange standards to a wearable point-of-care system" In; Conference Proceedings - 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, D2H2 2006, 2006, art. no. 1624807, pp. 101-104. DOI: 10.1109/DDHH.2006.1624807.

[55] Lutter, N., Norgall, T., Mell, J., Weigan, C., Schuettler, J., "Point of care: New connectivity standards and novel technologies in intensive care", In; Proceedings of (2005) International Journal of Intensive Care, 12 (4), pp. 175-185.

[56] Oh, H., Rizo, C., Enkin, M., Jadad, A., ¨What is eHealth (3): A systematic review of published definitions¨, (2005) Journal of Medical Internet Research, 7 (1), art. no. e1, DOI: 10.2196/jmir.7.1.e1.

[57] Koch, S. ¨Home telehealth-Current state and future trends¨,(2006) International Journal of Medical Informatics, 75 (8), pp. 565-576,DOI: 10.1016/j.ijmedinf.2005.09.002.

[58] Pagliari, C., Sloan, D., Gregor, P., Sullivan, F., Detmer, D., Kahan, J.P., Oortwijn, W., MacGillivray, S., ¨What is eHealth (4): A scoping exercise to map the field¨, (2005) Journal of Medical Internet Research, 7 (1), art. no. e9, DOI: 10.2196/jmir.7.1.e9.

[59] Black, A.D., Car, J., Pagliari, C., Anandan, C., Cresswell, K., Bokun, T., McKinstry, B., Procter, R., Majeed, A., Sheikh, A., ¨The impact of ehealth on the quality and safety of health care: A systematic overview¨, (2011) PLoS Medicine, 8 (1), art. no. e1000387, DOI: 10.1371/journal.pmed.1000387.

[60] Bennett, G.G., Glasgow, R.E.,"The delivery of public health interventions via the internet: Actualizing their potential", (2009) Annual Review of Public Health, 30, pp. 273-292, DOI: 10.1146/annurev.publhealth.031308.100235.

[61] Chan, M., Estève, D., Fourniols, J.-Y., Escriba, C., Campo, E., “Smart wearable systems: Current status and future challenges,” (2012) Artificial Intelligence in Medicine, 56 (3), pp. 137-156,DOI: 10.1016/j.artmed.2012.09.003.

[62] Hesse, B.W., Shneiderman, B., "eHealth Research from the User's Perspective", (2007) American Journal of Preventive Medicine, 32 (5 SUPPL.), pp. S97-S103, DOI: 10.1016/j.amepre.2007.01.019.

[63] Ahern, D.K., Kreslake, J.M., Phalen, J.M. "What is eHealth (6): Perspectives on the evolution of eHealth research", (2006) Journal of Medical Internet Research, 8 (1), pp. 12P, DOI: 10.2196/jmir.8.1.e4.

[64] Hardinge, M., Rutter, H., Velardo, C., Shah, S.A., Williams, V., Tarassenko, L., Farmer, A., "Using a mobile health application to support self-management in chronic obstructive pulmonary disease: A six-month cohort study eHealth/ telehealth/ mobile health systems", (2015) BMC Medical Informatics and Decision Making, 15 (1), art. no. 46,DOI: 10.1186/s12911-015-0171-5 .

[65] Bensley, R.J., Mercer, N., Brusk, J.J., Underhile, R., Rivas, J., Anderson, J., Kelleher, D., Lupella, M., de Jager, A.C., "The eHealth Behavior Management Model: a stage-based approach to behavior change and management", (2004) Preventing chronic disease, 1 (4), pp. A14.

[66] Iyngkaran, P., Toukhsati, S.R., Biddagardi, N., Zimmet, H., J.Atherton, J., Hare, D.L., "Technology-Assisted Congestive Heart Failure Care", (2015) Current Heart Failure Reports, 12 (2), pp. 173-186, DOI: 10.1007/s11897-014-0251-3.

[67] Gee, P.M., Greenwood, D.A., Paterniti, D.A., Ward, D., Miller, L.M.S., "The eHealth enhanced chronic care model: A theory derivation approach", (2015) Journal of Medical Internet Research, 17 (4), p. e86, DOI: 10.2196/jmir.4067.

[68] Zulman, D.M., Jenchura, E.C., Cohen, D.M., Lewis, E.T., Houston, T.K., Asch, S.M. "How Can eHealth Technology Address Challenges Related to Multimorbidity? Perspectives from Patients with Multiple Chronic Conditions", (2015) Journal of General Internal Medicine, 30 (8), pp. 1063-1070,DOI: 10.1007/s11606-015-3222-9.

[69] Rimoldi, C.M.;"Standardizing USB in medical applications", Electronic Products (2010) (Garden City, New York), 52 (2), ISSN: 00134953.

[70] Anthony G. A. Aggidis, Jeffrey D. Newman, George A. Aggidis, ¨Investigating pipeline and state of the ar tblood glucose biosensors to formulate next steps¨, 2015, Biosensors and Bioelectronics, 74, pp. 243-262,DOI: 10.1016/j.bios.2015.05.071.

[71] Azucena, H., Ríos, E., Peña, R.D., Díaz, J. ¨Design and implementation of a simple portable biomedical electronic device to diagnose cardiac arrhythmias", Sensing and Bio-Sensing Research, 2015, 4, pp. 1-10, DOI: 10.1016/j.sbsr.2015.01.001.

[72] Neal, R.E., II, Kavnoudias, H., Thomson, K.R. ¨An “Off-the-Shelf” System for Intraprocedural Electrical Current Evaluation and Monitoring of Irreversible Electroporation Therapy¨, CardioVascular and Interventional Radiology, 2015, 38 (3), pp. 736-741, DOI: 10.1007/s00270-014-0961-7.

[73] Sarkar, D., Chowdhury, A.¨Low cost and efficient ECG measurement system using PIC18F4550 microcontroller¨, International Conference on Electronic Design, Computer Networks and Automated Verification, EDCAV 2015, art. no. 7060529, pp. 6-11, DOI: 10.1109/EDCAV.2015.7060529.

[74] ZigBee Standards, last visit: September 2015. [Online]. Available: http://www.zigbee.org/

[75] Continua Health Alliance, Continua Guidelines 2010, last visit: September 2015. [Online]. Available: http://www.continuaalliance.org/products/design-guidelines.

[76] Dheerendra S. Gangwar, "Biomedical Sensor Network for Cardiovascular Fitness and Activity Monitoring", in Procedings of IEEE Point-of-Care Healthcare Technologies (PHT), 2013, pp 279-282, DOI:10.1109/PHT.2013.6461339.

[77] Clarke, M.; de Folter, J.; Palmer, C.; Verma, V.," Building point of care health technologies on the IEEE 11073 health device standards", in Proceedings of IEEE Point-of-Care Healthcare Technologies (PHT),2013, pp 117-119, DOI: 10.1109/PHT.2013.6461298.

[78] Steve Warren, Jeffrey Lebak, Jianchu Yao, Jonathan Creekmore, Aleksandar Milenkovic, and Emil Jovanov,"Interoperability and Security in Wireless Body Area Network Infrastructures", in Proceedings of the 27th Annual International Conference of the IEEE-EMBS 2005, pp 3837-3840,DOI: 10.1109/IEMBS.2005.1615297.

[79] Jiho Kim, Ohyoung Song,"ISO/IEEE 11073 interoperability for personal health devices based on ZigBee healthcare service", in Proceedings of the 2015 IEEE International Conference on Consumer Electronics (ICCE), pp 263-264,DOI: 10.1109/ICCE.2015.7066406.

[80] Park, Chan-Yong; Joon-Ho Lim; Soojun Park,"ISO/IEEE 11073 PHD standardization of legacy healthcare devices for home healthcare services", in Proceedings of the 2015 IEEE International Conference on Consumer Electronics (ICCE), pp 547-548, DOI: 10.1109/ICCE.2011.5722731.

[81] Bluetooth Special Interest Group, Bluetooth Standard, last visit: September 2015. [Online]. Available: https://www.bluetooth.org/

[82] Bluetooth Special Interest Group, Health Device Profile, last visit: September 2015. [Online]. Available: https://www.bluetooth.org/en-us/specification/assigned-numbers/health-device-profile

[83] Led, S.; Fernandez, J.; Serrano, L.," Design of a Wearable Device for ECG Continuous Monitoring Using Wireless Technology", in Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004, vol. 2, pp 3318-3321,DOI: 10.1109/IEMBS.2004.1403933.

[84] Escayola, J. ; Martínez-Espronceda, M. ; Serrano, L. ; Trigo, J.D. ; Led, S. ; García, J.,"Implementation experiences of ISO/IEEE11073 standard applied to new use cases for e-health environments", in Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009,pp 1679-1682, DOI:10.1109/IEMBS.2009.5333883.

[85] Santiago Led, Jorge Fernández and Luis Serrano."Design of a Wearable Device for ECG Continuous Monitoring Using Wireless Technology ",in Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004. IEMBS '04, vol. 2, pp 3317-3321, DOI:10.1109/IEMBS.2004.1403933.

[86] ISO/IEEE 11073-10404TM-2008,"Health informatics-Personal health device communication (X73-PHD). Device Specialization-Pulse Oximeter."

[87] ISO/IEEE 11073-10406^{TM}-2011,"Health informatics-Personal health device communication (X73-PHD). Device Specialization – Basic electrocardiograph (ECG)."

[88] ISO/IEEE 11073-10407^{TM}-2008,"Health informatics-Personal health device communication (X73-PHD). Device Specialization-Blood Pressure Monitor."

[89] ISO/IEEE 11073-10408^{TM}-2008,"Health informatics-Personal health device communication (X73-PHD). Device Specialization-Thermometer."

[90] ISO/IEEE 11073-10415^{TM}-2008,"Health informatics-Personal health device communication (X73-PHD). Device Specialization-Weighing Scale."

[91] ISO/IEEE 11073-10417^{TM}-2015,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Glucose meter.”

[92] ISO/IEEE 11073-10418^{TM}-2011,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-International Normalized Ratio (INR) Monitor.”

[93] ISO/IEEE 11073-10419^{TM}-2015,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Insulin Pump.”

[94] ISO/IEEE 11073-10420^{TM}-2010,”Health informatics-Personal health device communication (X73-PHD). Device Specialization- Body composition analyzer.”

[95] ISO/IEEE 11073-10421^{TM}-2010,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Peak expiratory flow monitor (peak flow).”

[96] ISO/IEEE 11073-10424^{TM}-2014,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Sleep Apnea Breathing Therapy Equipment (SABTE).”

[97] ISO/IEEE 11073-10425^{TM}-2014,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Continuous Glucose Monitor.”

[98] ISO/IEEE 11073-10441^{TM}-2013,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Cardiovascular fitness and activity monitor.”

[99] ISO/IEEE 11073-10442^{TM}-2008,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Strength fitness equipment.”

[100] ISO/IEEE 11073-10471^{TM}-2008,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Independent living activity hub.”

[101] ISO/IEEE 11073-10472^{TM}-2010,”Health informatics-Personal health device communication (X73-PHD). Device Specialization-Medication monitor.”

[102] IEEE P11073-10423,”Draft Standard for Health informatics. -Personal health device communication (X73-PHD). Device Specialization-Sleep Monitor.”

[103] IEEE P11073-10426,”Draft Standard for Health informatics. -Personal health device communication (X73-PHD). Device Specialization-Home Healthcare Environment Ventilator.”

[104] IEEE P11073-10427,”Draft Standard for Health informatics. -Personal health device communication (X73-PHD). Device Specialization-Monitoring Power Status of Critical Care Devices Equipment.”

[105] IEEE 11073-20601a-2010 - IEEE Health informatics--Personal health device communication Part 20601: Application profile--Optimized Exchange Protocol Amendment 1.

[106] 11073-20601-2014 - Health informatics--Personal health device communication - Part 20601: Application profile- Optimized Exchange Protocol.

[107] E. Mattila, J. Parkka, M. Hermersdorf, J. Kaasinen, J. Vainio, K.Samposalo, J. Merilahti,J. Kolari, M. Kulju, R. Lappalainen, I. Korhonen, ”Mobile diary for Wellness Management- Results on Usage and Usability in Two user studies”, IEEE Transactions on Information Technology in biomedicine, 2008, vol. 12, Issue: 4, pp 501-512, DOI: 10.1109/TITB.2007.908237.

[108] S. Adibi, “Ling Technologies and BlackBerry Mobile Health (mHealth) Solutions: A Review”, IEEE Transactions on Information Technology in Biomedicine, 2012, vol. 16, Issue: 4, pp 586-597, DOI: 10.1109/TITB.2012.2191295.

[109] Hao Chen, Xueqin Jia, “New requirements and trends of mHealth”, In: Proceedings of IEEE 14th International Conference on e-Health Networking, 2012, pp 27-31, DOI: 10.1109/HealthCom.2012.6380060.

[110] P. Ming-Zher, K. Kyunghee, A. Goessling, N. Senwon, R. Picard, “Cardiovascular Monitoring Using Earphones and a Mobile Device”, IEEE Pervasive Computing, 2012, vol. 11, Issue:4, pp 18-26, DOI: 10.1109/MPRV.2010.91.

[111] A. Minutolo, G. Sannino, M. Esposito, G. De Pietro, “A rule-based mHealth system for cardiac monitoring”,In; Proceedings of EMBS Conference on Biomedical Engineering and Sciences (IECVES), 2010, pp 144-149, DOI: 10.1109/IECBES.2010.5742217.

[112] D. Hayn, B. Jammerbund, G. Schreier, “ECG quality assessment for patient empowerment in mHealth applications”, Computing in Cardiology, 2011, pp 353-356, ISSN: 0276-6547, ISBN: 978-1-4577-0612-7.

[113] R.K. Lomotey, S. Jamal, R. Deters, “SOPHRA: A Mobile Web Services Hosting Infrastructure in mHealth”, In: Proceedings of IEEE First International Conference on Mobile Services (MS), 2012, pp 24-29, DOI: 10.1109/MobServ.2012.14.

[114] Antonio J. Jara, Miguel A. Zamora-Izquierdo, Antonio F. Skameta, “Interconnection Framework for mHealth and Remote Monitoring Based on the Internet of Things”, IEEE Journal on Selected Areas in Communications,2013, pp 47-65, DOI: 10.1109/JSAC.2013.SUP.0513005.

[115] S.G. Mouglakakou, I. Kourls, D. Lliopoulou, A. Vazeou, “Mobile technology to empower people with Diabetes Mellitus: Design and development of a mobile application”, In; Proceedings of 9th International Conference on Information Technology and Applications in Biomedicine (ITAB 2009), pp 1-4, DOI: 10.1109/ITAB.2009.5394344.

[116] S. Dumanli, S. Gormus, I.J. Craddock, “Energy efficient body area networking for mHealth applications”, In; Proceedings of 6th International Symposium on Medical Information and Communication Technology (ISMICT), 2012, pp 1-4, DOI: 10.1109/ISMICT.2012.6203037.

[117] W. Zhang, P. Passow, E. Jovanov, R. Stoll, K. Thurow, “A secure and scalable telemonitoring system using ultra-low-energy wireless sensor interface for long-term monitoring in life science applications”, In; Proceedings of IEEE International Conference on Automation Science and Engineering (CASE), 2013, pp 617-622, DOI: 10.1109/CoASE.2013.6653979.

[118] V. Misra, A. Bozkurt, B. Calhoun,T Jackson,” Flexible Technologies for Self-Powered Wearable Health and Environmental Sensing”, In; Proceedings of IEEE, 2015, pp 665-681, 10.1109/JPROC.2015.2412493.

[119] Miguel Martínez-Espronceda, Jesús D. Trigo, Santiago Led, H. Gilberto Barrón-González, Javier Redondo, Alfonso Baquero, Luis Serrano, “Event-Driven, Pattern-Based Methodology for Cost-Effective Development of Standardized Personal Health Devices”, Computer Methods and Programs in

Biomedicine, 2014, vol 117(2), pp. 168-178, DOI: 10.1016/j.cmpb.2014.07.008.

[120] G. Lenz, M. Frohner, S. Sauermann, M. Forjan, in: A. Hörbst,et al. (Eds.), LUMOR: An App for Standardized Control and Monitoring of a Porcine Lung and its Nutrient Cycle, eHealth2014 – Health Informatics Meets eHealth, 2014, http://dx.doi.org/10.3233/978-1-61499-397-1-79.

[121] Nicolae Goga, Stefania Costache, "A formal analysis of ISO/IEEE P11073-20601 standard of medical device communication", In; Procedings of the 3rd Annual IEEE Systems conference, 2009, pp 163-166, DOI: 10.1109/SYSTEMS.2009.4815792.

[122] Lee, Y.-F., "An interoperability solution for legacy healthcare devices", (2015) IT Professional, 17 (1), art. no. 7030146, pp. 51-57, DOI: 10.1109/MITP.2015.2.

[123] Lee, Y.-F.,"Personal medical monitoring system: Addressing interoperability", (2013) IT Professional, 15 (5), art. no. 6297970, pp. 31-37, DOI: 10.1109/MITP.2012.90.

[124] Piniewski, B., Muskens, J., Estevez, L., Carroll, R., Cnossen, R., "Empowering healthcare patients with smart technology", (2010) Computer, 43 (7), art. no. 5510876, pp. 27-34, DOI: 10.1109/MC.2010.200.

[125] Carroll, R., Cnossen, R., Schnell, M., Simons, D., "Continua: An interoperable personal healthcare ecosystem," (2007) IEEE Pervasive Computing, 6 (4), pp. 90-94. Cited 83 times, DOI: 10.1109/MPRV.2007.72.

[126] Baig, M.M., Gholamhosseini, H.,"Smart health monitoring systems: An overview of design and modeling", (2013) Journal of Medical Systems, 37 (2), art. no. 9898, DOI: 10.1007/s10916-012-9898-z.

[127] L. Caranguian, S. Pancho-Festin, L. Sison, Device interoperability and authentication for telemedical appliance based on the ISO/IEEE 11073 personal health device (PHD) standards, in: Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE, pp. 1270-1273.

[128] A. Egner, A. Soceanu, F. Moldoveanu, Managing secure authentication for standard mobile medical networks, in: 2012 IEEE Symposium on Computers and Communications (ISCC), IEEE, pp. 390-393.

[129] A. Kliem, M. Hovestadt, O. Kao, Security and communication architecture for networked medical devices in mobility-aware, in Proceedings of the 2012 IEEE First International Conference on Mobile Services (MS), 2012, pp. 112 - 114, DOI:10.1109/MobServ.2012.15.

[130] S. S. Kim, Y. H. Lee, J. M. Kim, D. S. Seo, G. H. Kim, Y. S. Shin, Privacy protection for personal health device communication and healthcare building applications, Journal of Applied Mathematics 2014 (2014), Article ID 462453, 5 pages [Online] Available from: URL:http://dx.doi.org/10.1155/2014/462453. Accessed: 13/08/15.

[131] Spin model checker available [Online]: http://spinroot.com/spin/whatispin.html :

[132] N. Goga, S. Costache, F. Moldoveanu, A formal analysis of ISO/IEEE P11073-20601 standard of medical device communication, in Proceedings of the 2009 IEEE International Systems Conference Proceedings,2009, art. no. 4815792, pp. 163-166.

[133] mCRL2 model checker toolset available [Online]: http://www.mcrl2.org/release/user_manual/index.html

[134] J. J. A. Keiren, M. D. Klabbers, Modelling and verifying IEEE Std 11073-20601 session setup using mCRL2, in Proceedings of the 12th International Workshop on Automated Verification of Critical Systems (AVoCS 2012, ISSN 1863-2122), v. 53, 2012.

[135] Héctor Gilberto Barrón-González, Miguel Martínez-Espronceda, Jesús Daniel Trigo, Santiago Led, Luis Serrano, "Lessons learned from the implementation of remote control for the interoperability standard ISO/IEEE11073-20601 in a standard weighing scale", Computer Methods and Programs in Biomedicine, 2015, DOI: 10.1016/j.cmpb.2015.09.015.

[136] FreeRTOS [Online] Available from: http://www.freertos.org/

[137] FreeRTOS API Reference. [Online] Available from: http://www.freertos.org/a00106.html

[138] Hercules SETUP utility. [Online] Available from: http://www.hw-group.com/products/hercules/index_en.html

[139] ARM Architecture for the Digital WorldContact ARM English [Online] Available from: URL: http://www.arm.com/products/processors/cortex-m/cortex-m3.php

[140] Low Power Processors [Online] Available from: URL: http://www.ti.com/general/docs/gencontent.tsp?contentId=46880