



# ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación :

INGENIERO INDUSTRIAL

Título del proyecto:

DISEÑO DE UN PLC INDUSTRIAL USANDO  
HARDWARE LIBRE.

Autor: Raúl Rivero Leo

Tutor: Ángel María Andueza Unanua

Pamplona, 26 junio de 2015

# TABLA DE CONTENIDO

Tabla de contenido.....	3
Índice de Figuras .....	7
Índice de Tablas.....	13
Objetivo del proyecto. ....	15
Estado del arte de los PLC.....	17
Contexto PLC.....	17
Open Source y Open Source Hardware .....	19
Arduino.....	19
Raspberry Pi .....	22
PLC OpenSource .....	24
Arquitectura .....	33
Definición de necesidades.....	34
Lista de Hardware .....	35
Arduino Mega 2560 .....	36
Raspberry Pi .....	37
Pantalla LCD de 2 líneas.....	38
Teclado Keypad Numérico.....	39
Relé de 8 canales .....	39
Buzzer de aviso sonoro .....	40
Entradas digitales .....	41
Salidas digitales.....	41
Entradas Analógicas .....	41
Entradas de lazo de Corriente.....	42
Fuente de alimentación.....	42
Diseño de Esquemáticos .....	43
Esquemático Raspberry Pi. ....	43
Esquemático Arduino Mega.....	45

Esquemático LCD .....	48
Esquemático Keypad Numérico .....	48
Esquemático Relé 8 Canales .....	49
Esquemático Buzzer de aviso sonoro .....	51
Esquemático Entradas digitales .....	51
Esquemático Salidas digitales .....	53
Esquemático Sensor Analógico .....	55
Esquemático Entradas Lazo de Corriente .....	56
Esquemático Fuente de Alimentación .....	58
Esquemático PCB Current Tester .....	59
Diseño del Layout .....	61
Reglas de Diseño .....	63
Colocación de los componentes en la PCB .....	65
Layout por zonas .....	65
Layout Arduino Mega 2560 .....	66
Layout Raspberry Pi .....	67
Layout Pantalla LCD de 2 líneas .....	67
Layout Keypad Numérico .....	68
Layout Relé 8 canales .....	68
Layout Buzzer .....	69
Layout Salidas Digitales .....	69
Layout Entradas Digitales .....	69
Layout Sensor Analógico .....	70
Layout Entrada lazo de corriente. ....	70
Layout Entradas fuente de alimentación .....	71
Programación .....	73
Arduino .....	73
Raspberry Pi .....	80

Programa Gestión del sistema .....	80
Flujo del programa. ....	83
Comunicación entre dispositivos.....	85
Interfases de comunicación Circuitos integrados .....	85
Interfaz linea Serie .....	86
Interfaz I2C .....	88
Interfaz SPI.....	90
Otros Interfases de comunicación.....	91
Lineas Futuras.....	93
Difusión del proyecto.....	93
Mejora del diseño .....	94
mejora del hardware .....	97
mejora del software .....	99
Conclusiones.....	101
Situación al inicio del proyecto.....	101
Situación al final del proyecto .....	102
Anexos .....	105
Anexo A: Esquemáticos.....	107
Índice de esquemáticos.....	107
Anexo B: Layout Completo .....	115
Índice de Layout.....	115
Anexo C: Datasheet Hardware.....	123
Entradas y salidas .....	123
Conexión de la Alimentación PCB .....	123
Conexión de las Entradas Digitales.....	124
Conexión de las Salidas Digitales .....	125
Conexiones de relé NA / NC.....	126
Conexiones Lazo de corriente .....	128

Conexiones Sensor de corriente .....	129
Elementos adicionales .....	129
Anexo D: Datasheet Software.....	131
Lista de funciones y codigos del sistema. ....	131
Página web de monitorización. ....	134
Anexo E: Presupuesto. ....	137
Anexo F: Listado BOOM de componentes.....	143
Anexo G: Cálculos Técnicos.....	147
Estimación de consumos eléctricos .....	147
Calculo de optoacopladores de entradas digitales.....	147
Calculo de optoacopladores de salidas digitales .....	148
Anexo H: Análisis de los PLC industriales.....	149
Siemens .....	151
Schneider Electric. ....	153
Anexo I: Análisis de los sistemas de comunicación Industriales .....	155
Sistemas de comunicación Cableadas .....	155
Sistemas de comunicación Inalámbrica. ....	159
Anexo J: LIsta de material contenido en el dvd .....	165
Bibliografía.....	167
Referencias .....	169

# ÍNDICE DE FIGURAS

Figura 1 Símbolo de Hardware libre .....	19
Figura 2 Símbolo de Software libre.....	19
Figura 3 Arduino Leonardo 2014 .....	20
Figura 4 Arduino Uno con 3 Shield acopladas. Una Shield de comunicación xbee, una Shield para montar prototipos y una última para el control de motores. ....	21
Figura 5 Imagen de la última versión disponible de Raspberry PI 2. Actualmente es la versión más avanzada. ....	22
Figura 6 Ejemplo de acople de Shield acoplables a una Raspberry PI B+.....	23
Figura 7 PLC hardware libre versión básica de Industrial Shield. Está basada en un Arduino Uno y electrónica adicional.....	24
Figura 8 PLC de hardware libre basado en un Arduino Mega 2560 así como en electrónica adicional. Está versión superior de Industrial Shield.....	25
Figura 9 PLC hardware libre versión Mega. Es el modelo superior de Controllino, un PLC financiado mediante kickstarter. ....	25
Figura 10 Descripción de la conectividad que presenta el PLC versión Mega de Controllino. Es una solución muy completa.....	26
Figura 11 Ejemplo de evolución del prototipo de un PLC de hardware libre. En este caso de Controllino.....	27
Figura 12 PLC de hardware libre de Industruino, es una versión principalmente educativa.....	28
Figura 13 Industruino hace uso de una zona de prototipado y optoacopladores para dar una completa solución educativa.....	28
Figura 14 Aspecto de la versión final de la PCB industrial de Industruino. ....	29
Figura 15 PLC Open Hardware mostrando el ejemplo de montaje en una caja comercial de carril DIN.....	30
Figura 16 Todos los componentes constituyentes de un Open Source PLC.....	30
Figura 17 Open Source PLC versión ampliada con un número aumentado de entradas y salidas, así como mayor potencia que su versión inferior.....	31
Figura 18 Bloques funcionales principales del prototipo.....	35
Figura 19 Ejemplo de conectores para un PLC Industrial compacto.....	36
Figura 20 Arduino Mega 2560 vista inferior.....	36
Figura 21 Arduino Mega 2560 vista superior.....	37
Figura 22 Raspberry PI B+ usada en el desarrollo de este proyecto.....	37

Figura 23 Interfaz Gráfica debian ejecutada en la Raspberry.....	38
Figura 24 Pantalla LCD, Keypad usada para la visualización de datos en el proyecto. ....	38
Figura 25 Keypad de membrana flexible de 12 posiciones usado en el proyecto.	39
Figura 26 PCB adicional de 8 relés para el control de cargas. ....	39
Figura 27 Ejemplo de conexión del circuito integrado que gestiona los relés. ....	40
Figura 28 Buzzer para emitir avisos sonoros.....	40
Figura 29 MCP3208 Conversor Analógico Digital de precisión. Diagrama de bloques.....	42
Figura 30 Fuente de alimentación del proyecto.....	42
Figura 31 Logotipo de DesignSpark programa para el diseño de PCB.....	43
Figura 32 Esquemático del Conexionado de la Raspberry Pi B+.....	44
Figura 33 Esquemático de conexión de un Arduino Mega 2560 .....	46
Figura 34 Diagrama completo de conexiones de un Arduino Mega 2560 .....	47
Figura 35 Esquemático de conexión de la pantalla LCD. ....	48
Figura 36 Esquemático de conexiones del Keypad de 12 posiciones. ....	49
Figura 37 Detalle de la PCB con 8 relés.....	49
Figura 38 Esquemático de conexiones del control de relés.....	50
Figura 39 Esquemático de conexiones del circuito indicador de sonido.....	51
Figura 40 Esquemático de los canales digitales de entrada 1 .....	52
Figura 41 Esquemático de los canales digitales de entrada 2.....	52
Figura 42 Esquemático de las salidas digitales a 5 Voltios.....	53
Figura 43 Esquemático de las salidas digitales a 5 voltios con PWM .....	54
Figura 44 Esquemático de las salidas digitales a 12 Voltios.....	54
Figura 45 Esquemático salidas digitales a 12 Voltios con PWM.....	54
Figura 46 Esquemático de conexiones del conversor analógico digital de precisión. ....	55
Figura 47 Esquemático de conexión de los 8 sensores de corriente según la versión previa. ....	56
Figura 48 Esquemático INA3221 principal.....	57
Figura 49 Esquemático conectores INA3221 .....	57
Figura 50 Diagrama de conexiones del conector de fuente de alimentación .....	58
Figura 51 Esquemático de generador de test de lazos de corriente. ....	59
Figura 52 Características dimensionales de un Circuito Integrado de agujero pasante. ....	61

Figura 53 Esquema layers de una PCB. ....	62
Figura 54 Logotipo de Euro Circuits fabricante de la PCB.....	63
Figura 55 Herramienta online de Euro Circuits. PCB Checker para comprobar la fabricabilidad de las placas.....	64
Figura 56 Herramienta online de Euro Circuits. PCB Configurator para seleccionar las características de fabricación de nuestra PCB. ....	64
Figura 57 Layout superior completo extraído de EC [EuroCircuits] y resultado final de la PCB montada .....	66
Figura 58 Montaje final del Arduino Mega 2560.....	66
Figura 59 Detalle del Layout Superior del Raspberry Pi. Extraído de EC.....	67
Figura 60 Detalle del conector de la pantalla LCD. ....	67
Figura 61 Montaje del conector del keypad. ....	68
Figura 62 Detalle del Layout Superior del conector de Relé comparado con el montaje final.....	68
Figura 63 Detalle del circuito de aviso sonoro Buzzer. Extraído de EC .....	69
Figura 64 Detalle del Layout Superior comparado con el montaje final de las salidas digitales.....	69
Figura 65 Detalle del Layout y el montaje final de las entradas Digitales. ....	70
Figura 66 Detalle del Layout Superior del Circuito Integrado AD de precisión...70	
Figura 67 Detalle del Layout Superior del sistema de sensado de corriente. Extraído de EC. ....	70
Figura 68 Detalle del layout del conector de entrada de alimentación. Extraído de EC .....	71
Figura 69 Diagrama de comunicaciones principales. ....	73
Figura 70 Diagrama de estados de configuración del Arduino Mega 2560. ....	77
Figura 71 Diagrama de estados principal del Arduino Mega 2560.....	78
Figura 72 Diagrama de flujo del Raspberry Pi .....	84
Figura 73 Interfaces de comunicación entre los diferentes bloques. ....	85
Figura 74 Este módulo conversor lógico de Sparkfun permite la conversión de 6 señales diferentes entre 3.3 y 5 voltios. ....	86
Figura 75 Diagrama de conexión para un solo canal de un conversor lógico mediante el empleo de un Mosfet de canal N.....	86
Figura 76 Secuencia de transmisión I2C. ....	88
Figura 77 Clock Stretching en comunicaciones I2C. ....	90
Figura 78 Diagrama de conexión de dispositivos SPI.....	91



Figura 79 Diagrama de tiempos de una comunicación SPI.....	91
Figura 80 Próximo encapsulado del prototipo.....	94
Figura 81 Ejemplo de PLC con carcasa estándar. ....	95
Figura 82 Ejemplo diseño nivel 1. Electrónica de potencia. ....	95
Figura 83 Ejemplo de diseño nivel 2. Conectores y aislamiento.....	95
Figura 84 Ejemplo de diseño nivel 3. Microcontroladores y elementos extras.....	96
Figura 85 Modulo Linux Aria G25.....	97
Figura 86 ARM 32 bits Sam D20. M0+ Atmel. ....	97
Figura 87 Modem GPRS SIM 900 de bajo coste. ....	98
Figura 88 Ejemplo programación interactiva de un PLC. ....	100
Figura 89 IDE4PLC. Interfaz libre para programación de PLC usando la norma IEC 61131-3 usando programación Ladder. ....	100
Figura 90 Componentes de Montaje Superficial SMT / SMD .....	102
Figura 91 Conectores de alimentación .....	124
Figura 92 Conexión de Entradas Digitales .....	124
Figura 93 Conexión de Salidas Digitales 1 .....	125
Figura 94 Conexión de Salidas Digitales 2 .....	125
Figura 95 Conexión de Salidas Digitales 3 .....	125
Figura 96 Explicación de conexión de una carga a una salida digital. ....	126
Figura 97 Relé SRD-5VDC-SL-C usado en el los relés.....	127
Figura 98 Sistema de funcionamiento y protección del array de relés. ....	127
Figura 99 Circuito en placa de gestor de relés.....	128
Figura 100 Numeración de los relés en la PCB.....	128
Figura 101 Circuito sensado de lazos de corriente 4-20 mA más sencillo.....	129
Figura 102 Panel de visualización de las variables del sistema.....	135
Figura 103 Calculo Resistencias optoacopladores de entradas digitales.....	147
Figura 104 Cálculos de las resistencias de salidas digitales.....	148
Figura 105 Imagen de un PLC siemens Logo .....	151
Figura 106 Imagen de un PLC siemens S7-200 .....	152
Figura 107 Imagen de un PLC Schneider M238, actual Modicon. ....	153
Figura 108 PLC Schneider completo mostrando su sistema de ampliación.....	153
Figura 109 Descripción de un paquete Ethernet.....	155
Figura 110 Descripción pines de un puerto serie. ....	157
Figura 111 Logotipo Wifi.....	159
Figura 112 Ejemplo de PLC con Wifi integrado.....	159

Figura 113 Diagrama de conexión de Zigbee en un entorno industrial.....	160
Figura 114 Emisores Zigbee con diferentes configuraciones de antena. ....	161
Figura 115 Sistema de conexión mallada de sensores.....	161
Figura 116 GPRS GL865-Dual de Telit.....	162
Figura 117 PLC con GPRS integrado.....	162
Figura 118 Logotipo de Bluetooth 4.0 Low Energy.....	163
Figura 119 Circuito integrado de Texas Instrument apto para el Bluetooth 4.0	163



# ÍNDICE DE TABLAS

Tabla 1 Código de colores del Arduino mega 2560. ....	45
Tabla 2 Parámetros técnicos fuente de alimentación.....	58
Tabla 3 Capas de una PCB .....	62
Tabla 4 Ficheros librería RaspberryPi-gpio.....	81
Tabla 5 Flujo de información en una comunicación I2C. ....	89
Tabla 6 Entradas y Salidas del PLC. ....	123
Tabla 7 Pin Map de las salidas digitales. ....	126
Tabla 8 Características relés. ....	127
Tabla 9 Mensajes recibidos por el Slave Arduino .....	131
Tabla 10 Lista de códigos recibidos por el Master Raspberry Pi .....	132
Tabla 11 Estimación de potencias del sistema. ....	147



## OBJETIVO DEL PROYECTO.

El objetivo de este proyecto es el desarrollo de un nuevo sistema para el control de procesos industriales mediante el nuevo diseño de un PLC<sup>1</sup>. Para ello nos basaremos en soluciones de hardware libre ya existentes como puede ser Arduino y microordenadores de bajo coste. Un ejemplo de esto es “Raspberry Pi<sup>2</sup>”. El mercado objetivo de este nuevo PLC es el ámbito educativo en el entorno universitario y en la comunidad Maker<sup>3</sup> con el objetivo de acercar el diseño electrónico y el entorno de los PLC.

La decisión de usar Arduino<sup>4</sup> se basa en los conocimientos previos de los que parto así como su gran comunidad de usuarios existente detrás de ella. Podemos basarnos en Arduino por ser una solución robusta, potente y muy activa. Arduino es un movimiento de hardware libre creado en Italia donde se parte de microprocesadores Atmel para facilitar el desarrollo de prototipos. Su estilo abierto permite su fácil modificación y creación de PCB compatibles y acoplables entre sí para poder añadir nuevas capacidades.

El segundo elemento principal a usar es el microordenador Raspberry Pi, es un ordenador con un ARM<sup>5</sup> y 512 de Ram que dispone de todo lo necesario para su funcionamiento. Dispone de conector HDMI, USB, Ethernet y líneas de comunicación. Su uso será conectado mediante una interfaz I2C<sup>6</sup> con el Arduino.

Gracias al gran número de productos compatibles con Arduino conocidos como Shield permiten configurar nuestro proyecto según las necesidades extras requeridas. De esta forma se puede añadir conectividad GPRS<sup>7</sup>, ZigBee<sup>8</sup>, como forma de comunicación inalámbrica. Trataremos de diseñar y fabricar un sistema lo más polivalente, ya que este proyecto es un prototipo donde queremos explorar el máximo de posibilidades existentes en un único entorno de desarrollo aplicado a un uso en entornos industriales.

Por otro lado tenemos un segundo objetivo que es el de profundizar en el diseño electrónico, ampliando y aplicando una serie de conocimientos adquiridos a lo largo de este proceso. En el desarrollo de este proyecto se aplicaran conocimientos de: Diseño de PCB, Diseño electrónico, Programación en C, Programación Web entre otros muchos.

El proyecto busca cumplir una serie de características en el desarrollo de este prototipo.

- Lograr un sistema fácilmente ampliable mediante la conexión de nuevos dispositivos conectados en red. Por ello los dispositivos tendrán conectividad.

- Construcción de un sistema modular basado en Shield y funcionalidades básicas para lograr adaptabilidad.
- Lograr un tamaño compacto y lograr un bajo coste.
- Sistema abierto para facilitar la libertad de modificación y adaptación a cualquier proceso.
- Difusión en la comunidad Maker para poder facilitar su uso, distribución y mejora por parte de la comunidad web.

Dentro de la elaboración de este proyecto buscamos adaptarnos a procesos de pequeña escala industrial fácilmente controlables junto con la posibilidad de lograr una monitorización lo más completa posible del sistema. Mediante una página web se podrá visualizar, controlar y programar el sistema de la forma más intuitiva posible.

Dentro de las limitaciones que va a tener nuestro sistema será la no utilización de electrónica apta para entornos industriales. Eso implica que no se podrá garantizar su perfecto funcionamiento en entornos agresivos debido a la presencia de humedad, gases o partículas metálicas. Esta etapa corresponden a posteriores versiones de este prototipo.

El proyecto está estructurado en tres apartados.

- El primer punto será el diseño de una PCB para introducir toda la electrónica necesaria. El Arduino y la Raspberry irán acoplados a esta PCB. Para ello requerirá del diseño de los esquemáticos y el diseño del layout del mismo.
- El segundo apartado del proyecto, será la programación del sistema, el cual viene englobado en tres programaciones independientes.
  - Programación del Arduino realizada en C usando librerías programadas.
  - Programación del Raspberry Pi realizada en C, el cual corresponde al grueso de la programación.
  - Página web para la monitorización y control del sistema. Está formado por un servidor Apache junto con una base de datos MySQL<sup>9</sup>.
  - El tercer apartado es el diseño de una carcasa que pueda encerrar el conjunto de todo el hardware.

Al finalizar este proyecto será difundido por la comunidad de hardware y software libre, al mismo tiempo que se mejorara sus funcionalidades. El fin será convertirlo en un producto apto para su uso industrial y educativo rediseñando el producto a un sistema de caja carril DIN. La solución final buscara ser ampliable mediante la incorporación de nuevos sistemas permitiendo una fácil integración industrial

# ESTADO DEL ARTE DE LOS PLC.

## Contexto PLC

Llamamos PLC a un controlador lógico programable, son dispositivos electrónicos que permiten ser usados en procesos de automatización industrial con el fin de mejorar la productividad de las líneas de fabricación

Todos los PLC comparten una serie de características comunes como son la disponibilidad de canales de entrada y actuadores para interactuar con las líneas de fabricación. Estos dispositivos están debidamente preparados para trabajar de forma segura y robusta. Son compatibles con una gran cantidad de estándares para permitir la ampliación de sus funcionalidades.

La historia de los PLC se remonta a los años 60 donde se crearon para sustituir a los obsoletos sistemas de control basados en relés, que servían para realizar el control de los procesos. La empresa que empezó con el desarrollo de un nuevo sistema de automatización provino de una de las grandes empresas de la época General Motors. El primer PLC creado fue el Modicon actualmente bajo la marca Schneider Electric.

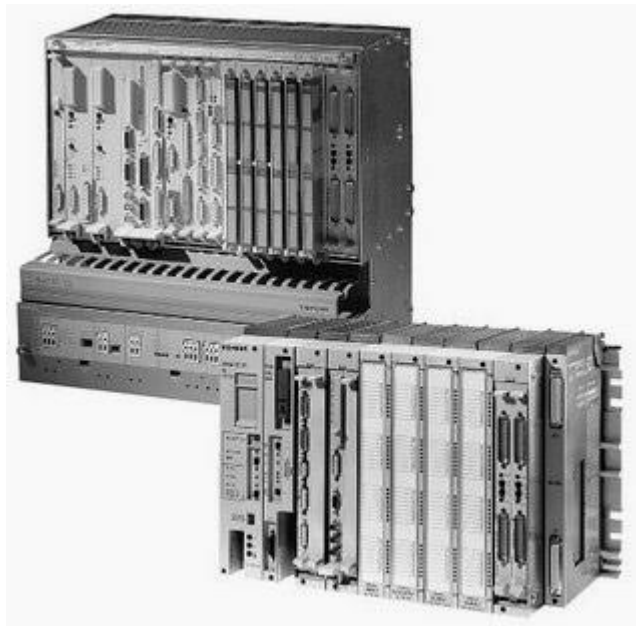


Figura 1 del primer PLC comercial Modicon

Para desempeñar esta labor fue creado los primeros lenguajes de programación orientados para su funcionamiento en PLC. El ejemplo más antiguo es la programación Ladder<sup>10</sup>. Actualmente los PLC tienen una capacidad de programación prácticamente ilimitada que simplifica la puesta en marcha de los mismos. El ejemplo más claro es la



programación gráfica usando entornos similares a LabView que facilitan enormemente la programación y monitorización.

Uno de los factores que más influyen en un PLC es su grado de seguridad y confiabilidad. Las empresas requieren sistemas completamente fiables respaldados por grandes marcas. De esta forma depositan el pleno control de la fábrica en manos de estos sistemas.

Es frecuente el aprendizaje de estos sistemas en entornos educativos y pequeñas instalaciones en los que se usan a una escala menor. Una variante de este tipo de elementos está orientado al entorno de la Domótica.

Dentro de los PLC existen distintas variantes según su complejidad y funcionalidades. Un ejemplo son los PLR o relés lógicos programables. Un PLR permite programar unos relés para actuar bajo los estímulos adecuados. Son una solución barata para cierto tipo de aplicaciones. Todos comparten en común sistemas de montaje similar y sistemas de comunicación compatibles entre sí. Estos dispositivos incluyen elementos de comunicación visual con el usuario así como una cierta interactividad.

## Open Source y Open Source Hardware

El mundo de la electrónica se ha empezado a democratizar en los últimos 10 años, gracias a la participación de la comunidad electrónica. Está ha ido generando proyectos de libres que permitan su modificación, distribución y uso. La mayoría de estos proyectos se comparten bajo licencias Open que permiten estas características que las diferencian de los elementos propietarios.



Figura 2 Símbolo de Software libre

Figura 1 Símbolo de Hardware libre

En el desarrollo de este tipo proyectos hay que diferenciar dos categorías diferentes. La primera la Open Source, permite la libre difusión, modificación y uso de software proporcionado por la comunidad. Se distribuye en forma de códigos fuente que facilitan el uso e implementación de nuevas funcionalidades de software. Originalmente es el movimiento de más antigüedad y que mayor volumen de información mueve.

Por otro lado el Open Source Hardware consiste en diseños de hardware disponibles bajo licencias de libre acceso, modificación y fabricación de los diseños de hardware así licenciados. Esto permite la fácil modificación de hardware de libre acceso para la adaptación del mismo a las necesidades particulares de cada usuario. Esto se puede unir con el Open Source para crear una perfecta sintonía entre ambos movimientos.

Actualmente se ha disparado el número de diseños y proyecto que trabajan bajo esta filosofía entre ellos el más conocido es Arduino.

## ARDUINO

Arduino es una herramienta de diseño y prototipado de electrónica que une el Open Source Hardware y el Open Source en un solo producto. Esto ha generado una gigantesca comunidad detrás de ella, con miles de diseños a lo largo de los años. Se pueden encontrar todo tipo de proyectos, desde el tracking mediante GPS<sup>11</sup> y GPRS al control y monitorización de redes eléctricas, o como este proyecto, un PLC industrial.

Se tratan de soluciones de bajo coste y de gran sencillez que unen el lenguaje de programación Processing con su hardware para desarrollar su entorno de desarrollo. El origen de Arduino tiene lugar en Italia en 2006 y surge ante la necesidad de desarrollar un sistema de aprendizaje de programación para el entorno universitario que fuera más económico que las soluciones ya existentes. En la actualidad estas placas tienen un precio muy inferior a los 30 euros.

Arduino está presente en una amplia gama de formas y aplicaciones entre ellas la más común tiene este formato que se puede ver en la figura inferior.

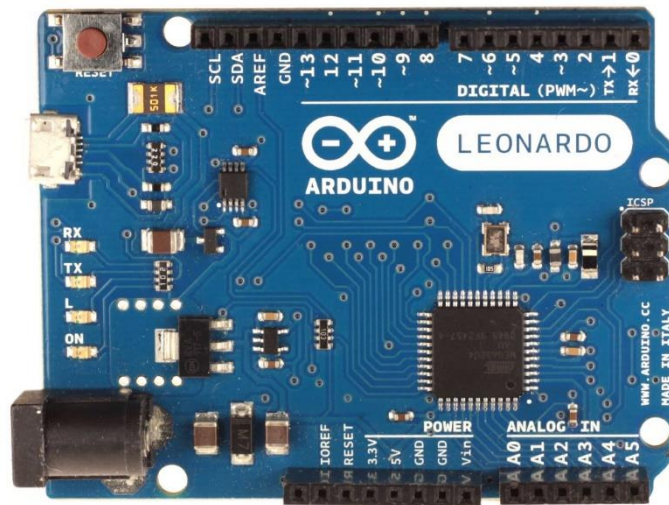
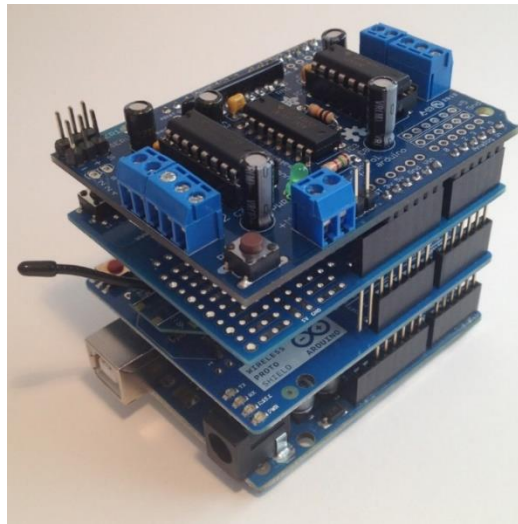


Figura 3 Arduino Leonardo 2014

Como forma de potenciar estos módulos existen PCB adicionales conocidas como SHIELD que amplían las funcionalidades de estos módulos mediante la adición de módulos insertables entre sí.



**Figura 4** Arduino Uno con 3 Shield acopladas. Una Shield de comunicación xbee, una Shield para montar prototipos y una última para el control de motores.

Desde la aparición de estos primeros proyectos de programación educativa han surgido decenas de proyectos independientes de electrónica y programación que tienen el objetivo de la difusión de conocimiento desarrollando sus propias soluciones para facilitar este aprendizaje, de entre todas ellas no todas licencian sus desarrollos de hardware bajo licencias Open Source Hardware como por ejemplo Raspberry.

## RASPBERRY PI

Raspberry Pi es un miniordenador de bajo coste del tamaño de una tarjeta de crédito. Entre sus características está técnicas está un ARM 700MHz y 512 de RAM que permiten ejecutar un entorno gráfico Linux completo. Se creó con la finalidad de introducir la programación al entorno educativo. Actualmente tiene una potencia comparable a la de la mayoría de teléfonos móviles pudiendo hacer tareas de gestión de servidores, programación, reproductores multimedia sin problemas de limitaciones técnicas.

Este proyecto fue creado por la fundación Raspberry Pi con los fines antes descritos, en el año 2009 finalmente se las primeras placas de testeo en el 2011 y empieza su distribución en 2012.

Como en el caso de Arduino este proyecto dispone de una amplia comunidad detrás que desarrolla software y lo distribuye libremente para facilitar el aprendizaje y difusión del conocimiento. Igualmente existen SHIELD Raspberry que amplían las funcionalidades de este proyecto, por ejemplo haciéndolo compatible y comunicable con Arduino.

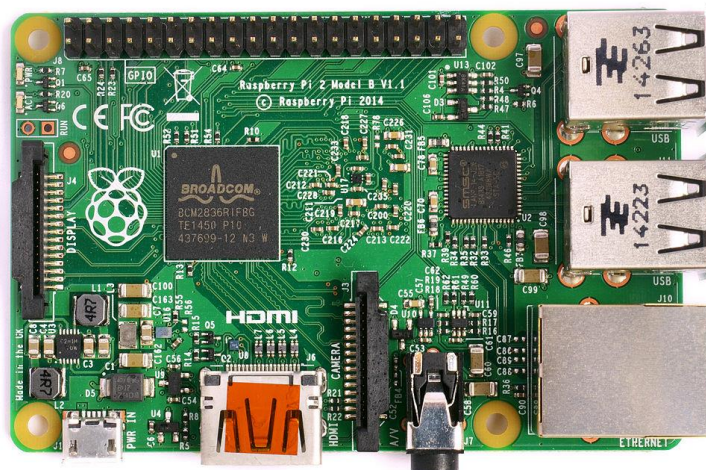


Figura 5 Imagen de la última versión disponible de Raspberry PI 2. Actualmente es la versión más avanzada.

Raspberry junto con Arduino son los dos proyectos más activos del ámbito del hardware y software libre, que se engloban como proyectos referencia de este ámbito, cumplen con los requisitos para facilitar la enseñanza de programación, uso en herramientas de prototipado, o más interesante aún es darle la utilidad deseada por cada individuo



**Figura 6 Ejemplo de acople de Shield acoplables a una Raspberry PI B+**

En este proyecto vamos a aprovechar todas las capacidades que presentan estas soluciones libres para desarrollar nuestra propio PLC industrial, siguiendo el tipo de filosofía de estos proyectos que es la de difusión del conocimiento y la mejora continua.

## PLC OPENSOURCE

Dentro del marco del Open Source y Open Source Hardware han aparecido infinidad de proyectos orientados al ámbito industrial y educativo. Se trata de diseños de PLC compatibles con la programación de Arduino o basados en el Hardware Arduino. Se trata de soluciones de bajo precio adecuadas para su funcionamiento en pequeñas y medianas líneas de producción industriales, con plena capacidad de comunicaciones y control.

Entre estas opciones existe ya un amplio abanico de PLC que cumplen estas características.

### *Industrial Shield*

La marca industrial Shield fue fundada en octubre de 2012 es la mayor empresa de electrónica industrial basada en Open, está íntegramente basado en Arduino y son originarios de España.

Su objetivo es generar una amplia gama de productos a un precio razonable y comparable con las opciones tradicionales. Se trata de un producto de montaje en carril DIN con las capacidades para trabajar en un entorno industrial. Aporta una solución perfecta para pequeñas instalaciones y potente al disponer de conectividad Ethernet.



**Figura 7 PLC hardware libre versión básica de Industrial Shield. Está basada en un Arduino Uno y electrónica adicional.**

Entre sus características está funcionar a tensiones estándar industriales de 12 y 24 voltios y disponer de una amplia gama de comunicaciones serie, Modbus<sup>12</sup> y Ethernet.

Proporciona de un amplio número de conexiones digitales y analógicas para cubrir todo tipo de necesidades generales.

Disponen de un modelo educativo y otro profesional con mayores capacidades de conectividad.



Figura 8 PLC de hardware libre basado en un Arduino Mega 2560 así como en electrónica adicional. Está versión superior de Industrial Shield.

### *Controllino*

Controllino es un PLC compatible con Arduino con fácil conexión a la IoT<sup>13</sup>. El origen de este proyecto tiene lugar en una campaña de crowdfunding a través de kickstarter. Este modelo es un PLC Open Source en tres formatos, el primero es el formato adaptado a entornos industriales, junto con dos variantes más para Domótica y usos educativos.



Figura 9 PLC hardware libre versión Mega. Es el modelo superior de Controllino, un PLC financiado mediante kickstarter.

Entre sus ventajas tiene estar fabricado respecto a ISOS industriales y de seguridad, lo que valida el producto junto al resto de su potencialidad. Sus componentes internos han



sido elegidos para cumplir los requisitos de funcionamiento en entorno industriales, gracias al uso de electrónica con cualificación industrial.

Las características técnicas de Controllino son muy completas y con un acabado del producto completamente similar al resto de productos industriales.

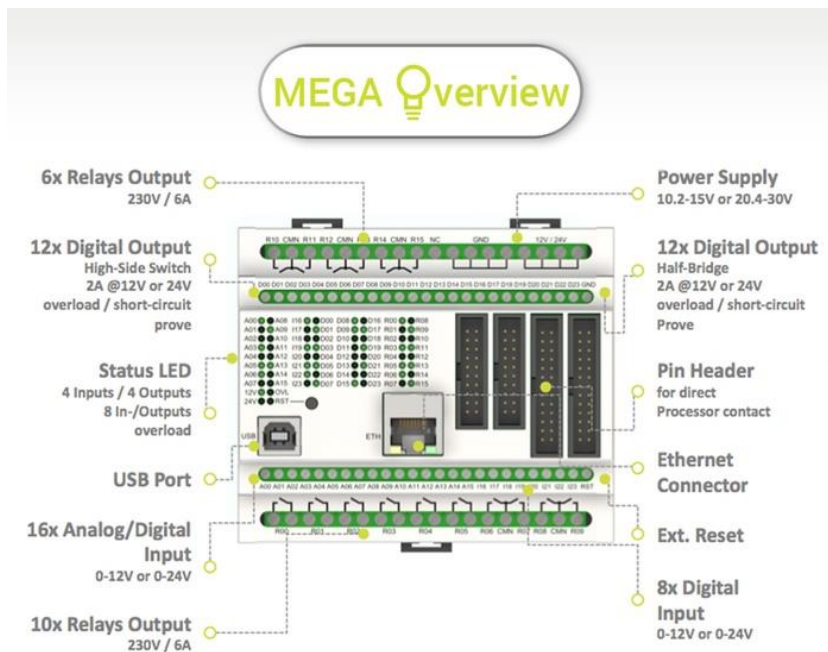


Figura 10 Descripción de la conectividad que presenta el PLC versión Mega de Controllino. Es una solución muy completa.

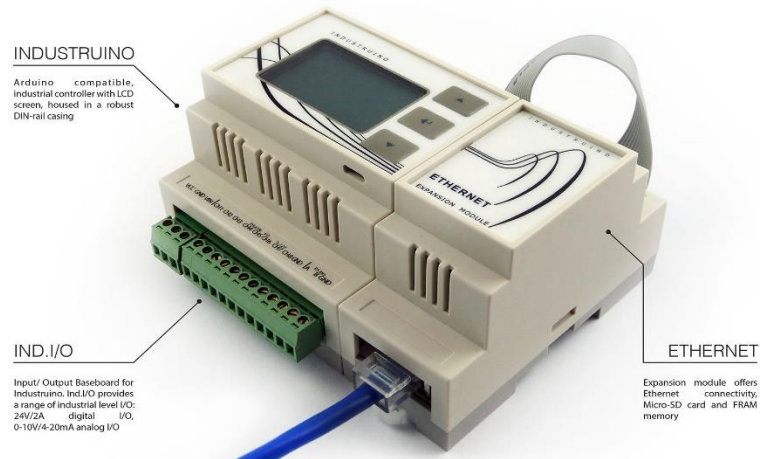
El diseño de un proyecto de este tipo pasa por varias etapas, desde que se prueba el primer prototipo funcional hasta que se llega a una versión final apta para su uso industrial. En las etapas intermedias de mejora el diseño y se mejora las capacidades técnicas. Esta solución de PLC tiene un precio más elevado que el resto de soluciones similares. En este caso vale 325 \$ la versión industrial y 139 la versión educativa.



**Figura 11** Ejemplo de evolución del prototipo de un PLC de hardware libre. En este caso de Controllino.

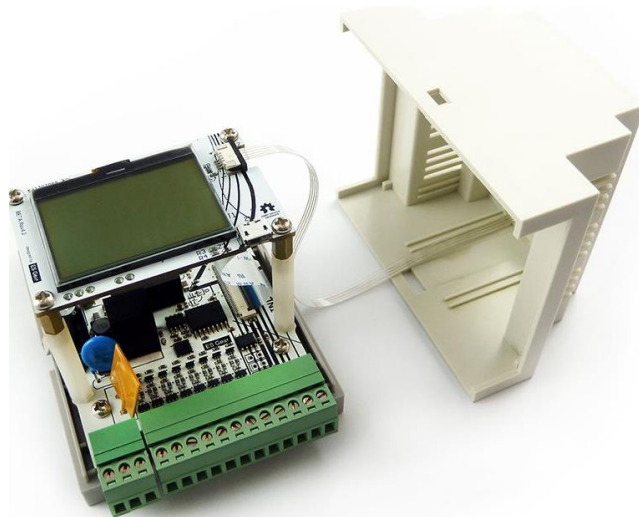
### **Industruino**

Otra solución PLC de hardware libre existente es Industruino. Se trata de un PLC con plenas funcionalidades y conexión Ethernet. Le diferencia del resto de PLC de hardware libre es disponer de pantalla y área de prototipado interna.



**Figura 12** PLC de hardware libre de Industruino, es una versión principalmente educativa.

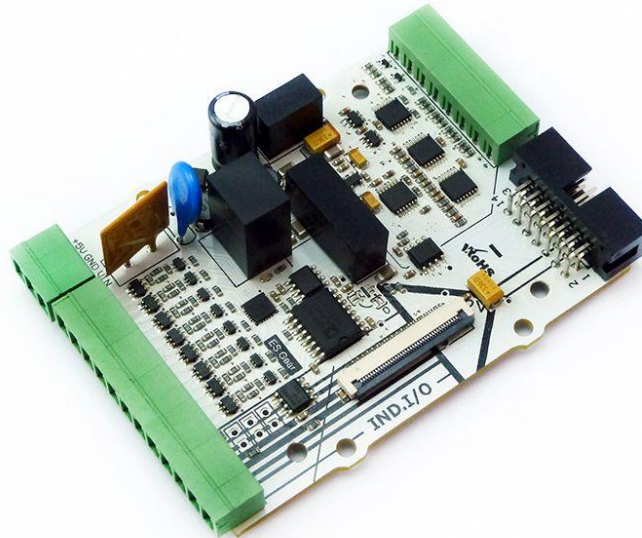
Viene en dos modelos, el primero listo para prototipado y modificación de las prestaciones de las que dispone y el segundo es de nivel industrial, que aporta aislamiento para asegurar el perfecto funcionamiento. Este dispositivo dispone de capacidades adecuadas para su uso en entornos educativos y pequeñas áreas de producción.



**Figura 13** Industruino hace uso de una zona de prototipado y optoacopladores para dar una completa solución educativa.

Dispone de entradas digitales y analógicas con el extra de sensores de corriente industrial y aislamiento como medida de protección.

El precio de la versión industrial son de 110 €, un precio muy inferior al resto de soluciones libres, con buena capacidad para ser instalado y utilizado en un amplio abanico de situaciones. Está disponible un módulo conectable para aportar conexión Ethernet. Se puede considerar como una solución muy económica.



**Figura 14** Aspecto de la versión final de la PCB industrial de Industruino.

### ***SMALL BRICK<sup>1</sup>***

Small Brick es otro PLC open-source que surgió en 2007 y ha ido mejorando desde su creación.

La idea de este proyecto era crear un dispositivo listo para usar basado en un PLC y que fuera comparable a un PLC comercial. Al estar programado en C es fácilmente programable por cualquier usuario aunque requiriendo mayor tiempo de programación.

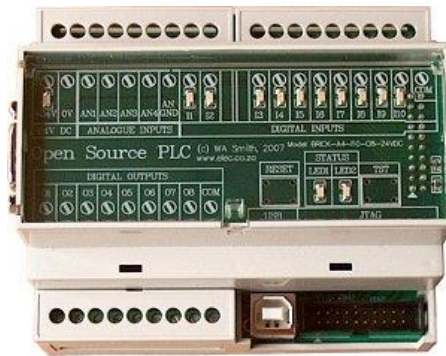


Figura 15 PLC Open Hardware mostrando el ejemplo de montaje en una caja comercial de carril DIN.



Figura 16 Todos los componentes constituyentes de un Open Source PLC.

Como todos los proyectos de hardware libre se proporciona toda la información necesaria para poder modificar y fabricar este modelo.

Funciona a tensiones de alimentación industrial de 12 y 24 voltios. Es una solución completa y profesional completamente Open Source Hardware que facilita el uso de este producto.

<sup>1</sup> <http://startingelectronics.org/projects/small-open-source-PLC/>

Su programación se realiza en Arduino siendo completamente compatible con este producto. Dispone de una segunda versión más potente conocida como Large Brick.



Figura 17 Open Source PLC versión ampliada con un número aumentado de entradas y salidas, así como mayor potencia que su versión inferior.



## ARQUITECTURA

Para la elaboración de este proyecto hemos pasado por varias etapas: Elección de funcionalidades, diseño de los esquemáticos, creación del layout y fabricación.

Una de las etapas más importantes en el desarrollo del proyecto ha sido la elección de las funcionalidades técnicas que debe tener este prototipo. Para determinar las funcionalidades básicas de nuestro diseño se han revisado los productos ya existentes. Para elegir los sistemas de comunicación se ha elegido solo los sistemas de comunicación más sencillos no por ellos menos potentes que otros sistemas de intercambio de datos.

Una vez elegidas las funcionalidades el siguiente paso es la elaboración de los esquemáticos que implican la creación de los diagramas de conexión eléctrica de los componentes. Para la elaboración de estos diseños usamos el programa DesignSpark 6.1 de RS. Con la finalización de los esquemáticos podemos pasar a la creación del layout.

El desarrollo del layout implica definir donde se colocarán los diferentes componentes en la placa de circuitos impreso así como el tamaño final que tendrá la PCB. Según el resultado final de la placa será diferente la disposición de las entradas y salidas del sistema que determinará el montaje de la carcasa. La elaboración de esta etapa igualmente se realiza con el DesignSpark 6.1 de RS.

La última etapa de la fabricación, es el montaje de los componentes en la placa así como el testeo de funcionamiento de todos los elementos del sistema. La aparición de errores cometidos en etapas previas tendrá su repercusión en este punto del proyecto, donde habrá que realizar correcciones sobre la placa, para lograr el perfecto funcionamiento.



## Definición de necesidades

En la creación de las funcionalidades de nuestro sistema decidimos que tuviera las siguientes características.

Las características de entradas y salidas son las siguientes:

- 8 Salidas de Relé para el accionamiento de dispositivos de potencia.
- 6 Entradas Digitales a 5 Voltios.
- 6 Salidas Digitales a 5 Voltios, 3 de ellas con la posibilidad de generar salida mediante PWM y poder regular la tensión de salida entre 0 y 5 Voltios.
- 6 Salidas Digitales a 12 Voltios, 3 de ellas con la posibilidad de generar salida mediante PWM y poder regular la tensión de salida entre 0 y 12 Voltios.
- 8 Entradas Analógicas de Precisión entre 0 y 5 Voltios.
- 8 Canales de entrada de Sensores de lazo de corriente entre 4 y 20 mA.

Las posibilidades de interacción de un usuario con el dispositivo son:

- Buzzer de aviso sonoro.
- Display LCD de dos líneas.
- Keypad numérico para la introducción de comandos.
- Conexión de monitor.
- Conexión USB para teclado y ratón.
- Conexión Ethernet.
- Conexión remota para monitorización y control del sistema mediante un servidor web y la Base de datos correspondiente.

Otras de las características que incluiremos en nuestro proyecto son:

- Sistema de Reloj RTC<sup>14</sup> para la gestión y control de la hora y fecha.
- Fuente de alimentación conmutada para proporcionar alimentación al sistema. Será necesario disponer de suministro de 12 y 5 Voltios.

## Lista de Hardware

Para lograr las funcionalidades anteriores hemos elegido la electrónica apropiada

Para cumplir todas estas funcionalidades hemos introducido los elementos electrónicos necesarios. Los dos bloques principales que debe tener el sistema son un controlador para gestionar las entradas y las salidas del sistema y un procesador principal que coordine todas las características. Para la primera tarea se ha elegido un Arduino Mega 2560 y para la segunda un Raspberry PI B+. El diagrama de bloques indica el sistema de conexión.

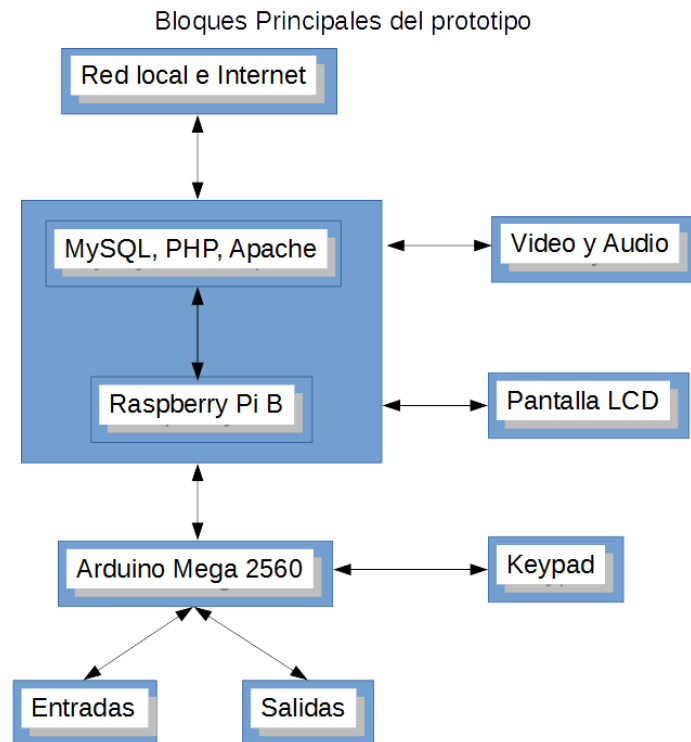


Figura 18 Bloques funcionales principales del prototipo.

Para el diseño de este proyecto se ha intentado que todos los componentes que se incluyen sean de montaje Pasante. Esto es debido a que no tenemos una limitación de espacio ni incluimos elementos demasiado complejos como giróscopos o acelerómetros.

Por otro lado para incluir un sistema robusto de conexión del cableado externo hemos incluido conectores fijos de montaje en PCB.



Figura 19 Ejemplo de conectores para un PLC Industrial compacto.

## ARDUINO MEGA 2560

El Arduino Mega 2560 hará las tareas de gestión de todas las entradas y salidas del sistema, de tal forma que será el elemento que accionará los relés, recibirá las señales externas y controlará otros dispositivos. Es un elemento altamente especializado en el control de dispositivos que desempeña un papel mejor que el que puede hacer Raspberry para esta tarea. Tiene la ventaja de que es más robusto y menos propensos a fallos de lo que puede ser un Raspberry. El dispositivo tiene un número de pines lo suficientemente elevado como para poder controlar por su propia cuenta un número realmente elevado de dispositivos.

Entre las características técnicas de este dispositivo están 54 pines de entradas / salidas digitales, de las cuales 15 de ellas permiten salidas PWM junto con 16 entradas analógicas y 4 puertos serie UART<sup>15</sup> y funciona bajo una frecuencia de 16 MHz, más rápida que los Arduinos de gama más básica. La gran mayoría de Shield existentes son compatibles con este modelo por lo que será muy fácil ampliar las características técnicas del proyecto con estas soluciones.

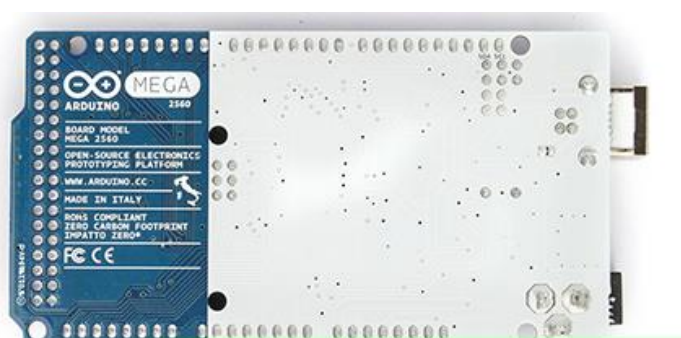


Figura 20 Arduino Mega 2560 vista inferior.

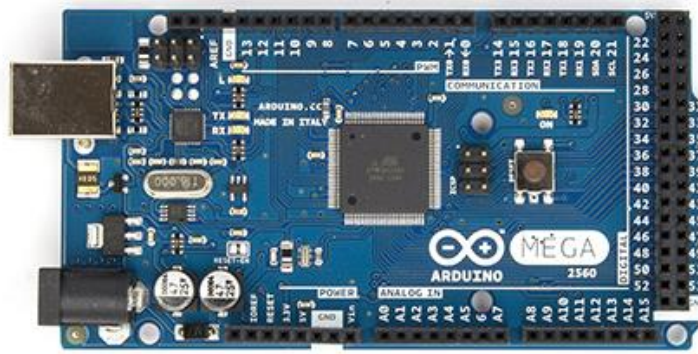


Figura 21 Arduino Mega 2560 vista superior.

El Arduino Mega hará todas las tareas de comunicación con las entradas y salidas y el Raspberry Pi se encargará de la gestión de las tareas más complejas.

## RASPBERRY PI

La Raspberry Pi integrada en nuestro proyecto, hace el papel de ordenador central comunicándose mediante I2C al Arduino. Mediante esta comunicación se realiza el envío y recepción de información de forma bidireccional. Esta comunicación es íntegramente gestionada por la Raspberry Pi al actuar como maestro de las comunicaciones.

Este dispositivo del tamaño de una tarjeta de crédito 85.6 x 53.98 x 17 mm tiene un gran número de funcionalidades. Esto posibilita hacer funcionar un sistema Linux integro con amplias funciones multimedia y de comunicación.

El sistema está gestionado por un SoC<sup>16</sup> Broadcom BCM2835 que integra la CPU, GPU<sup>17</sup>, la RAM<sup>18</sup> y el control de puertos USB en un único SoC. El chip es un ARM11 ARMv6 de un solo núcleo a 700 MHz del 2010.



Figura 22 Raspberry Pi B+ usada en el desarrollo de este proyecto.

En el ordenador hay disponibles 512 MB de RAM compartidos con la gráfica, por lo cual estaremos limitados para aplicaciones multimedia o para la gestión de múltiples aplicaciones simultáneas. No obstante permite ejecutar un completo interfaz gráfico gracias a sus prestaciones, tal y como se puede ver en la Figura inferior.

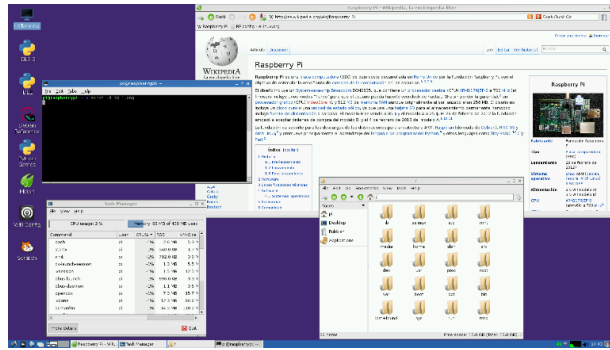


Figura 23 Interfaz Gráfica debian ejecutada en la Raspberry.

En las posibilidades de intercomunicación con otros dispositivos permite la comunicación gracias a la existencia de GPIO<sup>19</sup> en la placa entre los cuales incluyen puertos de comunicación Serie, I2C y SPI. Existen amplias librerías ya existentes y preconfiguradas para facilitar el uso de estas conexiones.

La Raspberry Pi utiliza como sistema operativo una distribución Raspbian basada en Linux. El arranque y carga de ficheros se realiza desde una tarjeta SD. El uso de este dispositivo permite la gestión de una Base de Datos basada en MySQL así como el servidor Apache para gestionar el servidor de comunicación con el usuario.

## PANTALLA LCD DE 2 LÍNEAS.

Como Elemento de interacción con el usuario hemos introducido una pantalla LCD con Keypad incorporado para permitir una gestión a los usuarios, así como una rápida monitorización.

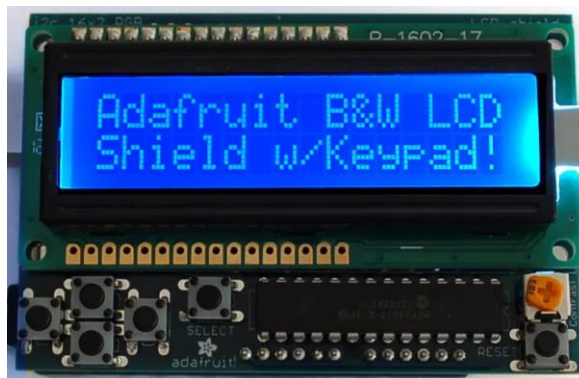


Figura 24 Pantalla LCD, Keypad usada para la visualización de datos en el proyecto.

Este dispositivo se controla mediante el uso del puerto de comunicaciones I2C junto los 5 botones existentes. Es completamente compatible con el modelo de Arduino usado en nuestro proyecto. El control de este dispositivo solo es capaz de controlarlo el Raspberry Pi al ser el único que ostenta el pleno control de las comunicaciones.

## TECLADO KEYPAD NUMÉRICO

El Teclado de membrana permite interactuar con el sistema introduciendo códigos gracias a sus caracteres numéricos. Es una solución muy simple y poco sofisticada que usa un total de 7 cables para la detección de pulsaciones. Como contrapartida tiene el problema de que es muy ineficiente al requerir el uso de 7 cables, cuando generalmente se puede simplificar su conexión mediante el uso de algún circuito integrado para requerir únicamente el uso de 2 pines. Es una solución muy barata comparada con el resto de opciones posibles.



Figura 25 Keypad de membrana flexible de 12 posiciones usado en el proyecto.

Al ir conectado directamente al Arduino será este quien gestione y detecte las pulsaciones. Posteriormente los comandos serán enviados al Raspberry mediante I2C.

## RELÉ DE 8 CANALES

Para el accionamiento de dispositivos de potencia se incluyen un módulo de 8 Relés para la gestión de este tipo de aplicaciones. Se trata de un módulo que incluye optoacopladores y elementos aislamiento para proteger las señales digitales de la red eléctrica.



Figura 26 PCB adicional de 8 relés para el control de cargas.

La conmutación de los relés se hace pin a pin a través de 8 pines digitales de entrada. Los relés disponen de contactos Normalmente cerrados y Normalmente Abiertos y permiten el control de un máximo de 10 Amperios.

Para la mejor gestión de este dispositivo se ha usado un Circuito integrado extra conocido como Shifter Register con Latch<sup>20</sup> mantenido. La característica de Latch mantenido permite dejar fijo el estado de los relés sin tener que activarlos en cada ciclo de reloj. Esto es un requisito fundamental para esta aplicación de lo contrario no se puede garantizar el funcionamiento de los relés. El circuito integrado usado para esta tarea ha sido un 74HC595 que realiza la conversión de serie a paralelo para controlar los 8 Relés. En la Figura 28 se aprecia un ejemplo de conexión de este circuito sobre una placa de prototipos.

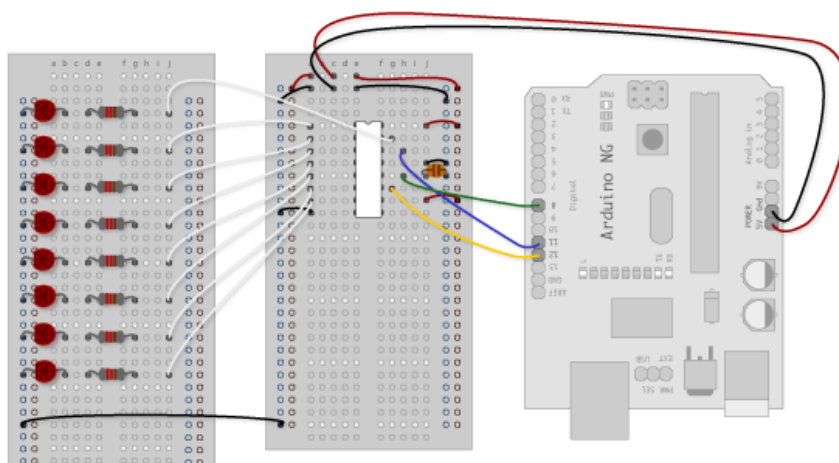


Figura 27 Ejemplo de conexión del circuito integrado que gestiona los relés.

## BUZZER DE AVISO SONORO

Al proyecto se le ha añadido un elemento para generar avisos sonoros mediante la incorporación de un Buzzer. Este dispositivo es un piezoeléctrico que mediante la excitación con señales de una frecuencia determinada se genera un tono de una cierta intensidad y con una frecuencia igual a la usada para excitar el Buzzer. En la Figura 29 se observa un piezoeléctrico para generar alertas sonoras.



Figura 28 Buzzer para emitir avisos sonoros.

## ENTRADAS DIGITALES

Para la interacción de dispositivos se ha añadido 6 entradas digitales las cuales se encuentran protegidas mediante optoacopladores para proteger la integridad del dispositivo. Para la tarea de aislamiento se han elegido optoacopladores de un solo canal ya que no tenemos limitaciones de espacio. Las señales se activan a 5 voltios.

## SALIDAS DIGITALES

Para el control de dispositivos se han añadido una serie de salidas para poder controlar dispositivos que funcionan a 5 y 12 voltios. Entre las características clave de estos elementos es su optoaislamiento. Este aislamiento se produce gracias a unos optoacopladores que aíslan la parte de potencia externa al PLC y la zona de electrónica digital.

Entre otras de las características de estas salidas es la capacidad de aportar una alta intensidad de corriente, gracias al uso de Mosfet de canal N para permitir dar altos valores. La limitación de esta intensidad solo está limitada por la fuente de alimentación y el resto de dispositivos conectados.

El sistema cuenta con 6 salidas digitales a 5 voltios y otras 6 salidas digitales a 12 voltios. Para aportar una mayor flexibilidad a esta característica se ha conectado varias de ellas a las salidas del Arduino Mega con capacidad PWM.

De esta forma disponemos de 3 salidas a 5 voltios con capacidad PWM que permiten obtener valores entre 0 y 5 voltios. Al mismo tiempo el mismo sistema aporta otras 3 salidas a 12 voltios con capacidad PWM que permiten obtener valores entre 0 y 12 voltios.

## ENTRADAS ANALÓGICAS

Otra de las funcionalidades incluidas en este primer prototipo permite recibir entradas externas analógicas como puede ser valores de sensores externos. De esta forma es posible el sensado de hasta 8 sensores externos siempre y cuando aporten señales comprendidas entre 0 y 5 voltios.

Como característica adicional de esta funcionalidad está la capacidad de realizar un sensado Analógico de precisión de 12 bits, de esta forma aporta 4 veces la resolución básica que aporta un Arduino Mega.

El elemento que aporta esta característica es un MCP3208 un circuito integrado de la casa de semiconductores Microchip. Es un dispositivo que realiza la conversión Analógico Digital que aporta 8 canales con una precisión de 12 Bits. El modo de comunicación con este dispositivo es mediante una interfaz SPI<sup>21</sup>.



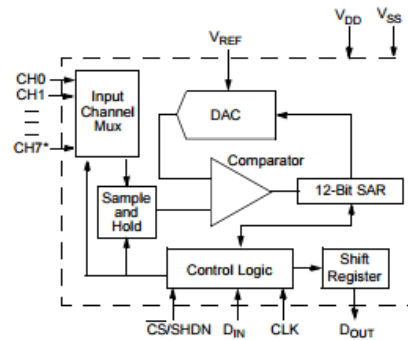


Figura 29 MCP3208 Conversor Analógico Digital de precisión. Diagrama de bloques.

## ENTRADAS DE LAZO DE CORRIENTE

Como sistema de sensado de sensores industriales es frecuente incorporar la posibilidad de sensar señales de entre 4 a 20 mA para mapearlas a valores entre 0 y 5 Voltios que acepte el Arduino. La forma más sencilla es utilizar la Ley de Ohm para mapear esta información y convertirla.

Para este proyecto se ha intentado un sistema más sofisticado mediante el empleo de un circuito de sensado energético para medir corrientes y voltajes.

La explicación del uso de este tipo de información en sensores industriales está en usar un sistema que no se base en la tensión para transmitir la información. Un cable cortado proporcionaría como valor de entrada un cero, por lo cual se confundiría con un valor real. Por otro lado para transmitir corriente se pone como mínimo 4 mA de corriente para indicar un valor de cero y un valor de 20 mA para transmitir el valor máximo. Este sistema además evita las pérdidas de información por cables muy largos.

## FUENTE DE ALIMENTACIÓN

Para el sistema hemos introducido una fuente de alimentación conmutada para la transformación del voltaje de red a 12 y 5 Voltios. Hemos introducido una fuente de alimentación de elevada potencia para poder mover elementos pesados de potencia. Este sistema alimenta íntegramente a todo el sistema.



Figura 30 Fuente de alimentación del proyecto.

## Diseño de Esquemáticos

Como ya hemos comentado hemos diseñado los esquemáticos mediante el uso de DesignSpark 6.1. Para el diseño de los esquemáticos requiere el uso de la lectura de los datasheet de los diferentes elementos incluidos como hardware para saber cómo se debe realizar las conexiones de los diferentes bloques.



Figura 31 Logotipo de DesignSpark programa para el diseño de PCB

El diseño de los esquemáticos está agrupado según PCB a la que pertenecen. Al mismo tiempo cada PCB está ordenada según funcionalidades.

La primera hoja del esquemático incluye la información sobre las conexiones del Arduino Mega y el Raspberry, es decir el bloque de control y lógica del proyecto. La segunda hoja incluye el diagrama de conexiones de las entradas analógicas de la PCB. El tercer esquemático incluye la información sobre las entradas y salidas digitales, así como las conexiones de relé. Finalmente la última hoja de los esquemáticos agrupa la información del LCD, el Keypad y el Buzzer.

Como se ha comentado en el diseño de los esquemáticos es necesario entender perfectamente el funcionamiento de los componentes, para ellos hay que estudiar el datasheet y buscar ejemplos de aplicaciones a así como decidir la forma más adecuada de conexión eléctrica.

## ESQUEMÁTICO RASPBERRY PI.

El Raspberry Pi B tiene un conector de 26 Pines que proporciona acceso a todos los pines de GPIO que tiene disponibles, entre ellos los relativos a la conexión I2C y las alimentaciones.

La alimentación eléctrica de la Raspberry Pi se realiza a través de los pines que presenta de GND y 5 Voltios. La razón de realizar el suministro eléctrico de esta forma es que está limitada la corriente máxima accesible a través del conector microusb de alimentación debido a un polifusible de 1.1 amperio. En caso de que se requiera un consumo eléctrico mayor o mayor consumo de los periféricos usb el sistema se vería incapaz de hacerlo funcionar, provocando reinicios inesperados. El

punto negativo de esta forma de alimentación es que perdemos la protección contra sobrecorriente de la alimentación principal.

Las conexiones existentes entre el Raspberry Pi y el resto de elementos únicamente permiten las comunicaciones mediante una interfaz I2C y una conexión para forzar el Reset del Arduino Mega. La Raspberry Pi se alimenta en este sistema a través de la conexión de 5 Voltios.

La Raspberry PI posee dos puertos I2C de los cuales solo hemos usado uno de ellos, el otro está disponible para soldar un conector a la placa. Los puertos I2C presentan un Pull-up requerido en los dispositivos con comunicaciones I2C.

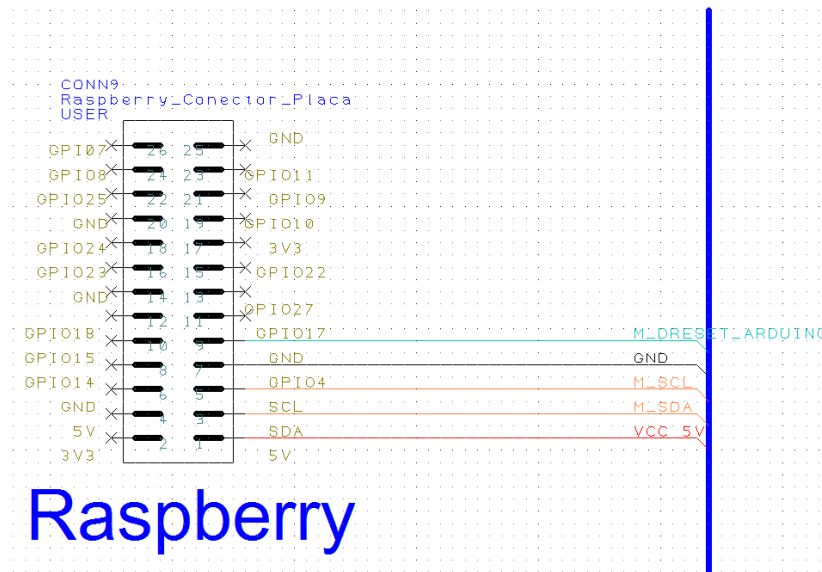


Figura 32 Esquemático del Conexionado de la Raspberry Pi B+

## ESQUEMÁTICO ARDUINO MEGA

El Esquemático del Arduino Mega 2560 es el más complejo del sistema al ser el elemento que tiene mayor número de conexiones de toda la PCB. Un Arduino Mega presenta aproximadamente 75 conexiones eléctricas, de las cuales en nuestro proyecto habremos usado la gran mayoría de ellas para una u otra aplicación.

A este elemento es adonde se han conectado la mayoría de los periféricos para comunicarse con él. Este dispositivo se comunica mediante I2C con el Raspberry PI a través del pin 20 y 21 que corresponden a SDA y SCL.

El suministro eléctrico al Arduino Mega se realiza mediante el suministro directo a través del pin GND y 5Voltios presente en la PCB. Igualmente está comunicado el Arduino Mega con el pin de Reset a la Raspberry Pi, para forzar el reinicio del módulo Arduino. Esta activación se realiza a nivel bajo.

La conexión de los esquemáticos está marcado en colores según el tipo de conexión. En la tabla inferior se puede ver el significado del código de colores.

**Tabla 1 Código de colores del Arduino mega 2560.**

Color	Tipo de conexión
Negro y Rojo	Alimentaciones
Azul	Conexiones digitales
Naranja	Conexiones analógicas
Marrón	Conexiones de comunicación

Este bloque de DesignSpark presenta un elevado número de conexiones para representar exactamente el mismo número de pines físicos insertables que tiene un Arduino Mega.

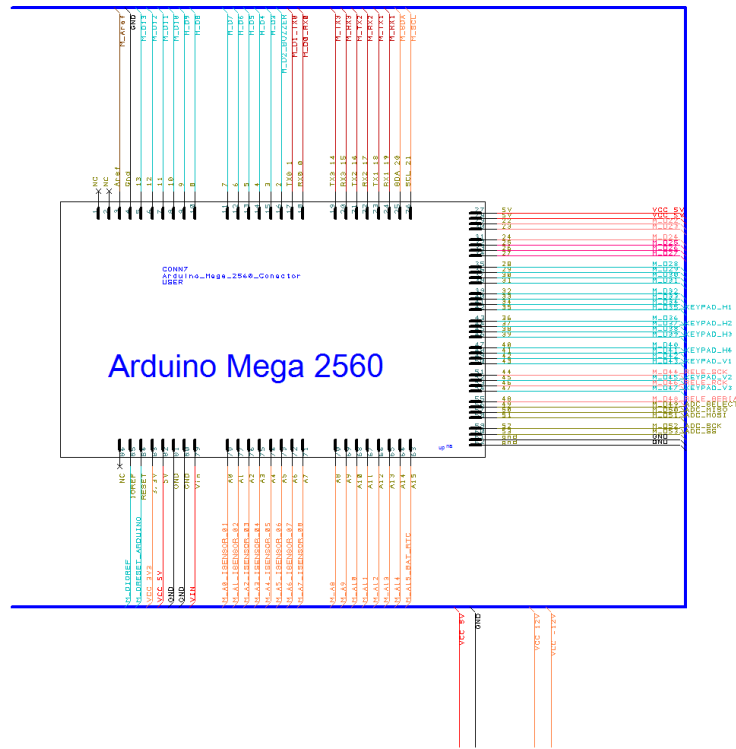


Figura 33 Esquemático de conexión de un Arduino Mega 2560

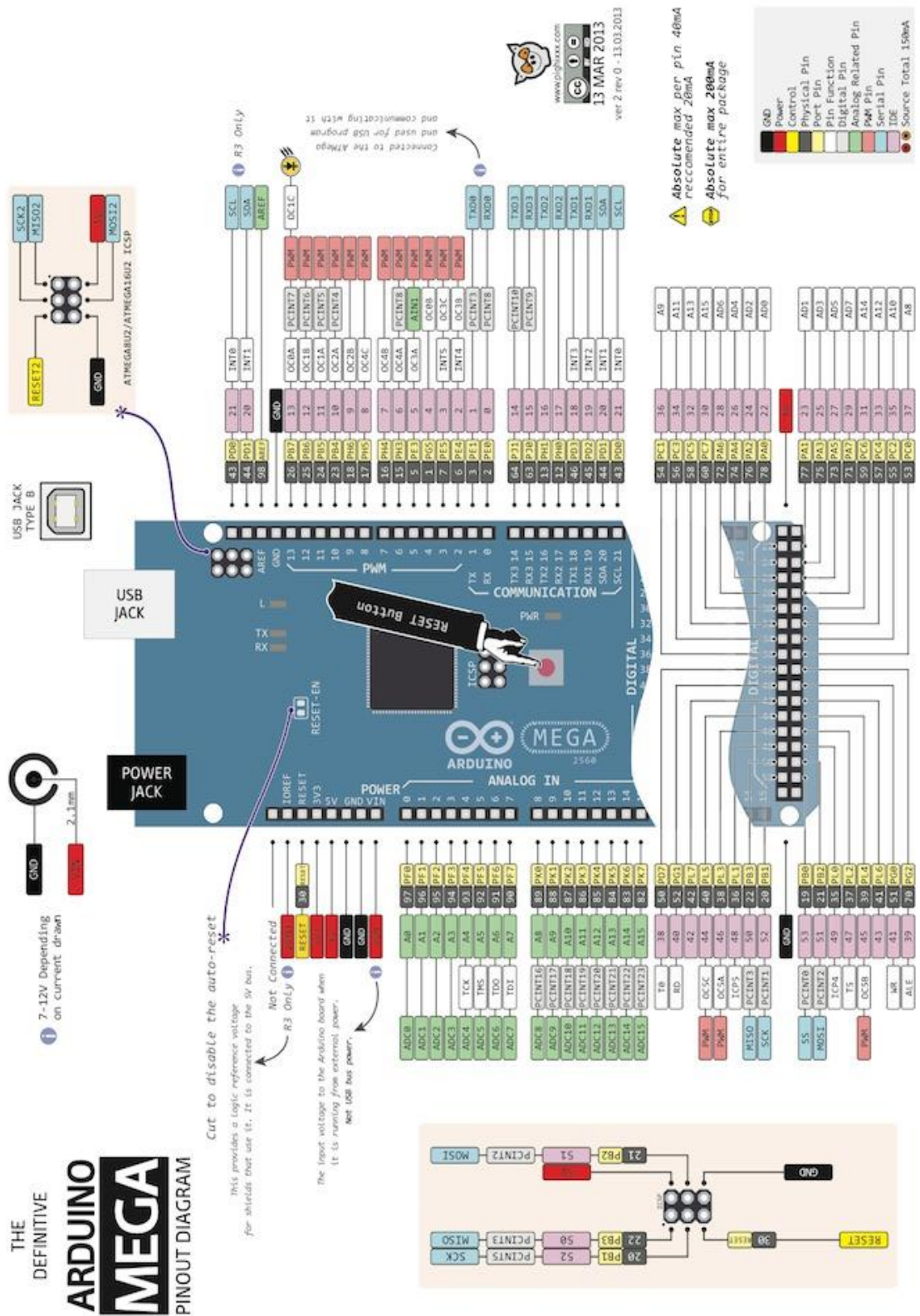


Figura 34 Diagrama completo de conexiones de un Arduino Mega 2560

## ESQUEMÁTICO LCD

El esquemático de la pantalla LCD que usamos es muy sencilla. Se trata de un conector de cuatro posiciones que agrupa GND, VCC, 5V, SCL y SDA en un único conector para hacer funcionar la pantalla LCD y que se alimenta a 5 voltios empleando comunicación I2C.

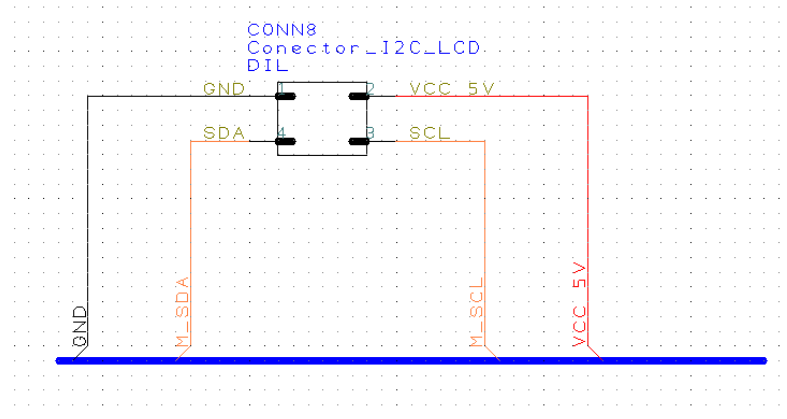


Figura 35 Esquemático de conexión de la pantalla LCD.

La conexión física entre el conector y la pantalla LCD se realiza mediante cuatro cables soldados y unidos a la PCB.

Este dispositivo aporta un fácil control y muestreo de información desde su panel LED. No obstante su uso se encuentra eclipsado por el uso de un monitor de ordenador conectado al Raspberry.

## ESQUEMÁTICO KEYPAD NUMÉRICO

Para la conexión del keypad numérico usamos un cableado sencillo formado por un sistema de 7 cables. Se usan 3 pines de conexión para las filas verticales y 4 pines de conexión para las filas horizontales. El Arduino va realizando un mapeado de los 7 siete cables para localizar que filas y columnas están pulsadas, determinando los botones pulsados.

Aunque existen otras formas de conectar este teclado numérico está es la más sencilla y factible al tener disponible pines sobrantes del Arduino Mega 2560. La mejora de este sistema implica un montaje usando dos circuitos integrados que hagan la tarea de cambiadores de registro de entrada y de salida.

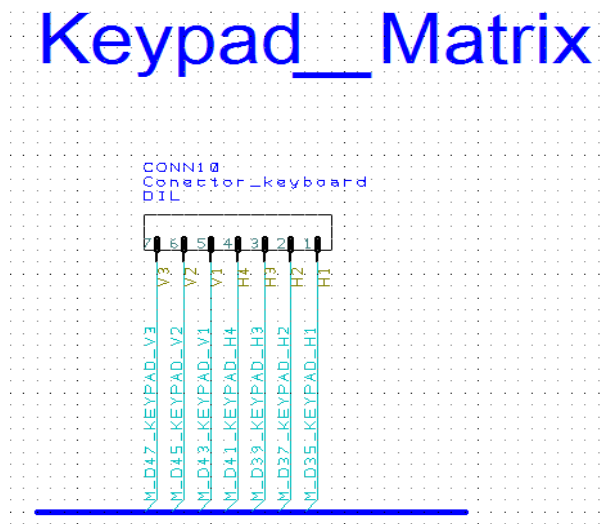


Figura 36 Esquemático de conexiones del Keypad de 12 posiciones.

## ESQUEMÁTICO RELÉ 8 CANALES

Otro de los esquemáticos diseñados para nuestro proyecto es el que se encarga de la gestión del array de 8 relés. Este array se controla mediante el uso de 8 entradas digitales. Para no usar 8 pines digitales del Arduino Mega hemos usado un Shift Register (Cambiador de registro) con Latch mantenido para poder controlarlo únicamente mediante 3 pines digitales junto con las conexiones de alimentación

La funcionalidad de Latch mantenido permite dejar fijo el estado sin tener que estar recargando continuamente el estado de los relés. Además el uso de una opción que no implica el Latch mantenido implicaría el funcionamiento defectuoso de los relés, seguramente sin fuerza suficiente para poder llegar a activarlos.



Figura 37 Detalle de la PCB con 8 relés.

En el diseño del circuito de control de los relés hemos usado un “MM74HC595” de ST<sup>22</sup>. Está configurado para activar cada uno de los canales del módulo de relés a través de un conector en la PCB. A través de este conector de 10 posiciones suministramos alimentación, GND y las 8 señales lógicas. Para el correcto funcionamiento además es necesario una serie de condensadores para



estabilizar las tensiones y mejorar la velocidad de acceso al dispositivo, así como eliminar interferencias que puedan afectar a su correcto funcionamiento.

En el anexo correspondiente se añaden los esquemáticos y documentos de fabricación de este módulo. Como detalle, este dispositivo presenta led de indicación de funcionamiento así como optoacopladores y diodos de protección para los relés. De esta forma aislamos perfectamente señal digital de potencia producida por el relé. Esta PCB presenta igualmente cortes en la misma para aislar las interferencias electromagnéticas generadas al conmutar los relés.

Este dispositivo está presente en forma de módulo de 16 relés, 8, 4, 2 y 1 relé. La Figura inferior muestra el diagrama de conexión del sistema de control de relés.

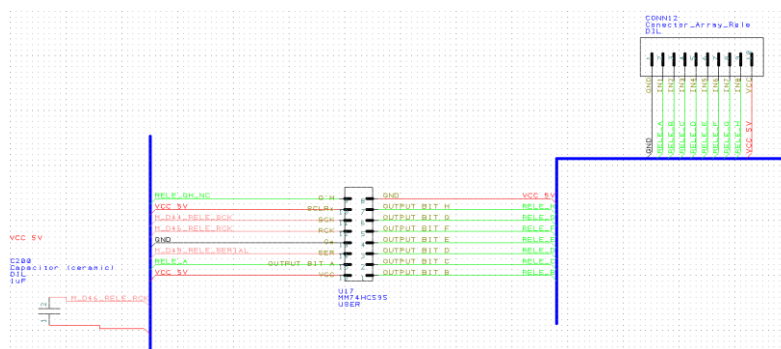


Figura 38 Esquemático de conexiones del control de relés.

## ESQUEMÁTICO BUZZER DE AVISO SONORO

Nuestra PCB lleva montado un Buzzer basado en un piezoeléctrico para avisar en caso de fallo del sistema o según las notificaciones oportunas. Se encuentra conectado directamente a un pin digital con capacidad PWM de Arduino. Su frecuencia se puede modular para emitir el tono del pitido deseado.

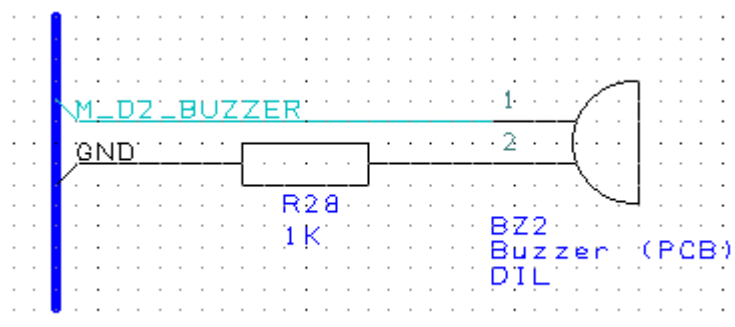


Figura 39 Esquemático de conexiones del circuito indicador de sonido.

La intensidad del sonido se regula según el valor de la resistencia instalada de esta forma será un sonido más o menos intenso.

## ESQUEMÁTICO ENTRADAS DIGITALES

Para la conexión con otros dispositivos hemos añadido 6 entradas digitales. Estas conexiones externas permiten recibir información de sensores. Estos sensores deben ser dispositivos con solo dos valores lógicos posibles “1” y “0”.

Para el aislamiento entre dispositivos externos y el PLC hemos incluido una etapa de optoacopladores. De esta forma la conexión de dispositivos externos convertirá los valores de entrada superiores a 4 voltios en salidas lógicas de 5 Voltios.

Este aislamiento optoeléctrico es fundamental en todo tipo de dispositivo que se usa en el entorno industrial, en circuitos en los que se van a usar tensiones y corrientes muy elevadas. Omitir este dispositivo puede fundir el prototipo entero por un simple descuido.

El funcionamiento del optoacoplador es muy sencillo. La entrada digital del PLC se conecta a un Diodo LED que conducirá al llegar a un mínimo de corriente momento en el cual activará la base del fototransistor interno presente en el encapsulado. Las resistencias presentes se encargan de limitar la corriente y tensión que pasa por el sistema. De esta forma los parámetros eléctricos del sistema estarán contenidos dentro de los parámetros aceptables del sistema. Los emisores de los optoacopladores están configurados en modo pull-down para dejar fijada la tensión en reposo a GND. El optoacoplador utilizado para proteger la electrónica es el CNY750C. Cada uno de ellos únicamente tiene un único canal.

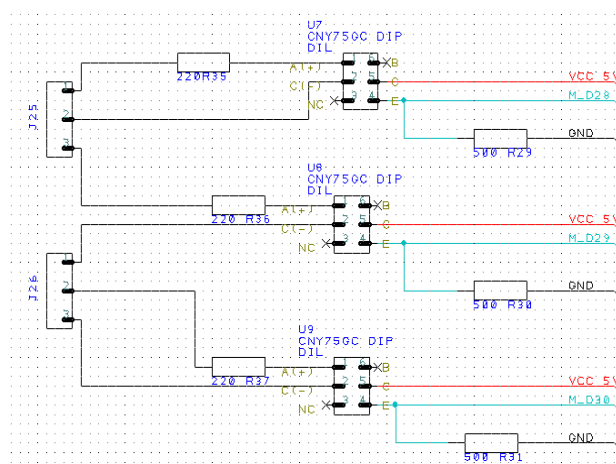


Figura 40 Esquemático de los canales digitales de entrada 1

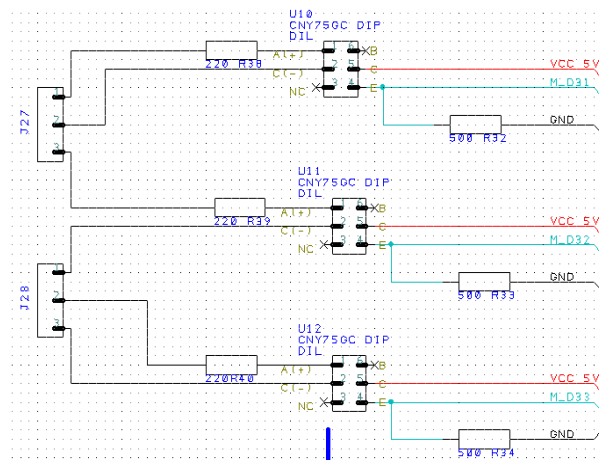


Figura 41 Esquemático de los canales digitales de entrada 2

## ESQUEMÁTICO SALIDAS DIGITALES

Otra de las características de conexiones de nuestro proyecto es la capacidad para generar señales digitales y analógicas que permitan la conexión con dispositivos externos. Esto permite activar relés externos o actuar sobre otro tipo de dispositivos, un ejemplo de actuación puede ser la lectura de estas señales para iniciar una transmisión de radiofrecuencia.

Con este sistema disponemos de un total de 12 salidas. Estas salidas funcionan a 2 valores de tensión diferentes, 6 salidas a 5 Voltios y 6 salidas a 12 Voltios. Las salidas a 12 voltios están preparadas para conectarse a dispositivos que usen estas tensiones como valores de entrada. Por otro lado las salidas a 5 Voltios son más apropiadas para conectarse a dispositivos digitales.

Por otro lado es más complejo el aislamiento que presentan las salidas digitales que las entradas. En este bloque funcional de salidas digitales / analógicas disponemos de un primer aislamiento formado por un optoacoplador para aislar las señales digitales del Arduino y la etapa de potencia.

El segundo elemento que forma parte de este bloque cumple la función de sumidero de corriente. Mediante el uso de un Mosfet de potencia podemos proporcionar la corriente necesaria para activar dispositivos externos. El límite de esta corriente está únicamente limitada por la fuente de alimentación.

El Mosfet que se usa en este sistema es un Mosfet de canal N IRF520NPBF y un optoacoplador CNY750C para el aislamiento. Otra de las características del sistema es la capacidad para generar salidas analógicas gracias al uso de pines digitales con capacidad PWM. Estos pines se encuentran conectados a los optoacopladores, como resultado final generamos salidas analógicas comprendidas entre 0 y 5 voltios y entre 0 y 12 voltios.

Para el correcto funcionamiento de esta electrónica es necesario usar resistencias que limiten los parámetros eléctricos del sistema y condicionen el funcionamiento del sistema bajo los parámetros necesarios.

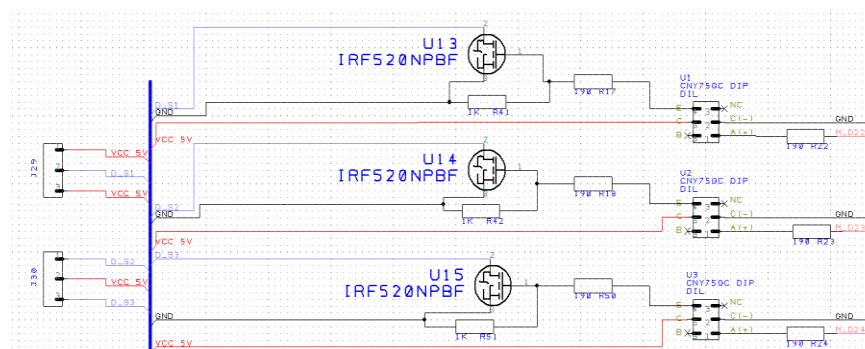


Figura 42 Esquemático de las salidas digitales a 5 Voltios

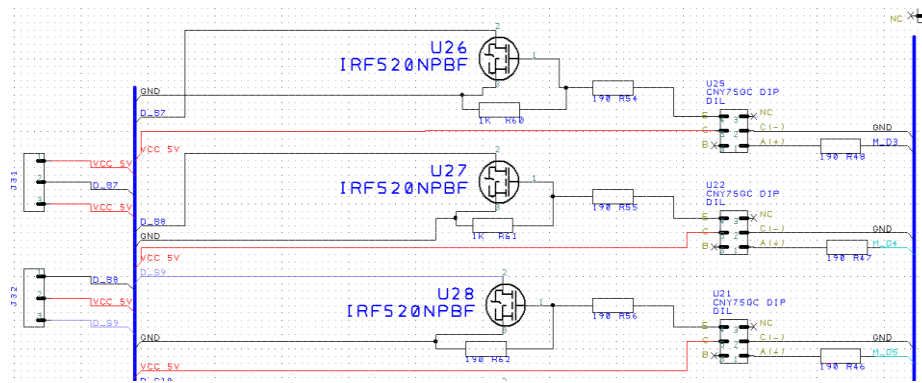


Figura 43 Esquemático de las salidas digitales a 5 voltios con PWM

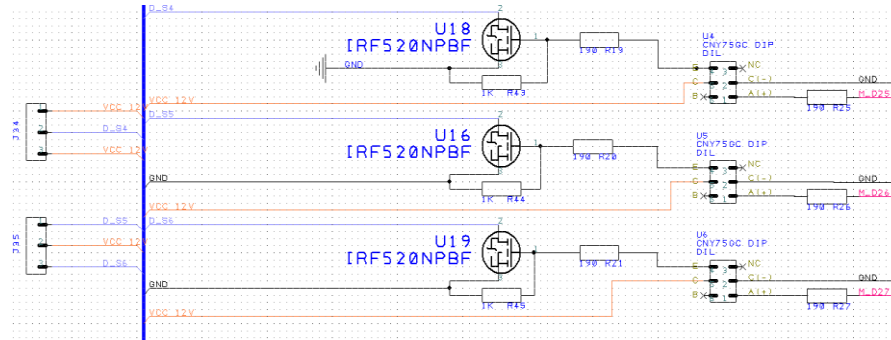


Figura 44 Esquemático de las salidas digitales a 12 Voltios

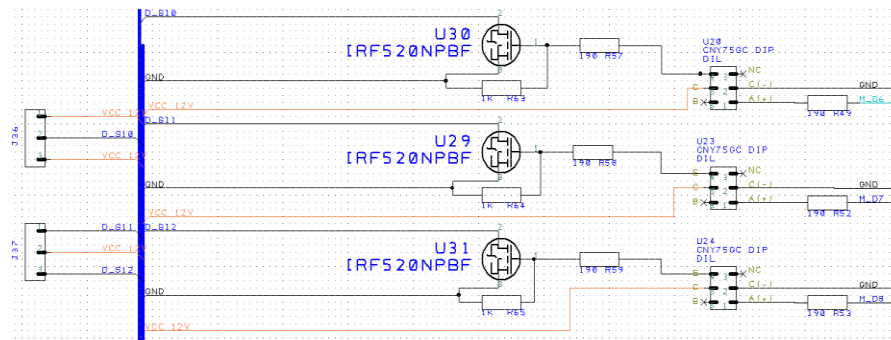


Figura 45 Esquemático salidas digitales a 12 Voltios con PWM

## ESQUEMÁTICO SENSOR ANALÓGICO

Para el sensado de canales analógicos hemos introducido un circuito integrado MCP3208 para realizar la conversión Analógica Digital con 12 bits de precisión. La precisión que aporta este elemento es superior a la capacidad de conversión analógica digital que presenta un Arduino. Los Atmega328P o similares que suelen integrar los Arduinos tienen una menor precisión formada por 10 bits.

Este sistema es capaz únicamente de sensar valores comprendidos entre 0 y 5 Voltios debido al sensado directo de las entradas analógicas al circuito integrado. Para la mejora del sistema se debe modificar este circuito para ser capaz de sensar entradas comprendidas entre 0 y 10 voltios.

La comunicación de este dispositivo se realiza mediante una línea de comunicación SPI con el Arduino. Esto permite una fácil comunicación usando un sistema de 4 hilos. Por otro lado para la simplificación del dispositivo esta conexión no se encuentra aislada mediante optoacopladores.

La imagen inferior muestra el esquemático de conexión que tiene nuestro dispositivo. Los 8 canales se encuentran conectados directamente a los conectores de conexión externa.

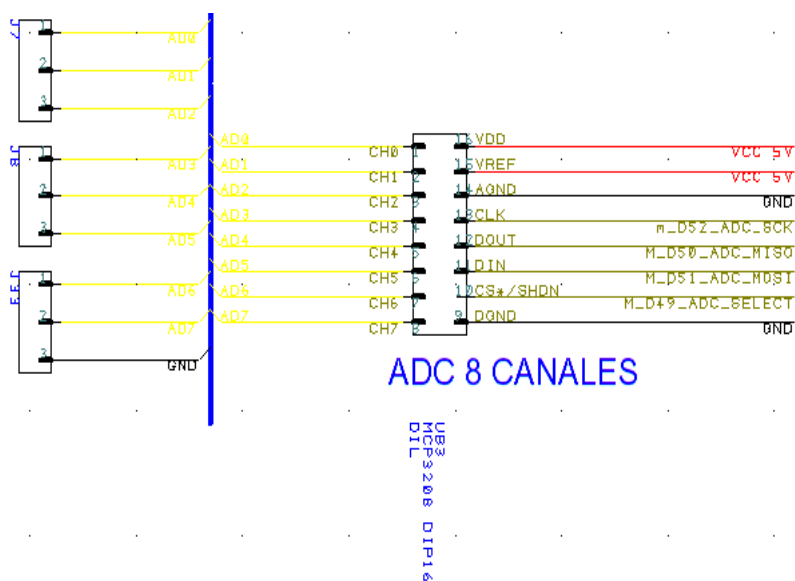


Figura 46 Esquemático de conexiones del conversor analógico digital de precisión.

## ESQUEMÁTICO ENTRADAS LAZO DE CORRIENTE

En el diseño del esquemático para el lazo de corriente originalmente se ha intentado un diseño usando una configuración que implementa un amplificador de instrumentación para realizar un sensado de precisión mediante resistencias. Esta solución no ha funcionado tal y como debería, por lo tanto hemos tenido que cambiar el diseño empleando una conversión analógica mediante una resistencia de 250 Ohm.

El Amplificador de instrumentación se ha intentado usar para realizar la amplificación del voltaje generado en una resistencia. En esta resistencia se genera el voltaje tras el paso de la corriente de los sensores de corriente externos.

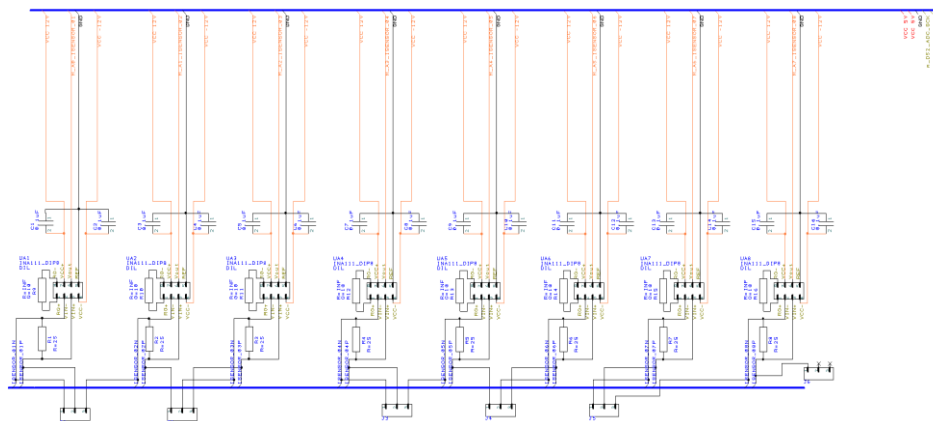


Figura 47 Esquemático de conexión de los 8 sensores de corriente según la versión previa.

Como solución finalmente implementada para solucionar el fallo de diseño, se ha diseñado una PCB externa un circuito eficaz para realizar este sensado. Consiste en un circuito integrado formado por un INA3221 de triple canal que permite sensar valores de corriente comprendida entre 4 y 20 mA así como valores analógicos y de potencia.

Este complemento al PLC se comunica mediante una línea I2C. Las interfaces y protocolos de comunicación se explican más adelante en el [capítulo de comunicaciones](#). El siguiente diseño del prototipo implementara este sistema de medición de corrientes.

La Figura inferior muestra el esquemático del circuito adicional para el sensado de corriente. Unas resistencias se encargan de la conversión de corriente a voltaje.

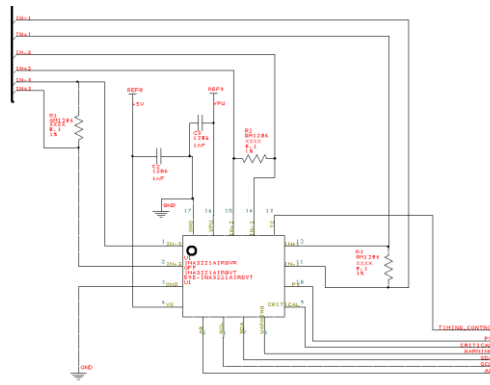


Figura 48 Esquemático INA3221 principal

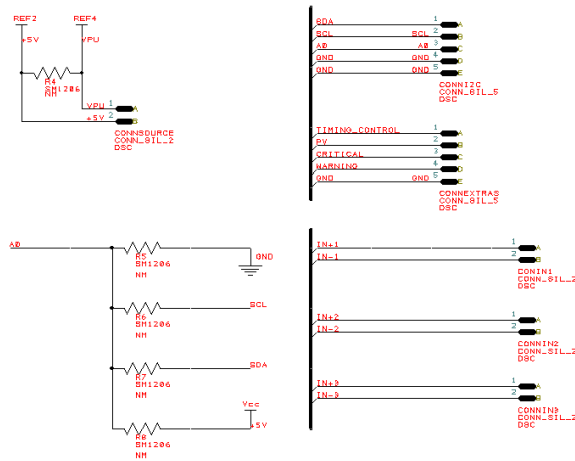


Figura 49 Esquemático conectores INA3221



## ESQUEMÁTICO FUENTE DE ALIMENTACIÓN

Para realizar la alimentación desde la fuente de alimentación hemos añadido dos conectores fijos para PCB de donde suministramos las señales de GND, VCC 5V, VCC 12V y VCC -12V. Estas conexiones se instalaran en un borde de la PCB para poder alimentar el sistema. La fuente elegida es la LPT23 del fabricante Astec. Cuenta con fusibles de protección y diferentes medidas de seguridad.

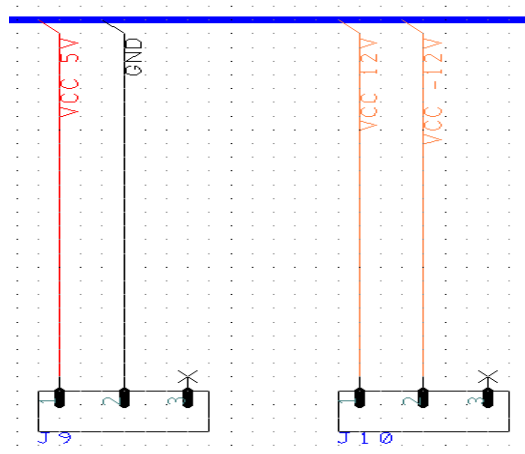
La fuente de alimentación ha sido elegida en base a las tensiones de salida de las que dispone. La alimentación a 5 Voltios del circuito es fundamental para poder alimentar al Arduino Mega y al Raspberry Pi. Las alimentaciones a 12 Voltios permiten generar salidas analógicas con un valor máximo de 12 Voltios. Inicialmente la alimentación a -12 Voltios estaba destinada para suministrar la tensión necesaria para el funcionamiento de los sensores de corriente iniciales.

La tabla inferior muestra los parametros técnicos de la fuente de alimentación.

**Tabla 2 Parámetros técnicos fuente de alimentación.**

Salida 12 Voltios	0.7 Amperios
Salida -12 Voltios	0.7 Amperios
Salida 5 Voltios	5 Amperios
Eficiencia	70%
Rizado y Ruido	120mV Pk-Pk, 50 mv Pk-Pk
Tensión entrada	120-300V DC, 85-264V AC
Potencia nominal	40 W
Protecciones	Sobretensión, Sobrecorriente

La figura inferior se muestra el esquemá de conexión de la fuente de alimentación.



**Figura 50 Diagrama de conexiones del conector de fuente de alimentación**

## ESQUEMÁTICO PCB CURRENT TESTER

Para poder hacer pruebas de sensores externos de corriente se ha creado un diseño que permita simular un sensor de corriente comprendido entre 4 y 20 mA. Este diseño se basa en el uso de un amplificador operacional junto con un transistor para actuar como generador de corriente.

Un potenciómetro es el elemento que permite gestionar la corriente que se desea generar. Este dispositivo puede generar como salida una corriente comprendida entre 4 y 18 mA

Con el fin de poder testear el funcionamiento del sensor de corriente se ha creado un esquemático extra que permite sacar como salida una corriente deseada y así poder simular los sensores de corriente. Los sensores de corriente generan señales comprendidas entre 4 y 20 mA.

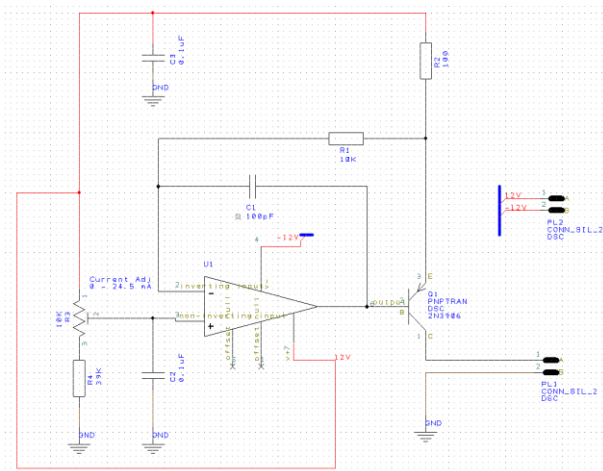


Figura 51 Esquemático de generador de test de lazos de corriente.



## Diseño del Layout

Teniendo finalizado el diseño de los esquemáticos y habiendo realizado los cálculos de los diferentes componentes. El siguiente paso es la creación del Layout mediante DesignSpark. Una de las etapas más importantes a la hora de generar el Layout es la creación de las huellas necesarias para cada elemento. Para diseñar correctamente las footprint de los correspondientes elementos se debe seguir la información que aparece en los datasheet de cada componente. Los footprint son diferentes según el encapsulado utilizado en el diseño.

Un ejemplo de huella de un optoacoplador de 4 canales es la que se puede ver en la figura inferior, donde aparecen diferentes tipos de encapsulados para un mismo componente.

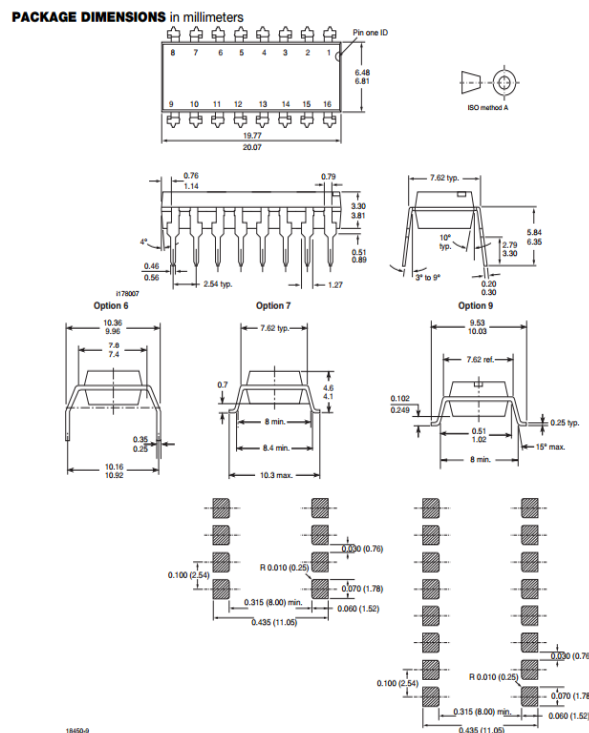


Figura 52 Características dimensionales de un Circuito Integrado de agujero pasante.

Los fabricantes dan siempre las dimensiones de los encapsulados para facilitar el correcto diseño electrónico. Esto implica las dimensiones de los componentes junto con las normas y requisitos necesarios en el diseño de los footprint.

Uno de los puntos importantes en la creación del layout es tener en cuenta la existencia de las diferentes capas que componen una PCB. La tabla inferior muestra las diferentes capas existentes y su utilidad.

Tabla 3 Capas de una PCB

Nombre de la Capa	Finalidad de la capa
Cobre superior	Está capa de la cara superior, contiene el cobre conductor que permite la conducción eléctrica. Forma conexiones y planos de tierra.
Cobre interno	Es un conjunto de capas internas de la PCB comprendidas entre 0 y 12 que añaden capas extras para la conexión eléctrica de componentes. Es necesario en diseños complejos.
Mascara superior	La capa de la cara superior contiene el barniz que protege y evita la exposición del cobre al exterior. Evita la oxidación del cobre y permite la conexión en las aperturas correspondientes para la soldadura de componentes.
Serigrafía superior	Está capa de la parte superior de la PCB contiene la serigrafía con tinta de textos y diagramas. Es un componente informativo para facilitar la lectura y montaje de la PCB.
Montaje Superior	Está capa contiene la información para facilitar el ensamblaje de capas por parte de máquinas de montaje automática.
Documentación superior	Es una capa adicional que incluye información textual sobre el diseño.

En la figura inferior se puede ver un esquema de montaje de las diferentes capas de una PCB

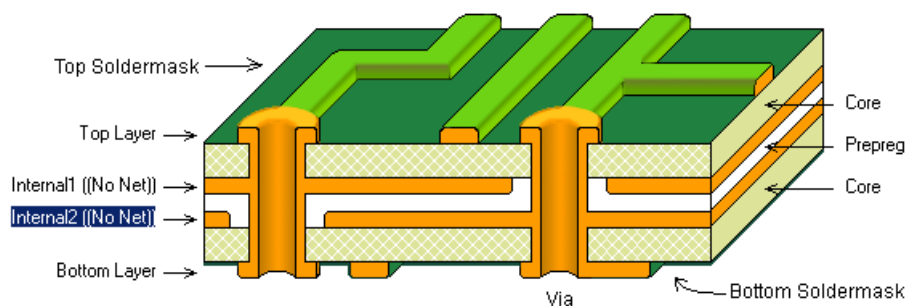


Figura 53 Esquema layers de una PCB.

## REGLAS DE DISEÑO

Se ha decidido que por las características de la PCB vamos a fabricarla en [www.eurocircuits.com](http://www.eurocircuits.com) una empresa europea que se dedica a la fabricación industrial de PCB. Esta empresa está especializada en la fabricación de pocas unidades, llamados prototipos. Tienen un precio asequible para la fabricación de pocas cantidades, con tiempos relativamente cortos de fabricación. El precio de estas fabricaciones cae rápidamente con el número de unidades fabricada.



Figura 54 Logotipo de Euro Circuits fabricante de la PCB.

Para diseñar este dispositivo hemos aplicado las guías de diseño de PCB que suministran desde eurocircuits, de esta forma nos aseguramos de la correcta fabricación de nuestra PCB. En este punto se tienen en cuenta dimensiones mínimas de todos los elementos como ancho de pista mínimo, diámetro de vía mínimo o separación mínima entre pistas. Las reglas de diseño se pueden consultar en esta dirección <http://www.eurocircuits.com/PCB-design-guidelines>.

Una de las mayores ventajas al diseñar una PCB mediante Eurocircuits es la potente herramienta de validación y comprobación online que proporcionan en su página web. Mediante su página web se puede asegurar la fabricabilidad y el correcto diseño de la PCB mediante su software. Igualmente una vez enviada a fabricar y comprobada que no tiene fallos con su herramienta online, los propios técnicos revisan el diseño para comprobar que no existe ningún error adicional.

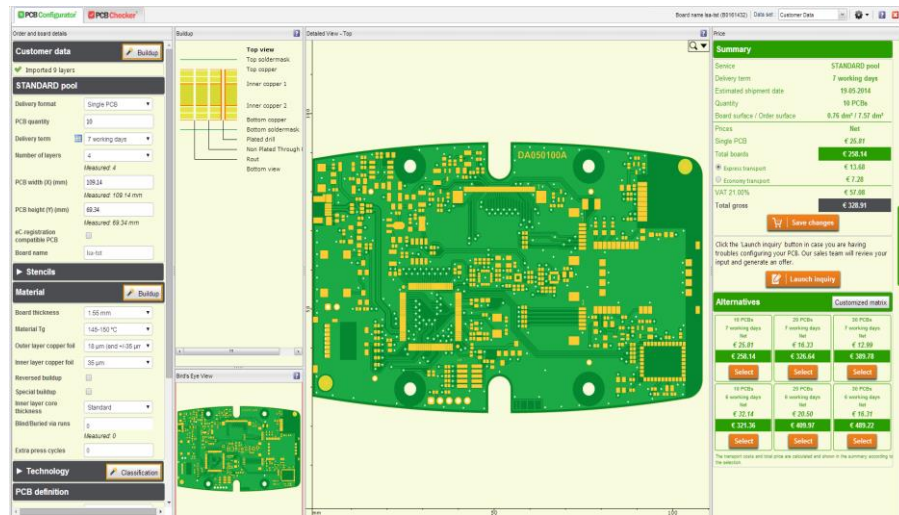


Figura 55 Herramienta online de Euro Circuits. PCB Checker para comprobar la fabricabilidad de las placas.

Por otro lado hemos seguido condiciones de diseño adicionales. Estas condiciones de diseño buscan establecer el ancho de las pistas de cobre. Para ello se ha distinguido entre conexiones eléctricas de señal y potencia. El ancho nominal de pistas para señal es de 0.254 mm y 0.512 mm para señales de potencia y alimentaciones.

Una buena regla de diseño es sobredimensionar las pistas de suministro eléctrico. La regla mínima de diseño es usar la ley de Ohm para evitar el sobrecalentamiento de las pistas debido al paso de corriente.

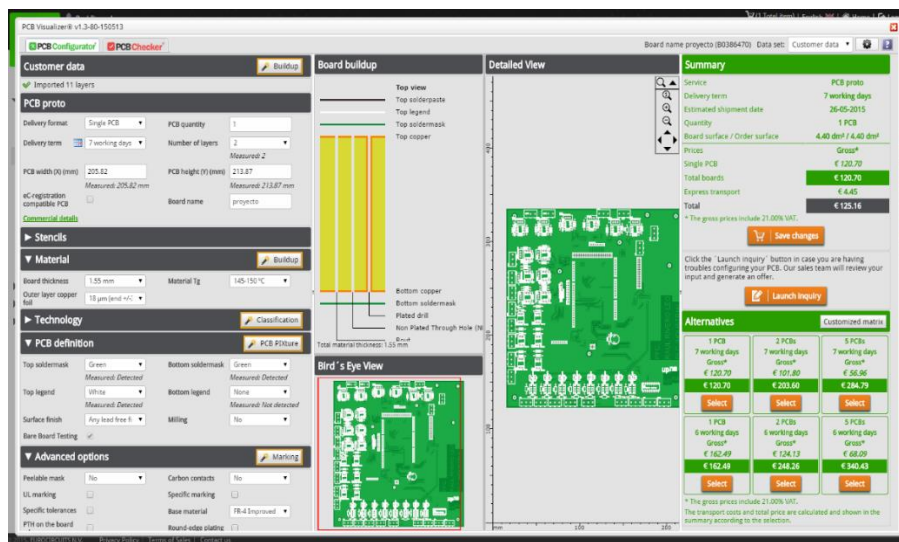


Figura 56 Herramienta online de Euro Circuits. PCB Configurator para seleccionar las características de fabricación de nuestra PCB.

Hemos creado esta PCB diseñándola a doble cara. Debido a la gran cantidad de espacio disponible no ha sido difícil realizar las conexiones eléctricas. No obstante en diseños más complicados en los que las dimensiones requeridas son más reducidas es fundamental el diseño a varias capas.

## COLOCACIÓN DE LOS COMPONENTES EN LA PCB

Otro de los puntos más críticos en el diseño de la PCB es la colocación de cada elemento en una zona determinada de la PCB. Para ello debemos seguir un patrón de colocación de los elementos.

Hemos intentado agrupar los componentes según la funcionalidad. De esta forma hemos juntado todos los Conectores fijos en los bordes de la PCB y colocados juntos según la funcionalidad que desempeñan.

Los circuitos integrados que actúan como optoacopladores igualmente los hemos colocado cerca de los terminales correspondientes para reducir las distancias. De esta forma dividimos la PCB en zonas según la función que desempeñan. Como defecto de esta colocación resulta en un gran desaprovechamiento del espacio que podría reducir el área de la PCB un 75%

Otra característica en la colocación de la PCB ha sido la forma de conectar los módulos externos, de tal manera que hemos usado conectores para poder insertar los módulos, y de esta manera no se queden fijos siendo posible intercambiarlos.

Es especialmente útil para poder colocar el Arduino Mega 2560, Raspberry PI o el módulo RTC. La forma de conexión del Arduino mega permite que se inserten nuevas Shield acoplables al conjunto de Arduino Mega y PCB.

## LAYOUT POR ZONAS

En este apartado se muestra el layout de cada elemento mostrando los elementos cercanos y detalles sobre las condiciones de su diseño. Las dimensiones finales que tiene la PCB son de 205.82 mm de ancho y 213.87 mm de alto. El número de capas que tiene la PCB es de 2 diseñadas a doble cara. El material que se ha utilizado para su fabricación ha sido en fibra de vidrio FR4.

El resultado final de la PCB se puede apreciar en las dos imágenes inferiores. En las imágenes se puede comparar el diseño simulado por Eurocircuits y el acabado final con los componentes montados.



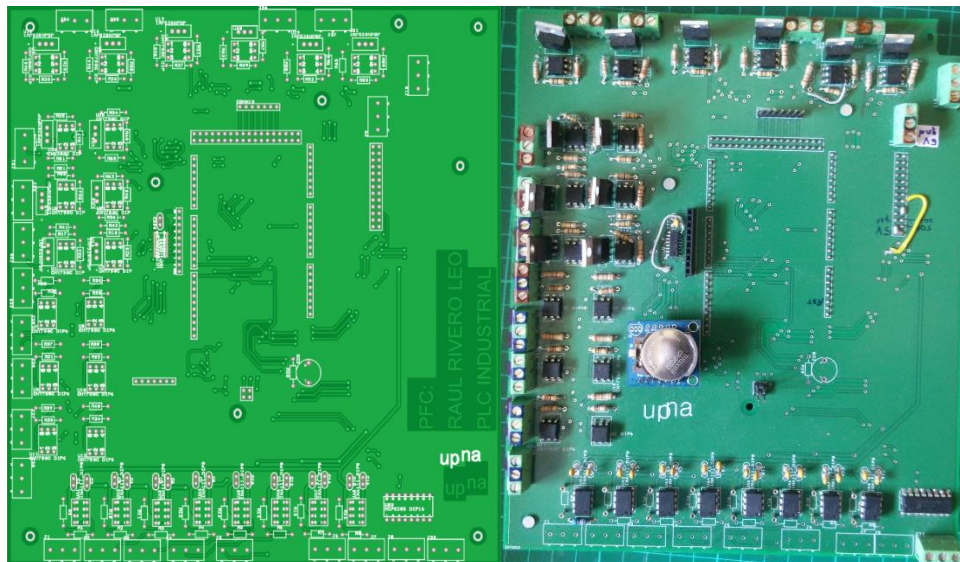


Figura 57 Layout superior completo extraído de EC [EuroCircuits] y resultado final de la PCB montada

## LAYOUT ARDUINO MEGA 2560

Para el montaje del Arduino Mega se ha utilizado varios conectores de montaje en PCB para insertar el módulo Arduino. El montaje por el lado inferior minimiza el espacio ocupado pero hace imposible el montaje de Shield compatibles. Para hacer esto posible requiere modificar los conectores del Arduino Mega.



Figura 58 Montaje final del Arduino Mega 2560.

## LAYOUT RASPBERRY PI

El montaje de la Raspberry en nuestra PCB se ha realizado aprovechando el conector de 26 pines presente en la Raspberry. Esto simplifica la comunicación de la Raspberry con el resto de la placa, al mismo tiempo que se aprovecha para alimentar al módulo principal.

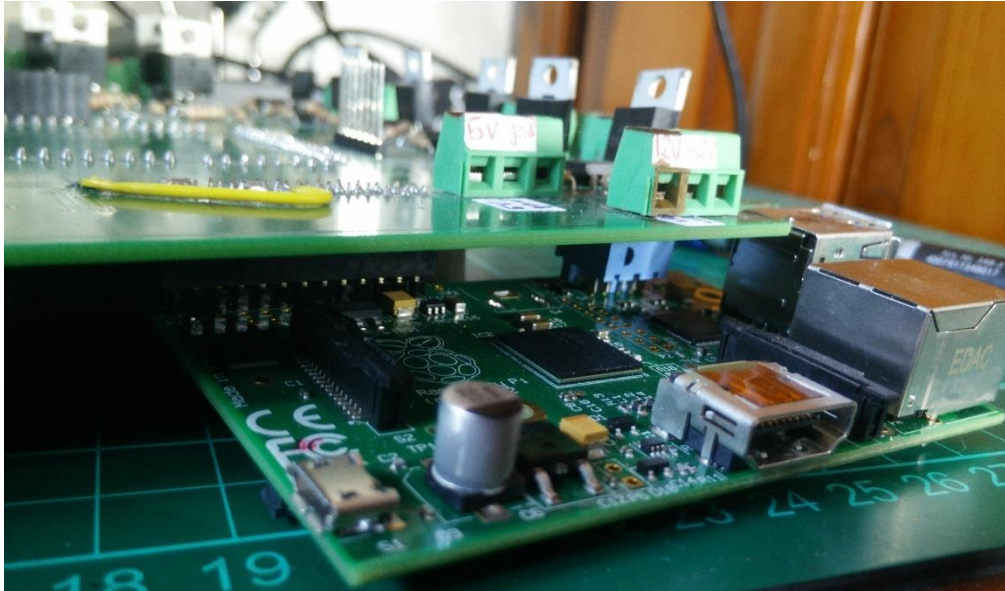


Figura 59 Detalle del Layout Superior del Raspberry Pi. Extraído de EC

## LAYOUT PANTALLA LCD DE 2 LÍNEAS.

En nuestra PCB hemos introducido un conector de 4 posiciones para gestionar el funcionamiento de la pantalla LCD. Se ha buscado colocar un conector lo más compacto posible situado en la mitad de la PCB. Este Conector permite la conexión de las alimentaciones y las conexiones I2C.

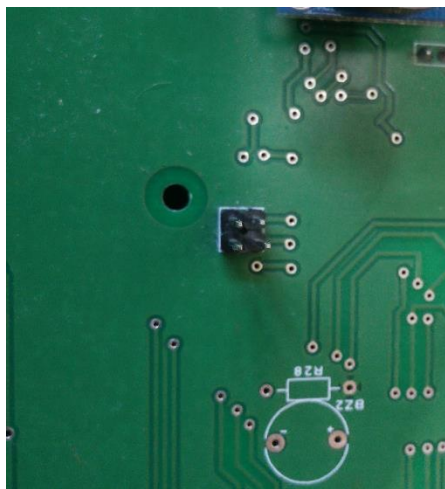


Figura 60 Detalle del conector de la pantalla LCD.

## LAYOUT KEYPAD NUMÉRICO

Como elemento para conectar el Keypad numérico hemos introducido un conector de 7 posiciones conectado directamente a unos pines digitales del Arduino mega 2560. Se ha elegido situar el conector en la parte central de la PCB, lo más cerca posible de los pines digitales del Arduino Mega 2560.

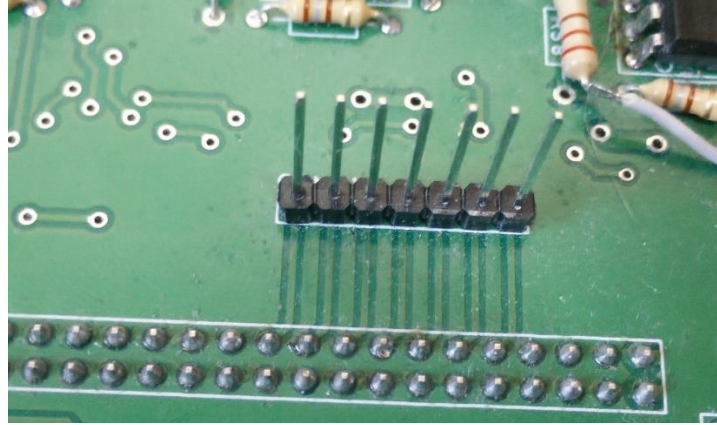


Figura 61 Montaje del conector del keypad.

## LAYOUT RELÉ 8 CANALES

Otro de los bloques funcionales que debemos alojar en la PCB corresponde al circuito de gestión de los relés. Este circuito está formado por un circuito integrado Shifter Register junto con condensadores para estabilizar la tensión en el circuito.

Como elementos adicionales tenemos disponibles el conector de 10 posiciones que envía las señales y la alimentación al módulo de relés.

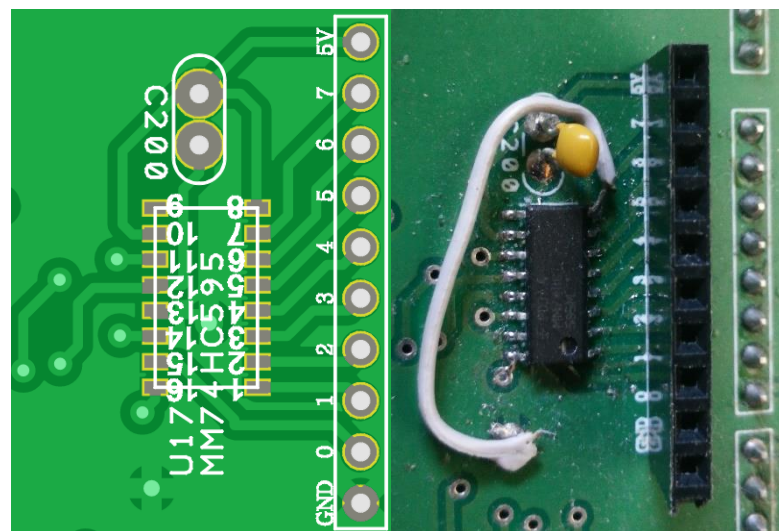


Figura 62 Detalle del Layout Superior del conector de Relé comparado con el montaje final.

## LAYOUT BUZZER

Otro de los elementos introducidos en nuestro layout ha sido el Buzzer formado por los dos elementos necesarios para su funcionamiento, el piezoeléctrico y su resistencia asociada. En la figura inferior vemos la comparativa entre el diseño simulado y el montaje en placa.

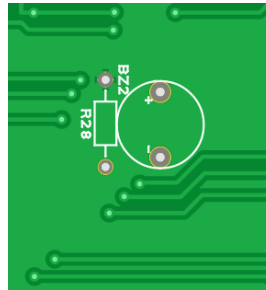


Figura 63 Detalle del circuito de aviso sonoro Buzzer.  
Extraído de EC

## LAYOUT SALIDAS DIGITALES

En la figura inferior vemos el diseño simulado de una parte de las salidas digitales del sistema comparado con el diseño final montado en PCB.

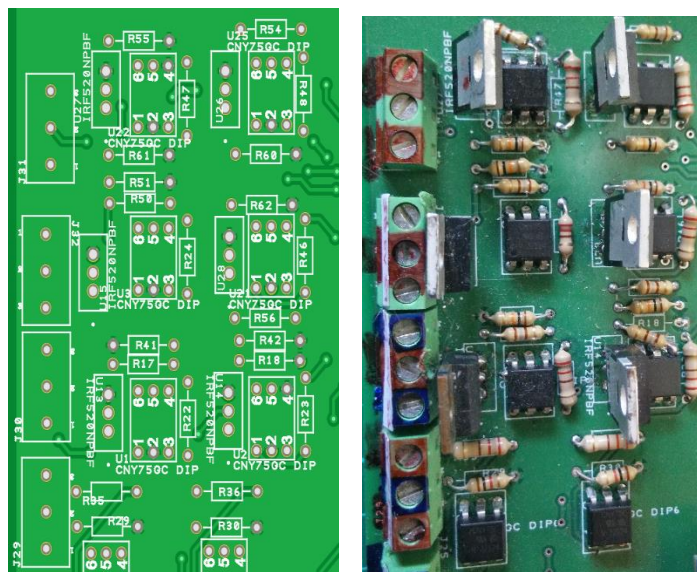


Figura 64 Detalle del Layout Superior comparado con el montaje final de las salidas digitales.

## LAYOUT ENTRADAS DIGITALES

En la figura inferior se muestra el resultado final de las entradas digitales del sistema comparadas con el montaje sobre la PCB.

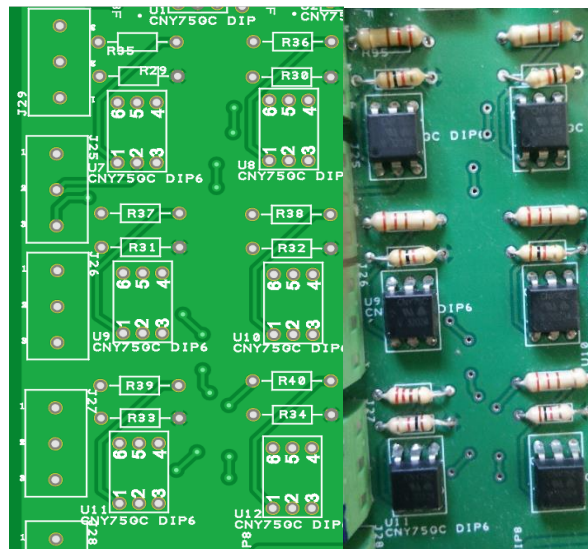


Figura 65 Detalle del Layout y el montaje final de las entradas Digitales.

## LAYOUT SENSOR ANALÓGICO

Otro de los layout diseñados para gestionar las entradas del PLC es el que emplea el circuito integrado MCP3208. Este se presente en formato DIP16 al ser más fácil de fabricar y sobre todo al no tener limitaciones de espacio en nuestra PCB.

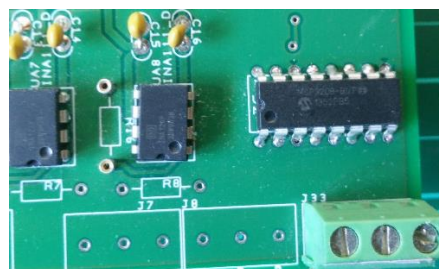


Figura 66 Detalle del Layout Superior del Circuito Integrado AD de precisión.

## LAYOUT ENTRADA LAZO DE CORRIENTE.

En el siguiente apartado vemos el montaje inicial para el sensado por lazo de corriente en nuestra PCB. Usamos como sensor un amplificador de instrumentación INA para realizar el sensado. Sin embargo estos dispositivos no funcionaron como debían así que se ha tenido que recurrir a otra solución como ha sido explicada en el apartado de esquemáticos.

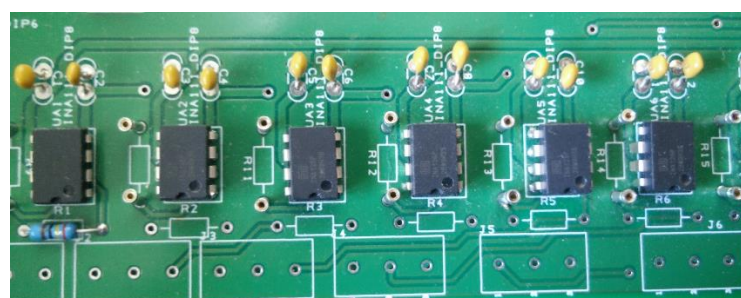
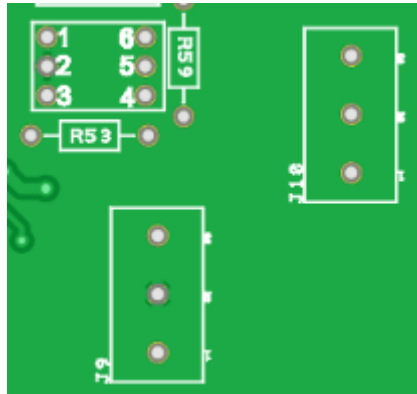


Figura 67 Detalle del Layout Superior del sistema de sensado de corriente. Extraído de EC.

## LAYOUT ENTRADAS FUENTE DE ALIMENTACIÓN

En este punto se muestra el layout de los conectores utilizados para alimentar al sistema a través de los conectores de 5, 12 y -12 Voltios



**Figura 68 Detalle del layout del conector de entrada de alimentación.**  
Extraído de EC



# PROGRAMACIÓN

La programación realizada en este proyecto ha sido una de las claves para conseguir este prototipo operativo. Nuestro proyecto está formado por 3 bloques diferentes que se comunican entre ellos usando diferentes protocolos. En este proyecto formado por los dos módulos con capacidad lógica, el Arduino mega 2560 y el Raspberry Pi, se ejecuta diferentes códigos diferenciados en cada una de ellas.

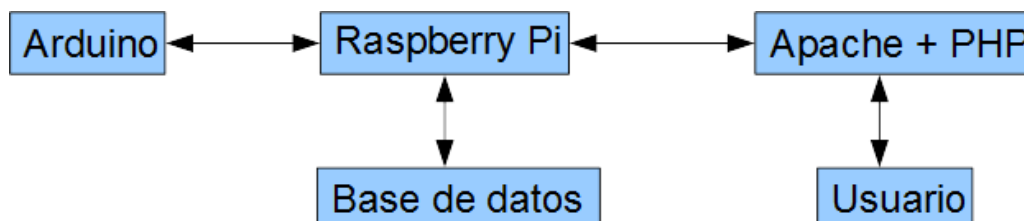


Figura 69 Diagrama de comunicaciones principales.

## Arduino

La programación del Arduino mega 2560 se ha realizado programándolo en Arduino. El lenguaje de programación Arduino es un lenguaje de programación de alto nivel con un potente nivel de abstracción, que está basado en el lenguaje de programación Processing el cual es similar a C++.

Processing es un lenguaje de programación y entorno de desarrollo de código abierto que está basado en Java de fácil aprendizaje que tiene su origen en proyectos multimedia y aplicaciones interactivas.

La programación de este lenguaje se puede realizar en numerosas plataformas siendo las dos más usadas Arduino IDE<sup>23</sup> y Atmel Studio.

Atmel Studio es la aplicación ofrecida por el fabricante ATMEL fabricante de los chip Arduino para programar sus microcontroladores. Contiene toda la información para poder programar estos micros en C++. Para poder hacer compatible la programación de Arduino con Atmel Studio requiere una pequeña configuración del programa para incluir ficheros include para permitir su uso. Es una solución muy potente al ser la única que permite funciones de Debug.

La otra solución posible y la más extendida entre otras cosas por su sencillez es el uso de Arduino IDE. Esta interfaz de programación ha sido la utilizada para programar nuestro Arduino, concretamente la versión Arduino IDE 1.5.3 Para realizar la programación se ha seleccionado como tipo de placa Arduino una Arduino Mega 2560 con el puerto COM<sup>24</sup> correspondiente.

Por otro lado para poder realizar la programación desde Atmel Studio requiere el uso de un programador especializado como AVR Dragon<sup>25</sup>. Esta herramienta de desarrollo permite realizar tareas de programación y debugging de software.



El código fuente escrito se estructura en 9 ficheros de programación entre ficheros de código y ficheros de referencia. La inclusión de ficheros adicionales permite mejorar la legibilidad, desarrollo y mantenimiento del programa al tener separada en diferentes ficheros según funcionalidades. De esta forma al mismo tiempo los ficheros creados son ligeramente reutilizables para otros programas. La inclusión de estos ficheros se hace mediante la llamada en el fichero principal de estos incluye.

Como elemento adicional a esta solución hemos creado algunas librerías propias para facilitar la automatización de algunas tareas como la inicialización del programa y la configuración previa de todos los elementos presentes.

Para la gestión del programa y el uso de un método de programación optimizado y fiable hemos usado un diagrama de estados. De esta forma logramos un código fluido perfectamente legible.

El fichero fundamental del programa es el fichero “Principal.ino”

```
/*
  Código principal que va instalado en el Arduino Mega,
  Este firmware se encarga de ejecutar las órdenes recibidas
  Desde el Raspberry, así como leer cambios y comunicarlos
  De vuelta al Raspberry para ser escrito sobre la BD.
*/
//Librerías
#include <Init.h>      //Librería Init debe ir antes que setup.h. Gestiona las
inicializaciones
#include <Wire.h>      //Librería i2c
#include <comunicacionI2C_PFC.h> //Librería i2c para gestionar el proyecto

//Includes del programa
#include "setup.h"     //Gestiona la llamada a las diferentes inicializaciones
#include "i2cControl.h" //Gestiona los callback i2c y las funciones de control de
la información.
#include "constantes.h" //Gestiona todas las constantes
#include "loopState.h"  //Gestiona los estados del loop

void setup() {
  // put your setup code here, to run once:

  while (EstadoSetup != InicializacionFin) {
    //Inicializamos variables
    switch (EstadoSetup) {
      case InicializarVariables:
        InicializarSistema();
        EstadoSetup = InicializarComunicacion;

        break;
      case InicializarComunicacion:
        ConfigurarComunicacion();
        EstadoSetup = InicializarDebugMode;
```

```

        break;
    case InicializarDebugMode:
        //Fin de la finalización.
        ConfigurarDebugMode();
        EstadoSetup = InicializacionFin;
        break;
    }
}
}
void loop() {
    // put your main code here, to run repeatedly:
    //Bucle principal según diagrama de estados.

    switch (EstadoLoop) {
        case CheckInput:
            CheckCambios(DebugMode); //Comprobamos las entradas del sistema
            EstadoLoop = CheckDos;
            break;

        case CheckDos:
            //ReadBufferModule(DebugMode); //Leemos la información procedente de
los buffer.
            CheckAnalog(DebugMode);
            EstadoLoop = CheckTres;
            break;

        case CheckTres:
            //ActuadorModule(DebugMode); //Estado actuar sobre el sistema
            EstadoLoop = CheckCuatro;
            break;

        case CheckCuatro:
            //Comunicandoi2cModule(DebugMode); //estado comunicando i2c.
            EstadoLoop = CheckInput;
            break;
    }
    delay(1000); //Metemos un delay para ralentizar el sistema
}

```

En el funcionamiento de este programa hacemos primero las llamadas a las librerías propias de Arduino y posteriormente se llama a los ficheros include creado para este desarrollo.

Entre las llamadas a las librerías de Arduino se encuentran las siguientes:

- `#include <Init.h>` Esta es la librería creada por nosotros para poder hacer llamada a la configuración del sistema. De esta forma configuramos las entradas, salidas y líneas de comunicación para nuestro código.
- `#include <Wire.h>` Esta librería llama a las clases propias de las funciones i2c. Sin estos ficheros no se puede realizar comunicaciones i2c.

- #include <comunicacionI2C\_PFC.h> Este fichero incluye todas las funciones de configuración e inicialización de las comunicaciones

Entre el resto de ficheros incluidos en el proyecto están:

- #include “setup.h” Este fichero gestiona la llamada a las diferentes inicializaciones del sistema.
- #include “i2cControl.h” Este fichero gestiona los callback i2c y las funciones de control de la información como puede ser la codificación y decodificación de mensajes mediante i2c.
- #include “constantes.h” Aquí se registran todas las variables del sistema de forma que sean compartido para todas las funciones del sistema.
- #include “loopState.h” Este fichero finalmente controla las llamadas a las funciones de cambio de estado para ejecutar las tareas requeridas por cada módulo.

El programa de Arduino usa dos diagramas de estados. El primero es un diagrama de estados para la gestión de las inicializaciones y el segundo es el diagrama de estados usado en el funcionamiento normal del programa.

En el primer diagrama de estados el de configuración del sistema se usa la variable “EstadoSetup” para controlar la evolución en el Diagrama de estados.

En el proceso “InicializarVariables” se realizan las siguientes tareas:

- Configuración de cada pin como entrada o salida
- Configuración de los pines de salida según el estado requerido. Nivel bajo o alto.
- Tras esta configuración la variable Estadosetup entra al estado “InicializarComunicación”.

En el proceso “InicializarComunicación” se realizan las siguientes configuraciones.

- Se inicializa la comunicación serial y se configura a una velocidad de 9600bps
- Se configura la comunicación i2c uniéndose a la línea en modo esclavo con la dirección 0x04. Esta dirección es configurable desde el fichero constantes.h en la línea 27
- Envía por i2c un comando al Raspberry para indicar el encendido del Arduino e indicar de que está operativo.
- La variable “Estadosetup” entra al estado InicializarDebugMode.

Finalmente en el último proceso “InicializarDebugMode” se realiza las siguientes tareas:

- Se configura el “modoDebug” del sistema tanto del Arduino como del Raspberry Pi. La configuración prevista está mediante teclado para activar el modo “DebugMode”.
- Entrada en “InicializaciónFin”.

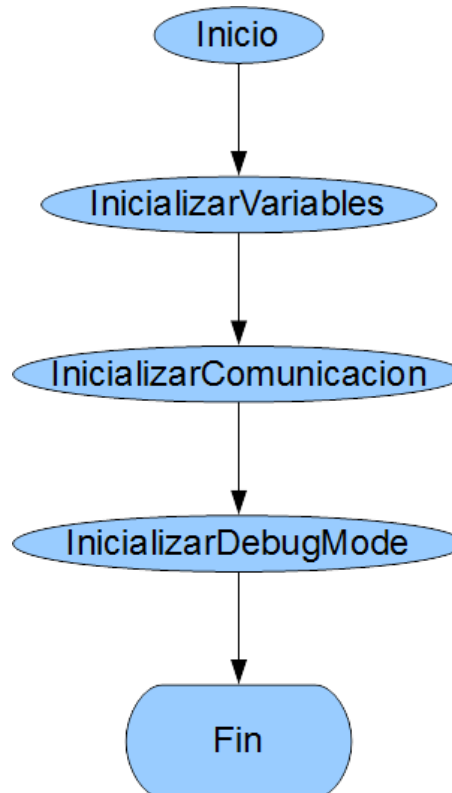
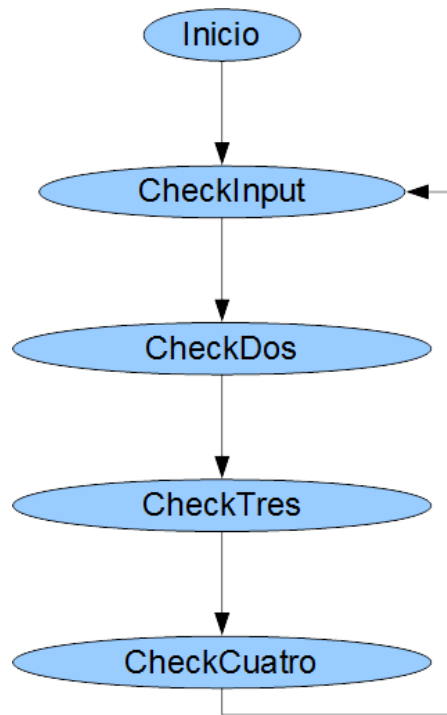


Figura 70 Diagrama de estados de configuración del Arduino Mega 2560.

Al entrar en “InicializaciónFin” el sistema ya está configurado y entra al bucle loop, momento en el cual entra en el diagrama de estados principal del sistema.

En el diagrama de estados principal la variable que gestiona el estado en el que se encuentra el sistema es “EstadoLoop”. Esta variable pasa por cuatro tareas: “CheckInput”, “CheckDos”, “CheckTres” y “CheckCuatro”.

En cada bucle principal que ejecuta el Arduino solo se ejecuta una de estas cuatro tareas principales. Al finalizar cada iteración del bucle el sistema pasa a la siguiente tarea. De esta forma se logra una rápida ejecución del código casi sin interrupciones y simulando una programación de ejecución simultánea.



**Figura 71 Diagrama de estados principal del Arduino Mega 2560.**

El sistema empieza con el proceso “CheckInput” que se encarga de la gestión de las siguientes tareas:

- El sistema llama a “CheckCambios” para comprobar si se han producido cambios en las entradas del sistema respecto a la última comprobación. En el caso de detectar cambios envía una transmisión i2c al Raspberry para actualizar la base de datos.
- El sistema entra en el proceso “CheckDos”

El proceso “CheckDos” se encarga de realizar las siguientes tareas:

- El sistema llama a la función “CheckAnalog” para realizar la medición de los 8 canales analógicos. Tras la medición el sistema envía por i2c al Raspberry Pi una trama con el estado actual de los 8 canales en formato hexadecimal con el fin de comprimir la información en la forma más compacta posible.
- Finalmente el sistema cambia al proceso “CheckTres”

En el proceso “CheckTres” se encarga de gestionar acciones adicionales En un principio leer pulsaciones de teclado procedentes del Keypad. Actualmente no implementado. Posteriormente se llama al proceso “CheckCuatro”.

En el proceso “CheckCuatro” está preparado para gestionar otra tarea separada como puede ser la lectura de los canales de Corriente ya sea analógicos o mediante un INA3221. Finalmente devuelve al sistema al “EstadoLoop” = “CheckInput” inicial.

Estas son las acciones directamente controladas por el Arduino pero existen dos funciones adicionales no directamente controladas por el Arduino que se activan cuando se requiere información o se envía información procedente del maestro de la línea i2c, en este caso el Raspberry PI.

La activación de estas peticiones procedentes del Raspberry PI se ejecuta en el Arduino por medio de dos callback presentes en el Arduino.

La función “sendDataRequest” envía por i2c la información en el sistema que se encuentra pendiente por enviar. Está función se activa cuando el maestro solicita información para recibirla.

El otro callback existente es la función “recibirDatos” que recibe información procedente del Raspberry PI cuando el maestro entra en modo envió de datos. El Arduino recibe está información y la descodifica enviando la trama a la función “i2cGestionarCodigo“ para descodificar la información en el código de tarea y la información de dicho código. Está función de “GestionCodigo” hace posteriormente la llamada a la función objetivo.

La explicación detallada de las funciones existentes así como su funcionalidad y los argumentos se explican en el datasheet de programación. Por otro lado también se explica el protocolo utilizado para comunicarse por i2c con la lista de códigos y los parámetros requeridos.

## Raspberry Pi

El objetivo del código que se ejecuta en la Raspberry Pi es gestionar la información procedente del Arduino y de las entradas del usuario y hacer un uso apropiado de dicha información. Al mismo tiempo permite una visualización del estado actual para un usuario.

Para lograr estos objetivos hay dos programaciones separadas. La primera de ellas es un código escrito en C que se ejecuta en la Raspberry Pi de forma no automática que gestiona las tramas de información que llegan desde el Arduino la descodifica y la envía a la base de datos para su almacenamiento. De igual forma recoge los cambios producidos en la base de datos y los codifica para mandar un nuevo mensaje por i2c al Arduino con la información actual de los Relés y de las salidas digitales.

El segundo código está formado por una base de datos MySQL que almacena la información de todas las variables de configuración y estados del sistema. Esta base de datos se inicia automáticamente con el encendido del sistema. Como forma de mostrar la información al usuario se utiliza una página web alojada en un servidor Apache dentro de la Raspberry Pi. Para poder cargar la información de la base de datos se ha utilizado PHP 5 de esta forma se ha logrado una página web que permita la lectura y escritura sobre la base de datos. Esta página web está alojada en el servidor local de la Raspberry Pi por lo cual se tiene acceso a dicha página web desde la red a la que esté conectada la Raspberry PI.

## PROGRAMA GESTIÓN DEL SISTEMA

El programa principal del sistema que coordina toda la información disponible es un código escrito y compilado sobre la misma Raspberry PI, no dispone de ninguna interfaz gráfica y su funcionamiento, lectura y ejecución es completamente desde una terminal Linux.

Para escribir este código se ha utilizado varias herramientas para facilitar la programación. Se ha utilizado un ordenador conectado a la misma red que el Raspberry PI, de esta forma se ha usado una línea de comandos conectada por ssh al Raspberry Pi para poder trabajar de forma nativa en la línea de comandos de Linux. De esta forma se ha podido compilar, ejecutar el código y ver los resultados de Debug. El uso de un notepad++<sup>26</sup> conectado por ssh al Raspberry Pi permite programar de una forma muy cómoda y completa siendo automática la actualización de los ficheros en la Raspberry. Por último se ha usado el programa WinCSP para realizar la transferencia entre ficheros.

Por otro lado está la compilación del código. La compilación se ha realizado de forma nativa en la misma Raspberry PI usando la herramienta de compilación “gcc”. Este compilador es el más utilizado para este tipo de entornos por su bajo consumo de recursos y simplicidad. El código del

proyecto se ha alojado en una carpeta contenida en la memoria del Raspberry Pi “~/Desktop/proyectos/PFC/programa”

### Librerías

Para acelerar el desarrollo en este entorno se ha realizado una búsqueda previa de librerías preexistentes para Raspberry PI. Concretamente el objetivo era encontrar una librería compatible con “C / C++” que permitiera conectarse a los pines digitales existentes en la Raspberry Pi, de esta forma podemos usar la comunicación i2c del Raspberry Pi para comunicarnos con Arduino. Igualmente hemos usado otra librería disponible en internet, que permite la comunicación entre el programa y la base de datos en MySQL

### RaspberryPi-gpio

La librería utilizada para permitir la conexión entre el programa en C y el periférico de puertos digitales se llama RaspberryPi-GPIO.

Para el funcionamiento de esta librería se requiere la presencia de los siguientes ficheros.

**Tabla 4 Ficheros librería RaspberryPi-gpio**

Bcm2835_gpio.h	Contiene la dirección de los puertos del micro principal del Raspberry PI un BCM2835 Broadcom. Aquí se encuentran las direcciones de los registros para poder leerlos y escribirlos según las necesidades.
rpiGpio.h	Este fichero aumenta en un nivel la abstracción sobre el anterior fichero haciéndolo más legible al usuario. Al mismo tiempo presenta las funciones disponibles en el sistema.
Gpio.c	Contiene un fichero de configuración y cambios en el programa de control de los GPIO
Gpio.h	Contiene información sobre el control de los GPIO así como su configuración y adaptación a diferentes versiones.
I2c.h	Este código fuente es el que gestiona el funcionamiento de las funciones i2c. Aquí se encuentra únicamente la definición de prototipos de funciones y variables.
I2c.c	Este código fuente es el que gestiona el funcionamiento de las funciones i2c. Aquí se encuentra las funciones que hacen uso de la comunicación i2c.

Uno de los mayores problemas encontrados en la programación del sistema de comunicación i2c con el Arduino se deben a un fallo de diseño del procesador del Raspberry Pi, Esta versión de



Raspberry PI hace uso de BCM2835 que integra en un único procesador la CPU, GPU, la memoria RAM y el control de puertos USB, Ethernet y una serie de pines digitales y de comunicación.

Para acelerar las comunicaciones i2c los procesadores y microcontroladores integran un sistema conocido como “clock stretching” que reduce los tiempos de comunicación por i2c bajo determinadas circunstancias. De esta forma se agiliza ciertas respuestas. Para su correcto funcionamiento requiere que ambos sistemas implementen esta funcionalidad.

El problema se debe a que el maestro de la comunicación i2c gestiona la generación de los pulsos de reloj. El clock Stretching <sup>27</sup>se produce cuando el esclavo de la comunicación no puede funcionar a la velocidad que le aporta el maestro y se reduce la velocidad de la comunicación a través de este mecanismo. El fallo de diseño reside en que no es posible la desactivación y configuración del clock stretching que genera el “Raspberry PI”. Este fallo no se produciría si el maestro de la comunicación i2c fuera el Arduino Mega.

Para solucionar este fallo se ha tenido que ajustar los tiempos de ejecución de las comunicaciones i2c para sincronizar manualmente las comunicaciones i2c con los tiempos que se generan por el clock stretching. Eso implica insertar delays en la comunicación i2c para ajustar las comunicaciones. La solución alternativa implicaba convertir el Arduino en maestro de la línea de comunicación o usar otro protocolo de comunicación.

## MySQL

Para realizar la comunicación con la base de datos MySQL hemos usado una librería disponible en la red para realizar esta comunicación desde nuestro programa en C. La librería correspondiente es MySQL Connector/C y está ampliamente disponible desde la página de desarrollo de MySQL. El modo de uso de esta librería es instalando la librería en el sistema y realizando posteriormente la llamada desde el código fuente, aunque también es posible compilarlo con el resto del código como fichero incluido.

Este programa dispone de una serie de funciones que permiten realizar la conexión con la base de datos en MySQL de forma que permite todas las funcionalidades que permite MySQL. El modo de funcionamiento principal consiste en enviar comandos completos entendibles por la base de datos para poder realizar cualquier funcionalidad para la cual está normalmente preparada. MySQL tiene sus propios comandos a través de los cuales se puede crear nuevas bases de datos, crear tablas, actualizar y registrar información así como otras gestiones.

Otro de los puntos de la gestión de la base de datos MySQL ha sido la utilización de PHPMyAdmin para facilitar el control y la gestión por parte del usuario de la base de datos. Permite acceder desde el servidor apache a dicha información en la misma red.

La interacción con la base de datos se produce cuando se requiere actualizar la información en la misma base de datos o cuando se comprueba la existencia de cambios producidos en la base de datos.

## **FLUJO DEL PROGRAMA.**

Para realizar la programación de este elemento hemos seguido una serie de códigos de ejemplo ya existentes sobre los cuales se han construido el programa principal. Este sistema ha sido programado sin seguir un diagrama de estados y en su lugar se usó un sistema de programación lineal que sigue de forma secuencial las diferentes tareas.

Una mejora de esta programación será pasar a un sistema de diagramas de estados que optimiza la evolución del código y facilita la programación y mantenimiento del sistema. El sistema tiene dos funciones especiales que se ejecutan en el orden correspondiente. Estas funciones sirven para enviar una orden de escritura i2c al Arduino, el cual recibe el callback correspondiente y se pone en modo lectura. La otra función permite poner en modo lectura al Raspberry PI como master de la comunicación i2c, de esta forma puede recibir información del Arduino u otro elemento conectado a las líneas de comunicación i2c. Los dispositivos comunicados por línea i2c son los siguientes: reloj RTC, la pantalla LCD y el sensor INA incluido en la placa extra de sensado de corriente.

A grandes rasgos el programa realiza una serie de procesos de lectura y comprobación del sistema así como una inicialización de las comunicaciones y precarga de la Base de Datos MySQL a través de la librería correspondiente.

Posteriormente el sistema entra en un bucle while que reitera la programación de comprobación de la base de datos y lectura de información procedente de las comunicaciones i2c. Para el correcto funcionamiento de este bloque y para solucionar el problema de clock stretching del procesador ha sido necesario añadir en este punto un delay al programa de 500 milisegundos. La secuencia de esta etapa es la siguiente:

1. Delay de 500ms para evitar los efectos del clock stretching (CS).
2. Recepción de información por i2c pendiente de enviar por parte del Arduino.
3. Delay de 500ms para evitar los efectos del “CS”.
4. Comprobación de cambios en la base de datos y envío de dichos cambios por i2c al Arduino activando el callback correspondiente en el Arduino.
5. Delay de 500ms para evitar los efectos del “CS”.

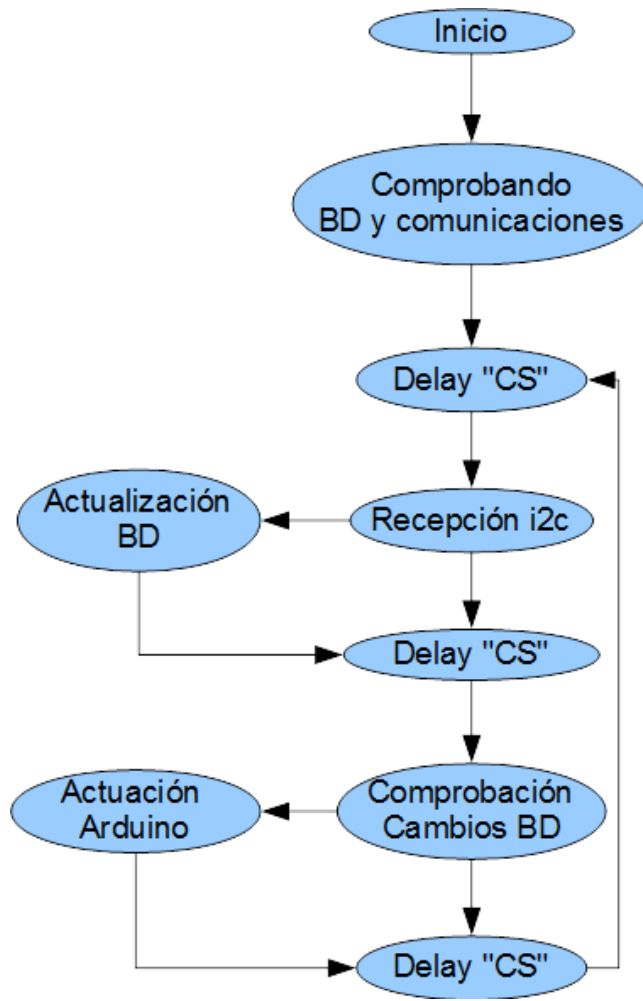


Figura 72 Diagrama de flujo del Raspberry Pi

# COMUNICACIÓN ENTRE DISPOSITIVOS

## Interfaces de comunicación Circuitos integrados

Todos los bloques funcionales de este diseño requieren una serie de interfaces y protocolos de comunicación para poder funcionar adecuadamente. Cada elemento usará una interfaz determinada de comunicación con los bloques direccionales. Esta comunicación puede ser únicamente unidireccional o bidireccional dependiendo de la naturaleza del bloque. En la imagen inferior se puede observar las diferentes interfaces de comunicación entre los diferentes bloques.

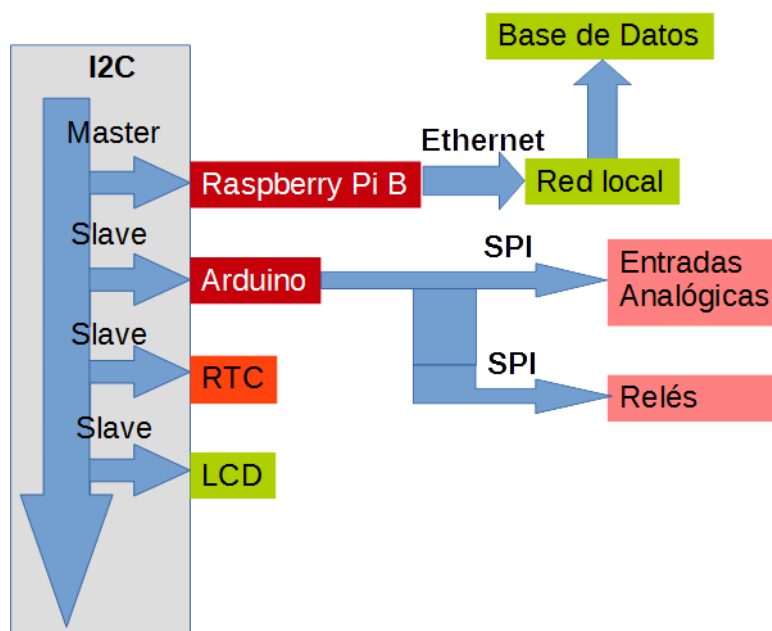


Figura 73 Interfaces de comunicación entre los diferentes bloques.

Estas comunicaciones se pueden separar en interfaces internas que permiten la comunicación entre diferentes componentes electrónicos e interfaces externas que permiten la comunicación con bloques o elementos externos al PLC. Entre las interfaces internas tenemos la comunicación serie, las interfaces I2C y la interfaz SPI.

Por otro lado las interfaces que permiten la comunicación con elementos externos al PLC es el Ethernet. Esta interfaz permite la conexión al resto de la red local, incluso su conexión a internet. Esto implica un gran potencial para obtener y distribuir la información contenida.

Al margen de las interfaces el otro elemento fundamental para las comunicaciones es un protocolo de comunicación. Los protocolos son el formato de envío de información a través de las interfaces de comunicación.

## INTERFAZ LINEA SERIE

La primera de las interfaces es la más simple de todas en cuanto a concepto y funcionalidades. En el formato más simple de esta interfaz consiste en el uso de dos cables para la transmisión de información. Estos cables se conocen como “RX” y “TX”, adicionalmente requiere compartir la misma señal de referencia.

Adicionalmente requiere que los elementos trabajen sobre el mismo nivel lógico. Esto en resultados finales requiere que trabajen bajo los mismos voltajes. Para adecuar el funcionamiento entre elementos que trabajen bajo diferentes voltajes o lógicas requiere el uso de convertidores de nivel.

Un convertidor de nivel es un circuito formado por una serie de transistores que convierten las señales a diferentes niveles de tensión para poder ser interpretada la información. La Figura inferior es un ejemplo de convertidor lógico en PCB junto con su diagrama de conexión.

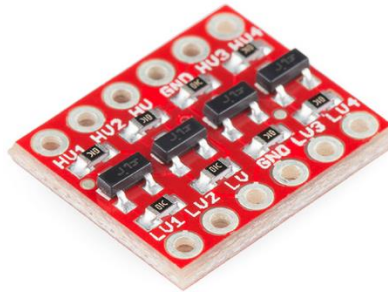


Figura 74 Este módulo convertidor lógico de Sparkfun<sup>28</sup> permite la conversión de 6 señales diferentes entre 3.3 y 5 voltios.

Poniendo el ejemplo del estado del arte: Un Arduino funciona a 5 voltios, mientras que un Raspberry funciona a 3.3 Voltios. Para la comunicación entre estos elementos requiere la incorporación de un circuito convertidor de niveles lógicos. La imagen inferior es una representación de la conexión requerida para un único canal de comunicación.

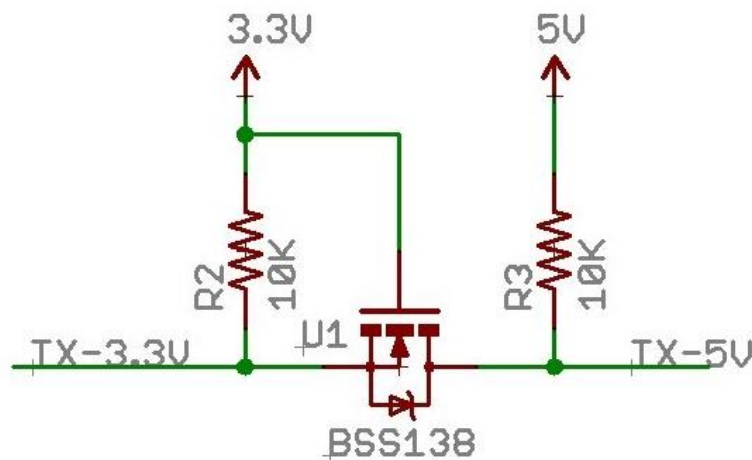


Figura 75 Diagrama de conexión para un solo canal de un convertidor lógico mediante el empleo de un Mosfet de canal N.

Adicionalmente en las comunicaciones serie existen conexiones adicionales que aportan mayor control a las comunicaciones. De esta forma se añade comprobación de errores, detección de datos pendientes y señales de reloj.

Otra de las características de las comunicaciones serie es que permite la transmisión de información a diferentes velocidades. Esta medición se realiza en baudios por segundo “bps<sup>29</sup>” que indica la cantidad de información transmitida en un segundo a dicha velocidad. Generalmente se trabaja entre 30000 y 115200 bps.

Para el funcionamiento se deben diseñar o aplicar protocolos de comunicación sobre estas líneas serie que sean lo más eficientes posibles en cuanto a codificación para minimizar las transmisiones necesarias. La codificación añade la ventaja adicional de proteger las comunicaciones frente al uso no autorizado por parte de elementos externos.

Entre los diferentes tipos de interfaces serie existentes existen las comunicaciones RS432 y RS232 aptas para ciertos tipos de aplicaciones y muy extendidas en los sistemas. No obstante han ido desapareciendo su uso gracias a la aparición de otras interfaces. En el anexo de comunicaciones industriales se puede ver más información sobre diferentes interfaces usadas actualmente.

## INTERFAZ I2C

Otras de las interfaces más comúnmente usadas consisten en el uso de un sistema de transmisión compuesto por 3 hilos. El cableado consiente en la conexión de una señal de reloj y una señal de datos, la conexión adicional implica la conexión de tierra.

La conexión de la señal de reloj se llama “SCL” y es una señal de reloj cuadrada generada por uno de los sistemas envueltos en la comunicación. El otro elemento que transfiere la señal de datos es “SDA”.

En está interfaz los elementos conectados pueden asumir dos roles diferentes en la comunicación. El elemento principal de esta comunicación es el Maestro de la comunicación. Este Maestro se encarga de la completa gestión de la comunicación, generando la señal de reloj y controlando las señales de respuesta entre cada mensaje. El otro rol corresponde al funcionamiento de un Esclavo, el cual recibe las órdenes procedentes del Maestro para iniciar la transmisión de información.

En una red I2C el número máximo de dispositivos conectados es 128 dispositivos. En esta red cada dispositivo tiene una dirección de 7 bit los cuales van comprendidos desde 0x00 hasta 0x80 en formato hexadecimal. Para formar completamente la dirección de los dispositivos se debe añadir un bit adicional formando de tal forma direcciones de 1 byte.

Este bit adicional corresponde al bit de menos importancia dentro de la dirección I2C. El significado de este bit indica si se trata de una operación de lectura por parte del maestro o de escritura. En estas comunicaciones es el maestro el que inicia todas las comunicaciones.

La comunicación I2C tiene un protocolo ya establecido de comunicación que se puede ver en la imagen inferior.

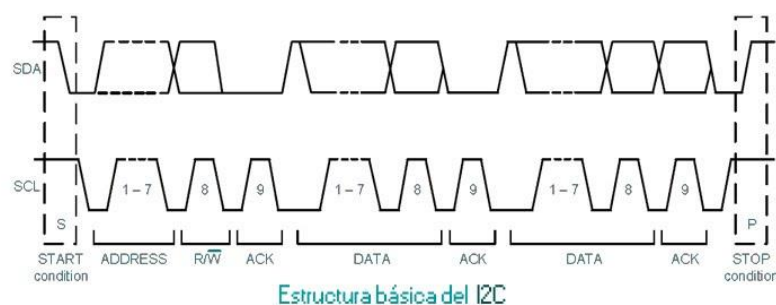


Figura 76 Secuencia de transmisión I2C.

En el protocolo hay una serie de pasos para el correcto funcionamiento. Un bloque de datos está formado generalmente por la siguiente información.

Tabla 5 Flujo de información en una comunicación I2C.

Información	Enviado por
Condición START	Master
7 Bits de dirección de esclavo	Master
1 Bit de RW, 0 es Leer y 1 Escribir	Master
1 Bit de Acknowledge	Slave
Byte de dirección de memoria	Master
1 Bit de Acknowledge	Slave
Byte de datos	Master / Slave (Escritura / Lectura)
1 Bit de Acknowledge	Slave / Master (Escritura / Lectura)
Condición STOP	Master

Inmediatamente después de un intento de Lectura o Escritura se produce el envío de un bit de Acknowledge (ACK) o NonAcknowledge (NACK). Significa la respuesta del receptor al transmisor. Este bit permite saber si el transmisor ha sido escuchado o ha fallado la comunicación. En caso de ausencia de respuesta se detecta un NACK.

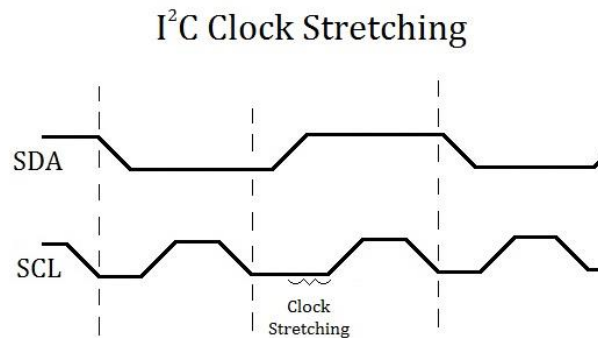
Para el funcionamiento de la interfaz I2C requiere el funcionamiento de unas resistencias “pull-up” entre las líneas SCL y SDA y alimentación. De esta forma las líneas I2C se encuentran a nivel alto mientras no exista intercambio de información.

Las líneas I2C igualmente funcionan a diferentes velocidades hasta 400 KHz. En la comunicación la velocidad será la del dispositivo con la velocidad máxima más baja de todos los dispositivos conectados a la línea.

La última de las características de las comunicaciones I2C se llama “Clock Stretching” es una funcionalidad que tiene la mayoría de los dispositivos con capacidad de comunicación I2C. El objetivo de esta funcionalidad es lograr que la velocidad final de la línea de comunicación no sea la del dispositivo con la velocidad más baja. De esta forma se permite comunicarse a la velocidad más alta posible con la mayoría de los dispositivos.

El funcionamiento se basa en el control puntual de las líneas I2C cuando se produce comunicación con dispositivos más lento que el resto de elementos. La imagen inferior muestra como es el funcionamiento del “Clock Stretching en las comunicaciones I2C”





**Figura 77 Clock Stretching en comunicaciones I2C.**

Como las comunicaciones serie se requiere el uso de convertidores de niveles lógicos cuando se comunican dispositivos que trabajan a diferentes voltajes lógicos. Como por ejemplo cuando se usan voltajes de 5 y 3.3 Voltios.

## INTERFAZ SPI

Otras de las interfaces utilizadas en nuestro proyecto son las comunicaciones mediante SPI, es una interfaz de comunicación que requiere el uso de varias conexiones eléctricas. En total son 4 cables los que se requieren como mínimo. En esta interfaz hay dos tipos de roles el funcionamiento como Maestro y el funcionamiento en modo Slave.

La primera conexión es el MISO, “Master Input Slave Output”, está conexión es la que realiza el flujo de información desde el esclavo hasta el maestro de esta comunicación. La segunda conexión necesaria es el MOSI, “Master Output Slave Input”, en esta comunicación fluye la información desde el maestro de la comunicación al esclavo.

Esta interfaz permite la conexión de ilimitados dispositivos conectados a la red. Sin embargo por cada dispositivo conectado requiere una conexión adicional llamada “Chip Select” “CS” la cual se activa siempre a nivel bajo. De esta forma se permiten ilimitados dispositivos conectados siempre y cuando se disponga de suficientes GPIO.

Las últimas dos conexiones necesarias para el funcionamiento son SCLK para transmitir la señal de Reloj generada por el Maestro de la comunicación y la misma señal de tierra.

Las ventajas de la Interfaz SPI es que permite un mayor número de dispositivos que las comunicaciones I2C y permite unas velocidades de comunicación mayores, al disponer de dos cables para la transmisión de información. Como contrapartidas frente a la interfaz I2C es que se requiere más conexiones eléctricas y requiere implementar un protocolo propio para el envío y comprobación de errores.

La imagen inferior muestra el diagrama de conexión de dispositivos conectados a una red SPI.

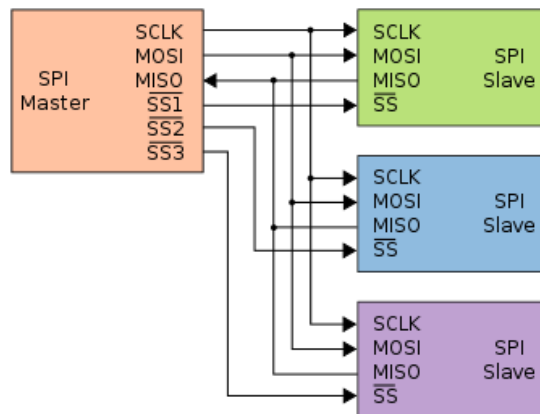


Figura 78 Diagrama de conexión de dispositivos SPI

Por las características de esta interfaz no se permite el uso para la transmisión de información a largas distancias, esta tarea está reservado para interfaces como RS232 o RS485. Adicionalmente como sucede en los casos anteriores se requiere el uso de conversores de niveles lógicos para unir dos dispositivos que trabajen bajo voltajes diferentes.

La imagen inferior es un ejemplo de diagrama de tiempos de una línea de comunicación SPI.

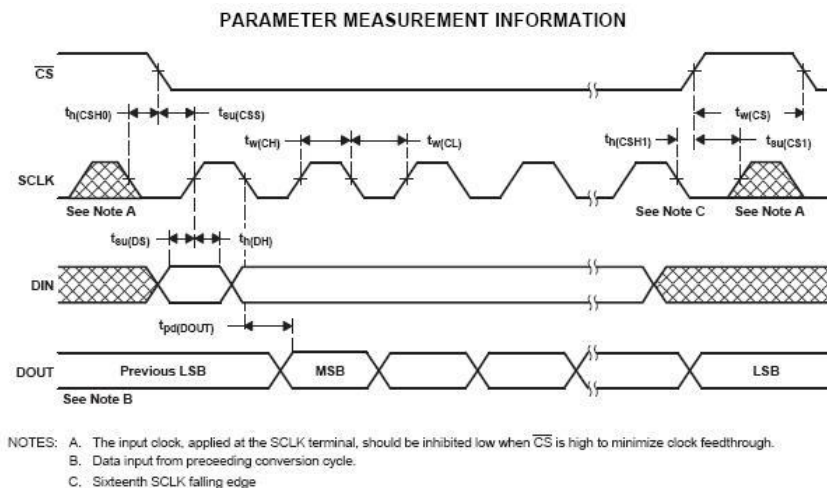


Figure 1. Timing Diagram

Figura 79 Diagrama de tiempos de una comunicación SPI.

## Otros Interfaces de comunicación

Existen otros protocolos de comunicación usados para realizar comunicaciones con dispositivos externos al PLC. Este es el caso de la interfaz Ethernet. Esta conexión permite la conexión a una red local de dispositivos siguiendo el protocolo IP. El punto fuerte de esta interfaz es la posibilidad de conexión internet.

Gracias a esta interfaz se puede acceder a la red local del dispositivo permitiendo el acceso a través de ella a bases de datos y servidores de páginas web. Para lograr este nivel de comunicación se utiliza conectores Ethernet o comunicación a través de Wifi.

En el [anexo de comunicaciones industriales](#) se encuentra información sobre diferentes interfaces y protocolos de comunicación extendidos en el ámbito industrial.

# LINEAS FUTURAS

## Difusión del proyecto

Una vez finalizado el diseño y las primeras pruebas de funcionamiento de este diseño, el siguiente objetivo del proyecto es la difusión de este trabajo en diferentes páginas web del mundo de la electrónica.

En la comunidad online se define Maker como aquella persona que sigue la filosofía DIY “Do It Yourself”. Este movimiento tiene mucha difusión en proyectos de electrónica. Entre sus ideas existe la difusión de conocimiento y el “how to”. Entre las páginas web en las que se produce está difusión de conocimientos está [www.blog.bricogeek.com](http://www.blog.bricogeek.com) junto con las páginas web de los creadores de dichos diseños.

La difusión se realizara por parte de estas páginas web externas así como la web personal del diseñador de este proyecto: [www.makertronica.com](http://www.makertronica.com). Para la correcta difusión se licenciara como open hardware el diseño de este proyecto, poniendo a disposición pública el diseño de hardware, la programación realizada en este proyecto y la memoria del mismo.

Igualmente por la misma razón la memoria de este proyecto quedara a disposición pública por parte de la universidad y todos los estudiantes.

Este proyecto queda acotado en un entorno principalmente educativo y didáctico. En la situación actual es atractivo como ejemplo de diseño electrónico viendo cómo se puede usar para el control y automatización de pequeños elementos.

Para su uso en aplicaciones industriales se realizaran una serie de mejoras en el prototipo para convertirlo en una solución atractiva y lo más segura posible para aplicaciones de producción.

## MEJORA DEL DISEÑO

En el desarrollo de un nuevo diseño es frecuente la aparición de nuevas ideas y soluciones para problemas surgidos en el diseño. No obstante por el estado actual del proyecto es difícil volver atrás sin rehacer todo el diseño. A lo largo del diseño ha sido necesario hacer modificaciones en el hardware ya fabricado para solucionar los problemas surgidos.

Por este motivo tras lograr un sistema básico funcional, se debe pasar a la siguiente versión del prototipo que cumpla las siguientes características.

Los PLC comparten unas características comunes relativas al montaje. El montaje más común que tiene este tipo de dispositivos es su instalación en un rail DIN. Al mismo tiempo en su aspecto tienen unas dimensiones y formas similares.

En la figura inferior se puede ver una caja preparada para incluir una electrónica similar a un PLC. Al mismo tiempo este encapsulado está preparado para ser montado en un carril DIN.

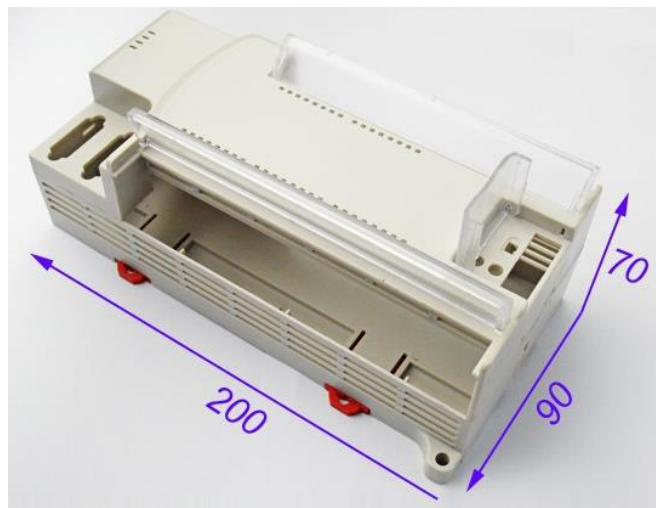


Figura 80 Próximo encapsulado del prototipo.

Como se aprecia en la foto este diseño está preparado para incluir dos niveles de conectores fijos lo cual permite incluir aproximadamente hasta 60 entradas y salidas. Al mismo tiempo este encapsulado dispone de dos salidas para conectores de línea serie como RS232 o RS485. Como conclusión el uso de un encapsulado de este tipo ahorraría tiempo de desarrollo de una carcasa y facilitaría la disposición del hardware.

Otra característica en este tipo de carcasas es que es característica la fabricación de la electrónica montada en tres niveles diferentes en el interior de la carcasa. Por ello requiere la fabricación de 3 PCB diferentes en las que cada una de ellas se encarga de una tarea determinada.

En la figura inferior se muestra un ejemplo de un diseño que usa una carcasa muy similar en las que se dan ejemplos de la electrónica diseñada.



**Figura 81** Ejemplo de PLC con carcasa estándar.

Este diseño está estructurado en 3 niveles diferentes. El nivel inferior incluye un circuito de potencia para realizar la conversión de tensión de red a 24 y 5 voltios para el correcto funcionamiento de la electrónica.



**Figura 82** Ejemplo diseño nivel 1. Electrónica de potencia.

El siguiente nivel de este diseño incorpora los conectores y las salidas y entradas correspondientes. Igualmente incluye los elementos para aislar el sistema mediante el uso de optoacopladores. Igualmente se encuentran incluidos los conectores de línea serie.



**Figura 83** Ejemplo de diseño nivel 2. Conectores y aislamiento.

Finalmente el último nivel de este diseño incluye a los microcontroladores que se encargan de gestionar el sistema. Este último nivel incluye indicadores LED para indicar el estado de las

entradas y salidas de una forma sencilla para el usuario final. Como elementos extras añade conectores de programación, RTC y circuitos para generar salidas de video gracias a un driver VGA.

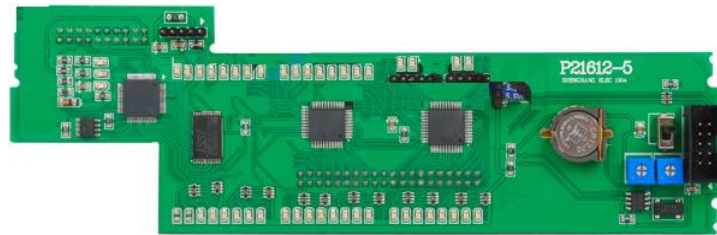


Figura 84 Ejemplo de diseño nivel 3. Microcontroladores y elementos extras.

En estos diseños se incluyen conectores para lograr la conexión entre diferentes niveles de la PCB. Por otro lado para facilitar la conexión entre dispositivos y facilitar la ampliación de dispositivos se añadirá un conector extra que facilita la comunicación.

## MEJORA DEL HARDWARE

Por otro lado en las mejoras que deben producirse en el hardware del dispositivo implican el rediseño completo. Estos cambios se deban tanto a fallos de diseño anteriores como la necesidad de incluir nuevas funcionalidades.

Por ese motivo están planteadas las siguientes mejoras de Hardware:

1. Sustitución del Raspberry Pi por otro hardware con similares capacidades técnicas, pero de un formato mucho más reducido. Esto facilita lograr un diseño más compacto.



Figura 85 Modulo Linux Aria G25

2. Sustitución del Arduino Mega 2560 por un microprocesador más pequeño y más potente. Se usará un ARM de 32 bits M0+ de la marca ATMEL. Con esta media minimizamos el tamaño del PLC al usar una solución integrada que nos permite ajustarnos al diseño de la caja. Al mismo tiempo este Microcontrolador es superior técnicamente al que lleva incorporado el Arduino Mega gracias a entre otras características el uso de 48 MHz de frecuencia de reloj.

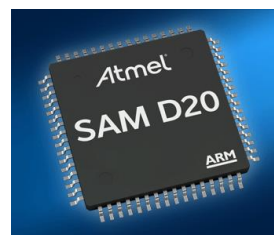


Figura 86 ARM 32 bits Sam D20. M0+ Atmel.

3. La siguiente característica nueva en el diseño será el uso de unos conectores modulares que minimice el espacio que ocupan actualmente los conectores. Al mismo tiempo dispondrán de indicadores luminosos para conocer el estado.
4. La siguiente característica técnica que tendrá el nuevo modelo respecto al anterior es la inclusión de tecnología inalámbrica de comunicación 868MHz mediante el uso de tecnología ZigBee. Esta tecnología es perfectamente para su



uso en entornos industriales donde se ven perjudicadas el resto de tecnologías inalámbricas.

5. Otro elemento adicional que se va añadir por su utilidad en equipos “Stand Alone” es la inclusión de un modem GPRS de bajo coste para permitir su conexión a internet. Igualmente requerirá la instalación de una tarjeta SIM.
6. Para facilitar la ampliación de los equipos se incluirá un conector para añadir equipos compatibles en un bus de comunicaciones de libre uso.
7. En el diseño y elección de componente se elegirán componentes aptos para el uso en entornos industriales. Entre las principales características de este tipo de electrónica es su capacidad para trabajar en condiciones ambientales más agresivas.
8. Se sustituirá el diseño actual que emplea tecnologías de agujero pasante para ser sustituido íntegramente el diseño con tecnología smd o equivalente. Esto facilitara el diseño ya que será posible incluir componentes a doble cara.



Figura 87 Modem GPRS SIM 900 de bajo coste.

9. Para reducir el tamaño se diseñara una fuente de alimentación para realizar la conversión de la tensión de red a 24 y 12 voltios. Se utilizara el estándar a 24 voltios para alimentar dispositivos industriales externos.
10. Se añadirán conexiones línea serie de RS232 y RS485 para la comunicación con otros equipos de las instalaciones industriales.

## MEJORA DEL SOFTWARE

Para la mejora de este prototipo también se plantean actualizaciones del software que se ejecuta en el sistema. Como primer punto tras la decisión de actualización del Microcontrolador que funciona en el sistema, este Microcontrolador ARM de 32 bits M0+ es perfectamente compatible con la programación Arduino. Esta compatibilidad está probada por uno de los últimos diseños de Arduino, el Arduino Zero usa este mismo Microcontrolador y es perfectamente programable desde la última IDE de Arduino.

La programación del Arduino se reestructurara para formar librerías externas que faciliten la configuración, comunicación y desarrollo del mismo. Esto agilizará la programación y aún más importante facilitará la configuración por parte de los usuarios al lograr librerías de alto nivel.

En el otro lado el microprocesador Aria G25 perderá la capacidad de interfaz gráfica sustituyendo íntegramente el control del dispositivo por una interfaz web que se encargue de la interacción con el usuario final. Al mismo tiempo seguirá albergando una base de datos que sincronice y monitorice la información disponible en el sistema.

Este módulo ARM Aria G25 ejecutará una distribución Linux ligera donde primará la estabilidad y seguridad del sistema. Para ello se usará un sistema operativo en modo lectura. Una tarjeta de almacenamiento SD será la encargada de almacenar la información y configuración del sistema operativo.

Sucediendo de igual forma que con la programación de Arduino se busque externalizar la programación creando librerías de alto nivel que simplifique la labor de puesta a punto por parte de los usuarios.

Tal vez el punto más difícil de la nueva programación sea lograr una interfaz gráfica que facilite la programación del PLC por parte del usuario final. Esto implica lograr una interfaz que permita configurar el funcionamiento del sistema según una serie de variables de entrada y estados. Se trataría de una programación muy similar a la que puede tener LabView.

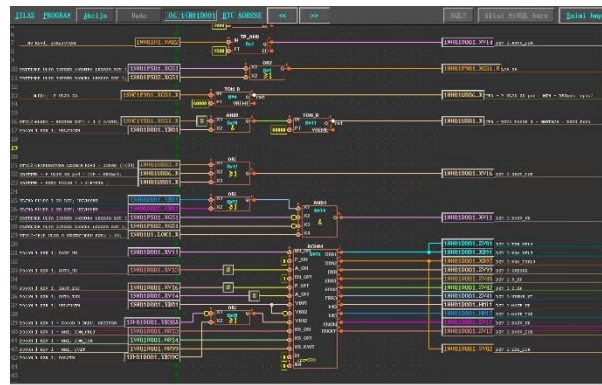


Figura 88 Ejemplo programación interactiva de un PLC.

Un ejemplo de esta forma de programación es una serie de librerías compatibles con LabView de esta forma permite la programación y comunicación con Arduino desde la interfaz gráfica de LabView.

Otro ejemplo de entorno de programación libre orientado a PLC es IDE4PLC diseñado por la Universidad Nacional de Quilmes. Es una IDE que usa la norma IEC 61131-3 el cual estandariza el modelo y los lenguajes de programación de los PLC.

Está interfaz gráfica permite la programación en lenguaje Ladder para microcontroladores NXP pero es fácilmente portable a otro tipo de arquitecturas como puede ser la de ATMEL.

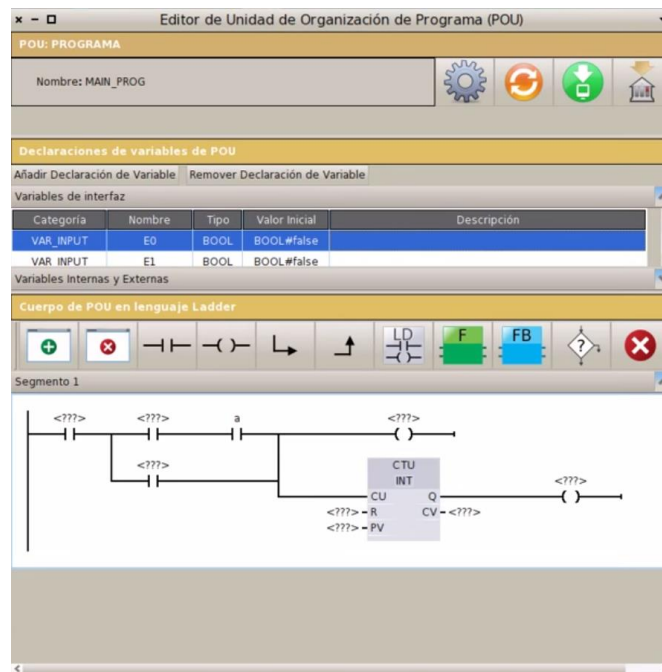


Figura 89 IDE4PLC. Interfaz libre para programación de PLC usando la norma IEC 61131-3 usando programación Ladder.

## CONCLUSIONES

Finalmente tras haber pasado por todo el proceso de diseño de un primer prototipo, solo queda finalmente hacer un balance del proyecto. De esta forma vamos a analizar la situación al inicio y al final de este diseño.

### Situación al inicio del proyecto

En la situación original de este proyecto, partíamos de conocimientos de nivel medio sobre la electrónica empleada. Un ejemplo de ello son los conocimientos referentes a las amplias posibilidades de Raspberry y Arduino. Estos dispositivos están diseñados para facilitar el aprendizaje, programación y diseño electrónico gracias a la gran cantidad de información existente.

Las principales dificultades técnicas al inicio de este proyecto ha sido el aprendizaje de diseño de PCB usando la herramienta DesignSpark. Este diseño de PCB cubre el aspecto de diseño de esquemáticos y diseño del Layout. Dentro de la etapa de diseño inicial se ha notado la falta de experiencia en el diseño de circuitos. Una mayor experiencia inicial habría eliminado la gran mayoría de fallos iniciales, tanto desde un punto de vista estético como funcional de la electrónica.

En el diseño inicial se decidió seguir las normas de diseño proporcionadas por el personal técnico de la universidad. Originalmente está prevista la fabricación de la PCB en la misma universidad mediante un Router CNC. Estos dispositivos tienen unas restricciones de fabricación más exigentes que el fabricante de PCB Eurocircuits. No obstante dada la complejidad final de la PCB, y el número de conexiones involucradas, se decidió su fabricación en Europa mediante Eurocircuits.

Por esa razón se decidió hacer originalmente un diseño que seguía los siguientes criterios: Montaje con componentes pasantes, libertad de dimensiones de diseño, uso de módulos en vez del diseño integrado. Estas libertades de diseño implicaron penalización a la hora de su fabricación externa.

Inicialmente también se ha partido de un nivel de programación medio. Inicialmente no ha supuesto barreras de entrada a la hora de adquirir conocimientos técnicos de programación. Pero una vez más la falta de experiencia ha provocado una ralentización por problemas técnicos de programación que implicaban un mejor estudio de los problemas. El elemento que más ha facilitado el desarrollo de programación ha sido el uso de diagramas de estados, elemento fundamental para realizar una buena programación.

Como opinión personal considero escasa la formación en electrónica impartida en la especialidad de electrónica de Ingeniería Industrial. El ejemplo más claro relativo al plan antiguo de

Ingeniería Industrial, es la no existencia de formación en el diseño de PCB o la falta de experiencia en diseños reales y casos prácticos.

## Situación al final del proyecto

Como balance final de este proyecto considero que he adquirido un conocimiento muy notable en el diseño electrónico y programación a lo largo de estos meses. Este conocimiento adquirido se debe en una gran parte a mi trayectoria profesional en la empresa [Embeblue](#) pionera en el diseño y fabricación ágil de electrónica.

En contraste a las primeras etapas del proyecto considero actualmente que habría realizado un diseño muy distinto debido a la diferencia de conocimientos inicial y final.

Como primera medida sería adquirir un encapsulado en formato carril DIN para montar todo el diseño electrónico en su interior, como se ha explicado ya en las líneas futuras.

Como segunda regla de diseño habría sido el uso de una tecnología de montaje superficial SMT<sup>30</sup> (Como se ven en la figura inferior.) para todos los componentes funcionales de este diseño. Junto con una reducción del área de la PCB y la adaptación de los conectores al encapsulado.

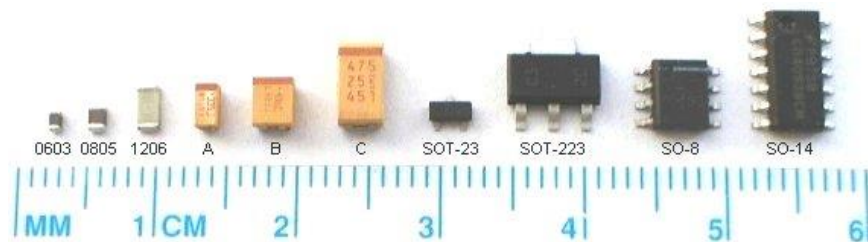


Figura 90 Componentes de Montaje Superficial SMT / SMD

Respecto al análisis referente a los objetivos iniciales del proyecto, podemos concluir que unos objetivos han sido alcanzados y otros no han sido posible lograrlos.

El primer objetivo era lograr un sistema fácilmente ampliable conectado en red. El uso de una interfaz de conexión Ethernet permite una fácil conexión y ampliación por medio de la red local. Igualmente es fácil lograr el control del sistema desde la misma red local.

El segundo objetivo era lograr un sistema modular basado en la conexión de nuevas Shield que permitiera la adaptabilidad y expansión del sistema por medio de estos módulos.

No ha sido posible lograr esta capacidad de adaptación al completo. Esto es debido a que el uso de módulos Shield, requiere el uso de conexiones digitales diferentes según la Shield conectada. Es por ello que nuestro diseño no está preparado para adaptarse a cualquier Shield. Nuestro diseño ya tiene ciertas conexiones asignadas para ciertas tareas del sistema, siendo imposible utilizarlos para otras finalidades.

Considero que se ha logrado un sistema con un coste no demasiado exagerado para un primer prototipo teniendo en cuenta con las siguientes versiones el coste se reducirá considerablemente, sobre todo teniendo en cuenta el número de conexiones y posibilidades que presenta este sistema. Lo que si que no se ha buscado finalmente es lograr un sistema compacto, objetivo reservado para el siguiente diseño.

Otro de los objetivos del proyecto era lograr un sistema abierto, que permita completa libertad de modificación y adaptación a cualquier proceso. Con este objetivo, por el amplio abanico de conexiones y la facilidad de programación del sistema, considero que se ha logrado un sistema de fácil configuración. Como limitación es que de momento requiere personal con conocimientos de programación. La implementación de una interfaz gráfica solventaría este problema.

El último de los objetivos es referente a la difusión del conocimiento y el diseño. Con el fin de facilitar el aprendizaje relativo al diseño electrónico y los PLC. En el nuevo diseño futuro se hará una completa difusión del diseño tras una mejora del diseño. La difusión de este conocimiento se realizará a través de la página web del autor de este proyecto. [www.makertronica.com](http://www.makertronica.com)

Finalmente me considero muy satisfecho tanto por mi proyecto, como por la evolución futura que tendrá por mi propia cuenta. Igualmente estoy muy satisfecho por mi evolución de conocimientos y puesta en práctica de ellos en el desarrollo de este proyecto



# ANEXOS

## INDICE de ANEXOS

Anexo A: [Esquemáticos](#)

Anexo B: [Layout](#)

Anexo C: [Datasheet Hardware](#)

Anexo D: [Datasheet Software](#)

Anexo E: [Presupuesto](#)

Anexo F: [Listado BOOM de componentes](#)

Anexo G: [Cálculos técnicos](#)

Anexo H: [Análisis PLC Industriales](#)

Anexo I: [Análisis Comunicaciones Industriales](#)

Anexo J: [Documentación DVD](#)





## Anexo A: Esquemáticos

### ÍNDICE DE ESQUEMÁTICOS

Los esquemáticos aquí presentes han sido creados usando el software de DesignSpark 6.1 de RS. Para la creación de estos esquemáticos han sido elaboradas partes especiales para representar los modelos usados.

- [Esquemático PCB](#)
  - [Esquemático Arduino Raspberry Pi.](#)
  - [Esquemático Lazo Corriente y convertor Analógico Digital.](#)
  - [Esquemático entradas, salidas digitales y relés.](#)
  - [Esquemático elementos extras.](#)
- [Esquemático Test Corriente](#)
  - [Esquemático generador lazo corriente 4-20 mA](#)
  - [Esquemático sensor corriente INA3221](#)
    - [Esquemático sensor corriente INA](#)
    - [Esquemático sensor corriente INA conectores](#)

#### ***Esquemático PCB Hardware libre***

Aquí se incluye los esquemáticos completos usados como primera versión del proyecto. Han sido diseñados íntegramente con DesignSpark 6.1 Las hojas de los esquemáticos se encuentran agrupadas según bloques funcionales.

- a. Arduino y Raspberry
- b. Entradas con componente de sensado analógico.
- c. Entradas y salidas con lógica digital.
- d. Pantalla LCD, Keypad y Buzzer

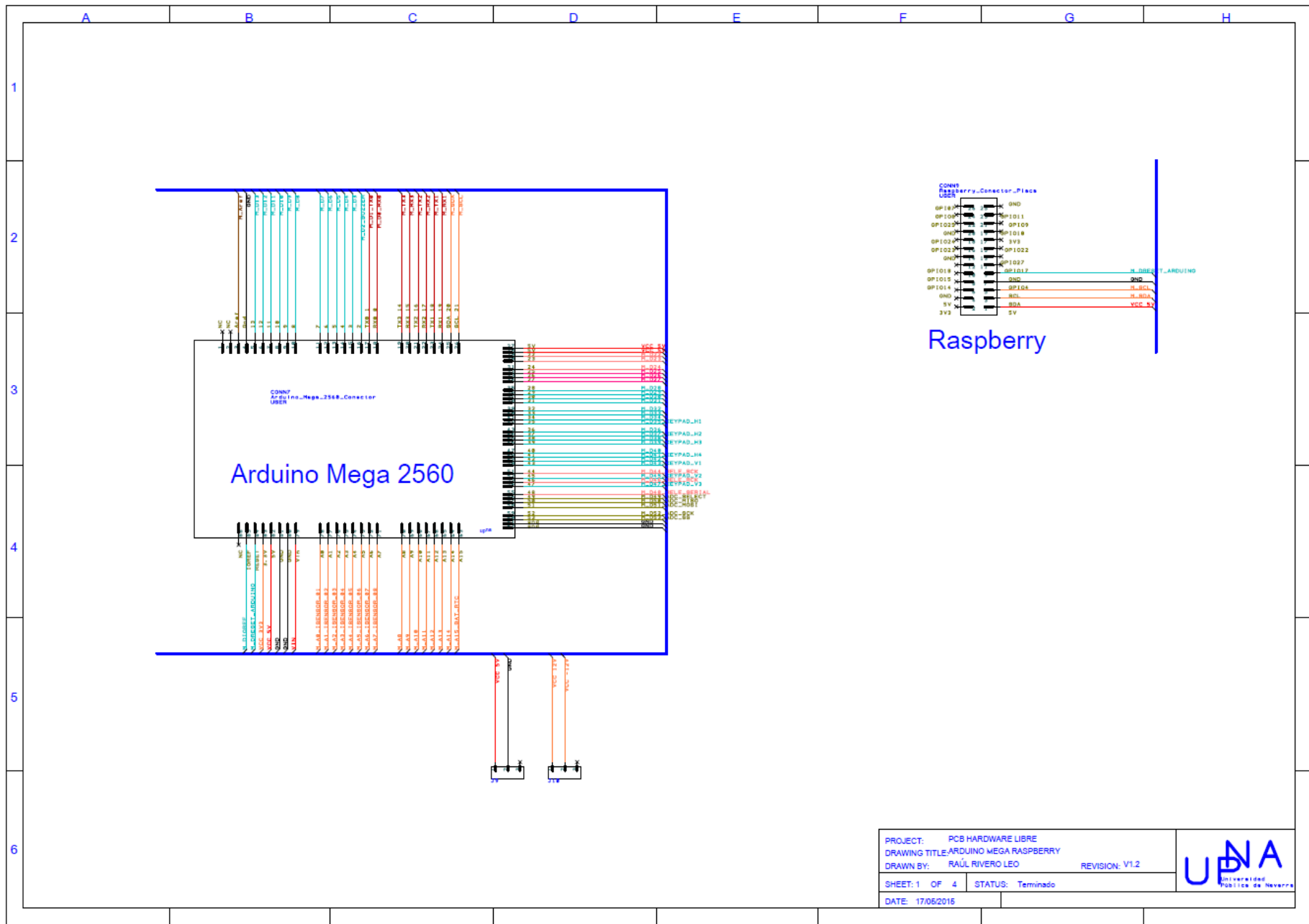
#### ***Esquemático Generador Corriente Test***

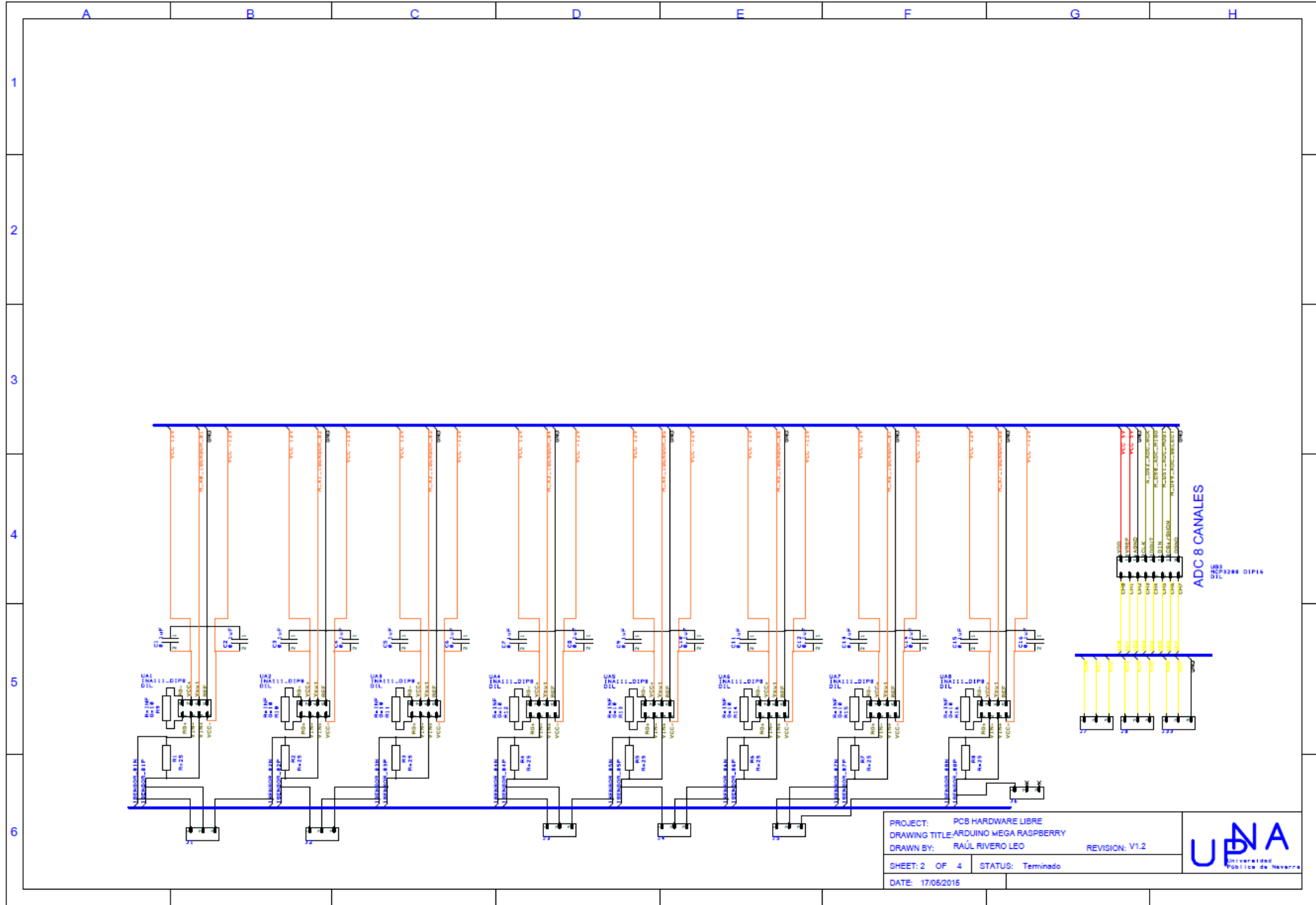
En este esquemático se representa el diseño un sistema para generar corrientes de test. De esta forma se pueden simular sensores externos que funcionen bajo lazos de corriente entre 4 y 20 mA

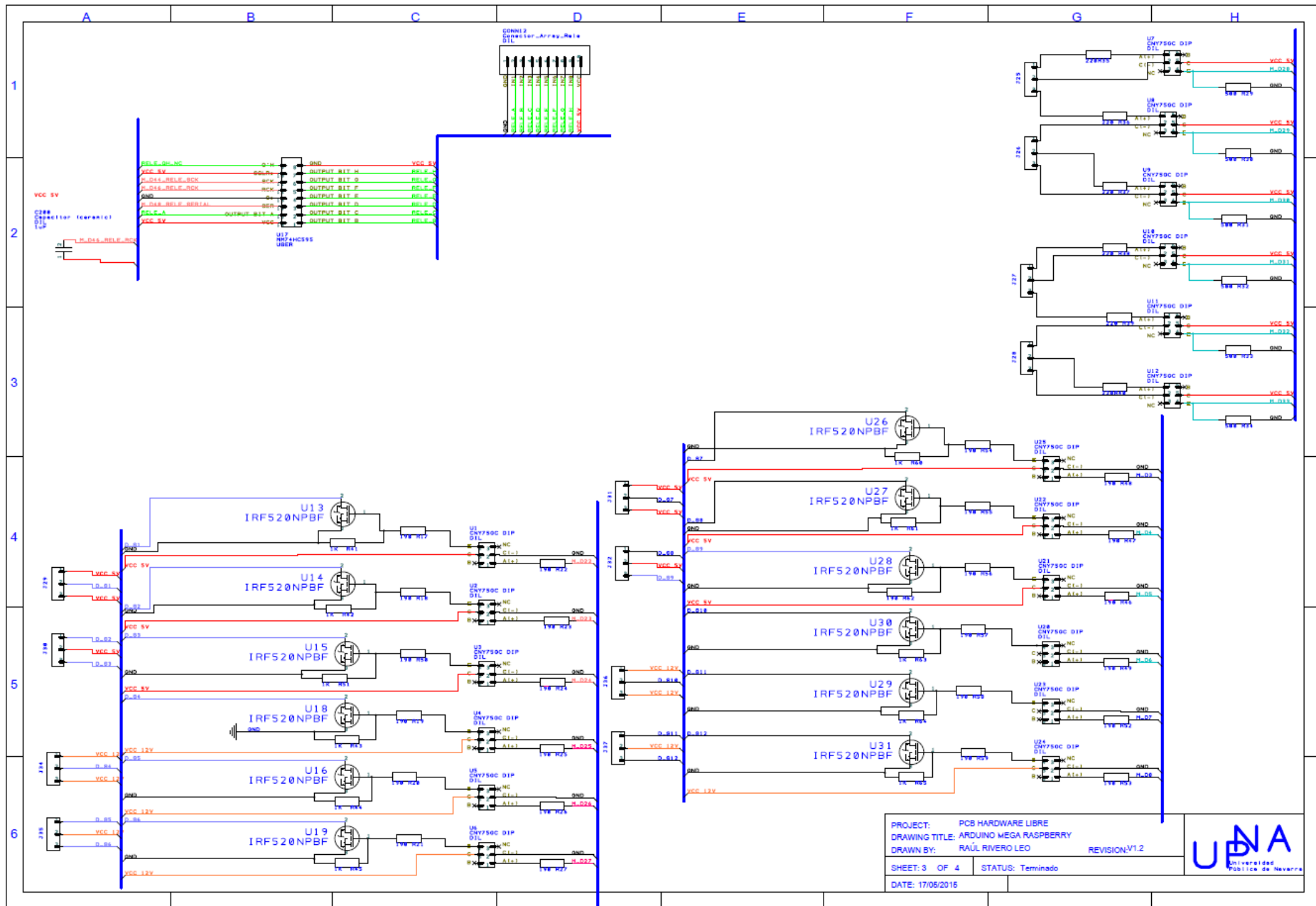
#### ***Esquemático Current Tester INA3221***

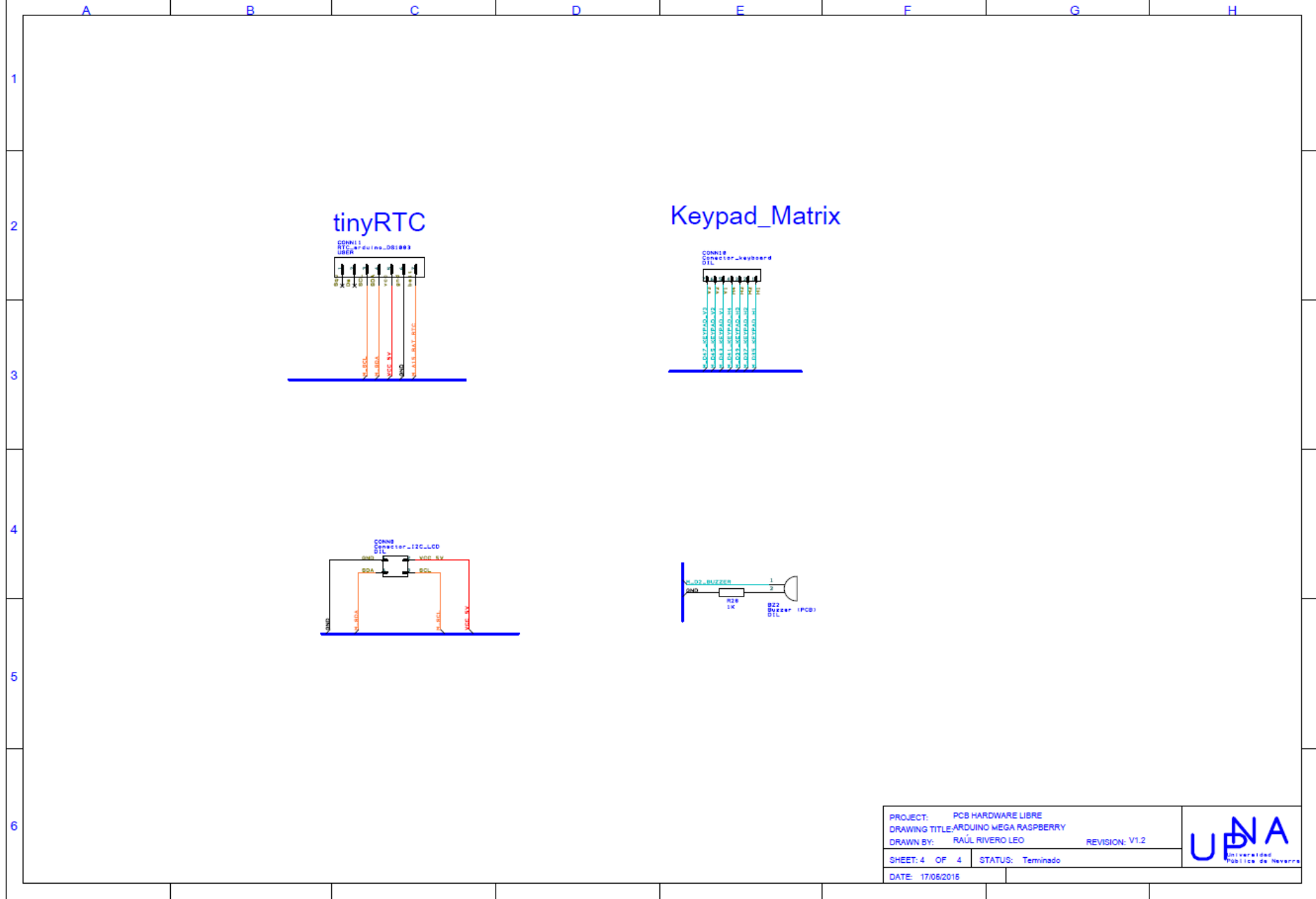
Este esquemático aporta una solución para implementar un sistema de sensado de corriente, tensión y potencia más eficaz que el uso de resistencias analógicas implementada finalmente en la PCB. De esta forma se aporta un componente de gran precisión.

- a. Diagrama del componente INA3221
- b. Diagrama del componente INA3221 Conectores y elementos adicionales.



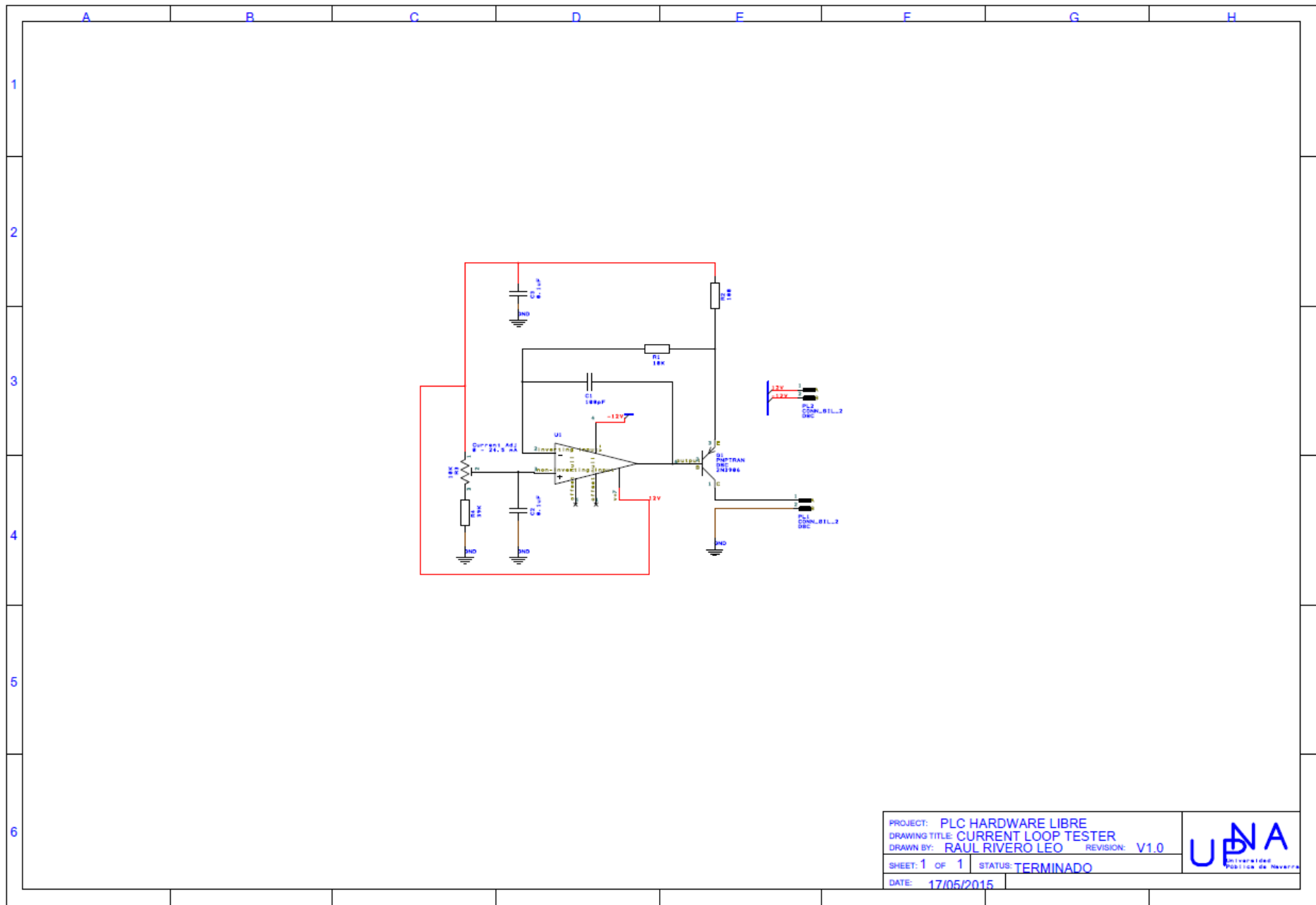






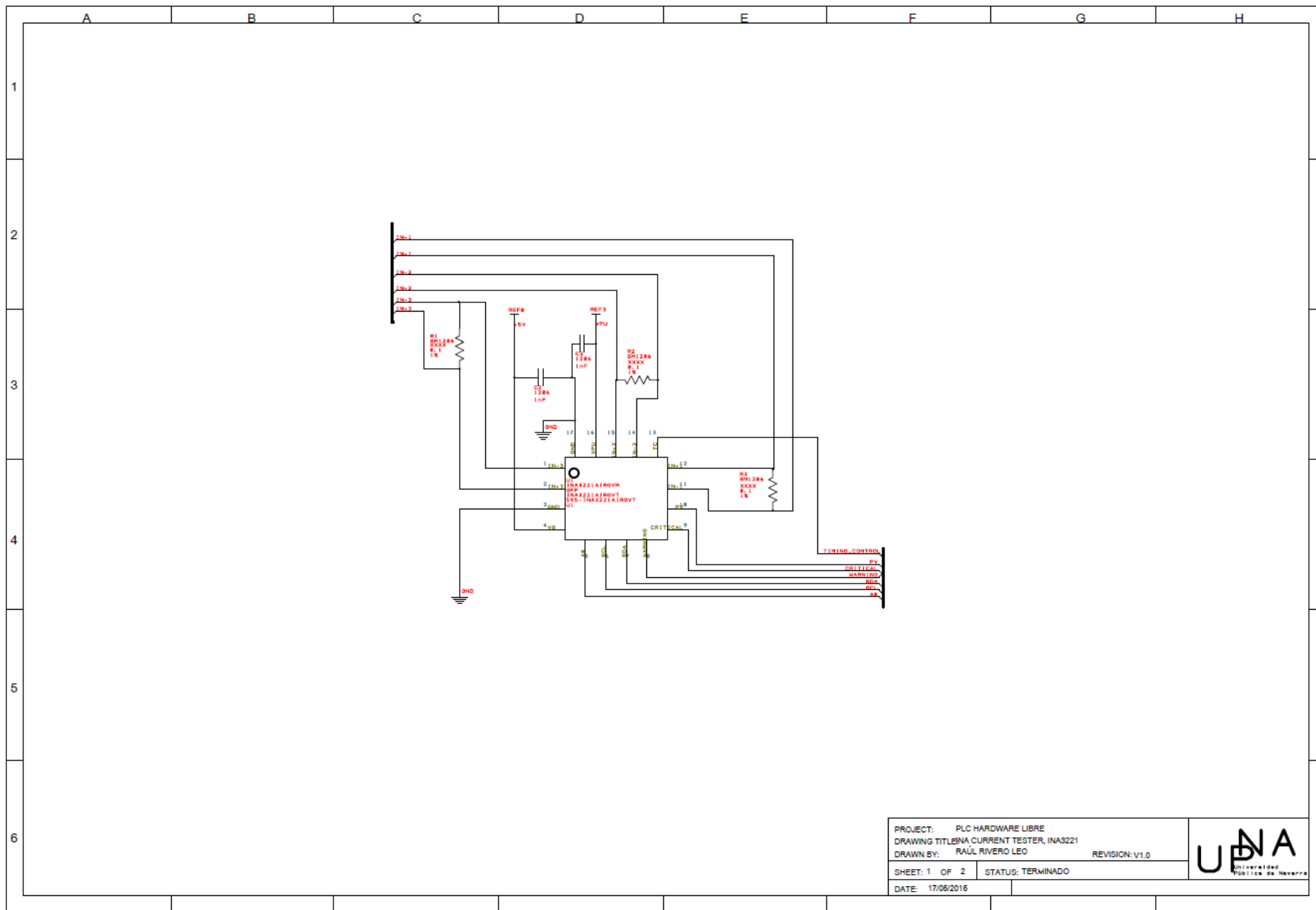
PROJECT:	PCB HARDWARE LIBRE	
DRAWING TITLE:	ARDUINO MEGA RASPBERRY	
DRAWN BY:	RAÚL RIVERO LEO	REVISION: V1.2
SHEET:	4 OF 4	STATUS: Terminado
DATE:	17/05/2016	





PROJECT: PLC HARDWARE LIBRE  
 DRAWING TITLE: CURRENT LOOP TESTER  
 DRAWN BY: RAUL RIVERO LEO REVISION: V1.0  
 SHEET: 1 OF 1 STATUS: TERMINADO  
 DATE: 17/05/2015

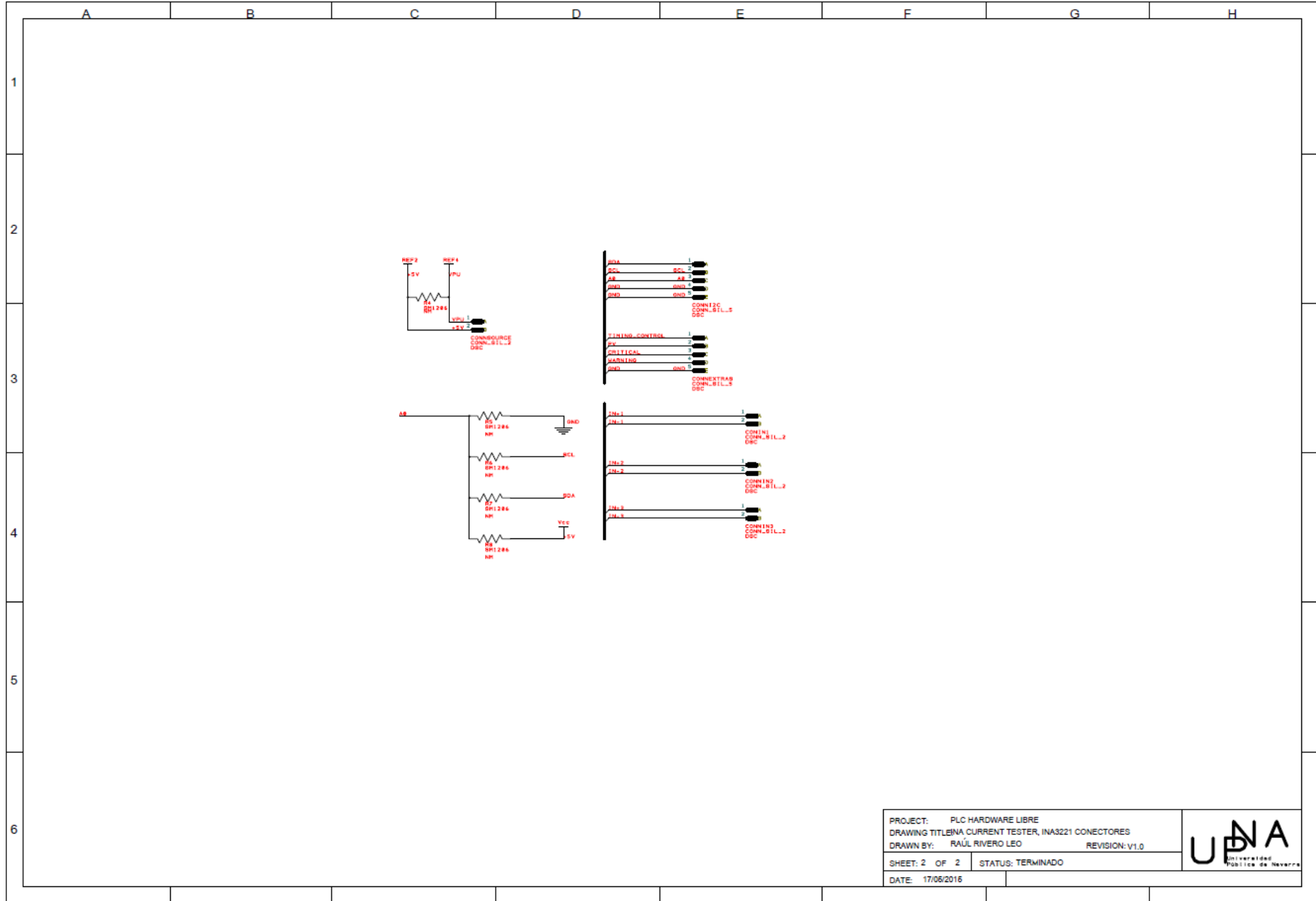




PROJECT: PLC HARDWARE LIBRE  
DRAWING TITLE: INA CURRENT TESTER, INA3221  
DRAWN BY: RAÚL RIVERO LEO REVISION: V1.0  
SHEET: 1 OF 2 STATUS: TERMINADO  
DATE: 17/06/2016







PROJECT: PLC HARDWARE LIBRE	
DRAWING TITLE: INA CURRENT TESTER, INA3221 CONECTORES	
DRAWN BY: RAÚL RIVERO LEO	REVISION: V1.0
SHEET: 2 OF 2	STATUS: TERMINADO
DATE: 17/06/2016	



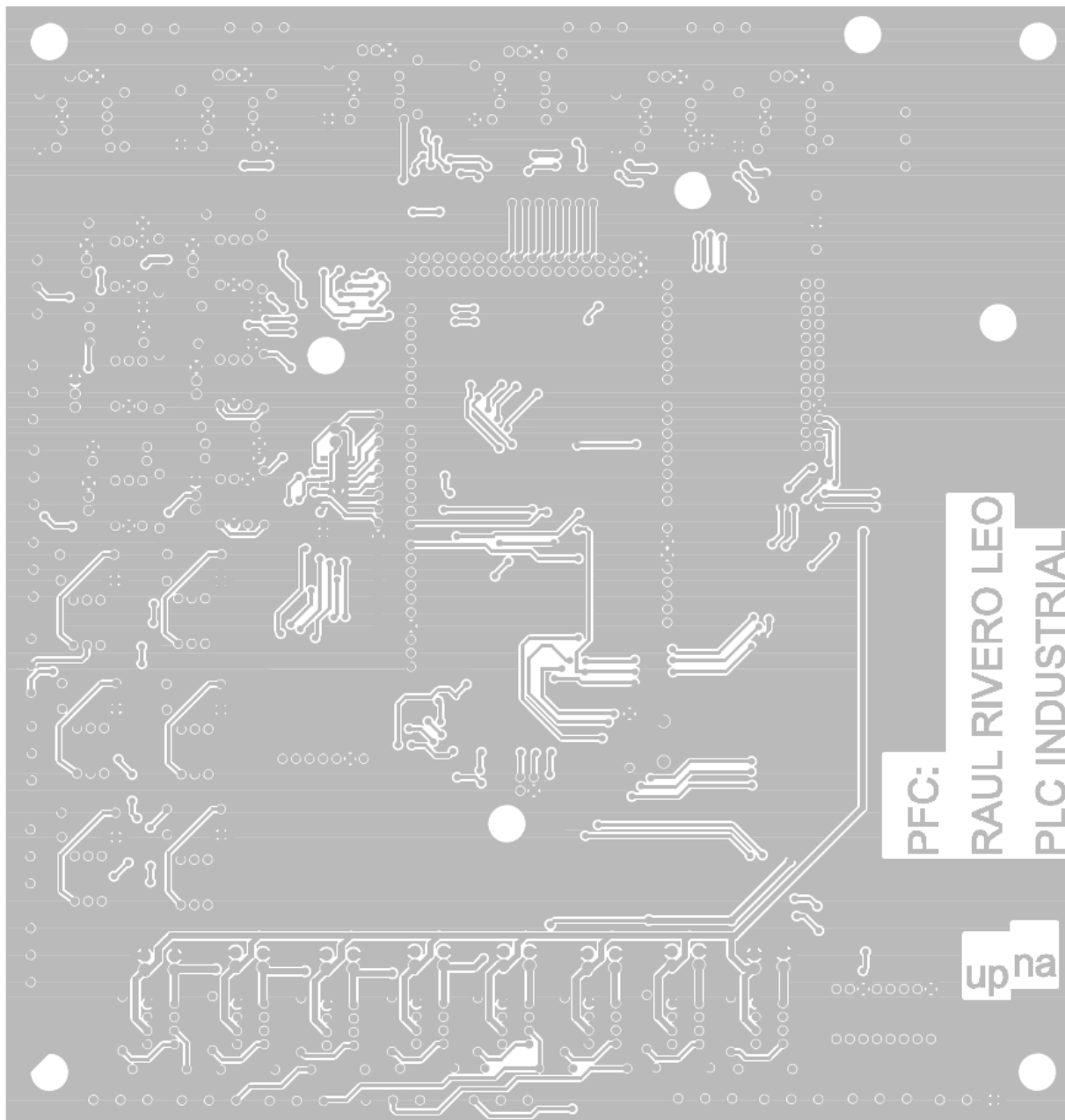
## Anexo B: Layout Completo

### ÍNDICE DE LAYOUT

Los layout aquí presentes han sido creados usando el software de DesignSpark 6.1 de RS. Cada elemento de layout contiene información sobre la hoja correspondiente.

La visualización por capas se ha obtenido mediante la página web [www.eurocircuits.com](http://www.eurocircuits.com) usando su herramienta “PCB Analyzer”.

- [Layout PCB](#)
- [Top Copper Layer.](#)
- [Bottom Copper layer.](#)
- [Plated & nonplated thruholes layer.](#)
- [Silkscreen & Documentation layer.](#)
- [Layout Test Corriente](#)
- [PCB Test Corriente.](#)
- [Layout sensor corriente INA3221](#)
- [Top Copper Layer INA.](#)
- [Bottom Copper Layer INA.](#)

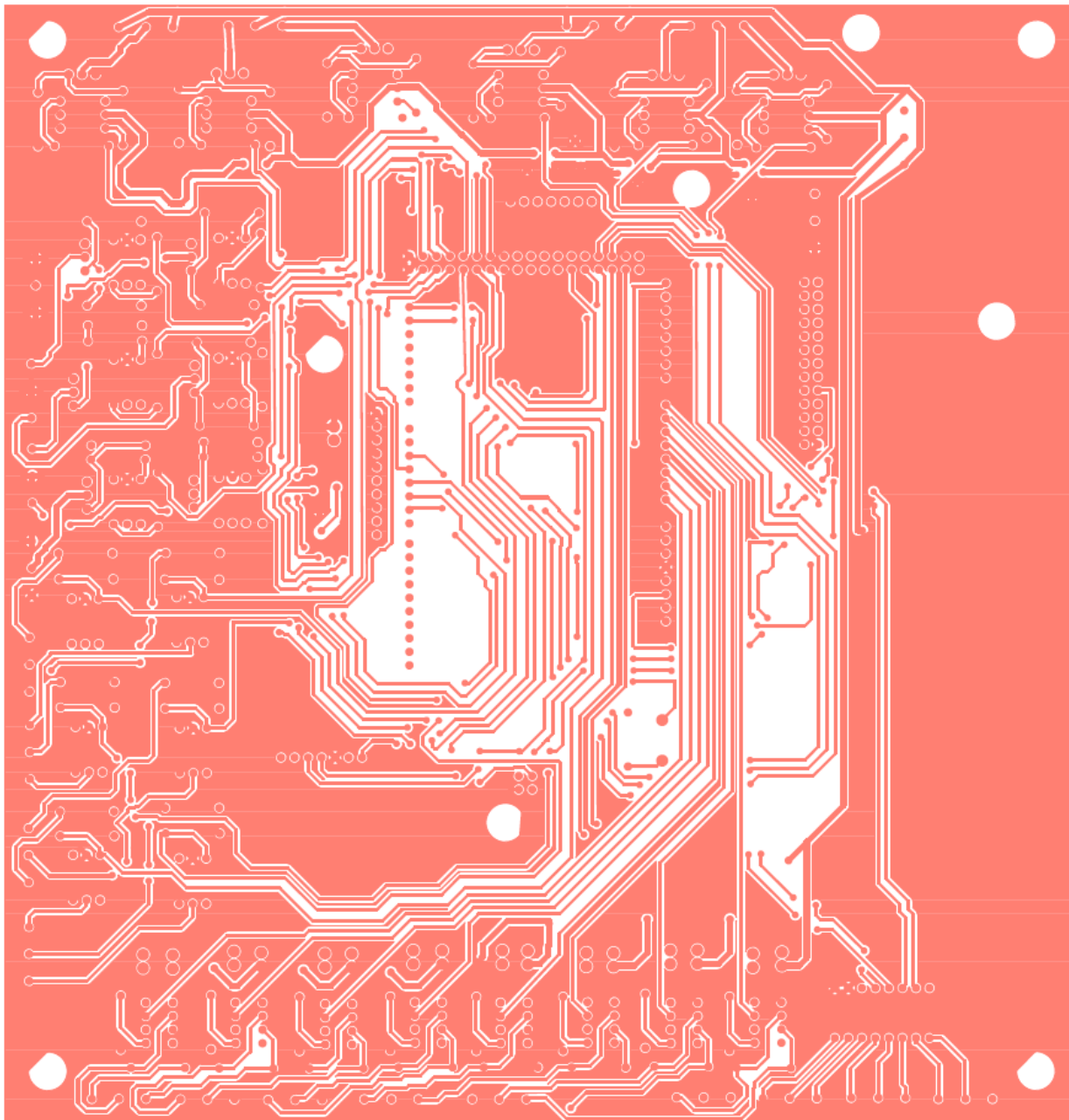


## COBRE SUPERIOR

La capa top de la PCB contiene el cobre que genera las pistas y los planos de tierra en la cara superior de la PCB.

En las especificaciones de diseño el diámetro de las vías es de 0.8 mm y una anchura mínima de pistas de 0.256mm. Por otro lado la distancia mínima entre obstáculos es de 0.256 mm.

El diseño en montaje DIP es debido a que inicialmente se esperaba una fabricación usando una CNC propia del departamento de electrónica de la universidad. Debido a la complejidad de la PCB se optó finalmente mandarla a fabricar a eurocircuits.

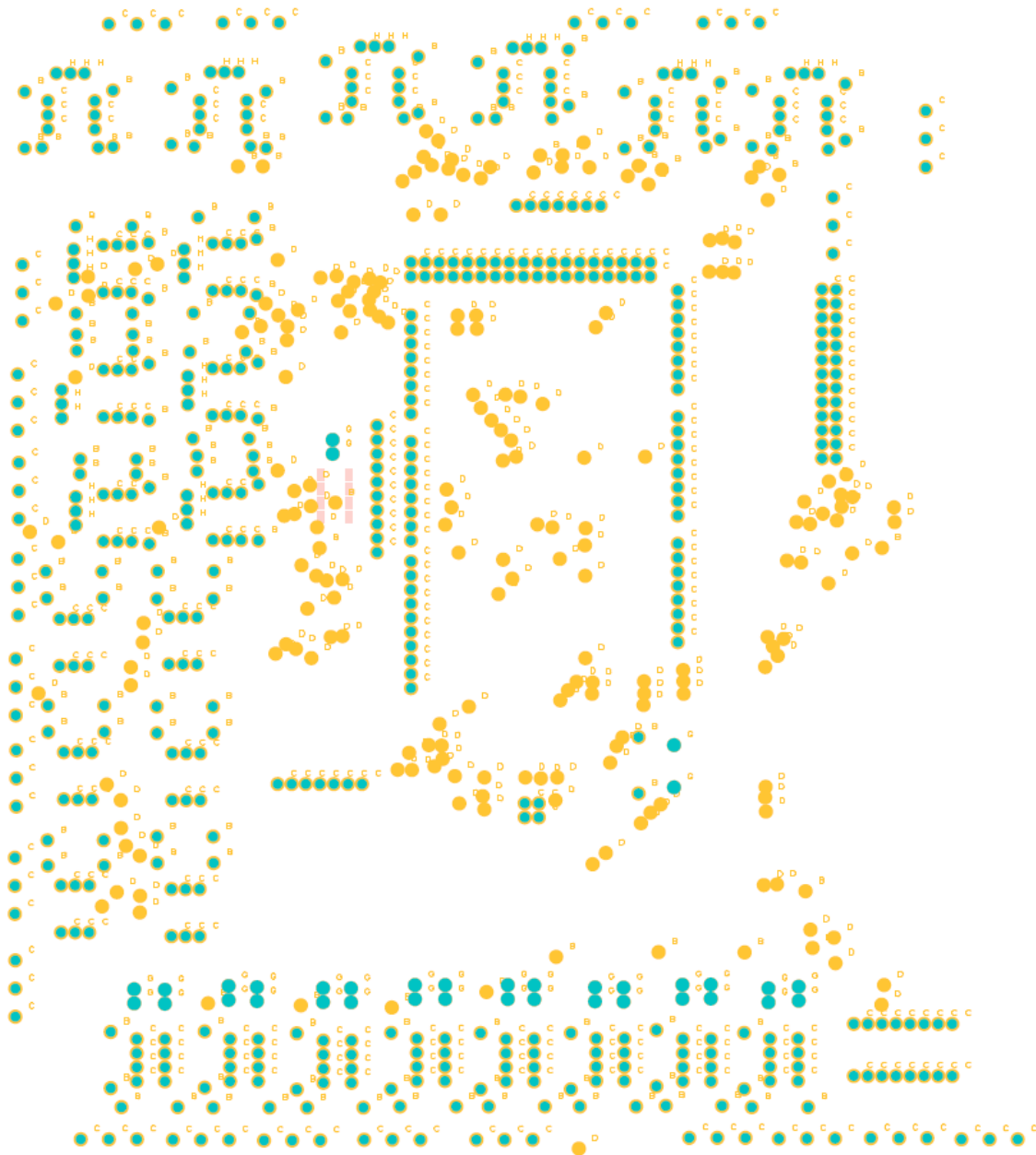


## COBRE INFERIOR

La capa opuesta a la anterior contiene el cobre contenido en la cara inferior de la PCB.

Esta capa cumple las mismas especificaciones de diseño. En esta capa se ha intentado hacer el conexionado de la mayor parte de conexiones posibles.

La otra Layer de la PCB que incluye el cobre es el bottom Copper. Esta capa contiene la mayoría de las conexiones buscando minimizar el trazado de las pistas.



## CAPA TALADROS

Está capa contiene la información de los taladros que se debe realizar sobre la PCB.

Existen dos tipos de taladros los plateados y los no plateados.

Los plateados contiene un cubierta de cobre en el agujero para hacer la unión entre la capa superior e inferior. En el plano se pueden ver en verde.

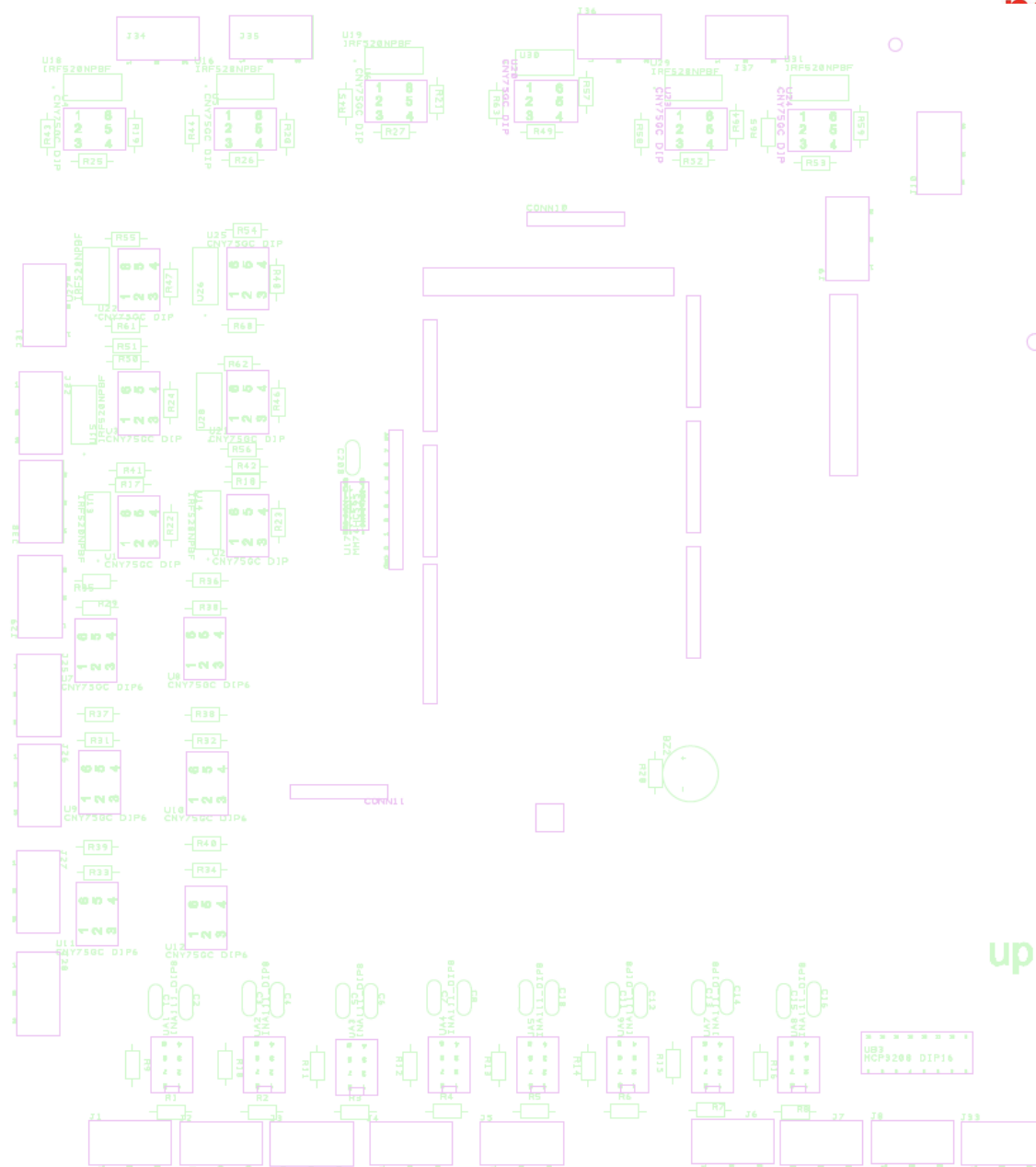
Los no plateados no contienen cobre interno en el agujero.

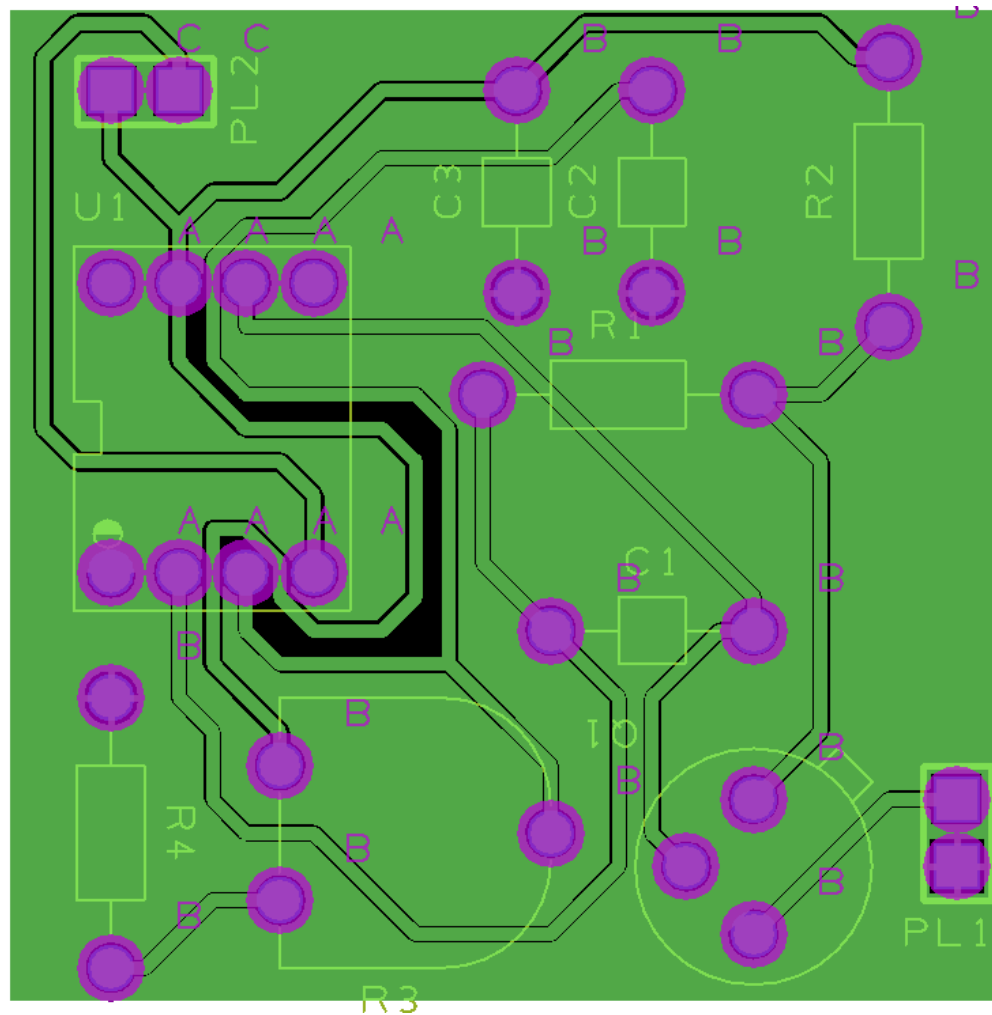
Los taladros en amarillo corresponden a taladros propios de vías.

# SERIGRAFÍA Y DOCUMENTACIÓN

En estas capas se puede ver la serigrafía realizada sobre la PCB en color verde.

Por otro lado también está presente en color morado la información de documentación. Esta información representa información extra para el diseño.





## GENERADOR DE TEST DE CORRIENTE.

Está PCB implementa un sistema para generar lazos de corriente comprendidos entre 0 y 20 mA.

Su funcionamiento consiste en el uso de un Amplificador Operacional junto con un potenciómetro y transistor para generar la corriente deseada.

Este diseño se ha realizado usando Orcad 16.6. Para todos los demás diseños se ha usado DesignSpark 6.1

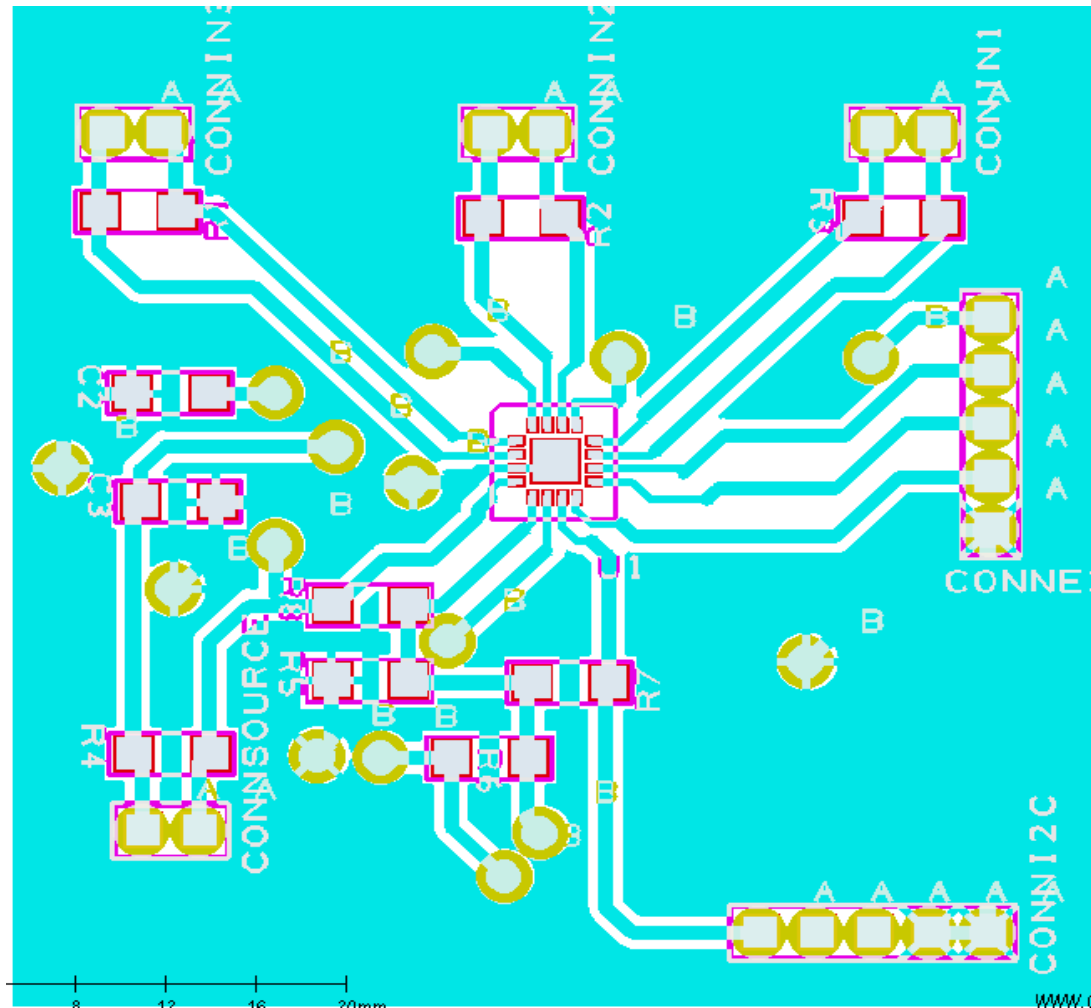
## SENSOR CORRIENTE. COBRE SUPERIOR

Esta capa contiene el cobre de la capa superior del sensor de corriente. Esta PCB implementa un INA3221 para la detección de corrientes.

Las condiciones de diseño son vías de 0.8mm de grosor y separación mínima de 0.256 entre obstáculos.

Debido al uso de un componente de encapsulado TQFP16, se ha requerido la modificación de las condiciones de diseño. De esta forma es posible la fabricación de esta PCB usando la CNC de la universidad.

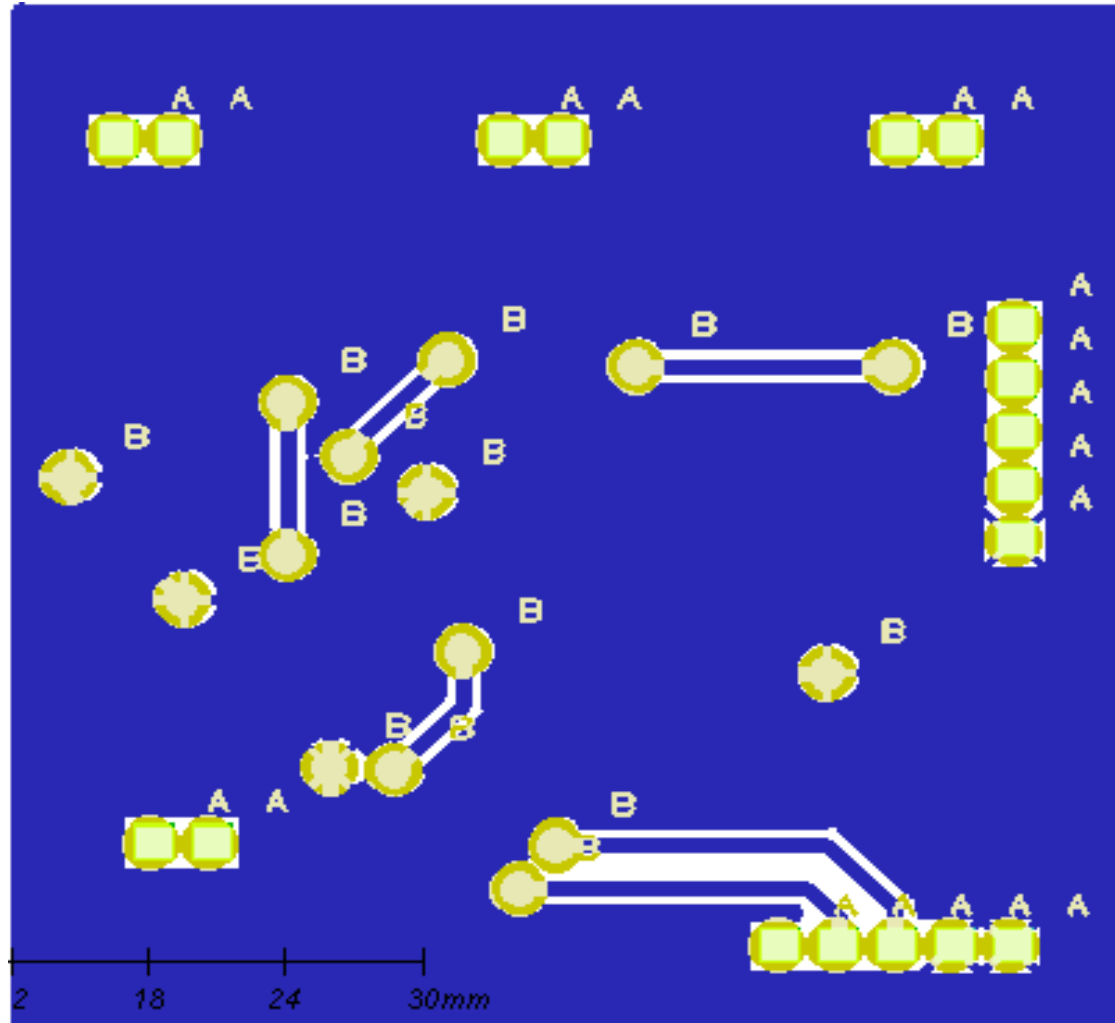
Para asegurar que es posible su fabricación, se ha utilizado conexiones rectas a las conexiones del encapsulado.





## SENSOR CORRIENTE COBRE INFERIOR

Esta es la capa inferior de la PCB anterior. Las condiciones de diseño son las mismas.



## Anexo C: Datasheet Hardware

Este prototipo busca dar solución a las aplicaciones industriales en las que prima el bajo coste y donde no se requieren grandes capacidades. Al mismo tiempo proporciona un entorno fácil de programar y muy completo en cuanto a posibilidades.

El sistema está formado por un Arduino Mega 2560 como controlador y gestor de puertos unido a un Raspberry PI mediante un protocolo I2C. El Raspberry ejecuta un Raspbian OS sobre una Raspberry PI B como entorno de sistema. Sobre este sistema se ejecuta un programa en C que gestiona la comunicación mediante I2C con Arduino. Por otro lado este mismo programa se comunica con una base de datos MySQL sobre la que se guarda la información del estado de la configuración del sistema y el estado actual.

La información del conexionado de cada dispositivo está en su apartado correspondiente y la información completa de diagramas en el anexo de esquemáticos.

### ENTRADAS Y SALIDAS

Tabla 6 Entradas y Salidas del PLC.

Entradas digitales a 5 Voltios	6
Salidas digitales a 5 Voltios	3
Salidas digitales a 5 Voltios PWM	3
Salidas digitales a 12 Voltios	3
Salidas digitales a 12 Voltios PWM	3
Entradas analógicas 12 bits	8
Entradas sensor lazo de corriente 4 – 20 mA	8
Salidas de Relé NA / NC	8

### CONEXIÓN DE LA ALIMENTACIÓN PCB

La alimentación a la PCB se realiza suministrando 5 Voltios, 12 Voltios y -12 Voltios junto con GND por líneas separadas. La fuente de alimentación que viene con el prototipo suministra estas tensiones mediante 4 cables independientes.

Para el asegurar el correcto funcionamiento del sistema se requiere un mínimo de 3 Amperios a 5 Voltios. Por otro lado la alimentación de 12 voltios solo sirve para activar los canales digitales a 12 voltios por lo tanto la corriente requerida dependerá de la carga instalada. La fuente de alimentación conectada solo es capaz de suministrar 6 Amperios a 12 voltios.

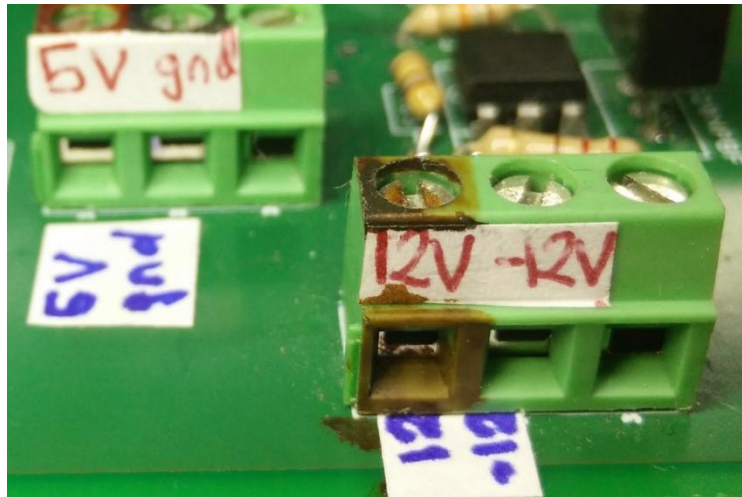


Figura 91 Conectores de alimentación

En caso de sobredemanda de potencia por parte de la fuente de alimentación existe la posibilidad de instalar un ventilador a 12 voltios controlable por PWM, se puede dejar activado a 12 voltios o conectarlo a uno de los puertos PWM del sistema. Este modo de funcionamiento permite aumentar la potencia demandada por el sistema.

La fuente de alimentación se conecta directamente a 360 voltios y dispone de dos fusibles de protección y un interruptor de apagado general. Se debe tener especial cuidado al tocar este elemento al trabajar con tensiones de red es igual de importante tener en cuenta que la fuente de alimentación conserva durante un tiempo de 5 minutos la tensión almacenada en los condensadores de la fuente conmutada.

## CONEXIÓN DE LAS ENTRADAS DIGITALES

El sistema proporciona 6 canales de entrada digitales optoacoplados a los cuales se accede a través de 4 conectores fijos para PCB. La corriente por estos puertos está limitada gracias al uso de los optoacopladores perfectamente diseñados.



Figura 92 Conexión de Entradas Digitales

Este sistema de montaje permite referir cada canal a un voltaje de referencia deseado que puede ser GND o cualquier otro externo proporcionado por la entrada. La conexión de cada canal por

lo tanto se hace con un par de cables. Los canales de izquierda a derecha van desde el 1+, 1- hasta el 6+ y 6-.

Cada canal va conectado como es visible su optoacoplador correspondiente. A este subsistema solo se le debe conectar tensiones de 5 Voltios.

## CONEXIÓN DE LAS SALIDAS DIGITALES

En el sistema hay disponibles una serie de salidas digitales que proporcionan tensiones entre 5 y 12 Voltios, de las cuales algunas de ellas también pueden ser reguladas mediante modulación de ancho de pulso PWM. El sistema se encuentra optoacoplado y posteriormente conectado a un Mosfet para proporcionar la tensión deseada. Los conectores fijos de PCB permiten dar suficiente corriente como sea necesario. Al mismo tiempo tienen la capacidad de fijar completamente los cables conectados.

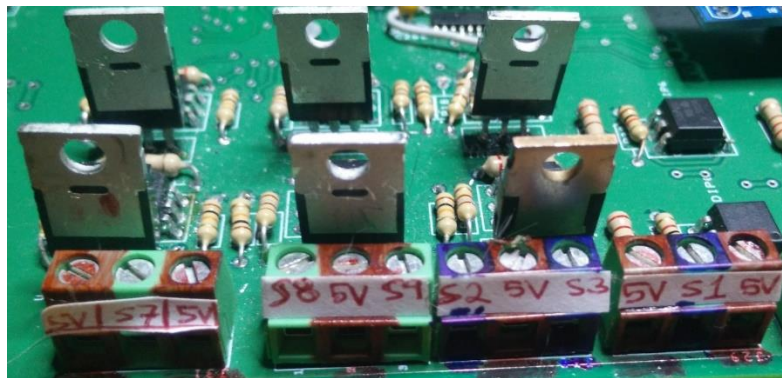


Figura 93 Conexión de Salidas Digitales 1

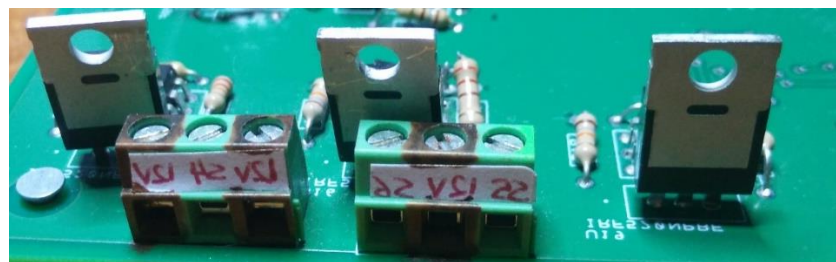


Figura 94 Conexión de Salidas Digitales 2

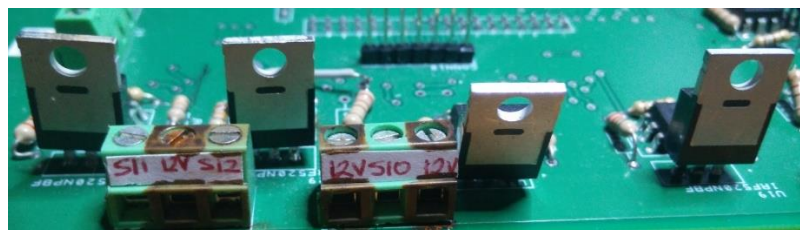


Figura 95 Conexión de Salidas Digitales 3

Las salidas aquí presentes son desde S1 hasta S12 en estas salidas se encuentran agrupadas las salidas a 5 Voltios y las salidas a 12 Voltios, incluyendo las salidas que disponen de PWM.

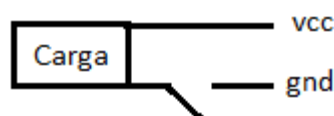
El mapeado de las salidas es el siguiente.

**Tabla 7 Pin Map de las salidas digitales.**

S1	Salida a 5 Voltios
S2	Salida a 5 Voltios
S3	Salida a 5 Voltios
S4	Salida a 12 Voltios
S5	Salida a 12 Voltios
S6	Salida a 12 Voltios
S7	Salida a 5 Voltios PWM
S8	Salida a 5 Voltios PWM
S9	Salida a 5 Voltios PWM
S10	Salida a 12 Voltios PWM
S11	Salida a 12 Voltios PWM
S12	Salida a 12 Voltios PWM

La corriente máxima que se puede aportar a través de estos pines está limitada por la fuente de alimentación más que por el Mosfet en sí. Se recomienda no exceder los 1000 mA por canal y los 2 Amperios en 5 Voltios. En el canal de 12 Voltios se puede suministrar más corriente al estar más liberado esta red, no se recomienda suministrar más de 4 Amperios en conjunto

El sistema de conexión está realizado de tal forma que sea el canal de drenaje el que se abre o se cierra según la orden de encendido o apagado. Se debe cablear un pin de alimentación correspondiente de 5 o 12 voltios junto con la salida correspondiente que hace de tierra. Este sistema de cableado se cambiara en próximas versiones para dejar siempre fijada la tierra de la salida.



**Figura 96 Explicación de conexión de una carga a una salida digital.**

## CONEXIONES DE RELÉ NA / NC

Otra de las salidas disponibles en nuestro proyecto está la posibilidad de usar los 8 canales de relé disponibles, estos relés tienen disponibles salidas normalmente abierto y normalmente cerrado. Por otro lado las características de estos relés son las siguientes:

Tabla 8 Características relés.

10 A 250 VAC	10 A 125 VAC
10 A 30 VDC	10 A 8 VDC

Los relés que usa esta placa tiene disponible los contactos de normalmente abierto y normalmente cerrado. La tensión de excitación de la bobina del relé son 5 voltios más fácil de manejar que los 12 voltios comúnmente utilizados.

El control de estos 8 relés se realiza por un lado alimentando a la placa mediante GND y VCC 5V y por otro lado con 8 pines digitales a 5 voltios que son los que excitan los fotodiodos de los optoacopladores que permiten la conducción de los opto transistores del mismo optoacoplador. De esta forma se pone en funcionamiento los Mosfet que permiten la excitación de las bobinas de los relés. Como elemento de protección adicional un diodo permite que se descargue la bobina al desconectar el relé.



Figura 97 Relé SRD-5VDC-SL-C usado en los relés.

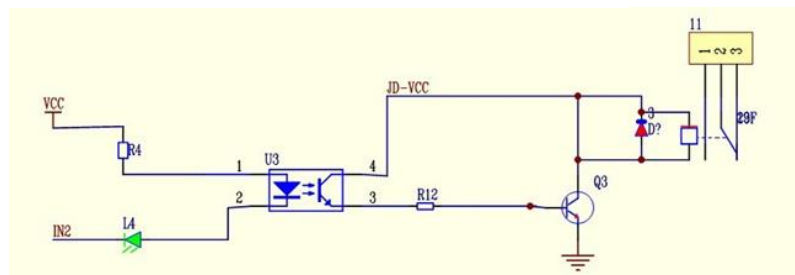


Figura 98 Sistema de funcionamiento y protección del array de relés.

Como forma de simplificar el control de este dispositivo se ha usado un circuito integrado 74HC595 en formato SMD que realiza la tarea de multiplexado para el control de los 8 relés de una forma más simple. Mediante el uso de un sistema de comunicación SPI podemos convertir un byte enviado desde el módulo de control Mega al circuito integrado para ser multiplexado a 8 bit correspondiente a los 8 canales. Es requisito fundamental para el funcionamiento de los mismos que este dispositivo presente Latch mantenido para que poder dejar fijado el estado de cada canal.

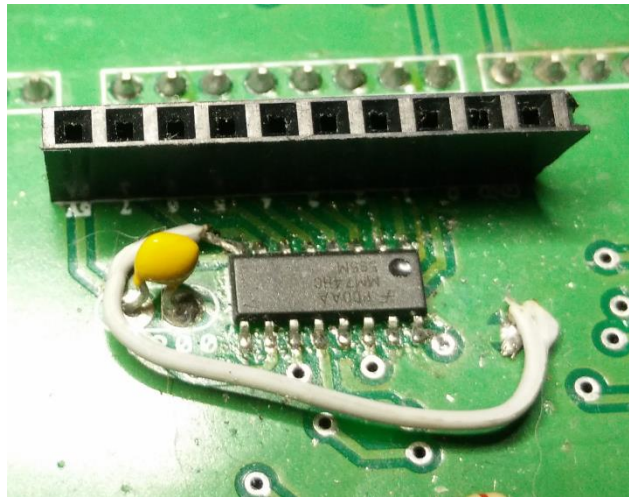


Figura 99 Circuito en placa de gestor de relés.

Por ultimo están disponible los esquemáticos y el diseño del layout de este componente en el anexo correspondiente. Es un diseño de hardware libre disponible de forma pública.

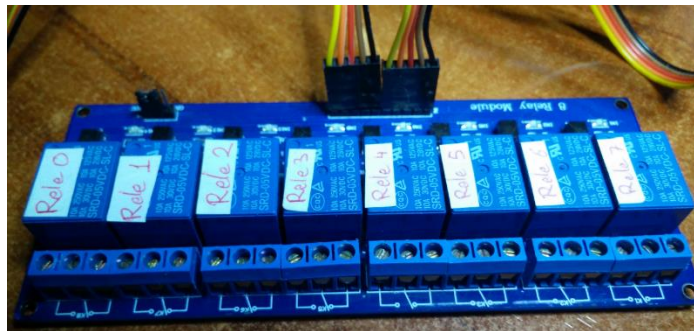


Figura 100 Numeración de los relés en la PCB.

## CONEXIONES LAZO DE CORRIENTE

Este prototipo dispone de 8 canales para realizar el sensado analógico de lazos de corriente entre 4 y 20 mA. El método final aplicado en el prototipo para realizar este tipo de medición es la conversión directa de corriente a voltaje. De esta forma convertimos una señal entre 4 y 20 mA a un valor comprendido entre 0.8 y 4.4 Voltios. Para ello el sistema está cableado mediante una resistencia de precisión de 220 Ohm y conectado entre los canales analógicos y GND. Este sistema de medición ha sido elegido al final por su simplicidad y fiabilidad.

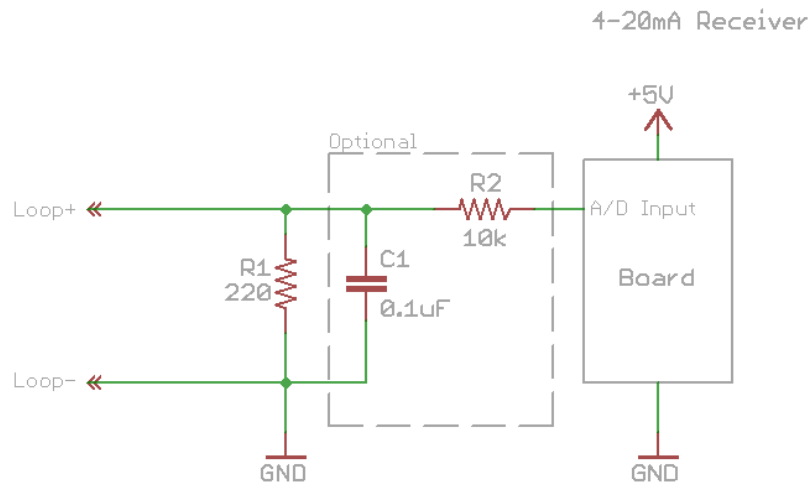


Figura 101 Circuito sensado de lazos de corriente 4-20 mA más sencillo.

## CONEXIONES SENSOR DE CORRIENTE

Este dispositivo presenta 8 canales de sensado de precisión con una resolución de 12 bits mejor que la que presenta por defecto un Arduino Mega. De esta forma en el sensado de valores entre 0 y 5 voltios hay disponibles 4096 valores posibles. Es un notable incremento de resolución respecto a la placa por defecto. Estos canales de sensado no se encuentran protegidos de ninguna forma salvo el mismo circuito integrado que realiza la tarea de sensado analógico un MCP3208.

Para el sensado de canales analógicos se requiere conectar un par de cables que son la GND del sensor y el cable con la señal analógica entre 0 y 5 voltios.

## ELEMENTOS ADICIONALES

Además de los elementos de conexiones descritos el sistema incluye una serie de elementos adicionales, reloj RTC para mantener hora del sistema. El respaldo se hace mediante el uso de una pila que mantiene la alimentación del RTC para guardar la hora del sistema. La conexión de este dispositivo se realiza mediante i2c siendo el Raspberry el que ostenta el pleno control de las comunicaciones i2c.

Otro de los elementos auxiliares presentes en esta PCB es una pantalla LCD con botones de control integrado que permite la visualización de mensajes a través de su pantalla de dos segmentos. El control se realiza mediante la comunicación por parte del maestro para el envío de mensajes y recepción de pulsaciones.

El último de los elementos disponibles es la membrana táctil de 9 botones numéricos que se controla mediante el uso de 7 cables para el control de filas y columnas. El control se realiza directamente a través del Arduino





## Anexo D: Datasheet Software.

### LISTA DE FUNCIONES Y CODIGOS DEL SISTEMA.

Para el funcionamiento del PLC se han implementado una serie de mensajes para realizar la lógica de comunicación entre los diferentes elementos lógicos. Esta comunicación se realiza íntegramente entre el Arduino Mega y el Raspberry Pi, a través de las líneas de comunicación I2C.

Dada la naturaleza de la información transmitida entre los dispositivos y la existencia de defectos de diseño en el microprocesador principal del Raspberry Pi, la comunicación entre los dispositivos requiere adecuar los mensajes a una serie de condiciones como pueden ser: Limitar el número de bytes máximos de los mensajes y adecuar los tiempos de sincronización de las comunicaciones.

Los mensajes de comunicación enviados mediante I2C entre el Arduino y el Raspberry son diferentes según quien sea el emisor y el receptor. La tabla inferior muestra el número de códigos diferentes y quien es el emisor correspondiente de dicho mensaje.

**Tabla 9 Mensajes recibidos por el Slave Arduino**

Código I = 9	checkDigitalOutput()
Código N = 14	funcionRele()

El código interno del Arduino comprueba el primer carácter en formato ASCII enviado en la trama I2C y comprueba si pertenece a uno de los dos códigos anteriores. Si coincide uno de los códigos activa la función correspondiente y en caso contrario saltara el aviso de código no encontrado. Cada código tiene su propio formato de trama de datos que será descodificado en la función correspondiente.

La explicación y trama de información de estas funciones es la siguiente:

CheckDigitalOutput es una orden recibida por el Raspberry Pi que activa las salidas del PLC según el mensaje recibido. La función “checkDigitalOutput()” está formado por 12 caracteres que pueden ser un 0 o un 1. La correspondencia entre carácter y la salida digital a la que se refiere es la siguiente: El primer dígito se refiere a la salida 1 y el último dígito a la salida número 12. Por otro lado el valor de 0 indica una orden de apagado y un valor de 1 expresa una orden de encendido. Finalmente se cierra el mensaje con un carácter Z para indicar el cierre del mensaje. Un ejemplo de código enviado sería el siguiente. “I000111000111Z”

La función “funcionRele” es una orden recibida por parte del Raspberry Pi. La función activa los relés según el comando los parámetros recibidos. El formato es el siguiente: Tras el carácter N para detectar está función recibidos un número que corresponde con el relé deseado seguido de un

número que puede ser 0 o 1. El valor 0 indica una orden de apagado y el valor 1 indica la orden de encendido. Finalmente se cierra con el carácter Z para indicar el cierre del mensaje. El formato completamente desarrollado sería así: “N0a1b2c3d4e5g6h7iZ”. Los siguientes casos serían mensajes validos: “N00112030Z”. En este caso se pediría el apagado de los relés 0, 2 y 3; junto con el encendido del relé 1.

Uno de los puntos a implementar en la programación es la recepción de comandos introducidos por keypad y la comunicación con la pantalla LCD.

Por otro lado están los mensajes recibidos por parte del Raspberry Pi y que proceden todos del Arduino Mega 2560.

**Tabla 10** Lista de códigos recibidos por el Master Raspberry Pi

Código A = 1	rxStatusOk()
Código B = 2	rxStatusError()
Código C = 3	mensajesPendientes()
Código D = 4	checkTime()
Código E = 5	checkArduino()
Código F = 6	loadRele()
Código G = 7	checkTeclado()
Código H = 8	checkDigitalInput()
Código I = 9	checkDigitalOutput()
Código J = 10	checkAnalogInput()
Código K = 11	checkCurrentSensor()
Código L = 12	checkBuzzerStatus()
Código M = 13	checkArduinoDebug()

La tabla superior contiene la información sobre todas las funciones posibles de control del sistema. En el caso de todos estos mensajes empiezan por un carácter que indica el código correspondiente, seguido de los parámetros asociados a ese comando y finalizados por un símbolo de cierra de código “Z”.

La información inferior contiene el funcionamiento y payload de los diferentes códigos.

La función rxStatusOk() recibe diferentes valores según el mensaje recibido por parte del Arduino El formato es el siguiente: A22Z. Este mensaje interior indica la recepción de un mensaje de Status correspondiente al número 22, finalmente se cierra el mensaje. Estos mensajes se guardan en la base de datos del sistema para poder leer y comprobar los mensajes recibidos. De momento no existe ninguna correspondencia con significados reales.

La función `rxStatusError()` es exactamente igual a la función anterior pero refiriéndose únicamente a mensajes que implican errores o situaciones de fallo en el sistema. Igualmente no existe ninguna correspondencia con significados reales.

La función `mensajesPendientes()` muestra por pantalla y los almacena en un array del sistema la lista de mensajes por comprobar Su formato es igual que los dos anteriores.

La función `checkTime()` es una función que se activa al recibir el `TimeStamp` por parte del Arduino lo cual servirá para comprobar la correspondencia de tiempos del sistema.

La siguiente función `checkArduino` recibe una serie de comprobaciones de estado y mensajes procedente del Arduino. Todavía no ha sido implementada.

La función `loadRele()` es una función que recibe automáticamente el estado de los relés del PLC para comprobar si se han producido cambios y en caso de haberlos registrar la nueva información en la base de datos del sistema. Su formato es el mismo que el que recibe el Arduino.

La función `checkTeclado()` se encarga de registrar los códigos introducidos por teclado para actuar en consonancia. Es una función actualmente no implementada.

La función `checkDigitalInput()` recibe por parte del slave la situación actual de las entradas digitales. Una vez descodificada se graba la información en la base de datos del sistema. El formato de recepción de los mensajes es el siguiente: `H0x1x2x3x4x5xZ` Donde los números indican la entrada correspondiente y el valor posterior el estado de la entrada digital. Un 0 indica apagado y un 1 indica encendido.

La función `checkDigitalOutput()` recibe por parte del Arduino la situación actual de las salidas digitales para comprobar la información y grabarla en la base de datos. El formato es el mismo que en el caso de Arduino Mega.

La función `checkAnalogInput()` recibe en formato hexadecimal por parte del Arduino la información de los 8 canales de sensado analógico. Esta codificación en hexadecimal permite recibir el valor de la tensión codificada entre 0 y 255, gracias al uso de dos caracteres hexadecimales por cada canal de sensado. De esta forma el formato que descodifica esta función es: `"J0HH1HH2HH3HH4HH5HH6HH7HHZ"`. Las H indican los valores hexadecimales.

La función `checkCurrentSensor()` recibe en formato hexadecimal por parte del Arduino la información de los canales de sensado de corriente y los graba en la base de datos del sistema. Actualmente es una función que no ha sido implementada al 100%.

La función `checkBuzzerStatus()` indica que se a activado algún tipo de alarma sonora y el motivo de la alarma. No ha sido implementada.

La función `checkArduinoDebug()` permite recibir mensajes de Debug y comprobar el estado del modo Debug por parte del Raspberry PI.

## PAGÍNA WEB DE MONITORIZACIÓN.

El sistema incorpora un servidor apache con soporte php para permitir la monitorización del sistema. El sistema Raspberry establece automáticamente la dirección 192.168.0.20 para la dirección del sistema. Podemos acceder a phpMyAdmin <sup>31</sup> accediendo a la página web <http://192.168.0.20/phpmyadmin/> con el usuario pi y contraseña raspberry.

Mediante el uso de phpMyAdmin podemos hacer backups del sistema y modificar parámetros con control total de la base de datos.

Por otro lado para la monitorización y control básico del sistema está implementada la página de monitorización <http://192.168.0.20//monitorizacion/>. Este sistema permite un control y visualización de las principales variables del sistema. Como elemento de navegación podemos movernos por 6 pestañas diferentes que corresponden a:

- Panel de control del sistema de monitorización Arduino. Es la página principal de esta interfaz que permite navegar entre las diferentes opciones que contiene el sistema.
- Monitorización del sistema. Esta pestaña permite la monitorización de las siguientes entradas y salidas del sistema: Entradas digitales, Entradas de sensado Analógico, Entradas de sensores de corriente, Estado de los relés, Estado de las salidas Digitales. Las únicas opciones para modificar son el estado de los relés y el estado de las salidas digitales.
- Programación del sistema. El objetivo previsto de está pestaña era cargar un programa a partir de un fichero de texto para programar el sistema. No está implementado.
- Listado de mensajes del sistema. Está pestaña recoge el estado de todos los mensajes de estado y errores del Raspberry y Arduino. Los códigos de mensajes y significados se recoge en esta pestaña.
- Variables de configuración del sistema. Desde está interfaz se puede configurar diferentes parámetros de configuración del sistema así como información de la versión y diferentes configuraciones.

- Modo Debug. Esta pestaña permite configurar el modo Debug del Raspberry Pi y Arduino.
- Información del proyecto. Contiene información de contacto y sobre el proyecto en sí mismo.

**Página de monitorización del sistema.**

Esta página web es una demo de la monitorización del sistema PLC. Aquí podemos ver el estado de las variables del sistema. Podemos ver el estado de las entradas salidas. Además permite la omutación de los estados.

Esta página web está escrita en HTML5, usando además lenguaje PHP5 para acceder a la Base De Datos MySQL y CSS como forma de dar formato a la página web.

<p><b>Entradas Digitales</b></p> <table border="1"> <thead> <tr> <th>ID</th> <th>INPUT</th> <th>STATE</th> <th>LEVEL</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>2</td><td>2</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>3</td><td>0</td><td>0</td></tr> <tr><td>4</td><td>4</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>5</td><td>0</td><td>0</td></tr> <tr><td>6</td><td>6</td><td>1</td><td>0</td></tr> </tbody> </table> <p><b>Entradas de sentido Analógico</b></p> <table border="1"> <thead> <tr> <th>ID</th> <th>CHANNEL</th> <th>VALUE</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>39</td></tr> <tr><td>1</td><td>1</td><td>29</td></tr> <tr><td>2</td><td>2</td><td>48</td></tr> <tr><td>3</td><td>3</td><td>54</td></tr> <tr><td>4</td><td>4</td><td>43</td></tr> <tr><td>5</td><td>5</td><td>239</td></tr> <tr><td>6</td><td>6</td><td>67</td></tr> <tr><td>7</td><td>7</td><td>68</td></tr> </tbody> </table> <p><b>Entradas de Sensores de Corriente</b></p> <table border="1"> <thead> <tr> <th>ID</th> <th>CHANNEL</th> <th>VALUE</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>512</td></tr> <tr><td>1</td><td>1</td><td>777</td></tr> <tr><td>2</td><td>2</td><td>888</td></tr> <tr><td>3</td><td>3</td><td>0</td></tr> <tr><td>4</td><td>4</td><td>0</td></tr> <tr><td>5</td><td>5</td><td>0</td></tr> <tr><td>6</td><td>6</td><td>0</td></tr> <tr><td>7</td><td>7</td><td>0</td></tr> </tbody> </table>	ID	INPUT	STATE	LEVEL	0	0	0	0	1	1	0	0	2	2	0	0	3	3	0	0	4	4	0	0	5	5	0	0	6	6	1	0	ID	CHANNEL	VALUE	0	0	39	1	1	29	2	2	48	3	3	54	4	4	43	5	5	239	6	6	67	7	7	68	ID	CHANNEL	VALUE	0	0	512	1	1	777	2	2	888	3	3	0	4	4	0	5	5	0	6	6	0	7	7	0	<p><b>Estado de Reles</b></p> <table border="1"> <thead> <tr> <th>ID</th> <th>Rele</th> <th>Numero</th> <th>Estado</th> <th>Set</th> <th>FuncionSet</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>2</td><td>2</td><td>2</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>3</td><td>3</td><td>3</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>4</td><td>4</td><td>4</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>5</td><td>5</td><td>5</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>6</td><td>6</td><td>6</td><td>0</td><td>0</td><td>FuncionSet</td></tr> <tr><td>7</td><td>7</td><td>7</td><td>0</td><td>0</td><td>FuncionSet</td></tr> </tbody> </table> <p>ID: <input type="text"/></p> <p>Status: <input type="text"/></p> <p><input type="button" value="Actualizar reles"/></p> <p><b>Estado de las salidas Digitales</b></p> <table border="1"> <thead> <tr> <th>ID</th> <th>OUTPUT</th> <th>VOLTAJE</th> <th>STATUS</th> <th>PWM</th> <th>FLAG</th> <th>SET</th> <th>Valor</th> <th>FuncionSet</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>5V</td><td>1</td><td>0</td><td>0</td><td>1</td><td></td><td>FuncionSet</td></tr> <tr><td>1</td><td>1</td><td>5V</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>2</td><td>2</td><td>5V</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>3</td><td>3</td><td>5V PWM</td><td>1</td><td>0</td><td>0</td><td>1</td><td></td><td>FuncionSet</td></tr> <tr><td>4</td><td>4</td><td>5V PWM</td><td>1</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>5</td><td>5</td><td>5V PWM</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>6</td><td>6</td><td>12V</td><td>1</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>7</td><td>7</td><td>12V</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>8</td><td>8</td><td>12V</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>9</td><td>9</td><td>12V PWM0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> <tr><td>A</td><td>A</td><td>12V PWM0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>FuncionSet</td></tr> </tbody> </table> <p>ID: <input type="text"/></p> <p>STATUS: <input type="text"/></p> <p><input type="button" value="Actualizar DigitalOutput"/></p>	ID	Rele	Numero	Estado	Set	FuncionSet	0	0	0	0	0	FuncionSet	1	1	1	0	0	FuncionSet	2	2	2	0	0	FuncionSet	3	3	3	0	0	FuncionSet	4	4	4	0	0	FuncionSet	5	5	5	0	0	FuncionSet	6	6	6	0	0	FuncionSet	7	7	7	0	0	FuncionSet	ID	OUTPUT	VOLTAJE	STATUS	PWM	FLAG	SET	Valor	FuncionSet	0	0	5V	1	0	0	1		FuncionSet	1	1	5V	0	0	0	0		FuncionSet	2	2	5V	0	0	0	0		FuncionSet	3	3	5V PWM	1	0	0	1		FuncionSet	4	4	5V PWM	1	0	0	0		FuncionSet	5	5	5V PWM	0	0	0	0		FuncionSet	6	6	12V	1	0	0	0		FuncionSet	7	7	12V	0	0	0	0		FuncionSet	8	8	12V	0	0	0	0		FuncionSet	9	9	12V PWM0	0	0	0	0		FuncionSet	A	A	12V PWM0	0	0	0	0		FuncionSet	<p><b>Debug_Mode</b> 0</p> <p><b>STATUS</b> 0</p> <p><b>V_Firmware</b> 5</p> <p><b>V_Hardware</b> 5</p>
ID	INPUT	STATE	LEVEL																																																																																																																																																																																																																																																							
0	0	0	0																																																																																																																																																																																																																																																							
1	1	0	0																																																																																																																																																																																																																																																							
2	2	0	0																																																																																																																																																																																																																																																							
3	3	0	0																																																																																																																																																																																																																																																							
4	4	0	0																																																																																																																																																																																																																																																							
5	5	0	0																																																																																																																																																																																																																																																							
6	6	1	0																																																																																																																																																																																																																																																							
ID	CHANNEL	VALUE																																																																																																																																																																																																																																																								
0	0	39																																																																																																																																																																																																																																																								
1	1	29																																																																																																																																																																																																																																																								
2	2	48																																																																																																																																																																																																																																																								
3	3	54																																																																																																																																																																																																																																																								
4	4	43																																																																																																																																																																																																																																																								
5	5	239																																																																																																																																																																																																																																																								
6	6	67																																																																																																																																																																																																																																																								
7	7	68																																																																																																																																																																																																																																																								
ID	CHANNEL	VALUE																																																																																																																																																																																																																																																								
0	0	512																																																																																																																																																																																																																																																								
1	1	777																																																																																																																																																																																																																																																								
2	2	888																																																																																																																																																																																																																																																								
3	3	0																																																																																																																																																																																																																																																								
4	4	0																																																																																																																																																																																																																																																								
5	5	0																																																																																																																																																																																																																																																								
6	6	0																																																																																																																																																																																																																																																								
7	7	0																																																																																																																																																																																																																																																								
ID	Rele	Numero	Estado	Set	FuncionSet																																																																																																																																																																																																																																																					
0	0	0	0	0	FuncionSet																																																																																																																																																																																																																																																					
1	1	1	0	0	FuncionSet																																																																																																																																																																																																																																																					
2	2	2	0	0	FuncionSet																																																																																																																																																																																																																																																					
3	3	3	0	0	FuncionSet																																																																																																																																																																																																																																																					
4	4	4	0	0	FuncionSet																																																																																																																																																																																																																																																					
5	5	5	0	0	FuncionSet																																																																																																																																																																																																																																																					
6	6	6	0	0	FuncionSet																																																																																																																																																																																																																																																					
7	7	7	0	0	FuncionSet																																																																																																																																																																																																																																																					
ID	OUTPUT	VOLTAJE	STATUS	PWM	FLAG	SET	Valor	FuncionSet																																																																																																																																																																																																																																																		
0	0	5V	1	0	0	1		FuncionSet																																																																																																																																																																																																																																																		
1	1	5V	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
2	2	5V	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
3	3	5V PWM	1	0	0	1		FuncionSet																																																																																																																																																																																																																																																		
4	4	5V PWM	1	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
5	5	5V PWM	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
6	6	12V	1	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
7	7	12V	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
8	8	12V	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
9	9	12V PWM0	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		
A	A	12V PWM0	0	0	0	0		FuncionSet																																																																																																																																																																																																																																																		

Figura 102 Panel de visualización de las variables del sistema.



## Anexo E: Presupuesto.

**RAÚL RIVERO LEO**  
**UNIVERSIDAD PÚBLICA DE NAVARRA**  
**PAMPLONA**

**FECHA:** 01-06-15  
**FACT N°:** 1

**CLIENTE:** Presupuesto para la fabricación de 10 unidades del PLC  
**DIRECCION:** Hardware libre V1.  
**POBLACION:**

### RESUMEN DE PRESUPUESTO

CAP. DESCRIPCION	TOTAL CAPITULO
3 FABRICACIÓN PCB	120,00 €
4 COMPRA COMPONENTES MOUSER	99,40 €
9 COMPRA MODULOS ELECTRÓNICA	97,67 €
<b>TOTAL PRESUPUESTO EJECUCION MATERIAL</b>	<b>317,07 €</b>
Gastos Generales y Beneficio Industrial I.V.A.	
<b>TOTAL PRESUPUESTO DE CONTRATA</b>	<b>317,07 €</b>

**RAÚL RIVERO LEO**  
**UNIVERSIDAD PÚBLICA DE NAVARRA**  
**PAMPLONA**

**FECHA:** 01-06-15  
**FACT N°:** 1

**CLIENTE:** Presupuesto para la fabricación de 10 unidades del PLC  
**DIRECCION:** Hardware libre V1.  
**POBLACION:**

### RESUMEN DE PRESUPUESTO

CAP. DESCRIPCION	TOTAL CAPITULO
3 FABRICACIÓN PCB	315,30 €
4 COMPRA COMPONENTES MOUSER	537,68 €
9 COMPRA MODULOS ELECTRÓNICA	976,70 €
<b>TOTAL PRESUPUESTO EJECUCION MATERIAL</b>	<b>1.829,68 €</b>
Gastos Generales y Beneficio Industrial I.V.A.	
<b>TOTAL PRESUPUESTO DE CONTRATA</b>	<b>1.829,68 €</b>



**RAUL RIVERO LEO**  
**UNIVERSIDAD PUBLICA DE NAVARRA**  
**PAMPLONA**

---

Presupuesto **Presupuesto para la fabricación de 10 unidades del PLC de control**  
 10 unidades **Hardware libre V1.**  
 PLC Hardware

FECHA: **01-06-15**  
 FACT N°: **1**

**MEDICIONES Y PRESUPUESTO**

PARTIDA	UD	CONCEPTO	DETALLE DEL CONCEPTO				PRECIO	TOTAL
			CANT.	LARGO	ANCHO	ALTO		
<b>CAP. 1</b>		<b>DOCUMENTACIÓN PREVIA</b>						
1.1	Horas	Documentación funcionalidades PLC	30				30	
1.2	Horas	Documentación comunicaciones	15				15	
1.3	Horas	Documentación funcionamiento electrónica	20				20	
1.4	Horas	Testeo funcionamiento electrónica	30				30	
1.5	Horas	Aprendizaje programa DesignSpark	30				30	
		<b>SUBTOTAL</b>						
<b>CAP. 2</b>		<b>DISEÑO PCB</b>						
2.1	Horas	Creación de componentes librería DesignSpark	15				15	
2.2	Horas	Creación de esquemáticos	20				20	
2.3	Horas	Creación de layout	40				40	
		<b>SUBTOTAL</b>						
<b>CAP. 3</b>		<b>FABRICACIÓN PCB</b>						
3.1	ud	Fabricación PCB EuroCircuits	1				120,00	120,00
		<b>SUBTOTAL</b>						<b>120,00</b>
<b>CAP. 4</b>		<b>COMPRA COMPONENTES MOUSER</b>						
4.1	ud	Terminal fijo montaje PCB.	15			15	1,40	21,00
4.2	ud	Female Pin 2.54	10			10	0,20	2,00
4.3	ud	Male Pin 2.54	5			5	0,20	1,00
4.4	ud	IRF520N	12			12	1,50	18,00
4.5	ud	MCP3208	1			1	3,50	3,50
4.6	ud	CNY75C	18			18	0,70	12,60
4.7	ud	INA 125P	8			8	3,50	28,00
4.8	ud	Buzzer	1			1	3,50	3,50
4.9	ud	Resistencias varias	60			60	0,10	6,00
4.10	ud	Condensadores varios	18			18	0,10	1,80
4.11	ud	Separadores y Tornillos	10			10	0,20	2,00
		<b>SUBTOTAL</b>						<b>99,40</b>
<b>CAP. 5</b>		<b>MONTAJE PCB</b>						
5.1	Horas	Montaje componentes PCB	30				30	
		<b>SUBTOTAL</b>						
<b>CAP. 6</b>		<b>TESTEO FUNCIONAL</b>						
6.1	Horas	Testeo comunicaciones Raspberry - Arduino	20				20	
6.2	Horas	Testeo funcionamiento RTC	2				2	
6.3	Horas	Testeo funcionamiento Keypad	2				2	
6.4	Horas	Testeo funcionamiento LCD	4				4	
6.5	Horas	Testeo funcionamiento Reles	6				6	
6.6	Horas	Testeo funcionamiento Digital Input	8				8	
6.7	Horas	Testeo funcionamiento Digital Output	12				12	
6.8	Horas	Testeo funcionamiento Analog Input	4				4	
6.9	Horas	Testeo funcionamiento Current Sensor	15				15	
		<b>SUBTOTAL</b>						
<b>CAP. 7</b>		<b>PROGRAMACIÓN</b>						
7.1	Horas	Programación Raspberry PI	75				75	
7.2	Horas	Programación Web y MySQL	8				8	
7.3	Horas	Programación Arduino	25				25	
		<b>SUBTOTAL</b>						
<b>CAP. 8</b>		<b>DISEÑO CARCASA</b>						
8.1	Horas	DISEÑO CARCASA	12				12	
8.2	Horas	FABRICACIÓN CARCASA						
		<b>SUBTOTAL</b>						
<b>CAP. 9</b>		<b>COMPRA MODULOS ELECTRÓNICA</b>						
9.1	ud	Arduino Mega 2560	1			1	25,00	25,00
9.2	ud	Raspberry PI B+	1			1	20,67	20,67
9.3	ud	Reloj TinyRTC	1			1	2,00	2,00
9.4	ud	Pantalla LCD	1			1	20,00	20,00
9.5	ud	Keypad Membrana	1			1	3,00	3,00
9.6	ud	Array 8 Relay Module	1			1	7,00	7,00

**RAUL RIVERO LEO**  
**UNIVERSIDAD PUBLICA DE NAVARRA**  
**PAMPLONA**

---

Presupuesto: **Presupuesto para la fabricación de 10 unidades del PLC de control**  
 10 unidades: **Hardware libre V1.**  
 PLC Hardware

FECHA: **01-06-15**  
 FACT N°: **1**

**MEDICIONES Y PRESUPUESTO**

PARTIDA	UD	CONCEPTO	DETALLE DEL CONCEPTO				PRECIO	TOTAL
			CANT.	LARGO	ANCHO	ALTO		
9.7	ud	Fuente de Alimentación	1				20,00	20,00
<b>SUBTOTAL</b>								<b>97,67</b>

<b>TOTAL PRESUPUESTO EJECUCION MATERIAL</b>	<b>317,07</b>
Gastos Generales y Beneficio Industrial	
<b>TOTAL PRESUPUESTO EJECUCION CONTRATA</b>	<b>317,07</b>

**RAUL RIVERO LEO**  
**UNIVERSIDAD PUBLICA DE NAVARRA**  
**PAMPLONA**

---

Presupuesto **Presupuesto para la fabricación de 10 unidades del PLC de control**  
 10 unidades **Hardware libre V1.**  
 PLC Hardware

FECHA: **01-06-15**  
 FACT N°: **1**

**MEDICIONES Y PRESUPUESTO**

PARTIDA	UD	CONCEPTO	DETALLE DEL CONCEPTO						
			CANT.	LARGO	ANCHO	ALTO	PARCIAL	PRECIO	TOTAL
<b>CAP. 1</b>		<b>DOCUMENTACIÓN PREVIA</b>							
1.1	Horas	Documentación funcionalidades PLC	30				30		
1.2	Horas	Documentación comunicaciones	15				15		
1.3	Horas	Documentación funcionamiento electrónica	20				20		
1.4	Horas	Testeo funcionamiento electrónica	30				30		
1.5	Horas	Aprendizaje programa DesignSpark	30				30		
		<b>SUBTOTAL</b>							
<b>CAP. 2</b>		<b>DISEÑO PCB</b>							
2.1	Horas	Creación de componentes librería DesignSpark	15				15		
2.2	Horas	Creación de esquemáticos	20				20		
2.3	Horas	Creación de layout	40				40		
		<b>SUBTOTAL</b>							
<b>CAP. 3</b>		<b>FABRICACIÓN PCB</b>							
3.1	ud	Fabricación PCB EuroCircuits	10				10	31,53	315,30
		<b>SUBTOTAL</b>							<b>315,30</b>
<b>CAP. 4</b>		<b>COMPRA COMPONENTES MOUSER</b>							
4.1	ud	Terminal fijo montaje PCB.	30				30	0,88	26,43
4.2	ud	Female Pin 2.54	100				100	0,10	10,00
4.3	ud	Male Pin 2.54	50				50	0,10	5,00
4.4	ud	IRF520N	122				122	0,73	88,69
4.5	ud	MCP3208	10				10	3,30	33,00
4.6	ud	CNY75C	180				180	0,29	52,56
4.7	ud	INA126P	80				80	3,01	240,80
4.8	ud	Buzzer	10				10	3,22	32,20
4.9	ud	Resistencias varias	600				600	0,05	30,00
4.10	ud	Condensadores varios	180				180	0,05	9,00
4.11	ud	Separadores y Tornillos	100				100	0,10	10,00
		<b>SUBTOTAL</b>							<b>537,68</b>
<b>CAP. 5</b>		<b>MONTAJE PCB</b>							
5.1	Horas	Montaje componentes PCB	30				30		
		<b>SUBTOTAL</b>							
<b>CAP. 6</b>		<b>TESTEO FUNCIONAL</b>							
6.1	Horas	Testeo comunicaciones Raspberry - Arduino	20				20		
6.2	Horas	Testeo funcionamiento RTC	2				2		
6.3	Horas	Testeo funcionamiento Keypad	2				2		
6.4	Horas	Testeo funcionamiento LCD	4				4		
6.5	Horas	Testeo funcionamiento Reles	6				6		
6.6	Horas	Testeo funcionamiento Digital Input	8				8		
6.7	Horas	Testeo funcionamiento Digital Output	12				12		
6.8	Horas	Testeo funcionamiento Analog Input	4				4		
6.9	Horas	Testeo funcionamiento Current Sensor	15				15		
		<b>SUBTOTAL</b>							
<b>CAP. 7</b>		<b>PROGRAMACIÓN</b>							
7.1	Horas	Programación Raspberry PI	75				75		
7.2	Horas	Programación Web y MySQL	8				8		
7.3	Horas	Programación Arduino	25				25		
		<b>SUBTOTAL</b>							
<b>CAP. 8</b>		<b>DISEÑO CARCASA</b>							
8.1	Horas	DISEÑO CARCASA	12				12		
8.2	Horas	FABRICACIÓN CARCASA							
		<b>SUBTOTAL</b>							
<b>CAP. 9</b>		<b>COMPRA MODULOS ELECTRONICA</b>							
9.1	ud	Arduino Mega 2560	10				10	25,00	250,00
9.2	ud	Raspberry PI B+	10				10	20,67	206,70
9.3	ud	Reloj TinyRTC	10				10	2,00	20,00
9.4	ud	Pantalla LCD	10				10	20,00	200,00
9.5	ud	Keypad Membrana	10				10	3,00	30,00
9.6	ud	Array 8 Relay Module	10				10	7,00	70,00

**RAUL RIVERO LEO**  
**UNIVERSIDAD PUBLICA DE NAVARRA**  
**PAMPLONA**

Presupuesto 10 unidades PLC Hardware  
 Presupuesto para la fabricación de 10 unidades del PLC de control Hardware libre V1.

FECHA: 01-06-15  
 FACT N°: 1

**MEDICIONES Y PRESUPUESTO**

PARTIDA	UD	CONCEPTO	DETALLE DEL CONCEPTO				PRECIO	TOTAL	
			CANT.	LARGO	ANCHG	ALTO			PARCIAL
9.7	ud	Fuente de Alimentación	10				10	20,00	200,00
<b>SUBTOTAL</b>								<b>976,70</b>	

**TOTAL PRESUPUESTO EJECUCION MATERIAL** 1.829,68  
 Gastos Generales y Beneficio Industrial  
**TOTAL PRESUPUESTO EJECUCION CONTRATA** 1.829,68



## Anexo F: Listado BOOM de componentes.

Report Written : Sunday, May 31, 2015 12:51

E:\GoogleDrive\Proyectos\PFC\PFC\_PLC\_Arduino\1\_PCB\_Design

(1)\PCB\_PLC\ArduinoPLC\_16Junio2014\_01.prj

Design Path :

E:\GoogleDrive\Proyectos\PFC\PFC\_PLC\_Arduino\1\_PCB\_Design

(1)\PCB\_PLC\ArduinoPLC\_2014\_10\_15.pcb

Created : 15/10/2014 17:54:24

Last Saved : 17/05/2015 20:08:48

Units : mm (precision 2)

Bill of Materials generated with DesignSpark PCB - [www.DesignSpark.com/pcb](http://www.DesignSpark.com/pcb)

-----  
 "Ref Name","Qty","Component","Value","Package","Manufacturer","MPN","RS  
 Part Number","Other Part Number","Description"

```

"CONN7","1","Arduino_Mega_2560_Conector","","USER","","","",""
"BZ2","1","Buzzer (PCB)","","DIL","","",""
"C5","1","Capacitor (ceramic)","","DIL","","",""
"C6","1","Capacitor (ceramic)","","DIL","","",""
"C7","1","Capacitor (ceramic)","","DIL","","",""
"C8","1","Capacitor (ceramic)","","DIL","","",""
"C9","1","Capacitor (ceramic)","","DIL","","",""
"C10","1","Capacitor (ceramic)","","DIL","","",""
"C11","1","Capacitor (ceramic)","","DIL","","",""
"C12","1","Capacitor (ceramic)","","DIL","","",""
"C13","1","Capacitor (ceramic)","","DIL","","",""
"C14","1","Capacitor (ceramic)","","DIL","","",""
"C15","1","Capacitor (ceramic)","","DIL","","",""
"C16","1","Capacitor (ceramic)","","DIL","","",""
"C1","1","Capacitor (ceramic)","","DIL","","",""
"C2","1","Capacitor (ceramic)","","DIL","","",""
"C3","1","Capacitor (ceramic)","","DIL","","",""
"C4","1","Capacitor (ceramic)","","DIL","","",""
"C200","1","Capacitor (ceramic)","","DIL","","",""
"U4","1","CNY75GC DIP","","DIL","","",""
"U3","1","CNY75GC DIP","","DIL","","",""
"U2","1","CNY75GC DIP","","DIL","","",""
"U5","1","CNY75GC DIP","","DIL","","",""
"U6","1","CNY75GC DIP","","DIL","","",""
"U1","1","CNY75GC DIP","","DIL","","",""
"U7","1","CNY75GC DIP","","DIL","","",""
"U8","1","CNY75GC DIP","","DIL","","",""
"U10","1","CNY75GC DIP","","DIL","","",""
"U9","1","CNY75GC DIP","","DIL","","",""
"U12","1","CNY75GC DIP","","DIL","","",""
"U11","1","CNY75GC DIP","","DIL","","",""
"U20","1","CNY75GC DIP","","DIL","","",""
"U21","1","CNY75GC DIP","","DIL","","",""
"U22","1","CNY75GC DIP","","DIL","","",""
"U23","1","CNY75GC DIP","","DIL","",""
    
```

"U24","1","CNY75GC DIP",,","DIL",,"",,"",,"",,""  
 "U25","1","CNY75GC DIP",,","DIL",,"",,"",,"",,""  
 "CONN12","1","Conector\_Array\_Rele",,","DIL",,"",,"",,"",,""  
 "CONN8","1","Conector\_I2C\_LCD",,","DIL",,"",,"",,"",,""  
 "CONN10","1","Conector\_keyboard",,","DIL",,"",,"",,"",,""  
 "UA3","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA4","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA5","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA6","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA7","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA8","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA1","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "UA2","1","INA111\_DIP8",,","DIL",,"",,"",,"",,""  
 "U13","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U14","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U15","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U16","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U18","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U19","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U26","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U27","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U28","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U29","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U30","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "U31","1","IRF520NPBF",,","TO-220AB",,"International  
 Rectifier","IRF520NPBF",,"5411180",,","N-channel MOSFET,IRF520N 9.5A 100V"  
 "UB3","1","MCP3208 DIP16",,","DIL",,"",,"",,"",,""  
 "U17","1","MM74HC595",,","USER",,"",,"",,"",,""  
 "CONN9","1","Raspberry\_Conector\_Placa",,","USER",,"",,"",,"",,""  
 "R28","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R41","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R42","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R51","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R43","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R44","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R45","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R60","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R61","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R63","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R64","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"  
 "R65","1","Resistor","1K","DSC",,"",,"",,"",,"Resistor"





```
"R10","1","Resistor","R=INF","DSC","","","","Resistor"
"CONN11","1","RTC_arduino_DS1003","","USER","","",""
"J7","1","screw_terminal3","","USER","","",""
"J8","1","screw_terminal3","","USER","","",""
"J33","1","screw_terminal3","","USER","","",""
"J1","1","screw_terminal3","","USER","","",""
"J2","1","screw_terminal3","","USER","","",""
"J3","1","screw_terminal3","","USER","","",""
"J4","1","screw_terminal3","","USER","","",""
"J5","1","screw_terminal3","","USER","","",""
"J6","1","screw_terminal3","","USER","","",""
"J25","1","screw_terminal3","","USER","","",""
"J26","1","screw_terminal3","","USER","","",""
"J27","1","screw_terminal3","","USER","","",""
"J28","1","screw_terminal3","","USER","","",""
"J29","1","screw_terminal3","","USER","","",""
"J30","1","screw_terminal3","","USER","","",""
"J34","1","screw_terminal3","","USER","","",""
"J35","1","screw_terminal3","","USER","","",""
"J9","1","screw_terminal3","","USER","","",""
"J10","1","screw_terminal3","","USER","","",""
"J31","1","screw_terminal3","","USER","","",""
"J32","1","screw_terminal3","","USER","","",""
"J36","1","screw_terminal3","","USER","","",""
"J37","1","screw_terminal3","","USER","","",""
"J11","1","UPNA","","USER","","",""
```

## Anexo G: Cálculos Técnicos

El objetivo de este apartado es ofrecer algunos cálculos utilizamos para estimar el consumo energético así como el valor de algunos condensadores y resistencias.

Para la estimación de condensadores integrados para estabilizar la tensión de circuitos integrados se han utilizado condensadores de 1uF ya que permiten proporcionar una buena capacidad para estabilizar la tensión. Por otro lado actúa como filtro para estabilizar la tensión en la entrada del circuito integrado. En el caso de circuitos integrados más complejos se requeriría diferentes valores de condensadores para estabilizar la tensión en respuesta a diferentes frecuencias.

## ESTIMACIÓN DE CONSUMOS ELÉCTRICOS

Para el cálculo aproximado de potencia demandada en el circuito a plena potencia hemos estimado los siguientes cálculos:

Tabla 11 Estimación de potencias del sistema.

Raspberry Pi más periféricos.	5V 2 Amperios	10Watios
Arduino Mega	5V 1 Amperio	5 Watios
Activación de Salidas digitales 12V	X6 12 Voltios 0.1 Amperios	7.2 Watios
Activación de Salidas digitales 5V	X6 5 Voltios 0.2 Amperios	12.5 Watios

Estos cálculos suponen una estimación hipotética en la que la demanda de consumo energético por parte del sistema va a ser la máxima que se ha considerado que podría consumir cada sistema. No obstante el sistema no está diseñado para trabajar bajo estas condiciones, ya que al superar un rango de potencia el fabricante pide el uso de ventilación forzada para refrigerar la fuente de alimentación.

## CALCULO DE OPTOACOPLOADORES DE ENTRADAS DIGITALES

Para el correcto diseño de los optoacopladores se ha requerido el uso de resistencias para limitar el paso de la corriente por el circuito. El objetivo es asegurarnos de que nos encontramos bajo las condiciones de diseño y al mismo tiempo el sistema se conecta al recibir una señal externo. El modelo del optoacoplador es un CNY750C montaje DIP.

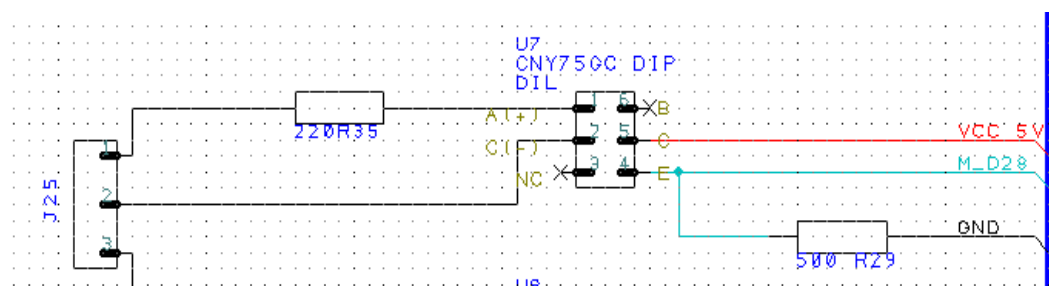


Figura 103 Calculo Resistencias optoacopladores de entradas digitales.

Las entradas digitales del sistema están preparadas para recibir estímulos entre 5 y 12 voltios. El sistema funciona mediante dos resistencias R35 y R29. El objetivo de la resistencia R29 es establecer una configuración “pull down” mientras no se recibe una señal, de esa forma el nivel lógico de la entrada digital se encuentra fijada ante perturbaciones eléctricas. Por otro lado la resistencia R35 limita el paso de corriente al optoacoplador asegurando el correcto funcionamiento al superar un cierto voltaje.

El diodo de optoacoplador posee una caída interna de 1.6 voltios, es por esa razón que la resistencia que limita la corriente deberá soportar como mínimo: una caída de voltaje de 3,4 Voltios al recibir una entrada a 5 Voltios y 10.4 Voltios al recibir entradas a 12 voltios. El sistema requiere de una corriente mínima de 10 mA para activar el fototransistor del optoacoplador. Con la resistencia R35 de 220 ohm permite el paso de 15 mA para las entradas a 5 Voltios. Por otro lado ante una entrada a 12 voltios el sistema conduce un máximo de 47 mA y disipando 500mW en la resistencia. Es por este último motivo que las resistencias montadas permiten la disipación de 500 mW de potencia.

## CALCULO DE OPTOACOPLADORES DE SALIDAS DIGITALES

Uno de los requisitos de desarrollo de las salidas digitales ha sido el cálculo de las resistencias para las salidas digitales.

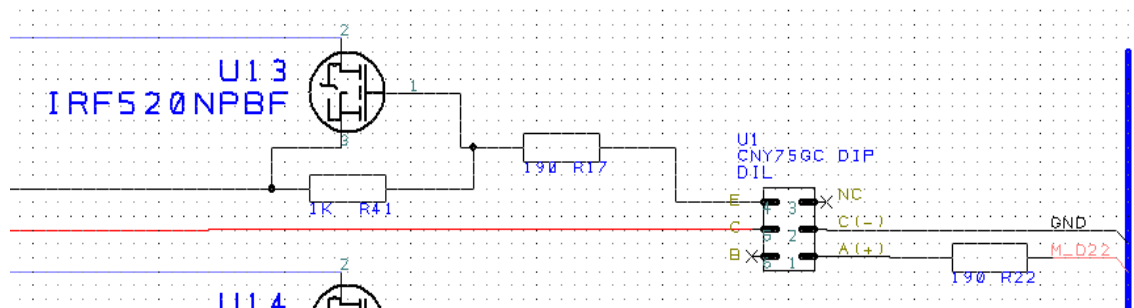


Figura 104 Cálculos de las resistencias de salidas digitales

En el diseño de las resistencias de las salidas digitales hay que modelar tres resistencias. La resistencia R22 permite limitar la corriente de activación del diodo del optoacoplador. Se ha elegido una resistencia de 190 ohm para forzar el paso de 18 mA por la salida digital, ya que la caída producida en el diodo del integrado es de 1.6 Voltios.

Por otro lado las resistencias R17 y R41 únicamente se encuentran como elemento de seguridad para mantener las tensiones dentro de unos niveles pero no son necesarias.

## Anexo H: Análisis de los PLC industriales.

Actualmente en el mercado existe una amplia gama de PLC industriales, las marcas más conocidas actualmente son Siemens, Schneider Electric junto con fácilmente 55 empresas más. Estas marcas son las que dan soluciones de automatización industriales a casi todas las fábricas del mundo.

En el sector industrial prima por encima del bajo coste la fiabilidad del producto así como el respaldo y la confianza que transmite un gran fabricante y detrás de un producto de estas características. Es un sector muy diferente al del consumo donde prima el precio entre otras características. Por estas razones en el mundo industrial no importa pagar miles de euros por una solución fiable, potente y de fácil integración en sus instalaciones y necesidades. Al mismo tiempo requieren una fácil programación de sus instalaciones y plena integración para el correcto funcionamiento.

En esto nuestro prototipo actualmente difiere de las características que tienen estos dispositivos. Nuestro producto busca lograr un precio más reducido que soluciones más completas, aunque será difícil lograr un precio menor que otras soluciones al ser el primer prototipo y sobretodo un coste de PCB superior al haber fabricado una sola unidad.

En este mercado de PLC industriales de alta complejidad técnica han empezado a aparecer proyectos de hardware libre orientado a PLC industriales. Junto con estos proyectos han surgidos otros que no tienen por qué seguir al filosofía del hardware libre pero que si hacen compatibles sus productos con la programación Arduino así como sus módulos. Esta compatibilidad democratiza los PLC industriales requiriendo un conocimiento menor para su puesta en marcha. Por regla general los proyectos que siguen esta filosofía tienen un precio menor que otros.

Por esa razón han surgido ya productos comerciales, como por ejemplo Industrial Shield, bieMme, Industruino... Por esa razón considero que este proyecto está a la par de las necesidades actuales en el mundo del hardware y software libre orientado a un cierto segmento industrial. Se han empezado a demandar en el entorno industrial productos compatibles con el entorno de Arduino, fáciles de programar usando el mismo lenguaje y de fácil ampliación. Esta gama de productos cobrará un gran protagonismo en mercados emergentes, y pequeña industria donde será fácil su implementación. Al mismo tiempo las grandes empresas empezarán a adoptar ideas y filosofías de este nuevo segmento de la electrónica.

Otro de los problemas o barreras entrada que tiene el sector de la automatización por parte de las grandes empresas, para hacer uso de sus productos se requiere el uso de carísimos paquetes de software únicamente compatibles con sus productos, siendo este software la única manera de poder programar los PLC. Generalmente las empresas proporcionan versiones Demo de su software para

facilitar el aprendizaje de sus entornos de programación. Generalmente este tipo de paquetes de software tienen un elevado precio que van desde 2.000 € a superar los 10.000 €.

Para la correcta integración de los PLC industriales y su uso en grandes líneas de producción se requiere una arquitectura modular que permita que el bloque principal de control o CPU con una serie de funcionalidades básicas de entradas y salidas permita integrarse mediante un BUS de comunicación a una serie de módulos auxiliares o expansores de puertos de esta forma se pueden añadir nuevas entradas y salidas o comunicaciones.

Generalmente los fabricantes distribuyen módulos separados que engloban diferentes tareas, como por ejemplo uno que incluya comunicación Ethernet, ModBus, CAN y RS-485 en un mismo bloque u otras que incluyan conexiones para sondas de temperaturas o RTD para monitorización de hornos.

Para el funcionamiento y comunicación entre el módulo principal y los módulos auxiliares conectados en un mismo carril DIN es necesario compartir un protocolo de comunicación. Generalmente las empresas desarrollan sus propios protocolos de comunicación generando nuevos estándares propios para su empresa, el objetivo del desarrollo de estos protocolos es proteger la compatibilidad e ingeniería inversa sobre sus productos.

Los protocolos de comunicación tradicionales como I2C, SPI generalmente son poco transparentes para un usuario normal por lo cual no es clara su información salvo que se utilicen herramientas especiales, por otro lado es frecuente que estén limitados a un determinado número de dispositivos conectados. Las líneas serie por otro lado son completamente transparentes para el usuario normal pero muy ineficientes en la transmisión de datos.

Generalmente los sistemas de comunicación tradicionales I2C, SPI, son más adecuadas para la comunicación interna entre electrónica. No son idóneos para transferir directamente información al requerir un mayor esfuerzo de codificación y decodificación de las señales aunque implementando unas buenas rutinas de programación y asegurando la sincronización entre dispositivos puede funcionar perfectamente.

## Siemens

Siemens es uno de los más fabricantes de PLC más conocidos con mayor tradición en la automatización. Siemens dispone de una amplia gama de productos orientados a la automatización, desde pequeñas instalaciones hasta las grandes fábricas industriales.

Para ello se ofertan diferentes modelos de diferentes tamaños y capacidades entre ellos está el modelo más básico actualmente LOGO, el cual es el que más se asemeja a nuestro prototipo en cuanto al sector de aplicación.

Entrando en revisión a los PLC es Siemens tenemos la familia más característica la ST 70, su modelo LOGO dispone entre sus características más notables es el funcionamiento a tensiones industriales estándar 12 o 24 V. Cuatro entradas digitales y cuatro entradas analógicas. Por otro lado dispone de 4 salidas de relé. En comparación con nuestro prototipo este modelo dispone de menos conectividad que nuestro prototipo así como menos funcionalidades. Como ventajas es que dispone de un tamaño más compacto así como protección IP contra agentes dañinos a nivel industrial. El uso de componentes electrónicos de alta calidad industrial aseguran su correcto funcionamiento a rangos entre -40 °C y 85 °C. Pero seguramente su gran ventaja sería el apartado de software con una gran y potente integración con el resto de la planta gracias al software disponible por parte de Siemens.

Éstos PLC que se clasificación como mini PLC se usan para pequeñas áreas de producción para automatizar determinadas tareas o procesos industriales que generalmente van a estar aislados del resto del área de producción.



Figura 105 Imagen de un PLC siemens Logo

Existen soluciones más profesionales y amplias por parte de Siemens como por ejemplo los simatic s7-200, este dispositivo PLC proporciona una solución compacta y súper potente con una gran potencialidad a nivel software. En el Hardware dispone de memoria interna para la gestión del programa y un máximo de 40 entradas y 38 salidas digitales así como máximo 8 entradas analógicas. Requiere de la conexión de módulos entre si comunicados mediante buses para lograr está máxima

conectividad. Según el modelo tendrá un modo u otro de conectividad. Los sistemas de comunicación más típicos son mediante Ethernet, Zigbee, GPRS, CANbus o algún otro BUS propietario.



Figura 106 Imagen de un PLC siemens S7-200

Por parte de Siemens también ha surgido soluciones modulares especializadas en ciertos sectores de la industria como puede ser en la industria manufacturera donde tratan de solucionar problemas típicos de este área de esta forma se logra en un mínimo espacio el máximo de conexiones posibles son segmentos bastante especializados en la automatización.

Aunque tradicionalmente no se utilizan ordenadores como PLC, surgen soluciones híbridas entre un PC tradicional y un PLC. Se trata de dispositivos con altas exigencias y estándares industriales para su resistencia a condiciones del ámbito industrial como calor, polvo o gases siendo capaces de trabajar de forma ininterrumpida. Es más propio de áreas de desarrollo y laboratorios.

## SCHNEIDER ELECTRIC.

Una de las ventajas de Schneider Electric es que disponen una amplia gama de controladores lógicos PLC, de pequeño tamaño y fácilmente ampliables entre sí tienen desde pequeños autómatas para controlar un determinado número de entradas y salidas, hasta el control de grandes autómatas programables.

La marca es Schneider Electric proporciona soluciones como su ya comentada familia Modicon, como por ejemplo el modelo M238, el cual es una controladora programable para máquinas que permite controlar desde 20 entradas en su modelo básico, hasta un número máximo de 228 entradas según el número de módulos expansores instalados a través de su bus.



Figura 107 Imagen de un PLC Schneider M238, actual Modicon.

Es Schneider Electric en cuanto a soluciones PLC más grandes tenemos el Modicon TM5 el cual es un sistema modular de hasta 2400 entradas y salidas, que permite el control de señales digitales y analógicas. Es una solución muy compacta y de gran conectividad.



Figura 108 PLC Schneider completo mostrando su sistema de ampliación.

Las soluciones más profesionales se dedican al control directo de autómatas, robots o aplicaciones muy específicas para entornos muy concretos.





# Anexo I: Análisis de los sistemas de comunicación Industriales

## SISTEMAS DE COMUNICACIÓN CABLEADAS

### Ethernet

Los PLC industriales generalmente se comunican con el resto del sistema haciendo uso de la interfaz Ethernet para tener acceso a la red local del sistema. Esto permite comunicarse con otros PLC y servidores conectados a la misma red.

Este sistema de conectividad es uno de los más potentes ya que permite acceso total a la información del sistema así como el mantenimiento de la red. Actualmente toda fábrica industrial dispone de redes cableadas para su control, por lo que es fácilmente la solución más extendida actualmente. Son soluciones poco costosas y fáciles de implementar.

Una de las debilidades que tiene este sistema es el mantenimiento y planificación de la seguridad del sistema como evitar la apertura de la red al exterior para evitar posibles intrusiones de seguridad que pongan en peligro la integridad del sistema.

Generalmente estos dispositivos van integrados en el PLC principal o son añadidos mediante tarjetas de comunicaciones Ethernet extras. El conector que usa estos dispositivos es un RJ45 de 8 hilos.

Teniendo una conexión Ethernet instalada hace falta montar sobre ella el protocolo de comunicación correspondiente, siendo entre ellas la más común la TCP/IP típica de internet.

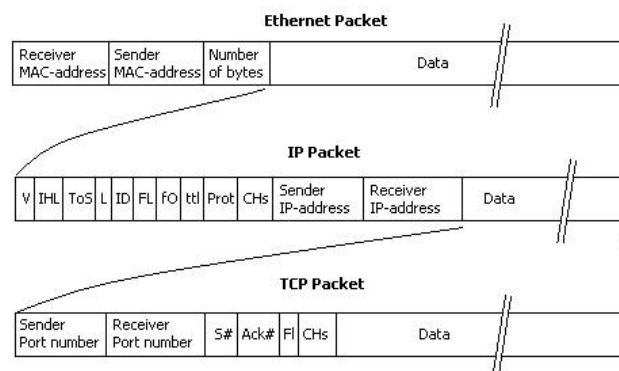


Figura 109 Descripción de un paquete Ethernet.

Habiendo establecida una comunicación Ethernet los PLC pueden usar instrucciones propias como SEND y RECEIVE para facilitar el intercambio de datos entre los autómatas. Para lograr todo este funcionamiento hay que configurar las direcciones IP, máscaras de la subred y otras configuraciones más.

Esta forma de comunicación dispone de protocolos propios similares a la comunicación http, usando sus propios comandos al estilo GET, POST, RECEIVE, típicos del protocolo http. Estos protocolos están especializados en comunicarse entre dispositivos PLC.

### ***CANBus***

El CANBus es un protocolo de comunicación desarrollado por la empresa Bosch en 1982, se basa en un sistema de comunicación distribuida. Este sistema de comunicación permite la comunicación directa entre múltiples CPUs.

Entre las ventajas de ese protocolo es el ser un sistema normalizado por lo que simplifica la comunicación entre sistemas que compartan la misma red.

En este sistema de comunicaciones el procesador principal cede el control de la comunicación a otros dispositivos para que delegan las tareas de gestión de la red según necesidades. Al compartir un mismo Bus reduce de forma considerable el cableado.

El protocolo CAN está orientado al envío de mensaje, tramas de información que se descomponen en bloques y se les asigna un identificador para ser enviados a la red. Cada mensaje está inequívocamente identificado de tal forma que solo los nodos destino intercepten ese mensaje para descodificarlo.

Es un robusto sistema multimaestro muy flexible y con bajos tiempos de latencia. Este protocolo de comunicación está orientado a automóviles en su origen que cumple con la certificación ISO.

El uso de este protocolo de comunicación en los PLC permite crear redes CANbus y la comunicación entre PLC o periféricos de expansión. Es una buena solución de comunicación en redes amplias donde se requiere unos mínimos tiempos de latencia en la recepción de la información. Se ha ido implantando en las soluciones más completas entre los PLC con el paso de los años, al mismo tiempo han salido varias variantes especializadas para el entorno industrial.

### ***Modbus***

Modbus es otro de los protocolos de comunicación existente en la actualidad para la comunicación entre PLC, es un sistema de comunicación con arquitectura Master/Slave diseñado en 1979 por Modicon para la comunicación entre PLC. Se ha convertido en uno de los protocolos estándar en la industria. Es un protocolo prácticamente disponible en todos los dispositivos electrónicos industriales.

Las razones de su éxito se deben a ser un protocolo de libre uso, con fácil implementación y coste de desarrollo y capaz de manejar amplios bloques de datos. Gracias a estas capacidades puede

controlar una amplia red de dispositivos, comunicar los resultados a un ordenador y comunicar a sistemas de supervisión de adquisición de datos. (SCADA). Este protocolo de comunicación tiene variantes para ser utilizados mediante puertos serie y Ethernet.

Existe una versión mejorada y cerrada del protocolo propio de los Modicon de Schneider Electric que se llama Modbus Plus que mejora la velocidad del procesamiento de la información. En este sistema cada dispositivo tiene su propia dirección única al estilo I2C pero logrando mayor direccionamiento de dispositivos.

Según la variante del protocolo la información transmitida es codificada no transparente para un humano o en ASCII transparente para un usuario.

### Línea serie

La comunicación de datos mediante puerto serie aunque no es propiamente un protocolo si no una interfaz de comunicaciones es ampliamente usada en computadoras y periféricos enviando la información bit a bit frente a un puerto paralelo donde se envía un byte de forma simultánea mediante otros cables.

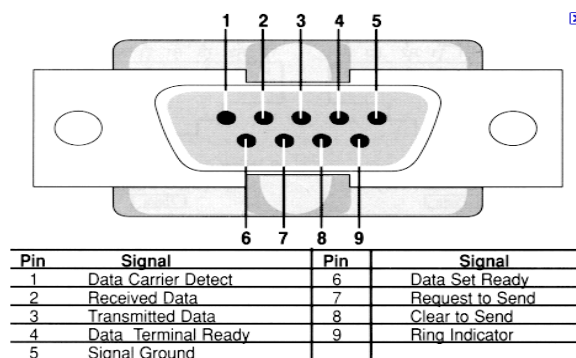


Figura 110 Descripción pines de un puerto serie.

El puerto Serie está formado por 9 cables entre los que están GND como referencia común de voltaje y TX y RX como líneas de transmisión de datos junto con una serie de pines adicionales para entender en qué momento se deben iniciar las comunicaciones.

El estándar más común es el **RS-232** diseñado para las comunicaciones que utiliza voltajes de puerto entre -12 y 12 voltios para la comunicación según la línea. Frente a la comunicación RS-232 más tradicionalmente ha sido usado en la industria el estándar RS-422 que permite conexiones cableadas con un límite físico de 1200 metros de cable. Esta variante permite mayores velocidades de transmisión adicionalmente trabaja en otro rango de voltajes de -6 a 6 Voltios. Es característico que usa comunicación Full Duplex es decir usa como líneas de transmisión **TX+**, **TX-**, **RX+**, **RX-** sin usar un cable GND de vuelta de corriente, por ello es un protocolo de comunicación diferencial.

Bajo esta arquitectura se pueden estructurar numerosos protocolos de comunicación para lograr un sistema robusto. De todas formas esta interfaz por razones de espacio y la aparición de mejores interfaces ha tendido a desaparecer en favor de por ejemplo de las interfaces USB más compactas y capaces de proporcionar alimentación y lograr altas velocidades. Generalmente este tipo de comunicaciones son temporales para realizar diagnosis y programación de los terminales correspondientes.

### ***Protocolos de comunicación propietarios.***

Es frecuente que los diferentes fabricantes intenten proteger sus entornos usando sus propios protocolos propietarios sin tener en cuenta la opinión del resto del ecosistema industrial, esto puede hacer que se compliquen las comunicaciones entre diferentes dispositivos, es por ello que se implementó la Tecnología OPC, como forma de interconectar los diferentes protocolos de comunicación convención a menudo propietarios.

La tecnología OPC elimina la necesidad de disponer de drives especializados para poder comunicar los diferentes protocolos existentes en la actualidad. Los conectores OPC intermedios reducen la carga del sistema y logran la máxima compatibilidad entre equipos convirtiendo toda la información a un lenguaje común. Este tipo de soluciones reducen el tiempo de desarrollo e implementación de las diferentes soluciones.

En resumidas cuentas el sistema de comunicaciones pasa a través de una única interfaz cumpliendo una serie de estándares. Lo cual permite obtener la información sin necesidad de entender los diferentes protocolos de comunicación nativos.

Para el correcto funcionamiento el sistema intercala un Cliente y Servidor para realizar todas las tareas de comunicación.

## SISTEMAS DE COMUNICACIÓN INALÁMBRICA.

### Wifi

El Wifi es uno de los medios de comunicación más extendidos en la actualidad, todos los hogares y trabajos disponen de una red Wifi, al trabajar sobre una red de 2.4 GHz y por las características de esta frecuencia que la hace altamente vulnerable frente a todo tipo de obstáculos presentes en el entorno no lo hacen una solución suficientemente robusta y fiable como para poder ser usada indiscriminadamente en la industria.



Figura 111 Logotipo Wifi

Si bien existen PLC disponibles con esta tecnología pero que requieren su uso junto con otros sistemas de comunicación para asegurar el perfecto funcionamiento. Es una solución adecuada para pequeños entornos industriales que no impliquen instalaciones grandes ni complejos obstáculos llenos de elementos metálicos.



Figura 112 Ejemplo de PLC con Wifi integrado.

## Zigbee

Por otro lado en la tecnología de las comunicaciones industriales es mucho más frecuente otro estándar formado por la tecnología ZibBee, el cual es un conjunto de protocolos de comunicación inalámbrica que se utilizan en sistemas de radiodifusión de bajo consumo energético. Cumple el estándar IEEE 802.15.1 de redes inalámbricas. Permite lograr una buena seguridad en las comunicaciones así como un consumo energético bajo.

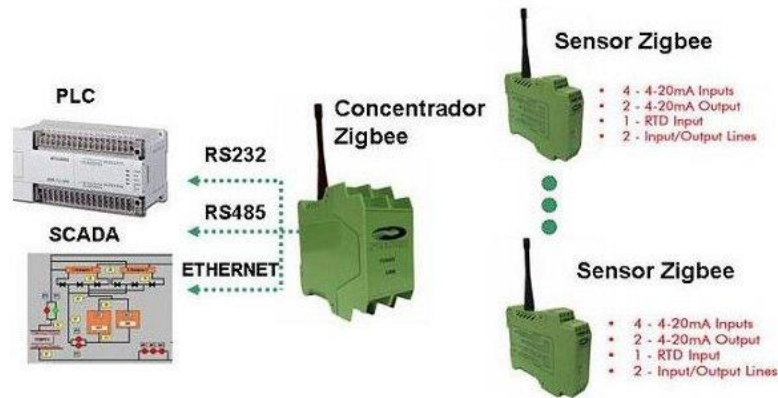


Figura 113 Diagrama de conexión de Zigbee en un entorno industrial.

Este tipo de tecnologías permiten construir una estructura mallada MESH, con una fácil integración de sus elementos. Esta tecnología dispone una banda ISM especial para usos industriales y científicos que depende del país de aplicación. En el caso de Europa la banda disponible es de 868 MHz y 915 MHz en estados unidos. Al mismo tiempo está disponible en todo el mundo la banda de 2.4 GHz. La ventaja de la banda de 868 MHz es que dispone de qué sufre menos apantallamiento por el terreno y los elementos estructurales, permitiendo un alcance de radiocomunicaciones mayor.

Entre las ventajas de una red ZigBee es ser capaz de trabajar con un máximo de 65535 nodos distribuidos en subredes de 255 nodos. Al mismo tiempo dispone de menor consumo energético que el Bluetooth tradicional.

En una red ZibBee de tipo Mesh cada dispositivo realiza un papel permitiendo que la información se distribuya módulo a módulo permitiendo logrando distancias mucho mayores que la solución alternativa de colocar dos únicos dispositivos. Las labores que pueden realizar los dispositivos son las de Coordinador para las labores de gestión y control de la red. El papel de Router permite hacer un puente de interconexión entre otras tecnologías y usuarios. El último papel disponible es el de “End Device” que gestiona la velocidad de los sensores correspondiente para posteriormente comunicarse con los routers o coordinadores.

Con el tiempo han salido estándares propietarios que combinan mejorar sobre la red tradicional ZigBee mejorando los protocolos y bloqueando el acceso al usar protocolos de comunicaciones diferentes. Es el ejemplo de Xbee.

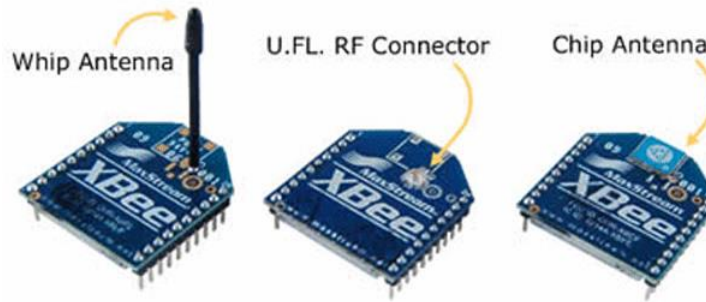


Figura 114 Emisores Zigbee con diferentes configuraciones de antena.

En su aplicación a la industria permite crear redes de sensores inalámbricos, de bajo coste y consumo energético que permite la monitorización y control bajo potentes soluciones. Se usa principalmente en entornos a los que no pueden acceder soluciones cableadas y van a estar sometidos a condiciones especiales. Es muy típica la interconexión de PLC con trabajadores o máquinas móviles que debido a sus características no son soluciones adecuadas para cablearlas.

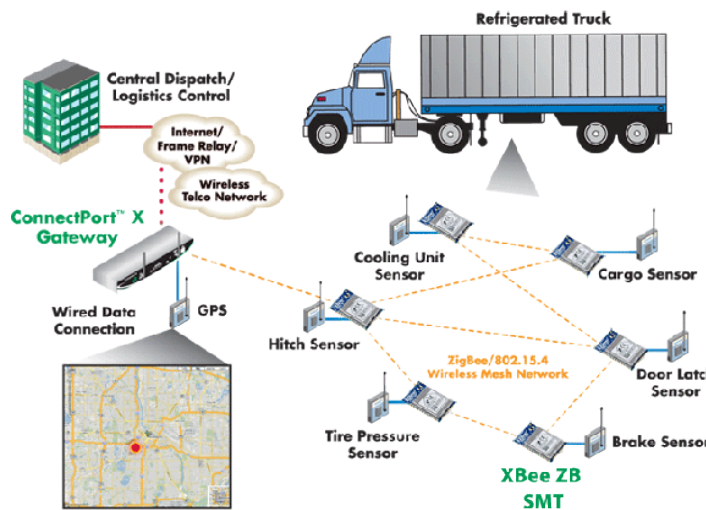


Figura 115 Sistema de conexión mallada de sensores.



## GPRS

Entre otra de los sistemas de comunicación empleamos en los PLC que resultan más útiles sobretodo en el caso de instalaciones aisladas o que requieran un sistema de respaldo en caso de fallo de las comunicaciones tenemos los dispositivos PLC que incluyen en su dispositivo comunicaciones GPRS mediante un Modem con dicha tecnología. En el diseño de estos dispositivos requiere que estén adaptados al país destino, o en su lugar usar tecnología cuatribanda para poder ser compatibles en cualquier lugar del mundo.



Figura 116 GPRS GL865-Dual de Telit

Se utilizan las redes de telefonía omnipresentes en todos los países para desarrollar el sistema de comunicaciones para permitir el intercambio de datos de forma bidireccional. Esta tecnología permite un sistema de respaldo en caso de caída de las comunicaciones, al mismo tiempo permite lanzar avisos y responder en este tipo de casos.



Figura 117 PLC con GPRS integrado.

Las soluciones de comunicación basadas en GPRS generalmente no implican la transmisión de una gran cantidad de datos como se puede utilizar en la telefonía de consumo, sus aplicaciones suelen ser el envío y recepción de datos puntuales. De esta forma se enmarca lo que se conoce como tecnología M2M o comunicación máquina a máquina. Esta forma de comunicación entre máquinas ha aparecido hace poco pero será uno de los sistemas que más va a crecer en volumen de usuarios

### ***Bluetooth Low Energy***

Entre las tecnologías de comunicación inalámbrica tenemos al tecnología Bluetooth, la cual lleva desde el año 2000 cuando fue lanzada su primera versión, con el paso de los años este protocolo ha ido evolucionando hasta llegar a la versión 4.0 o Bluetooth Low Energy BLE. Este protocolo de comunicación se caracteriza por unos bajos consumos energéticos un alcance relativamente pequeño comparado con el resto de soluciones inalámbricas disponibles. Está tecnología utiliza la banda de 2.4 GHz.



Figura 118 Logotipo de Bluetooth 4.0 Low Energy.

Tradicionalmente esta tecnología ha sido empleada en electrónica de consumo por sus características y bajo precio. De todos modos tiene zona de aplicación en la industria para la conexión de redes de sensores inalámbricos que permitan la comunicación bidireccional formando redes de sensores.

Una gran mayoría de aplicaciones de comunicación industrial requieren escasas distancias de comunicación que permite que esta tecnología sea óptima a estas distancias. Es un protocolo robusto, de bajo consumo energético y con amplias posibilidades.

En la actualidad esta tecnología empieza a surgir a mano de algunas empresas que aprovechan el potencial de esta tecnología para aplicarlo a las comunicaciones.



Figura 119 Circuito integrado de Texas Instrument apto para el Bluetooth 4.0



## ANEXO J: LISTA DE MATERIAL CONTENIDO EN EL DVD

Con el proyecto se entrega un único DVD con el siguiente contenido:

1. Carpeta **Memoría**
  - a. Memoria del proyecto en PDF (Este documento)
  - b. Carpeta Adicional con contenidos extras.
2. Carpeta **Diseño Hardware**
  - a. Carpeta **PCB Proyecto**
    - i. **Esquemáticos DesignSpark 6.1**
    - ii. **Diseño Layout DesignSpark 6.1**
  - b. Carpeta **Testeador Corriente**
    - i. **Esquemáticos DesignSpark 6.1**
    - ii. **Diseño Layout DesignSpark 6.1**
  - c. Carpeta **Sensor Corriente**
    - i. **Diseño Orcad 16.6**
      1. **Esquemáticos**
      2. **Diseño Layout**
    - ii. **Diseño DesignSpark 6.1**
      1. **Esquemáticos**
      2. **Diseño Layout**
  - d. Carpeta Documentación extra
3. Carpeta **Diseño Software**
  - a. Carpeta **Software Arduino**
    - i. **Código principal**
    - ii. **Librerías Arduino**
  - b. Carpeta **Software Raspberry Pi**
    - i. **Código principal Raspberry**
    - ii. **Librerías internas**
    - iii. **Librerías externas**
    - iv. **Imagen Linux comprimida**
4. Carpeta **Datasheet componentes**
  - a. **Datasheet Raspberry Pi**
  - b. **Datasheet Arduino Mega 2560**
  - c. **Datasheet Pantalla LCD Keypad**
  - d. **Datasheet TinyRTC**
  - e. **Datasheet Fuente de alimentación**
  - f. **Datasheet Circuitos integrados**



## BIBLIOGRAFÍA

- Álvarez, Miguel Angel. *Acceder a MySQL desde terminal en Linux*. s.f. <http://www.desarrolloweb.com/articulos/mysql-terminal-linux.html>.
- Arduino. *Arduino Mega 2560*. s.f. <http://www.arduino.cc/en/Main/arduinoBoardMega2560>.
- . *Mega 2560 Mega PinOut Diagram*. s.f. <http://www.arduteka.com/wp-content/uploads/2013/02/ArduinoMega.png>.
- Casas, Eduardo. *Comenzando con MySQL Connector/C++*. s.f. <http://www.eduardocasas.com/es/blog/27-02-2013/comenzando-con-mysql-connector->.
- Compare, PLC. *Research and Compare PLC*. s.f. <http://plccompare.com/>.
- Configurar servidor Apache para que nos muestre los errores de nuestro PHP*. s.f. <http://donnierock.com/2013/06/06/configurar-servidor-apache-para-que-nos-muestre-los-errores-de-nuestro-php/>.
- Current Loop Tester*. s.f. <http://www.discovercircuits.com/PDF-FILES/CURRENTSOURCE1.pdf>.
- DesdeLinux. *Acceder a MySQL usando C*. s.f. <http://blog.desdelinux.net/acceder-a-mysql-usando-c/>.
- DFrobot. *RTC DS1307*. s.f. [http://www.dfrobot.com/wiki/index.php/Real\\_Time\\_Clock\\_Module\\_\(DS1307\)\\_V1.1\\_\(SKU:DFR0151\)](http://www.dfrobot.com/wiki/index.php/Real_Time_Clock_Module_(DS1307)_V1.1_(SKU:DFR0151)).
- Fitzpatrick, Dennis. *Analog Design and Simulation using OrCAD Capture and PSpice*. Elsevier / Newnes, 2011.
- Hardware, Open. *Tutorial para conectar un arduino con raspberry pi por puerto serie*. s.f. <http://openhardware.pe/tutorial-para-conectar-un-arduino-con-raspberry-pi-por-puerto-serie/>.
- Insertar registros en MySQL*. s.f. <http://blog.unijimpe.net/insertar-registros-en-mysql/>.
- Instalación y configuración de PHPMyAdmin*. s.f. [http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/instalacin\\_y\\_configuracin\\_de\\_phpmyadmin.html](http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m5/instalacin_y_configuracin_de_phpmyadmin.html).
- Instrument, Texas. *Datasheet CD4051B*. s.f. <http://www.ti.com/lit/ds/symlink/cd4052b.pdf>.
- . *Datasheet CD74HC4067*. s.f. <http://www.ti.com/lit/ds/symlink/cd74hc4067.pdf>.
- . *INA 126*. s.f. <http://www.ti.com/lit/ds/symlink/ina126.pdf>.

King, K. N. *C PROGRAMMING*. Second Edition. New York: NORTON, 2008.

Labview, Proyectos Arduino y. *Configurando adecuadamente el módulo Tiny RTC*. s.f.  
<http://proyectosarduinoylabview.blogspot.com.es/2013/10/configurando-adecuadamente-el-modulo.html>.

Luauf. *¿Cómo conectarse a mySQL desde un programa en C?* s.f.  
<http://luauf.com/2011/07/09/%C2%BFcomo-conectarse-a-mysql-desde-un-programa-en-c/>.

Malik, Norbert R. *CIRCUITOS ELECTRÓNICOS*. Madrid: PEARSON, 2003.

Margolis, Michael. *Arduino Cookbook*. 2nd Edition. O'Reilly Media, 2011.

McCracken, Scott. *Curso de programación Web*. Barcelona: INFORBOOKS'S S. L. , 2011.

Mitzner, Kraig. *Complete PCB Design Using OrCad Capture and Layout*. Elsevier / Newnes, 2011.

Monk, Simon. *Raspberry Pi Cookbook*. O'Reilly Media, 2013.

MySQL. *20.6 MySQL C API*. s.f. <http://dev.mysql.com/doc/refman/5.0/en/c-api.html>.

Php5dp. *Get MySql and PHP to Work Together in Raspberry Pi*. s.f. <http://www.php5dp.com/get-mysql-and-php-to-work-together-in-raspberry-pi/>.

PhpBasico. *Curso de PHP básico*. s.f. [http://phpbasico.freevar.com/temas/php14\\_4.php](http://phpbasico.freevar.com/temas/php14_4.php).

ROHAUT, Sebastien. *Linux, Preparación para la certificación LPIC-1*. Barcelona: Ediciones ENI, 2012.

RS. *Designspark PCB Educational*. s.f. <http://www.rs-online.com/designspark/electronics/eng/page/designspark-pcb-educational>.

*Tutorial – Install PhpMyAdmin on your Raspberry Pi*. s.f.  
<http://www.raspipress.com/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/>.

ZetCode. *MySQL C API programming tutorial*. s.f. <http://zetcode.com/db/mysqlc/>.

## REFERENCIAS

<sup>1</sup> Controlador lógico Programable

<sup>2</sup> Raspberry Pi es un microordenador de bajo coste surgido inicialmente de fines educativos. Ha generado una gran comunidad de aprendizaje detrás suya.

<sup>3</sup> Se conoce como Maker a un movimiento surgido recientemente que sigue la filosofía de fabricar las cosas por uno mismo. También se conoce como “DIY” “Do It Yourself” hazlo por ti mismo.

<sup>4</sup> Arduino es un Microcontrolador diseñado para facilitar las labores de prototipado de diseños y facilitar el aprendizaje de electrónica. Se encuentra respaldado por una gran comunidad de usuarios y diseñadores. Pertenece al Hardware libre.

<sup>5</sup> Un ARM es un microordenador con una gran potencia y versatilidad que se utiliza actualmente en todo tipo de sistemas embebidos. Los teléfonos móviles es un ejemplo de diseño que integra un procesador ARM.

<sup>6</sup> Las interfaces de comunicación I2C es un tipo de comunicación ampliamente utilizada en la electrónica para el control de dispositivos. Se puede ver la información ampliada en la página de [Interfaz I2C](#).

<sup>7</sup> GPRS: General Packet Radio Service. Servicio General de Paquetes vía Radio.

<sup>8</sup> Zigbee: Conjunto de protocolos de alto nivel de comunicación inalámbrica. Se utiliza para la comunicación de bajo consumo en diferentes áreas.

<sup>9</sup> MySQL: Es un sistema de gestión de bases de datos. Como característica del sistema de gestión es su pertenencia al software libre. Actualmente es uno de los sistemas de gestión de bases de datos más utilizado.

<sup>10</sup> Ladder: Este lenguaje de programación es uno de los lenguajes de programación de PLC estándar. Su nomenclatura procede de la norma IEC 61131-3.

<sup>11</sup> GPS: Sistema de posicionamiento global.

<sup>12</sup> Modbus: Es un protocolo de comunicación diseñado para los primeros PLC Modicon. Actualmente se ha convertido en un estándar en la industrial.

<sup>13</sup> IoT: “Internet of Things” También conocido como internet de las cosas, consiste en la conexión de internet a todo tipo de dispositivos.

<sup>14</sup> RTC: Reloj en tiempo real. Es un dispositivo integrado que mantiene la hora actual del dispositivo gracias al uso de un cristal de cuarzo y una pila.

<sup>15</sup> UART: Son las siglas del inglés “Universal Asynchronous Receiver-Transmitter”. Es el dispositivo que controla los puertos y dispositivos serie.

<sup>16</sup> SoC: “System on Chip” También llamado microprocesador.

<sup>17</sup> GPU: “Graphics Processing Unit” Unidad de procesamiento gráfico.

<sup>18</sup> RAM: “Random-Access Memory” Memoria de acceso aleatorio. Es la Memoria principal para la gestión del sistema.

<sup>19</sup> GPIO: “General Purpose Input/Output” Entradas / Salidas de Propósito General. Conexiones eléctricas configurables de un circuito integrado.

<sup>20</sup> Shifter Register con Latch: Cambiador de registro con Latch mantenido, es un dispositivo que permite la ampliación de puertos de conexión. La característica de Latch mantenido permite dejar fijadas los estados eléctricos.

<sup>21</sup> SPI: “Serial Peripheral Interface” es un estándar de comunicaciones que se utiliza para la transferencia de información entre circuitos integrados. Se puede ver la información ampliada en la página de [Interfaz SPI](#).

<sup>22</sup> ST: STMicroelectronics: Fabricante de microcontroladores y circuitos integrados.

<sup>23</sup> Arduino IDE: Interfaz de programación de Arduino.

<sup>24</sup> Puerto COM: Similar a un puerto Serie.

<sup>25</sup> AVR Dragon: Herramienta de programación y Debug de ATMEL:

<sup>26</sup> Notepad++: Editor de textos especializado en el uso como herramienta de programación.



---

<sup>27</sup> Clock Stretching: Características de las Interfaces I2C que se usa para permitir la conexión simultánea de sistemas digitales de alta velocidad con otros dispositivos que trabajan a velocidad inferiores.

<sup>28</sup> Sparkfun: Es un distribuidor y diseñador de electrónica de internet. Sus productos se encuentran especializados al pequeño consumidor de electrónica.

<sup>29</sup> Bps: Baudios por segundo. Velocidad de comunicación.

<sup>30</sup> SMT: “Surface Mount Technology” Tecnología de montaje superficial.

<sup>31</sup> phpMyAdmin: Es una completa interfaz de control y mantenimiento de una base de datos MySQL.