

Implementación de un analizador de redes con Arduino



Grado en Ingeniería Eléctrica y Electrónica

Trabajo Fin de Grado

Autor: Barbáchano Cirión, Angel

Tutor: Andueza Unanua, Angel María

Pamplona, 3 de Mayo de 2018

Resumen

El objetivo de este proyecto es el diseño y construcción de un analizador de redes monofásicas para uso doméstico, capaz de medir parámetros como la corriente, tensión y factor de potencia; y partiendo de estos calcular la potencia y energía, de manera que se pueda supervisar totalmente los datos de la red doméstica, con la finalidad de poder gestionar la energía consumida y la generada.

El dispositivo está basado en un controlador de *hardware libre*, como es Arduino, construyéndose una PCB tipo *shield*, que se encargará de acondicionar las señales analógicas de entrada, dar conectividad a los periféricos, y proporcionar tensión de alimentación al controlador. Finalmente el sistema de control se programará empleando un *software de código abierto*.

Además de monitorizar los parámetros de la red mediante una pantalla local, el prototipo se configurará para que funcione como servidor web y así exportar los datos a la nube. De esta manera, desde cualquier punto de conexión a Internet, ya sea una IP privada (dentro del hogar) o una IP pública (desde el exterior), permitirá el acceso de forma remota a los parámetros de la red eléctrica doméstica.

Tratándose de un instrumento de medida, se someterá a ensayo para conocer el grado de precisión y calcular el error en la medida.

Palabras clave

- Arduino
- Prototipo
- Hardware libre
- Contador inteligente
- Analizador de redes de 4 cuadrantes
- IoT (Internet de las cosas)
- Sensor de corriente

Agradecimientos

- A mi familia por el apoyo incondicional, y a todos los amigos y amigas que han mostrado apoyo o incluso han ofrecido ayuda en determinados momentos, no solo durante el trabajo de fin de grado, sino a lo largo de toda la carrera.
- Al tutor del proyecto por el compromiso en los tiempos de la realización del mismo.
- A todos los profesores y profesoras de los departamentos de eléctrica y electrónica de la UPNA, que han aportado la flexibilidad necesaria para poder compaginar los estudios con la vida laboral.
- A mis compañeros de trabajo y jefes de las empresas con las que he trabajado por los cambios de horario de trabajo para facilitarme la asistencia a las clases y exámenes.

Abstract

The goal of this project is the design and implementation of a monophasic power network analyzer for home use. It measures parameters like current, voltage and power factor, and from those ones the power and energy for a full picture of the home power network, to help manage the consumed and generated energy.

The device is based on an Open Hardware controller (Arduino), on which a PCB shield is designed and attached. This shield's role is to condition the analog input signals, connect peripheral devices, and provide the supply voltage to the controller. Finally, the control system will be programmed by using an Open Source code.

Beyond local monitoring of the power network via a LED screen, the prototype will be configured as a web server, available anywhere on the Internet (be it a private home IP, or a public IP). Therefore, the home power network parameters will be thus available remotely.

As a measurement instrument, it will be subjected to extensive testing to know the precision and measurement error.

Key words

- Arduino
- Prototype
- Open hardware
- Smart counter
- Internet of Things
- Network analyzer
- Current sensor

Laburpena

Proiektu honen helburua etxean erabiltzeko sare aztertzaile monofasikoa diseinatzea eta eraikitzea da. Gailu hau korrontea, tentsio eta potentzia faktore neurtzeko gai da eta bide batez energia eta potentzia kalkulatzeko sarearen datuak guztiak kontrolatzeko gai da eta gainera kontsumitu eta sortutakoa energiaren kudeaketa ere egin dezake.

Gailuak, hardware kontroladore librean oinarrituta dago, Arduino bezalakoa, shield PCB tankerara eraiki egingo da eta honek sarrera analogikoen seinaleak egokituko egingo ditu eta aldi berean, albokoei konexioa eta hornikuntza emango die. Amaitzeko kontrol sistemak kode irekia duen software baten bitartez burutuko da.

Sareko parametroen jarraipena tokiko pantailan gain, prototipoak, web zerbitzari gisa lan egiteko konfiguratu da datuak hodeira igotzeko. Ondorioztat, sare elektronikoaren parametroak Interneteko edozein konexio puntutik; IP pribatutik (etxetik)naiz IP publikotik (kaletik) ikusteko gai da.

Neurketa-tresna honek, ariketa ezberdinak pasa beharko ditu zehaztasuna eta akatsen neurketak kalkulatzeko.

Hitz gakoak zerrenda

- Arduino
- Prototipoa
- Hardware doan
- Kontagailu adimenduna
- IoT (gauzak Internet)
- Sare analizatzailea
- Uneko sentsorea

Índice de contenidos

| | |
|---|----|
| Resumen..... | 1 |
| Palabras clave | 1 |
| Abstract | 2 |
| Key words | 2 |
| Laburpena..... | 3 |
| Hitz gakoak zerrenda..... | 3 |
| Índice de contenidos..... | 4 |
| Índice de ilustraciones | 7 |
| Índice de tablas..... | 11 |
| 1. Introducción | 12 |
| 1.1. Contexto tecnológico | 12 |
| 1.1.1. El Ahorro y eficiencia energética | 12 |
| 1.1.2. Generación de energía y modelo de “Balance neto” | 13 |
| 1.2. Estado del arte | 14 |
| 1.3. Motivación..... | 16 |
| 1.4. Aporte al día a día | 16 |
| 1.5. Origen de la idea..... | 17 |
| 1.6. Hardware libre | 17 |
| 1.7. ARDUINO | 17 |
| 1.8. ARDUINO YUN..... | 18 |
| 1.9. IDE Arduino..... | 22 |
| 2. Objetivos | 23 |
| 3. Diseño del <i>hardware</i> | 24 |
| 3.1. Módulo de reloj en tiempo real (RTC)..... | 26 |
| 3.2. Pantalla LCD..... | 27 |
| 3.3. Fuente de alimentación común del sistema..... | 28 |
| 3.4. Elementos de protección del sistema | 29 |
| 3.5. Transductores de señal | 30 |
| 3.5.1. ¿Qué es un transformador de tensión y/o corriente? | 30 |
| 3.5.2. Transductor de tensión | 32 |
| 3.5.3. Transductor de corriente..... | 33 |
| 3.6. Circuitos de acondicionamiento de señales | 34 |

| | |
|--|----|
| 3.6.1. Circuito de acondicionamiento de señal de tensión..... | 36 |
| 3.6.2. Circuito de acondicionamiento de señal de corriente | 39 |
| 3.7. Pulsadores y LEDs..... | 41 |
| 4. Prueba de funcionamiento y correcciones..... | 43 |
| 4.1. Tensión de salida en los amplificadores operacionales | 44 |
| 4.2. Ruido acoplado a la señal medida | 45 |
| 4.3. Tensión en el secundario del transformador..... | 46 |
| 4.4. Diseño final..... | 46 |
| 5. Diseño de la PCB | 47 |
| 5.1. Componentes de PCB..... | 47 |
| 5.2. Diseño de componentes en Design Spark | 51 |
| 5.3. Especificaciones del diseño de la PCB..... | 54 |
| 5.4. Diseño final de la PCB..... | 55 |
| 6. Fabricación y montaje de la PCB..... | 58 |
| 7. Primer encendido de Arduino y configuración..... | 61 |
| 7.1. Identificación de direcciones de módulos en I2C | 65 |
| 7.2. Ajuste de RTC..... | 67 |
| 7.3. Ajuste de LCD..... | 67 |
| 7.4. Tarjeta MicroSD | 68 |
| 7.5. Ajustes de servidor web | 69 |
| 8. Calibración y monitorización de señales | 73 |
| 8.1. Calibración y monitorización de señal de tensión | 74 |
| 8.2. Calibración y monitorización de señal de corriente..... | 75 |
| 8.3. Calibración del factor de potencia | 78 |
| 9. Desarrollo e implementación del software | 80 |
| 9.1. Cuerpo del programa | 80 |
| 9.2. Subrutina de muestreo de señales | 81 |
| 9.3. Subrutina de cálculos de potencia y energía..... | 83 |
| 9.4. Subrutina de registro de datos | 84 |
| 9.5. Subrutina de monitorización y exportación de datos | 85 |
| 10. Proyecto final..... | 86 |
| 11. Medida de error..... | 91 |
| 11.1. Error de medida en tensión..... | 92 |
| 11.2. Error de medida de corriente | 94 |

| | |
|--|-----|
| 11.3. Error de medida de factor de potencia | 96 |
| 12. Conclusiones..... | 98 |
| 13. Líneas futuras..... | 100 |
| 13.1. Ampliación de funciones | 100 |
| 13.2. Mejoras técnicas | 101 |
| 14. Referencias bibliográficas..... | 102 |
| ANEXOS | 103 |
| A.1. Cálculo de potencia total..... | 104 |
| A.2. Curvas de calibración sensor de corriente STC-013..... | 106 |
| A.3. Curva de calibración de factor de potencia..... | 107 |
| A.4. Medida de error absoluto y relativo | 108 |
| A.4.1. Medida de error de tensión..... | 108 |
| A.4.2. Medida de error de corriente | 109 |
| A.4.3. Medida de error de factor de potencia..... | 110 |
| A.5. BOM LIST (presupuesto)..... | 111 |
| A.6. Códigos de programación en IDE Arduino | 112 |
| A.6.1. <i>Sketch</i> "I2C_SCANNER" | 112 |
| A.6.2. <i>Sketch</i> "Puesta_en_hora_3231RTC" | 114 |
| A.6.3. <i>Sketch</i> "LCD" | 115 |
| A.6.4. <i>Sketch</i> "YunDiskSpaceExpander" | 116 |
| A.6.5. <i>Sketch</i> "Analizador_de_redes_CALIBRACION" | 122 |
| A.6.6. <i>Sketch</i> "Analizador_de_redes_FIRMWARE" | 127 |
| A.7. Planos..... | 142 |
| A.7.1. Diagrama esquemático PCB..... | 142 |
| A.7.2. Diagrama funcional | 142 |
| A.8. Datasheets..... | 142 |
| A.8.1. Efergy engage solar kit | 142 |
| A.8.2. Mirubee Mirubox V.2. | 142 |
| A.8.3. Polímetro PCE-PCM1..... | 142 |
| A.8.4. Transformador Block 230/6V 1,2VA | 142 |
| A.8.5. RECOM RAC06-05DC 6W | 142 |
| A.8.6. Pinza STC-013..... | 142 |
| A.8.7. Portafusibles 5x20 OGN | 142 |
| A.8.8. TEXAS Operational Amplifier TL08XX JFET | 142 |

Índice de ilustraciones

| | |
|--|----|
| <i>Figura 1: Efergy engage solar kit</i> | 14 |
| <i>Figura 2: Mirube Mirubox V.2</i> | 15 |
| <i>Figura 3: Arduino YUN vista perspectiva</i> | 18 |
| <i>Figura 4: Arduino YUN vista superior planta</i> | 19 |
| <i>Figura 5: Arduino YUN vista inferior planta</i> | 18 |
| <i>Figura 6: Arduino YUN esquema funcional de hardware</i> | 19 |
| <i>Figura 7: Arduino YUN esquema de hardware</i> | 19 |
| <i>Figura 8: Arduino YUN sockets</i> | 20 |
| <i>Figura 9: Arduino YUN LEDs</i> | 21 |
| <i>Figura 10: Arduino YUN Reset buttons</i> | 21 |
| <i>Figura 11: IDE Arduino</i> | 22 |
| <i>Figura 12: Leyenda de conexiones</i> | 24 |
| <i>Figura 13: Diagrama de bloques de hardware</i> | 24 |
| <i>Figura 14: RTC vista superior planta</i> | 26 |
| <i>Figura 15: RTC vista inferior planta</i> | 26 |
| <i>Figura 16: Pantalla LCD</i> | 27 |
| <i>Figura 17: Backpack I2C en pantalla LCD</i> | 27 |
| <i>Figura 18: Fuente de alimentación RECOM</i> | 28 |
| <i>Figura 19: Portafusibles</i> | 29 |
| <i>Figura 20: Fusible 5x20mm</i> | 29 |
| <i>Figura 21: Esquema protecciones</i> | 29 |
| <i>Figura 22: Transformador</i> | 30 |
| <i>Figura 23: Transformador PCB 230/6V</i> | 33 |
| <i>Figura 24: Esquema circuito equivalente transformador tensión</i> | 32 |
| <i>Figura 25: Pinza STC-013</i> | 34 |
| <i>Figura 26: Esquema circuito equivalente transformador de corriente</i> | 33 |
| <i>Figura 27: Esquema AO seguidor de línea</i> | 35 |
| <i>Figura 28: Esquema AO sumador no inversor</i> | 35 |
| <i>Figura 29: Esquema divisor de tensión de la señal de tensión alterna</i> | 36 |
| <i>Figura 30: Simulación PSIM</i> | 37 |
| <i>Figura 31: Esquema divisor de tensión señal continua</i> | 37 |
| <i>Figura 32: Simulación PSIM</i> | 37 |
| <i>Figura 33: Esquema circuito acondicionamiento de señal de tensión</i> | 38 |
| <i>Figura 34: Simulación PSIM</i> | 38 |
| <i>Figura 35 Esquema divisor de tensión de señal de corriente alterna</i> | 39 |
| <i>Figura 36: Simulación PSIM</i> | 39 |
| <i>Figura 37: Esquema de circuito de acondicionamiento de señal de corriente</i> | 40 |
| <i>Figura 38: Simulación PSIM</i> | 40 |
| <i>Figura 39: Esquema pulsador de Reset</i> | 41 |
| <i>Figura 40: Esquema pulsador de encendido de pantalla LCD</i> | 41 |
| <i>Figura 41: Esquema de conmutador de encendido</i> | 41 |
| <i>Figura 42: Esquema de LEDs</i> | 42 |
| <i>Figura 43: Fotografía de montaje en placa de pruebas</i> | 43 |

| | |
|---|----|
| <i>Figura 44: Fotografía de imagen de osciloscopio</i> | 43 |
| <i>Figura 45: Esquema de circuitos de acondicionamiento rediseñado</i> | 44 |
| <i>Figura 46: Simulación PSIM</i> | 44 |
| <i>Figura 47: Fotografía de imagen de osciloscopio (perturbacion)</i> | 45 |
| <i>Figura 48: Esquema de filtro RC paso bajo</i> | 45 |
| <i>Figura 49: Esquema divisor de tensión de señal de tension alterna rediseñado</i> | 46 |
| <i>Figura 50: Esquema de diseño final de circuitos de acondicionamiento</i> | 46 |
| <i>Figura 51: Transformador PCB 230/6V</i> | 48 |
| <i>Figura 52: Fuente de alimentación RECOM PCB</i> | 47 |
| <i>Figura 53: Arduino YUN</i> | 48 |
| <i>Figura 54: módulo RTC</i> | 47 |
| <i>Figura 55: Resistencia 1/8W 1% de tolerancia</i> | 48 |
| <i>Figura 56: Potenciómetro giratorio</i> | 48 |
| <i>Figura 57: Condensador de poliéster</i> | 48 |
| <i>Figura 58: TL084</i> | 49 |
| <i>Figura 59: TL084 pinout</i> | 48 |
| <i>Figura 60: Zócalo 14 pines</i> | 49 |
| <i>Figura 61: Portafusibles</i> | 49 |
| <i>Figura 62: Fusible 5x20mm</i> | 49 |
| <i>Figura 63: Pulsador PCB</i> | 49 |
| <i>Figura 64: LED 5mm verde</i> | 50 |
| <i>Figura 65: Clema</i> | 51 |
| <i>Figura 66: Conectores hembra</i> | 51 |
| <i>Figura 67: Conectores macho</i> | 50 |
| <i>Figura 68: Tornillería de nylon</i> | 50 |
| <i>Figura 69: Símbolo esquemático y footprint de portafusibles</i> | 51 |
| <i>Figura 70: Símbolo esquemático y footprint de fuente de alimentación RECOM</i> | 51 |
| <i>Figura 71: Símbolo esquemático y footprint de transformador PCB</i> | 52 |
| <i>Figura 72: Símbolo esquemático y footprint de clema</i> | 52 |
| <i>Figura 73: Esquemático y footprint de conector 6</i> | 53 |
| <i>Figura 74: Esquemático y footprint de conector 2x3</i> | 52 |
| <i>Figura 75: Esquemático y footprint de conector 8</i> | 53 |
| <i>Figura 76: Esquemático y footprint de conector 10</i> | 52 |
| <i>Figura 77: Símbolo PCB plantilla Arduino shield</i> | 53 |
| <i>Figura 78: Símbolo PCB módulo RTC</i> | 53 |
| <i>Figura 79: Calculadora Design Spark de anchura de pista</i> | 54 |
| <i>Figura 80: Diseño PCB Design Spark distribución de componentes</i> | 55 |
| <i>Figura 81: Diseño PCB Design Spark enrutado de pistas</i> | 56 |
| <i>Figura 82: Diseño PCB vista 3D cara superior</i> | 57 |
| <i>Figura 83: Diseño PCB vista 3D cara inferior</i> | 57 |
| <i>Figura 84: PCB vista superior planta</i> | 58 |
| <i>Figura 85: PCB vista inferior planta</i> | 58 |
| <i>Figura 86: PCB con componentes vista superior</i> | 59 |
| <i>Figura 87: PCB con componentes vista inferior</i> | 60 |
| <i>Figura 88: PCB completa</i> | 60 |

| | |
|---|----|
| <i>Figura 89: Arduino conectado por USB</i> | 61 |
| <i>Figura 90: IDE Arduino</i> | 61 |
| <i>Figura 91: Redes</i> | 62 |
| <i>Figura 92: IDE Arduino</i> | 61 |
| <i>Figura 93: Configuración de Arduino YUN</i> | 62 |
| <i>Figura 94: Configuración de Arduino YUN</i> | 62 |
| <i>Figura 95: Configuración de Arduino YUN</i> | 63 |
| <i>Figura 96: IDE Arduino</i> | 63 |
| <i>Figura 97: Configuración de Arduino YUN</i> | 63 |
| <i>Figura 98: PCB conectada a la red y micro USB</i> | 64 |
| <i>Figura 99: Arduino Power socket</i> | 64 |
| <i>Figura 100: Conexión Arduino y RTC</i> | 65 |
| <i>Figura 101: Conexión Arduino y pantalla LCD</i> | 65 |
| <i>Figura 102: Monitor serie</i> | 66 |
| <i>Figura 103: Conexión Arduino y pantalla LCD</i> | 66 |
| <i>Figura 104: Monitor serie</i> | 66 |
| <i>Figura 105: Monitor serie</i> | 67 |
| <i>Figura 106: Encendido pantalla LCD</i> | 67 |
| <i>Figura 107: Tarjeta micro SD</i> | 68 |
| <i>Figura 108: Monitor serie</i> | 68 |
| <i>Figura 109: Configuración servicios REST</i> | 69 |
| <i>Figura 110: Direcciones IP y MAC de Arduino</i> | 69 |
| <i>Figura 111: Pantalla de configuración del router</i> | 70 |
| <i>Figura 112: Asignación de IP estática</i> | 70 |
| <i>Figura 113: Configuración de puertos del router</i> | 71 |
| <i>Figura 114: Información de direcciones IP del router</i> | 71 |
| <i>Figura 115: Acceso a Arduino desde IP pública</i> | 71 |
| <i>Figura 116: IP dinámica asignada al dominio en "ChangelP"</i> | 72 |
| <i>Figura 117: Acceso a Arduino desde el dominio</i> | 72 |
| <i>Figura 118: Serial plotter</i> | 74 |
| <i>Figura 119: Banco de resistencias</i> | 76 |
| <i>Figura 120: Autotransformador</i> | 75 |
| <i>Figura 121: Resistencias en paralelo con autotransformador</i> | 75 |
| <i>Figura 122: Sonda en señal de corriente</i> | 77 |
| <i>Figura 123: Monitorización de señales con osciloscopio</i> | 76 |
| <i>Figura 124: Gráfico de curva de calibración STC-013 (V-I)</i> | 76 |
| <i>Figura 125: Serial plotter</i> | 77 |
| <i>Figura 126: Serial plotter</i> | 77 |
| <i>Figura 127: Fotografía de sonda conectada a PCB</i> | 78 |
| <i>Figura 128: Captura de osciloscopio de tensión y corriente</i> | 78 |
| <i>Figura 129: Gráfico de curva de calibración STC-013 (I-φ)</i> | 79 |
| <i>Figura 130: Diagrama de flujos cuerpo de programa</i> | 80 |
| <i>Figura 131: Representación gráfica continua Matlab</i> | 82 |
| <i>Figura 132: Representación gráfica discreta Matlab</i> | 81 |
| <i>Figura 133: Simulación PSIM</i> | 82 |

| | |
|--|-----------|
| <i>Figura 134: Diagrama de flujos de subrutina de muestreo de señales</i> | <i>82</i> |
| <i>Figura 135: Diagrama de flujos de subrutina de cálculos de potencia y energía</i> | <i>83</i> |
| <i>Figura 136: Diagrama de flujos de subrutina de registro de datos</i> | <i>84</i> |
| <i>Figura 137: Diagrama de flujos de subrutina de monitorización y exportación de datos.....</i> | <i>85</i> |
| <i>Figura 138: Caja de registro de montaje superficial</i> | <i>86</i> |
| <i>Figura 139: PCB montada en la caja de registro</i> | <i>86</i> |
| <i>Figura 140: Vista interna de la tapa de la caja de registro</i> | <i>87</i> |
| <i>Figura 141: Vista interna del montaje.....</i> | <i>87</i> |
| <i>Figura 142: Vista exterior del montaje en la caja de registro</i> | <i>88</i> |
| <i>Figura 143: Pantalla de valores instantáneos</i> | <i>88</i> |
| <i>Figura 144: Pantalla de valores instantáneos</i> | <i>88</i> |
| <i>Figura 145: Pantalla de máximos y acumulados</i> | <i>89</i> |
| <i>Figura 146: Pantalla de totalizador de consumo energético.....</i> | <i>89</i> |
| <i>Figura 147: Pantalla de totalizador de consumo energético.....</i> | <i>89</i> |
| <i>Figura 148: Pantalla de totalizador de inyección energética</i> | <i>89</i> |
| <i>Figura 149: Pantalla de totalizador de inyección energética</i> | <i>89</i> |
| <i>Figura 150: Pantalla de avisos.....</i> | <i>90</i> |
| <i>Figura 151: Acceso a los dato desde la red pública</i> | <i>91</i> |
| <i>Figura 152: Acceso a los datos desde la red privada</i> | <i>90</i> |
| <i>Figura 153: Polímetro PCE-PCM1.....</i> | <i>91</i> |
| <i>Figura 154: Polímetro conectado a autotransformador</i> | <i>92</i> |
| <i>Figura 155: Gráfico de error absoluto en tensión</i> | <i>92</i> |
| <i>Figura 156: Gráfico de error relativo en tensión.....</i> | <i>93</i> |
| <i>Figura 157: Conexión de las pinzas amperimétricas.....</i> | <i>94</i> |
| <i>Figura 158: Gráfico de error absoluto en corriente</i> | <i>95</i> |
| <i>Figura 159: Gráfico de error relativo en corriente</i> | <i>95</i> |
| <i>Figura 160: Condensador de 20mF</i> | <i>97</i> |
| <i>Figura 161: Condensador de 7μF</i> | <i>96</i> |
| <i>Figura 162: Gráfico de error absoluto en factor de potencia</i> | <i>97</i> |
| <i>Figura 163: Gráfico de error relativo en factor de potencia.....</i> | <i>97</i> |

Índice de tablas

| | |
|---|------------|
| <i>Tabla 1: Comparativa Mirubee Mirubox V.2. VS Efergy engage solar kit</i> | <i>15</i> |
| <i>Tabla 2: Características microcontrolador Arduino</i> | <i>20</i> |
| <i>Tabla 3: Características microprocesador Linux</i> | <i>20</i> |
| <i>Tabla 4: Comparación con productos del mercado</i> | <i>98</i> |
| <i>Tabla 5: Curva de calibración STC-013 (V-I)</i> | <i>106</i> |
| <i>Tabla 6: Curva de calibración STC-013 (I-ϕ)</i> | <i>107</i> |
| <i>Tabla 7: Medida de error de tensión.....</i> | <i>108</i> |
| <i>Tabla 8: Medida de error de corriente</i> | <i>109</i> |
| <i>Tabla 9: Medida de error de factor de potencia</i> | <i>110</i> |
| <i>Tabla 10: BOM LIST (presupuesto).....</i> | <i>111</i> |

1. Introducción

1.1. Contexto tecnológico

1.1.1. El Ahorro y eficiencia energética

Al hablar del ahorro y la eficiencia energética en global, todos asociamos ambos conceptos con la sostenibilidad energética y un respeto al medio ambiente. Los identificamos con un gasto energético menor del habitual; si bien, ambos conceptos identifican dos enfoques diferentes a esa reducción del gasto energético.

El ahorro energético conlleva un cambio en los hábitos de consumo humano los cuales repercuten en un menor gasto de energía. Son acciones directas que tomamos y que implican un menor consumo energético.

La eficiencia energética sin embargo implica un menor consumo de energía en nuestra vivienda o empresa pero sin que exista ningún cambio de hábito de consumo ^[1].

Así, la eficiencia energética se puede definir como la capacidad de un uso, equipo, instalación o proceso para realizar su función con el menor consumo energético posible ^[2].

De esta manera el ahorro energético es una parte fundamental en un sistema eléctrico de distribución; tanto por el consumidor, que se traduce en un ahorro económico en su tarifa eléctrica; como por la parte generadora, afectando al dimensionamiento de la infraestructura de la red de distribución (líneas, subestaciones, centrales,...), además del consumo de combustibles (uranio, carbón, gas,...) y el impacto medioambiental en el caso de centrales de energías no renovables, debido a las emisiones de CO₂.

Es evidente que el coste energético afecta en la economía a todas las escalas, con lo que debe ser analizado en el diseño de cualquier proyecto de ingeniería en cuyo funcionamiento sea necesario un abastecimiento energético.

La factura eléctrica en distribuciones de baja tensión depende prácticamente de dos variables: consumo energético y potencia de abastecimiento instantáneo. El consumo energético está relacionado con la potencia y el tiempo de funcionamiento de los aparatos eléctricos; mientras que en el abastecimiento solo influye la potencia instantánea de consumo.

Una forma de reducir la factura eléctrica es ajustando la potencia de abastecimiento contratada al valor adecuado a nuestras necesidades, o incluso gestionar el encendido de los aparatos eléctricos para minimizar ese valor de potencia contratada. El consumo energético se puede gestionar si tenemos contratada una tarifa con discriminación horaria, pero si no es así, solo se puede reducir evitando gastos innecesarios.

Por todo ello es de gran interés supervisar los parámetros de la red eléctrica, para poder gestionar el consumo energético con el fin de conseguir una mayor eficiencia energética.

1.1.2. Generación de energía y modelo de “Balance neto”

Hasta hace poco tiempo, en muchos países, legalmente sólo era posible instalar paneles solares para autoconsumo en viviendas, empresas o instalaciones que no tuvieran la posibilidad de conectarse a la red eléctrica. La producción de electricidad y su distribución sólo estaba autorizada a las compañías eléctricas, razón por la cual el modelo eléctrico imperante en el mundo es el de las grandes centrales de producción eléctrica, que a través de complejos sistemas de transporte, lo distribuyen a los consumidores.

Esta situación sin embargo ha empezado a cambiar en los últimos años y ya algunos países han avanzado legislativamente para permitir a particulares y empresas que no estuvieran en lugares remotos la instalación de sistemas de energías renovables con los que generar electricidad, para autoconsumo y/o para inyectarla a la red eléctrica.

Uno de los esquemas más interesantes, es el que se ha venido en llamar de *balance neto*. Es un modelo de conexión entre un pequeño productor de energía eléctrica renovable para autoconsumo (eólica o fotovoltaica) y la red eléctrica general, en el cual hay un intercambio de electricidad ^[3].

Consiste en extraer energía de la red cuando el consumo es superior a la producción energética, e inyectar a la red el excedente de energía generada cuando esta es mayor que la consumida. Al cabo de un periodo establecido, se hace un balance energético, y dependiendo del saldo final; la compañía suministradora cobrará con un recibo el consumo energético, o en caso de que se haya inyectado energía a la red, deberá compensar al productor económicamente, o acumulando el saldo, para descontarlo en meses menos favorables.

En este sistema se prescinde de las baterías, ya que no es necesario acumular carga; pero en cambio es indispensable tener un control absoluto de la energía consumida e inyectada a la red, y por ello es necesario contar en la instalación con un contador bidireccional o de 4 cuadrantes.

1.2. Estado del arte

Vamos a hacer un análisis de la oferta de mercado sobre dispositivos capaces de supervisar la red eléctrica del hogar para hacernos una idea del entorno de trabajo en el que nos moveremos. En concreto estudiamos 2 modelos de marcas diferentes:

- **Efergy Engage solar kit**

Es un contador inteligente capaz de medir valores instantáneos con factor de potencia en 4 cuadrantes, es decir, funciona como un contador de energía bidireccional, tanto de consumo como de generación (ver figura 1).

Dispone de conexión WIFI para exportar los datos a la nube y la posibilidad de monitorizar los parámetros desde una App para Smartphone o aplicaciones informáticas, pero no dispone de pantalla local.

Su instalación es descentralizada. Funciona con una central conectada al router, y esta hace de receptor de información de los distintos emisores colocados en los circuitos a medir. Cada emisor contiene una pinza amperimétrica y se instala en cada cuadro de medida. El problema de estos emisores es que se alimentan con pilas.

Su precio final es de 129,99 €



Figura 1: Efergy engage solar kit

- **Mirube Mirubox V.2**

Es un contador inteligente capaz de medir valores instantáneos y el consumo energético, pero no dispone de función de medida de generación de energía (ver figura 2).

Permite conexión WIFI y la exportación de datos a la nube, de manera que se puede monitorizar los parámetros desde una App para Smartphone o desde un programa informático, pero no dispone de una pantalla local para la comodidad del usuario.

Se instala cerca del cuadro de protección de la vivienda, para acoplar la pinza de medida de corriente al conductor de fase general, y se autoalimenta desde la medida de tensión de red.

Su precio final es de 109,95€



Figura 2: Mirube Mirubox V.2.

- **Tabla comparativa: Mirubee Mirubox V.2. VS Efergy engage solar kit**

| | <i>Mirubee Mirubox V.2.</i> | <i>Efergy engage solar kit</i> |
|--|------------------------------------|---------------------------------------|
| <i>Medida de tensión</i> | SI | SI |
| <i>Medida de corriente</i> | SI | SI |
| <i>Consumo energético</i> | SI | SI |
| <i>Generación energética</i> | NO | SI |
| <i>Conexión WIFI</i> | SI | SI |
| <i>Exportación de datos a la nube</i> | SI | SI |
| <i>Pantalla local</i> | NO | NO |
| <i>Alimentación</i> | TENSION DE RED | PILAS |
| <i>Precio</i> | 109,95 Euros | 129,99 Euros |

Tabla 1: Comparativa Mirubee Mirubox V.2. VS Efergy engage solar kit

1.3. Motivación

La razón por la cual se decidió realizar este proyecto procede principalmente del interés propio por la electrónica, añadido al diseño, construcción y puesta en marcha de proyectos dirigidos al uso cotidiano.

Se puede considerar un proyecto de interés general, ya que es aplicable a todas las instalaciones eléctricas domésticas, dispongan o no de una instalación de generación de energía eléctrica. También está destinado a todos los perfiles de usuarios interesados en reducir el gasto energético, ya sea por ahorro económico, o por contribución al medio ambiente, ya que solo es necesario comprender unos conceptos básicos de electrotecnia para interpretar los datos.

1.4. Aporte al día a día

El dispositivo aporta un control absoluto de los parámetros instantáneos de la red doméstica por parte del usuario, así como una visión global de la energía consumida y generada.

El mostrar los parámetros medidos en tiempo real es una gran ventaja, ya que además de ayudar a evitar malos hábitos de consumo o derroches controlables, permite a la vez demostrar al usuario la diferencia de consumo entre electrodomésticos con distintas potencia, o incluso comprobar la mejora de la eficiencia energética en el caso de que un electrodoméstico haya sido sustituido por otro de clase energética más elevada.

Para el usuario que no está familiarizado con términos como energía y potencia puede resultar algo abstracto la factura eléctrica. Por eso es interesante el hecho de tener una visión cuantificada de estos términos; ya que al comprobar las variaciones que se producen en ellos en función del uso de la red doméstica, ayuda a comprender el significado de la tarifa y cómo influye en esta nuestro consumo.

Este control de la red nos permite ser conscientes del consumo y reducir en la medida de lo posible la potencia máxima de abastecimiento instantáneo. Esto se puede traducir en un reajuste de la parte fija de la tarifa eléctrica, que corresponde con el abastecimiento; y ser más críticos a la hora de revisar la factura, ya que disponemos de un valor de referencia calculado.

A parte del control local desde el hogar, la posibilidad de monitorizar los parámetros desde un Smartphone o un PC conectado a cualquier red de comunicaciones es una ventaja añadida, sobretodo en el caso de generación de energía eléctrica, ya que permite la supervisión instantánea desde cualquier lugar.

1.5. Origen de la idea

Este tipo de dispositivos son muy comunes en el sector industrial y en la red de transporte eléctrica. Debido al alto consumo energético comparado al de una vivienda, en una instalación industrial, una pequeña mejora en la eficiencia energética, a largo plazo puede derivar en un gran ahorro energético.

En cambio, para el uso doméstico, puesto que los cambios no son tan significativos, no existen aparatos tan completos como los industriales, y aportan poca flexibilidad en la conexión a la red de datos. Por esto me pareció útil construir un dispositivo similar al de uso industrial, en cuanto a funciones, pero a menor escala, como es la del hogar, y con un coste de fabricación más asequible comparado a los productos del mercado.

1.6. Hardware libre

Cuando se cumple que los dispositivos de *hardware* cuyas especificaciones y diagramas esquemáticos son de acceso público es *hardware libre*, ya sea bajo algún tipo de pago o de forma gratuita. Siempre recordando que libre no es sinónimo de gratis. El *hardware libre* forma parte de la cultura libre.

El *hardware libre* toma las mismas ideas del *software libre* para aplicarlas en su campo, en lo referente a las cuatro libertades: libertad de uso, de estudio y modificación, de distribución, y de redistribución de las versiones modificadas. Su objetivo es crear diseños de aparatos electrónicos de forma abierta, de manera que todas las personas puedan acceder, como mínimo, a los planos de construcción de los dispositivos.

Al igual que en el *software libre*, la denominación de *hardware libre*, se refiere a la libertad de poder utilizar el dispositivo y su documentación, no a que sea necesariamente gratuito. Aunque comparta filosofía con el *software libre*, debido a la propia naturaleza de estos componentes físicos, muchos de sus preceptos son limitados.

Básicamente se considera que un *hardware* es abierto si cumple una serie de requisitos. La información sobre la manera de comunicarse con el *hardware*, el diseño del mismo y las herramientas utilizadas para crear ese diseño deben ser publicados para ser usadas libremente. De esta manera se facilita el control, implementación y mejoras en el diseño por la comunidad de desarrolladores ^[4].

Siguiendo lo anteriormente comentado, durante los últimos años ha sido muy extendido entre aficionados a la electrónica el Arduino.

1.7. ARDUINO

Arduino es una plataforma de prototipos electrónica de código abierto basada en *hardware* y *software* flexibles y fáciles de usar. Está pensado e inspirado en artistas, diseñadores, y estudiantes de computación o robótica y para cualquier interesado en crear objetos o entornos interactivo, o simplemente por *hobby*.

Arduino consta de una placa principal de componentes eléctricos, donde se encuentran conectados los controladores principales que gestionan los demás complementos y circuitos ensamblados en la misma. Además, requiere de un lenguaje de programación para poder ser utilizado, programado y configurarlo a nuestra necesidad ^[5].

1.8. ARDUINO YUN

El modelo a partir del cual se va a desarrollar este proyecto es el Arduino YUN (ver figura 3). Como toda la familia de controladores dispone de entradas/salidas digitales, y entradas analógicas para actuar con el entorno exterior, mediante sensores y actuadores, además de puertos de comunicaciones para conectar periféricos.



Figura 3: Arduino YUN vista perspectiva

La principal característica del nuevo Arduino YUN es su capacidad de conectarse a Internet. Esto lo logra ya sea a través del puerto de red que viene integrado a la placa o bien por medio de WIFI. La conexión con la computadora se realiza por medio de un puerto micro USB tal como el Arduino LEONARDO (de hecho, el YUN y el LEONARDO poseen el mismo micro controlador Atmega32u4).

A diferencia de la mayoría de los modelos de Arduino existentes, este no cuenta con un regulador de voltaje integrado por lo que la alimentación debe ser exclusivamente de 5 voltios. Se puede alimentar por medio del puerto micro USB conectado a la computadora o con un adaptador de 5 voltios.

Es vital para cualquier usuario del YUN ser precavido a la hora de alimentar el dispositivo. Un poco más de voltaje por encima de los 5 voltios hará que nuestro micro controlador se estropee y perdamos nuestra inversión.



Figura 4: Arduino YUN vista superior planta



Figura 5: Arduino YUN vista inferior planta

Aparte de la salida de Ethernet y la entrada micro USB Arduino cuenta con un host USB. ¿Para qué? Pues Arduino cuenta con un procesador Atheros AR9331, lo que le permite utilizar Linux, una versión especial para Arduino llamada Linino. El USB host puede ser utilizado para conectar memorias flash, cámaras y otros dispositivos para interactuar con el procesador Atheros.

En la parte inferior el Arduino YUN cuenta con un adaptador micro SD, el cual tiene conexiones físicas con el procesador Atheros pero no con el Atmega. Sin embargo, es posible que el Atmega utilice los recursos del micro SD por medio de la librería Bridge. En el esquema de la figura 6 se muestra esto de una forma gráfica.

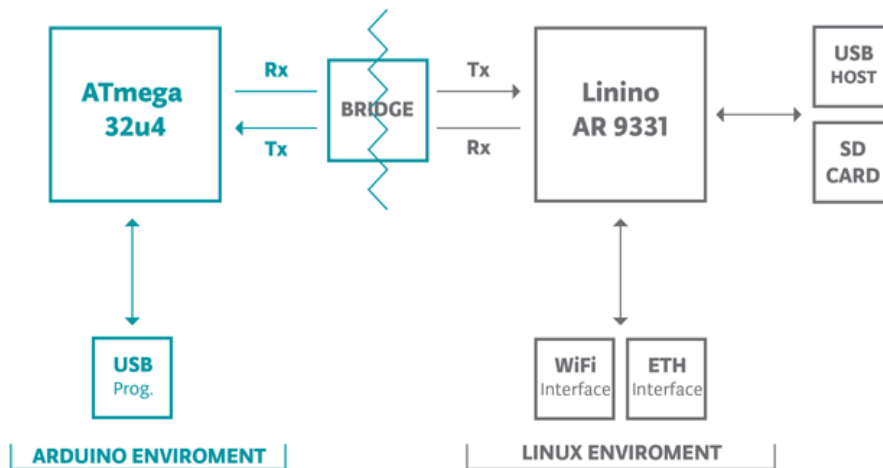


Figura 6: Arduino YUN esquema funcional de hardware

A partir del diagrama de la figura 7 podemos señalar las características del *hardware* que conforma la placa del Arduino YUN.

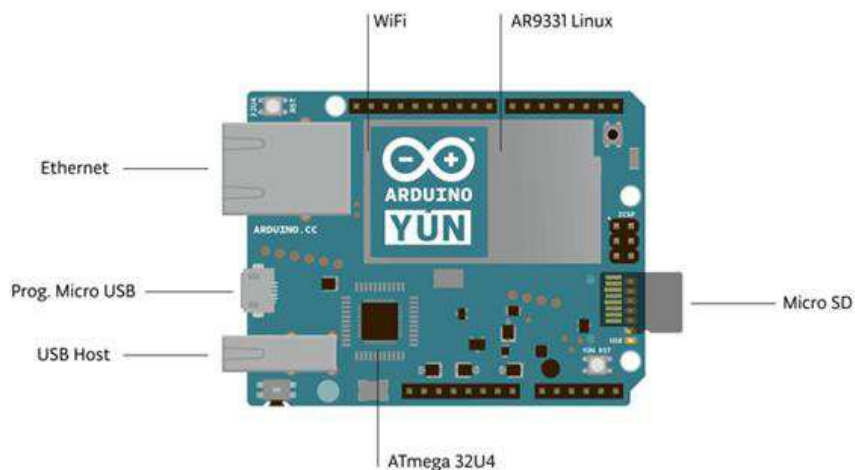


Figura 7: Arduino YUN esquema de hardware

En la imagen de la figura 8 se muestran las ubicaciones de los componentes arriba descritos. El *hardware* ha mejorado bastante en cuanto a versiones anteriores. Los componentes que se han agregado ocupan mucho más espacio que otros modelos de Arduino, por lo que ni siquiera fue posible imprimir los números de los pines en la placa. Lo que se ha hecho es que se ha colocado etiquetas sobre los socket de las salidas.



Figura 8: Arduino YUN sockets

- Ficha técnica del microcontrolador y microprocesador

| <i>Microcontrolador ARDUINO</i> | |
|--|-------------------------------------|
| Microcontrolador | AT MEGA 32u4 |
| Voltaje operativo | 5V |
| Voltaje de entrada | 5V |
| Pines digitales I/O | 20 |
| Canales PWM | 7 |
| Canales entrada analógica | 6 |
| Corriente DC pines 5V | 40mA |
| Corriente DC pines 3.3V | 50mA |
| Memoria Flash | 32KB (4KB usados por el BootLoader) |
| SRAM | 2.5 KB |
| EEPROM | 1 KB |
| Clock Speed | 16 MHz |

Tabla 2: Características microcontrolador Arduino

| <i>Microprocesador LINUX</i> | |
|-------------------------------------|--------------------------|
| Procesador | ATHEROS AR9331 |
| Arquitectura | MIPS @400 MHz |
| Voltaje operativo | 3.3V |
| Ethernet | IEEE 802.3 10/100 Mbit/s |
| Wifi | IEEE 802.11 b/g/n |
| USB Tipo-A | 2.0 Host/Device |
| Lector MicroSD | MicroSD only |
| RAM | 64 MB DDR2 |
| Memoria Flash | 16 MB |

Tabla 3: Características microprocesador Linux

- **Características de Memoria**

El ATmega32u4 posee 32 KB de memoria, de los cuales se puede utilizar 28KB (4KB son usados por el BootLoader). Posee 2,5KB de SRAM y 1KB de EEPROM.

El AR9331 no posee una memoria interna dentro del procesador. Las memorias RAM y de almacenamiento están conectadas de manera externa. El Arduino YUN posee 64MB de memoria RAM del tipo DDR2 y 16MB de memoria Flash.

En la memoria Flash viene instalada una distribución de Linux llamada *OpenWrt-Yun*, la cual ocupa 9 de los 16MB. Cuando se requiere más espacio para aplicaciones diversas se utiliza una memoria MicroSD.

- **Entradas y Salidas**

Como todos los modelos de Arduino, el YUN posee diferentes interfaces para entrada/salida de datos. A continuación el listado de pines con los números de los mismos.

- **Comunicación Serial:** 0 (RX) y 1 (TX)
- **I2C:** 2 (SDA) y 3 (SCL)
- **SPI:** en los pines ICSP

El Arduino YUN cuenta con algunos LEDs de estado que indican el estatus del dispositivo en determinado instante (ver figura 9).

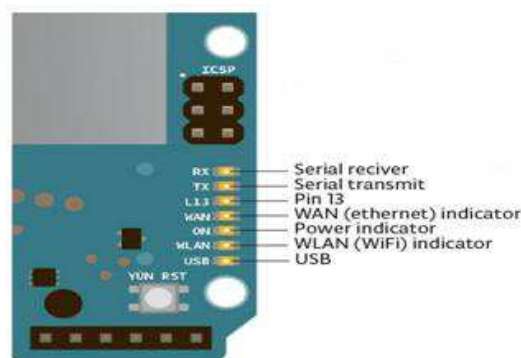


Figura 9: Arduino YUN LEDs

También cabe destacar que el Arduino YUN cuenta con 3 botones de Reset (ver figura 10), uno para el microcontrolador, uno para el microprocesador y otro para el Wifi [6].

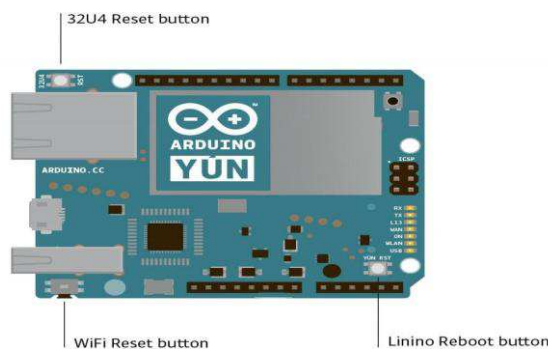


Figura 10: Arduino YUN Reset buttons

1.9. IDE Arduino.

El entorno de desarrollo integrado también llamado IDE (sigla en inglés de Integrated Development Environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios lenguajes.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Además en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del *hardware*.

La imagen de la figura 11 muestra el entorno de programación.

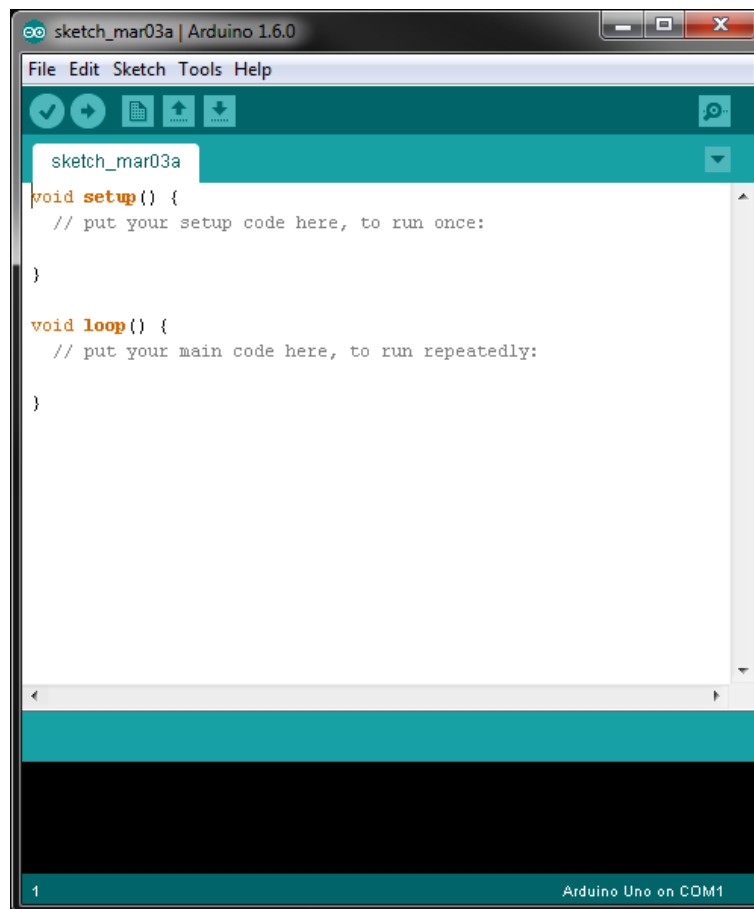


Figura 11: IDE Arduino

Los programas de Arduino están compuestos por un solo fichero con extensión "*ino*", aunque es posible organizarlo en varios ficheros. El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

La última versión del IDE de Arduino es la 1.8.5. También es destacable desde la aparición de la versión 1.6.2 la incorporación de la gestión de librerías y la gestión de placas muy mejorada respecto a la versión anterior y los avisos de actualización de versiones de librerías y cores ^[7].

2. Objetivos

En éste proyecto de fin de grado se construirá un analizador de redes de 4 cuadrantes para supervisar los parámetros de redes eléctricas monofásicas domésticas. El diseño y la construcción del aparato estarán basados en un controlador de *hardware libre*, Arduino.

Puesto que es únicamente un aparato de medida, sus funciones se basan en medición de magnitudes, y monitorización de los siguientes datos:

- Valores instantáneos de parámetros de la red doméstica como son la tensión, la corriente, el factor de potencia, y las potencias; activa, reactiva y aparente.
- Valores acumulados de energía; tanto consumida como generada, y registro de los mismos en una memoria externa no volátil.
- Valores máximos de corriente y potencia, acumulados en el mismo mes, y registrados en una memoria externa no volátil.
- Acceso a todos los parámetros de forma remota desde cualquier lugar mediante direccionamiento IP desde cualquier dispositivo con acceso a Internet.

Para conseguir los objetivos propuestos del proyecto, la implementación del prototipo se divide en diferentes etapas de diseño, construcción y configuración a partir de las cuales se desarrollará la línea de trabajo:

- **Diseño y simulación del hardware:** La primera fase del proyecto consiste en el diseño teórico del hardware apoyado con simulaciones de los circuitos de señales analógicas. El diseño se realiza en base a los componentes y periféricos principales elegidos.
- **Prueba de funcionamiento y corrección de errores:** Antes del volcado del diseño a una placa de circuito impreso, se comprueba el funcionamiento del mismo sobre una placa de pruebas.
- **Diseño y construcción de la PCB:** La etapa de diseño se realiza por medio de una herramienta informática para su posterior fabricación. En esta fase también se deciden los componentes genéricos para el montaje de la PCB.
- **Configuraciones del controlador y periféricos:** Antes de la implementación del software es necesario realizar ajustes de configuración en el hardware.
- **Calibración de los circuitos de acondicionamiento de señales:** Calcular las ecuaciones características de los circuitos de acondicionamiento, para interpretar las señales medidas en los valores reales de cada magnitud.
- **Implementación del software del controlador:** Se desarrolla un *sketch* en código abierto con las instrucciones a realizar por el controlador, para su funcionamiento autónomo.
- **Medida de error del prototipo:** Como etapa final, se realizaran ensayos para comprobar el grado de precisión en la medida de magnitudes por el prototipo.

3. Diseño del *hardware*

Se realiza un análisis y elección de los componentes principales necesarios para proceder al diseño teórico del *hardware* del sistema, con el fin de construir una PCB tipo *shield*, capaz de interconectar Arduino con los periféricos y circuitos de acondicionamiento de señal, creando un conjunto sólido y compacto.

El diseño se divide en bloques según su función a desempeñar, teniendo en cuenta los objetivos a cumplir por el dispositivo. En el diagrama de la figura 13 se representan los bloques de *hardware* y sus interconexiones.

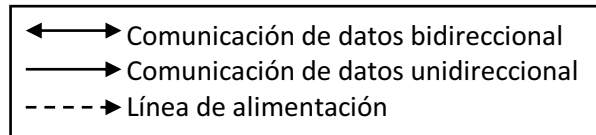


Figura 12: Leyenda de conexiones

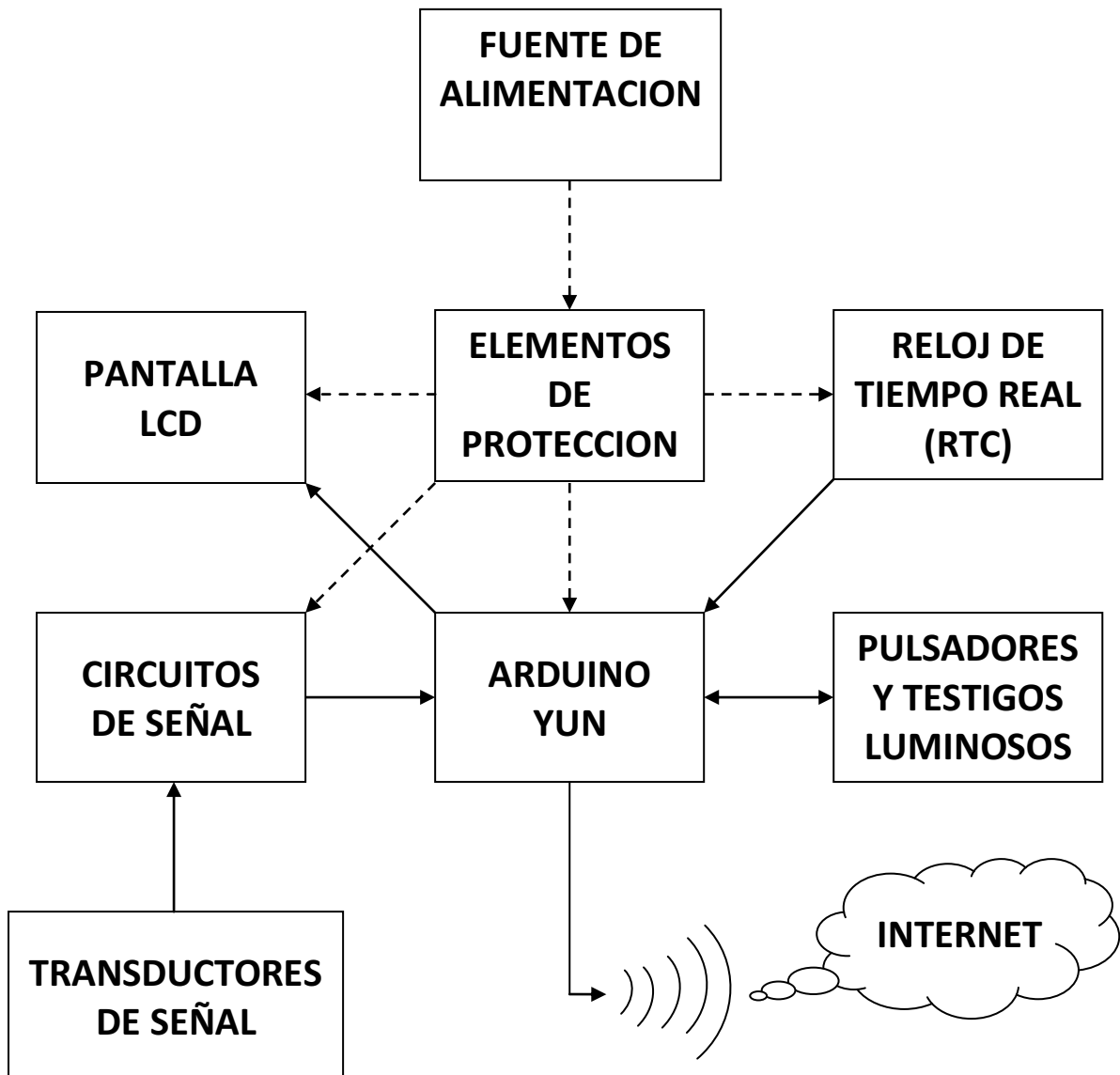


Figura 13: Diagrama de bloques de hardware

Para conseguir compactar en un bloque el mayor número de componentes posible, en la elección de ellos, se da prioridad a los de montaje en PCB. En algunos casos, por la función que desempeñan, su montaje es externo a la PCB, conectado a ella mediante regletas de conexión.

En el caso de componentes externos, como pulsadores, LEDs, pantalla LCD; se da libertad en el montaje, es decir, la PCB se puede instalar en el fondo de un armario o cuadro eléctrico y los componentes en el exterior, cableados a la PCB. Por lo tanto se deja al gusto del usuario.

A continuación se justifica la incorporación de los bloques de hardware representados en el diagrama de la figura 13.

- **Módulo de reloj de tiempo real (RTC):** Es necesario por su precisión de medida del tiempo para los cálculos de energía, ya que por software, la desviación de medida al cabo de un día puede llegar a varios minutos.
- **Módulo de pantalla LCD:** Muestra los datos calculados al usuario. Por aportar flexibilidad en cuanto al lugar y forma de montaje del sistema, la pantalla no se integra en la PCB, sino que se conecta mediante hilos y regletas de conexión.
- **Fuente de alimentación común del sistema:** Puesto que Arduino no se conecta a la red eléctrica, es necesario conectarlo a su tensión de alimentación, además de los periféricos adicionales y los circuitos de acondicionamiento de señales. Lo ideal es que se alimente todo el sistema a la misma tensión con una única fuente de alimentación.
- **Elementos de protección del sistema:** Protección tanto de los componentes como del mismo Arduino; para evitar daños por cortocircuito o sobrecarga de corriente.
- **Transductores de señal:** Su función es transformar la variable a medir, en una variable eléctrica. Aunque en este caso las variables a medir ya son eléctricas, los transductores harán la función de reducir el valor en amplitud, además de proporcionar aislamiento galvánico.
- **Circuitos de acondicionamiento de señales:** Se trata de transformar las magnitudes eléctricas de tensión y corriente, transducidas por los sensores, a la forma y amplitud en la que trabaja Arduino.
- **Componentes de control. Pulsadores y LEDs:** Para realizar maniobras por parte del usuario como RESET, ON/OFF, encendido de LCD, se utilizan pulsadores integrados a la PCB y otros externos conectados a esta por regletas, al igual que la pantalla LCD. Como testigos luminosos se utilizan LEDs para indicar el encendido del sistema.

Además del diseño y análisis teórico, se utiliza un programa informático de simulación de circuitos eléctricos, "PSIM", para comprobar el funcionamiento del diseño de los circuitos de acondicionamiento de señales.

3.1. Módulo de reloj en tiempo real (RTC)

El módulo de reloj RTC (Real Time Clock) está diseñado para controlar fecha y hora con alta precisión, por medio del micro controlador DS3231. Conectado al Arduino, el software dispone de acceso informativo a la fecha y hora actual (ver figura 14).

El software de Arduino es capaz de realizar instrucciones temporizadas, o contar el tiempo de un proceso; pero a diferencia del RTC, tiene una gran desviación del valor real del tiempo debido a que su hardware no fue diseñado con esa prioridad. Lo mismo sucede en ordenadores, servidores, etc. Aumenta el rendimiento si se dedica el procesador principal a operaciones más importantes y añadir por separado un módulo RTC.

Además tiene la ventaja de que cuenta con una pila de botón CR2032 en la parte trasera, que alimenta el módulo en caso de desconexión de la alimentación, manteniendo actualizados los datos de fecha y hora (ver figura 15).

El módulo es compatible con Arduino y se alimenta a la misma tensión que él. La comunicación funciona por el puerto I2C, es decir mediante los pines SCL (CLOCK) y SDA (DATA). Los pines sobrantes sirven para añadir un sensor de temperatura que en este caso no es necesario. Tratándose de un módulo de hardware interno, se diseña un acoplamiento a la PCB mediante conectores.

Especificaciones del módulo RTC

- Tensión de alimentación: 5VDC.
- Tensión de batería: 3,3VDC.
- Consumo máximo de corriente: 0,3mA.
- Comunicación: I2C.
- Dirección: 0x68.
- Dimensiones: 36x22mm.
- Tipo de montaje: PCB mediante conectores de agujero pasante.

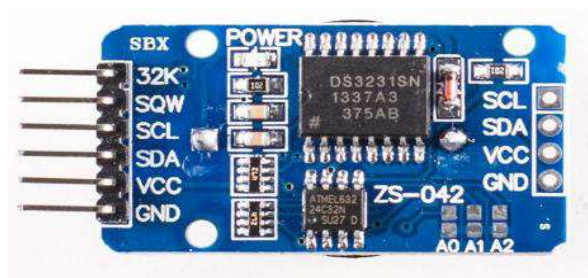


Figura 14: RTC vista superior planta



Figura 15: RTC vista inferior planta

3.2. Pantalla LCD

La pantalla LCD encargada de monitorizar los parámetros del dispositivo es de 20x4 caracteres (ver figura 16). Al ser compatible con Arduino comparten la misma tensión de alimentación. Este periférico en concreto, se conecta al puerto serie por los pines TX y RX, pero incluye en el módulo un adaptador (ver figura 17) para conectarla al Arduino por el puerto I2C, al igual que el módulo de reloj RTC; por lo que los dos periféricos compartirán el puerto con distintas direcciones.

Es una pantalla limitada puesto que no permite reproducir gráficos, pero tiene la ventaja de que el control para reproducir caracteres es muy sencillo. Este módulo no se integra en la PCB; de lo contrario se perdería la posibilidad de instalarlo en la tapa externa de un cuadro eléctrico o de una caja de registro.

Especificaciones de pantalla LCD

- Tensión de alimentación: 5VDC.
- Consumo máximo de corriente: 100mA.
- Comunicación: I2C.
- Dirección: 0x3F.
- Dimensiones: 99x60x22mm.
- Resolución: 20x4 caracteres.
- Ajuste de contraste.
- Ajuste de retro iluminación.
- Tipo de montaje: Superficial mediante separadores de nylon.



Figura 16: Pantalla LCD

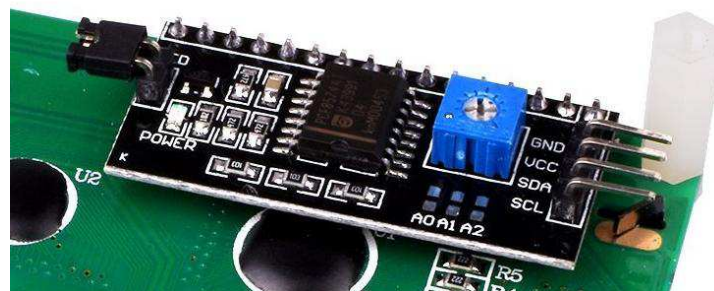


Figura 17: Backpack I2C en pantalla LCD

3.3. Fuente de alimentación común del sistema

Como la alimentación del controlador y los periféricos es de 5V, para simplificar el sistema, se diseñan los circuitos de acondicionamiento para ser alimentados a la tensión común de todo el conjunto, y así utilizar una sola fuente de alimentación.

En los circuitos de acondicionamiento de señales, las operaciones de escalado se realizan con amplificadores operacionales. Debido a las señales senoidales con las que operan los AO, estos requieren alimentación negativa, ya que para crear una tensión negativa en la salida respecto al punto cero (masa), el AO necesita su referencia negativa; por lo que la fuente debe suministrar tensiones de +5V y -5V.

Conviene que la fuente de alimentación pueda abastecerse con la tensión de red, evitando así el diseño de otra etapa de potencia de reducción y rectificación de tensión.

La fuente de alimentación que se elige es un modelo RECOM con una potencia suficiente para abastecer a todo el sistema teniendo en cuenta los consumos de Arduino y de los periféricos (ver figura 18). En el anexo A.1 se justifica la potencia total necesaria.

Especificaciones de fuente de alimentación RECOM RAC06-05DC

- Potencia de suministro: 6W.
- Tensión de entrada: 230VAC.
- Tensión de salida: +5VDC y -5VDC.
- Corriente máxima de salida: +600mA y -600mA.
- Tipo de montaje: PCB con orificio pasante.
- Dimensiones: 53,5x27,8x16,4mm



Figura 18: Fuente de alimentación RECOM

3.4. Elementos de protección del sistema

Como elementos de protección no rearmables, se instalan fusibles con calibre de 5x20mm para proteger el dispositivo contra sobrecargas de corriente y cortocircuitos, para evitar daños en los componentes de la PCB. El esquema de la figura 21 muestra la conexión de las protecciones.

Cada fusible quedará insertado en su propio porta fusibles de montaje en PCB por orificio pasante. En las figuras 19 y 20 se muestran los modelos de portafusibles y fusibles a montar en la PCB.



Figura 19: Portafusibles



Figura 20: Fusible 5x20mm

El fusible F1, es la protección de la alimentación principal del sistema, conectado en serie a la fase de la red eléctrica, a una tensión de 230V, situado aguas arriba del paralelo entre el transformador T1 y la fuente de alimentación. Se dimensiona para un valor de corriente nominal de 100mA.

$$I_{T1} = \frac{1,2VA}{230V} = 5,2mA$$

$$I_{FDA}(FULL\ LOAD) = 78mA$$

$$IAC_{MAX} = I_{FDA} + I_{T1} = 78mA + 5,2mA = 83,2mA$$

La corriente máxima de salida en la fuente de alimentación para las tensiones de +5V y -5V, es de 600mA. Estas dos líneas de alimentación quedan protegidas por los fusibles F2 y F3 respectivamente. Con lo cual, la corriente nominal de estos 2 fusibles es de 600mA.

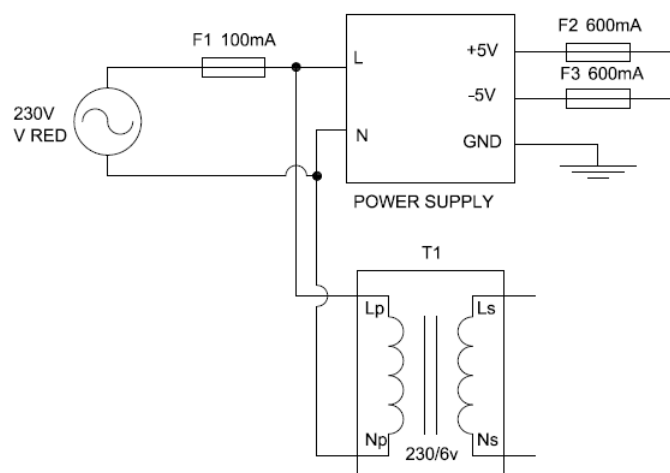


Figura 21: Esquema protecciones

3.5. Transductores de señal

Los transductores de señal deben actuar sobre la variable a medir y transformarla en una variable eléctrica con la que se pueda operar, para muestrearla en Arduino.

En el caso de la tensión, no debe transformar de una variable a otra, puesto que el rango de medida de Arduino es en voltaje, con lo cual, solo será necesario escalar el valor medido de la red (230V) a un valor menor. Para reducir tensión en corriente alterna se puede utilizar algo tan simple como un transformador de dos devanados.

En cambio la corriente, que también es una variable eléctrica, se debe transformar a otra variable eléctrica como es la tensión. Convertir el valor de corriente a tensión no es un problema que tenga que solucionarlo el transductor, ya que en el circuito de acondicionamiento no supondría ninguna complejidad.

Como la corriente a medir es del orden de decenas de amperios, con que el transductor sea capaz de reducir el valor medido a menor escala en la misma variable será suficiente. Con el mismo principio que el transductor de tensión, se puede medir corriente alterna mediante un transformador de corriente, conocida su relación de transformación.

Como sensor de medida de corriente se pueden utilizar métodos invasivos o no invasivos. La diferencia está en que con los métodos invasivos es necesario abrir el circuito para su aplicación; en cambio, con los no invasivos, no es así. Un transformador de corriente aporta simplicidad en el montaje, ya que como método no invasivo, no es necesario abrir el circuito.

3.5.1. ¿Qué es un transformador de tensión y/o corriente?

Un transformador o trafo es básicamente un dispositivo electro-magnético estático, constituido por dos arrollamientos o bobinas, que funciona según el principio de la ley de inducción de Faraday.

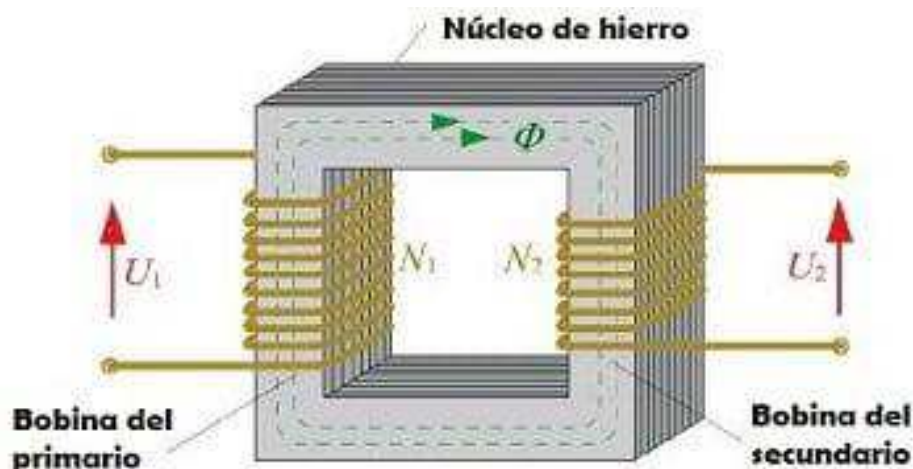


Figura 22: Transformador

El transformador es capaz de modificar, sin variar la frecuencia, los componentes de la energía eléctrica alterna de la entrada (voltaje y corriente) a otros distintos en la salida. Esto es posible gracias a la inducción mutua, es decir, el proceso por el cual una bobina de alambre induce magnéticamente un voltaje en otra bobina situada muy cerca de ella.

Sin entrar en detalles podemos desglosar un transformador de la siguiente manera:

- **Núcleo de chapas magnéticas:** este es el encargado de que el flujo magnético generado se mantenga dentro del transformador, evitando que se disperse por el aire lo cual provocaría pérdidas y reduciría la eficiencia del proceso. Se establece una configuración por láminas (o chapas) para evitar la aparición de corrientes de Foucault, ya que favorecen las pérdidas.
- **Bobinados primario y secundario:** están formados por conductores de cobre en forma de hilos redondos o rectangulares. Cuando una corriente eléctrica pasa a través del devanado primario, se desarrolla un campo magnético que induce una tensión en el devanado secundario.

Principio de funcionamiento de un transformador

Como dijimos anteriormente, el principio de funcionamiento de un transformador se basa en el fenómeno de la inductancia magnética entre dos circuitos. Recordemos que un transformador eléctrico tiene un devanado primario y uno secundario. Al conectar el bobinado primario a una fuente de tensión alterna, se produce una inducción de flujo magnético en el núcleo de hierro.

Si el segundo circuito está cerrado, por el principio de inducción magnética se generará una tensión inducida en él. Si suponemos que se trata de un transformador ideal, esta tensión inducida dependerá únicamente de la relación existente entre el número de espiras o vueltas del bobinado primario y del secundario, además de la tensión de entrada en el bobinado primario. Esta relación se conoce como "Relación de transformación". La fórmula sería la siguiente:

$$\frac{U_1}{U_2} = \frac{N_1}{N_2} = m$$

Si conocemos la relación entre el número de vueltas del bobinado primario y del secundario, podremos conocer la relación de transformación (m).

Vamos a suponer que trabajamos con un transformador ideal y que el devanado primario está formado por 2000 vueltas de hilo de cobre, mientras que el secundario está por 1000. En el circuito primario hay 220V y 3A. En base a ello tenemos que:

$$m = 2000 / 1000 = 2$$

Esto significa que en el secundario habrá la mitad de voltaje, es decir, 110V. Esta relación se representa, en este caso, como 2:1. Por otro lado, recordemos que no hay pérdida de potencia entre el primario y el secundario, por tanto la corriente en el secundario sería:

$$P_1 = P_2 \quad V_1 * I_1 = V_2 * I_2 \quad I_2 = (V_1 * I_1) / V_2 = m * I_1 = 2 * 3 \quad I_2 = 6A$$

Aquí se puede comprobar que la corriente aumenta si disminuye la tensión y disminuye si aumenta la tensión en la misma proporción ^[8].

3.5.2. Transductor de tensión

Se elige un modelo de transformador para montaje en PCB con orificio pasante, de núcleo de hierro, como sensor de tensión de red, con una relación de transformación de 230/6V.

La baja tensión del secundario facilita las operaciones de acondicionado de señal; y el aislamiento galvánico del transformador permite el acoplamiento magnético entre el circuito de señal del dispositivo, del de fuerza de la instalación, desacoplando los dos circuitos.

Especificaciones de transformador

- Relación de transformación: 230/6V.
- Potencia nominal: 1,2VA.
- Rango de frecuencia: 50-60Hz.
- Dimensiones (largo x ancho x alto): 32,3x27,3x23,8mm.
- Tipo de montaje: PCB de orificio pasante.

Las imágenes inferiores de las figuras 23 y 24, corresponden con la fotografía del transformador encapsulado elegido, y el esquema de su circuito equivalente respectivamente. Dicho circuito equivalente es con el que se modelará el circuito de acondicionamiento.

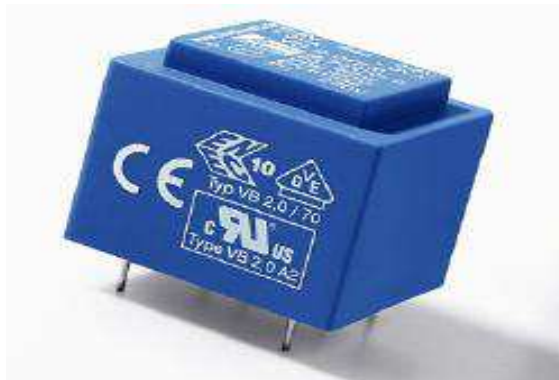


Figura 23: Transformador PCB 230/6V

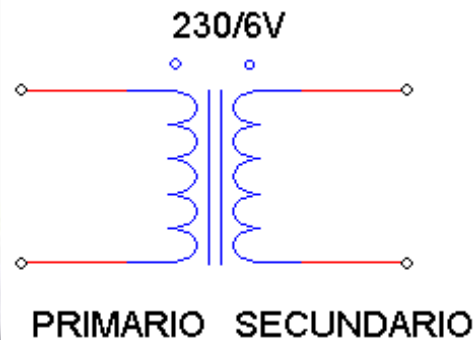


Figura 24: Esquema circuito equivalente transformador tensión

3.5.3. Transductor de corriente

En este caso se utiliza como sensor una pinza amperimétrica, modelo STC-013. Es un método de medida no invasivo. Está compuesto de un núcleo ferromagnético con un bobinado secundario. El conductor a medir, debe atravesar de forma perpendicular el núcleo ferromagnético, formando así, el bobinado primario de una sola vuelta, acoplado magnéticamente al bobinado secundario, creando el circuito equivalente de un transformador de 2 devanados.

Especificaciones de sensor de corriente STC-013

- Relación de transformación: 1/2000.
- Ratio de corriente: 100A: 0,5A.
- No linealidad: $\pm 3\%$.

En la imagen de la figura 25 se puede observar la forma de pinza del transformador de corriente. Con el entrehierro separado en dos piezas, permite abrirse para colocar el hilo conductor en su interior y volver a cerrarse.

La imagen de la figura 26 representa el esquema del circuito equivalente, viendo como único devanado el secundario.



Figura 25: Pinza STC-013

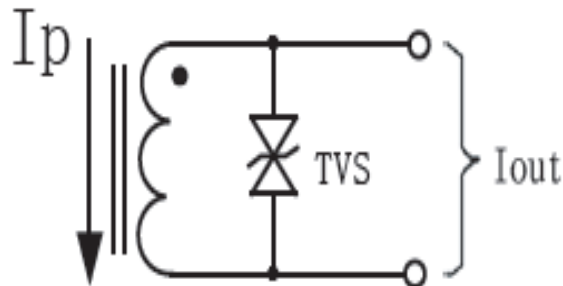


Figura 26: Esquema circuito equivalente transformador de corriente

3.6. Circuitos de acondicionamiento de señales

Las señales que se necesitan muestrear son dos: corriente y tensión de la red eléctrica. La señal de corriente se obtiene del hilo conductor de fase de la línea principal de alimentación de la vivienda; y la señal de tensión de la diferencia de potencial entre fase y neutro.

Son dos señales con forma de onda senoidal, que no se debe alterar, ya que al medir el desfase entre ellas para el cálculo de factor de potencia se toma como referencia el paso por cero. Se puede tomar cualquier otro punto de referencia, pero es necesario muestrearla en su forma natural, ya que en caso de rectificar la onda perderíamos toda la información respecto al tiempo.

Ambos circuitos de señal deben estar aislados de los circuitos de fuerza de la instalación eléctrica para evitar tanto anomalías eléctricas, como la perturbación de las señales analógicas con ruidos o armónicos generados por aparatos electrónicos conectados a la red.

La función de los circuitos de acondicionamiento de ambas señales es adaptar la señal a medir, al rango de señal de salida, es decir, al rango de entradas analógicas de Arduino, en cuanto a amplitud, siendo este entre 0 y 5V. Otro detalle a tener en cuenta es que las señales están formadas por semiciclos positivos y semiciclos negativos, y estos últimos no pertenecen al rango de entrada de Arduino. Con lo cual las señales se adaptan para conseguir un valor máximo de amplitud de 5Vpp pero desplazadas en el eje vertical, centradas en 2,5V, de forma que la señal completa esté en valor positivo para ser muestreable por el controlador.

Una forma de centrar las señales senoidales en un eje horizontal de 2,5V; es sumándoles una componente continua de dicha amplitud. Esta señal continua de 2,5V, al igual que las de tensión y corriente, será conducida al CAD de Arduino, de manera que el software la pueda muestrear continuamente en vez de interpretarla como un valor constante.

Para establecer relación entre el rango de señal de medida con el de salida, se impone un valor máximo para las dos señales. Cuanto mayor sea el límite máximo del valor a medir, peor será la precisión de la medida, ya que el CAD es de 10 bits para un rango de 0V a 5V, lo que significa que la cuantificación digital la realiza en pasos de 4,88mV.

Además, con la técnica utilizada, la señal a muestrear es simétrica en cuanto a amplitud, es decir, los valores que se muestrean en el semiciclo positivo son iguales al del semiciclo negativo pero desplazados 10 ms. En realidad equivale a muestrear la señal de 2,5Vp, en el rango de 2,5V a 5V, sobre 512 niveles, por lo que equivale a perder un bit de resolución en el CAD.

$$CAD = \frac{5Vpp}{2^{10} \text{ BITS}} = \frac{5Vpp}{1024 \text{ niveles}} = 4,88 \frac{mVpp}{nivel} \quad CAD' = \frac{2,5Vp}{2^9 \text{ BITS}} = 4,88 \frac{mV}{nivel}$$

Para el rango de corriente medida, se toma como referencia la potencia media contratada en una vivienda de 4,4KW, que se da con una corriente de 20A. Al añadir un margen a este valor medio, el límite máximo se impone en 25A, equivalente a una potencia máxima de 5750W.

$$Rango I_{MEDIDA} = [0 \sim 25]A \text{ RMS} \quad CAD = \frac{2*25*\sqrt{2}A}{2^{10} \text{ niveles}} = 0,07 \frac{A}{nivel}$$

En el caso de la tensión, el valor a medir en la red eléctrica es constante (230V), así que se establece con el máximo del rango de salida (5Vpp); de esta manera se aprovecha la máxima resolución del CAD de Arduino, para muestrear con claridad el paso por cero, y aumentar la precisión en el cálculo del desfase temporal.

$$Rango V_{MEDIDA} = [0 \sim 230]VRMS \quad CAD = \frac{2 \cdot 230 \cdot \sqrt{2}V}{2^{10}niveles} = 0,64 \frac{V_{pp}}{nivel}$$

En el desarrollo de los circuitos de acondicionamiento es necesario implementar bloques funcionales de seguidor de línea para evitar acoplamiento de impedancia entre señales y circuitos sumadores de señales. Estos bloques se desarrollan a partir de amplificadores operacionales.

En la elección de un modelo genérico de amplificador operacional para este circuito, se busca el mínimo desfase temporal entre la señal de entrada y salida, y el mínimo *offset*; para que la señal muestreada por Arduino quede lo menos perturbada posible.

Una familia de amplificadores operacionales que cumple los requisitos es la TL08XX, que dispone de diferentes modelos de integrados formados por 1, 2, 4 y 8 AO; para montaje en PCB. En función del número de AO necesarios, se decidirá en siguientes apartados el modelo a elegir.

Antes de continuar con los diseños de acondicionamiento, se procede al análisis de los bloques funcionales a implementar, ya que la configuración de los AO es la misma:

- **Seguidor de línea:** Con realimentación negativa para mantener la tensión de salida igual a la entrada V_{IN}^+ (ver esquema de la figura 27).

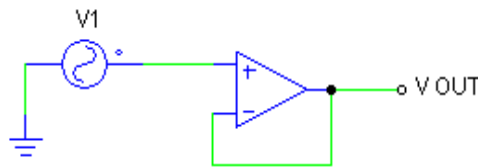


Figura 27: Esquema AO seguidor de línea

$$V_{IN}^+ = V_{IN}^-; \quad V_{OUT} = V_{IN}^-; \quad V1 = V_{IN}^+; \quad V_{OUT} = V1$$

- **Circuito sumador no inversor:** Sumatorio de señales con ganancia ajustable (ver esquema de la figura 28).

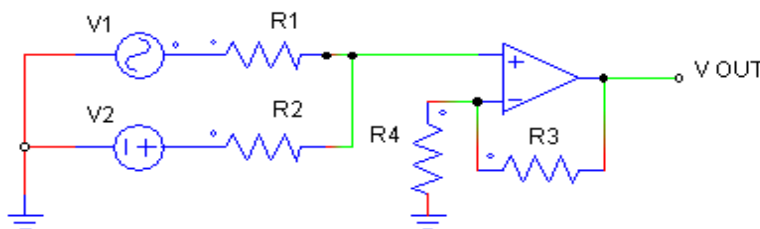


Figura 28: Esquema AO sumador no inversor

$$\text{Si } R1=R2; \quad V_{OUT} = \left[\left(\frac{1}{2} \right) * (V1 + V2) * \left(1 + \frac{R3}{R4} \right) \right]$$

$$\text{Si } R1=R2 \text{ y } R3=R4; \quad V_{OUT} = (V1 + V2)$$

3.6.1. Circuito de acondicionamiento de señal de tensión

El componente principal de este circuito es el transformador T1 que actúa como transductor de señal. Con un divisor de tensión alimentado con el secundario del transformador T1 se ajusta la señal senoidal mediante el potenciómetro P1, para conseguir en la entrada del AO una amplitud de 5Vpp equivalentes a la señal medida con el primario del transformador, de 230V eficaces de la red.

El AO configurado como seguidor de línea, evita que los siguientes componentes que se conecten al circuito puedan acoplarse al divisor de tensión y variar la caída de tensión ajustada en él. En el esquema de la figura 29 se muestra la disposición de los componentes enunciados hasta ahora.

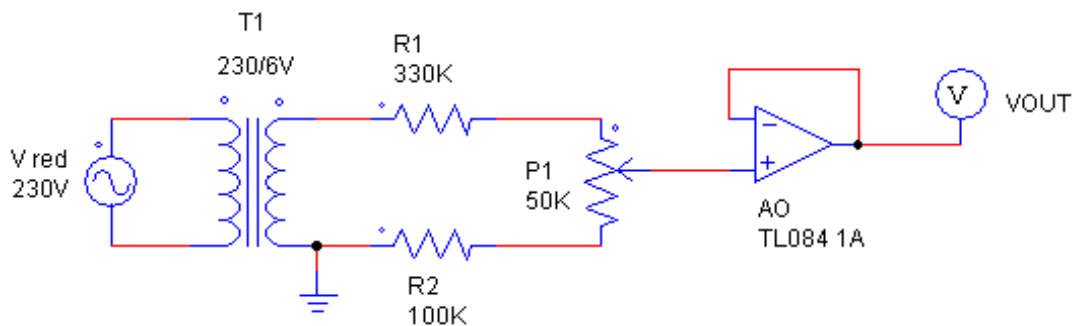


Figura 29: Esquema divisor de tensión de la señal de tensión alterna

$$I_{SECUNDARIO} = \frac{6V}{480K\Omega} = 12,5\mu A \quad S_{SECUNDARIO} = 6V * 12,5\mu A = 0.075mVA$$

$$V_{OUT_{MIN}} = 6V * \sqrt{2} * \frac{100K\Omega}{480K\Omega} = 1,76Vp \quad V_{OUT_{MAX}} = 6V * \sqrt{2} * \frac{150K\Omega}{480K\Omega} = 2,65Vp$$

Los valores de las resistencias y los potenciómetros de todos los divisores de tensión están calculados para que mediante el ajuste de los potenciómetros, se regulen los valores de amplitud de salida en un rango cercano a 1Vp para compensar imprecisiones y pérdidas en los componentes del circuito. Los valores de las resistencias conectadas a amplificadores operacionales se calculan para limitar las corrientes de entrada o salida a 20mA, debido a la no idealidad de los mismos.

Además, de forma general, para evitar calentamiento, pérdidas de potencia y sobredimensionado de componentes, se utilizan valores altos de resistencia para limitar la corriente a mA.

Se simula la señal con el potenciómetro P1 ajustado para dar salida a 2,5Vp, y se comprueba la salida del AO realimentado (ver gráfico de la figura 30).

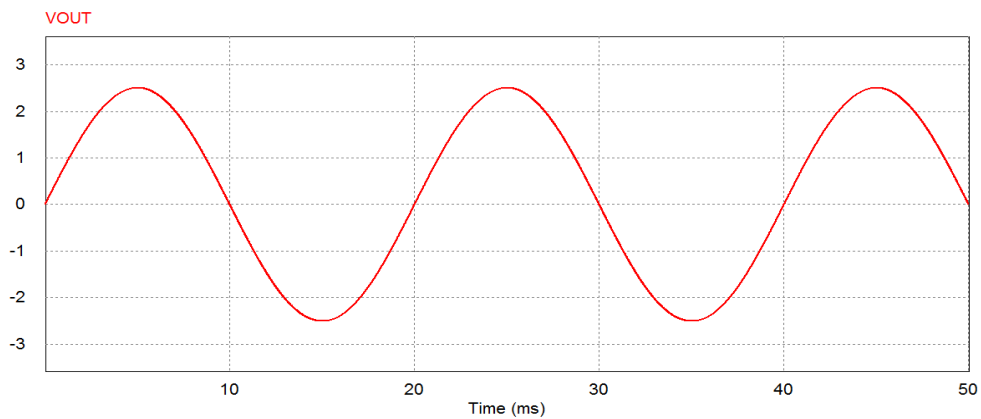


Figura 30: Simulación PSIM

El ajuste de la tensión de 2,5VDC se obtiene con el potenciómetro P0 de otro divisor de tensión alimentado entre los +5V y GND (masa, 0V) de la fuente de alimentación del sistema (ver esquema de la figura 31).

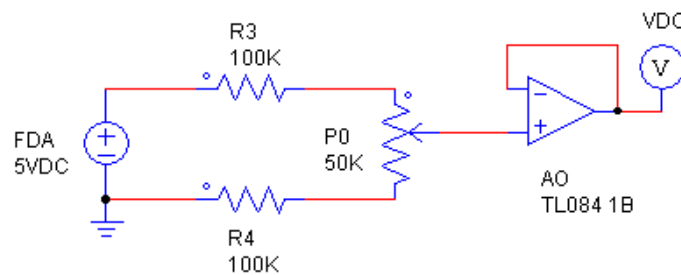


Figura 31: Esquema divisor de tensión señal continua

$$VDC_{MIN} = 5V * \frac{100K\Omega}{250K\Omega} = 2V$$

$$VDC_{MAX} = 5V * \frac{150K\Omega}{250K\Omega} = 3V$$

$$I_{DIVISOR} = \frac{5V}{250K\Omega} = 20\mu A$$

$$P_{DIVISOR} = 5V * 20\mu A = 0,1mW$$

Tras la simulación se obtiene la señal continua a frecuencia 0Hz (ver figura 32).

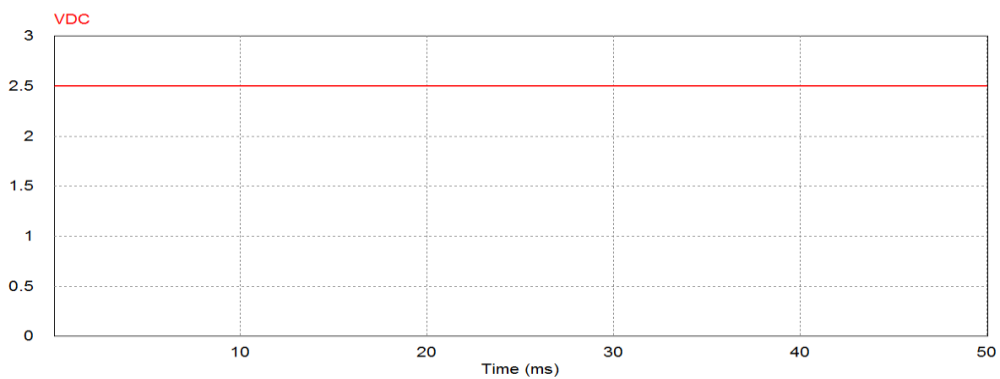


Figura 32: Simulación PSIM

El siguiente paso consiste en sumar las dos señales ajustadas en los divisores de tensión, mediante un bloque sumador no inversor (ver esquema de la figura 33).

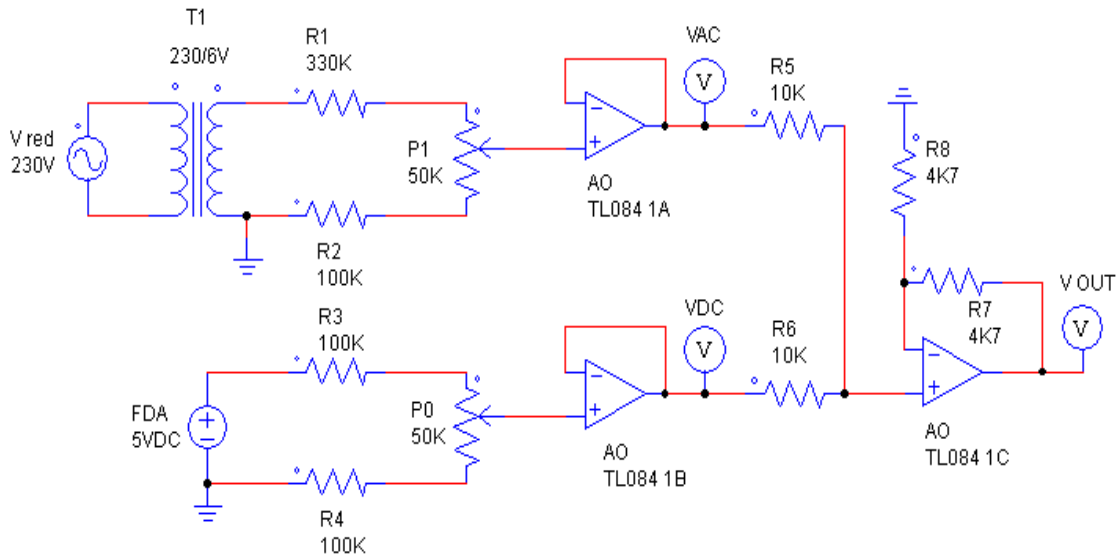


Figura 33: Esquema circuito acondicionamiento de señal de tensión

En el circuito sumador, se mantiene el criterio de $R5=R6$ y $R7=R8$, para simplificar el diseño. Las resistencias del circuito sumador podrían ser de cualquier otro valor, o incluso todas iguales, siempre que garanticen limitar las corrientes de entrada y salida de los amplificadores operacionales. La única razón por la que se imponen distintos valores es solo por diferenciación en los montajes.

$$IR5_{MAX} = IR6_{MAX} = \frac{5V}{10K\Omega} = 0,5mA \quad IR7_{MAX} = IR8_{MAX} = \frac{5V}{4K7\Omega} = 1,1mA$$

Como se puede observar en la simulación de la figura 34, la tensión de salida (VOUT) es la suma de las dos señales ajustadas en los divisores de tensión ($VDC + VAC$), centrada en un eje horizontal, a una amplitud de 2,5V. De esta forma la señal ya está acondicionada al rango de entradas analógicas de Arduino, suponiendo un valor de medida de tensión de 230V RMS.

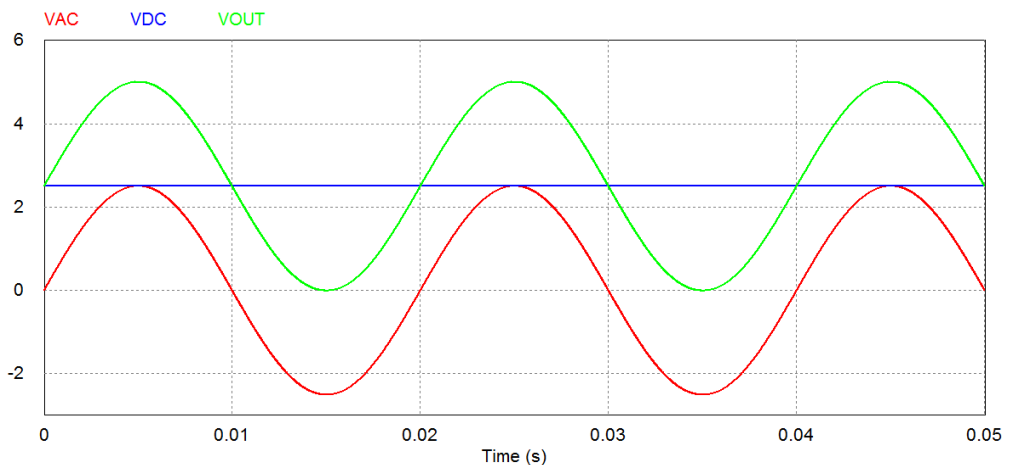


Figura 34: Simulación PSIM

3.6.2. Circuito de acondicionamiento de señal de corriente

El transformador de corriente funciona como una fuente de corriente, así que el primer paso es convertir la corriente inducida en tensión, acoplado una carga resistiva en paralelo, al secundario del transformador (ver esquema de la figura 35).

Con el mismo criterio que en el circuito de medida de tensión, se alimenta un divisor de tensión con el secundario del transformador de corriente. El potenciómetro P2 se dimensiona para una conseguir una caída de tensión ajustable para compensar pérdidas, y se añade un seguidor de línea.

En el análisis, se modela la corriente del circuito de la instalación eléctrica a medir, como una fuente de corriente senoidal, suministrando el valor máximo de corriente establecido en el rango de medida del sensor (25A).

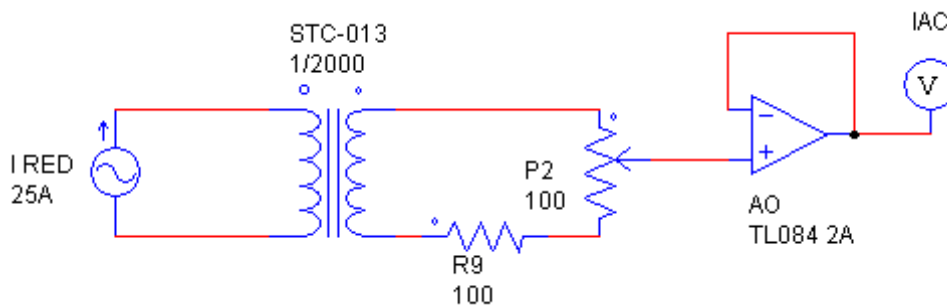


Figura 35 Esquema divisor de tensión de señal de corriente alterna

$$I_{SECUNDARIO} = 25 \frac{1}{2000} A = 12,5mA \quad S_{MAX SECUNDARIO} = 200\Omega * (12,5mA)^2 = 0,03VA$$

$$V_{IAC_{MIN}} = 12,5mA * \sqrt{2} * 100\Omega = 1,77Vp \quad V_{IAC_{MAX}} = 12,5mA * \sqrt{2} * 200\Omega = 3,53Vp$$

Después del ajuste del potenciómetro P2 a la mitad de su resistencia aproximadamente, se puede observar en la simulación del circuito (figura 36), que la señal de salida del AO es una senoidal de 5Vpp, para un supuesto caso de medida de un valor de corriente alterna de 25A.

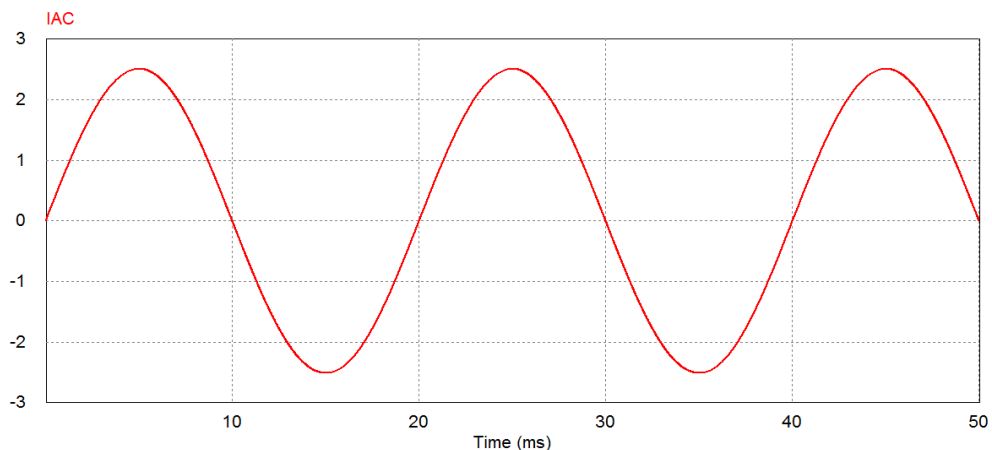


Figura 36: Simulación PSIM

Una vez escalada y transformada a voltaje la señal de corriente, se aplica el mismo método de acondicionamiento que en el circuito de medida de tensión, incluyendo el mismo divisor de tensión de corriente continua, y los mismos valores de resistencias en el bloque sumador (ver esquema de la figura 37).

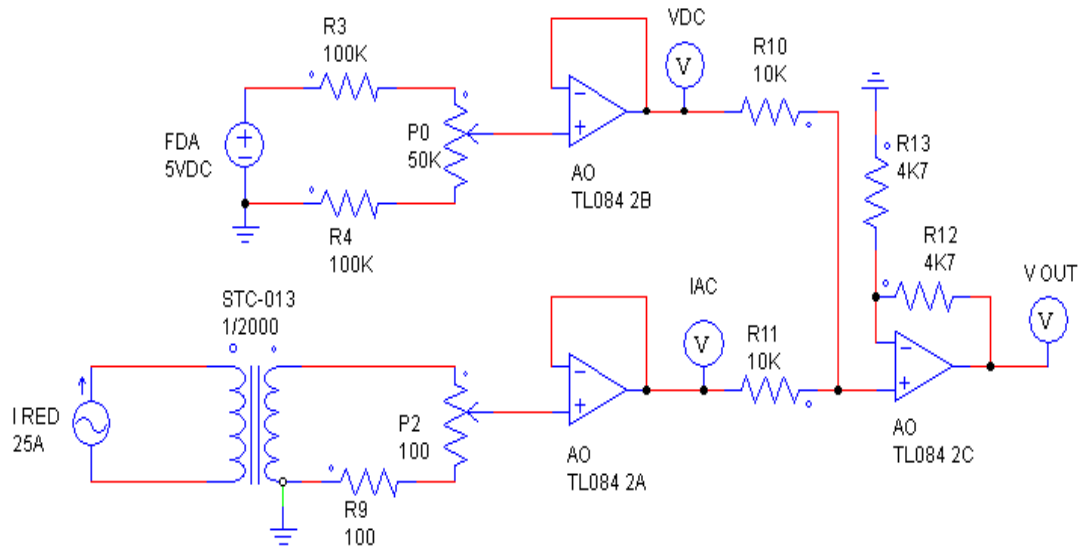


Figura 37: Esquema de circuito de acondicionamiento de señal de corriente

Una vez más, se comprueba la señal de salida final acondicionada (VOUT) del circuito por medio de la simulación informática de la figura 38, para dar por finalizado el diseño teórico de los circuitos de acondicionamiento de señal, y dar paso a la prueba práctica de funcionamiento.

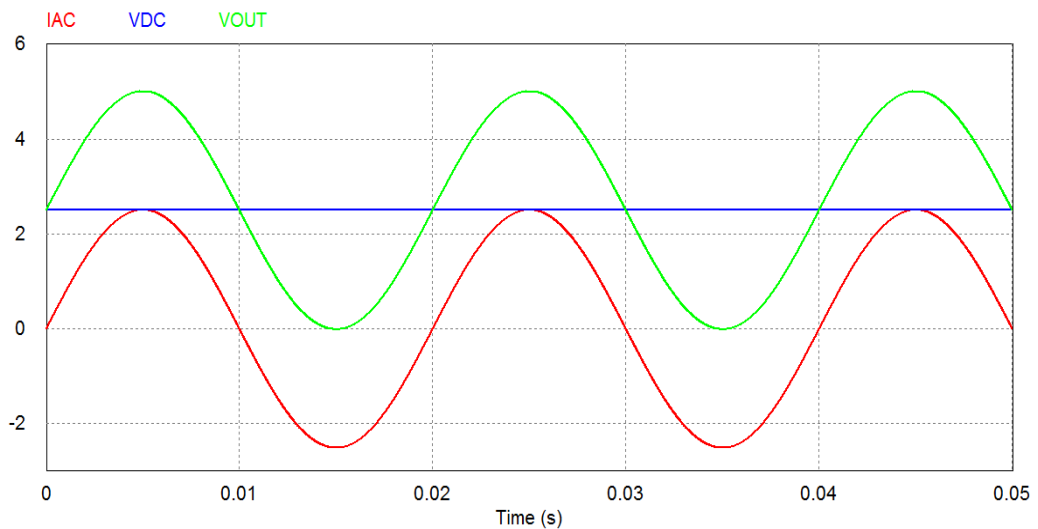


Figura 38: Simulación PSIM

3.7. Pulsadores y LEDs

Se implementan pulsadores para realizar maniobras de control por parte del usuario; tanto de montaje interno (integrado en la PCB), como de montaje externo (cableado a regleta de la conexión en la PCB):

- Pulsador de Reset de Arduino:** A pesar de que Arduino cuenta con su pulsador propio de Reset, la forma en que se conecta a la PCB impide el acceso a este. Con lo cual se integra un pulsador en la PCB para dicha función; conectado entre el pin de Reset de Arduino, y masa (ver figura 39).

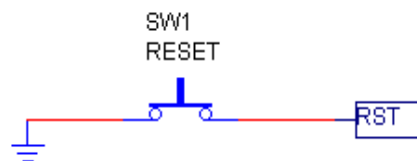


Figura 39: Esquema pulsador de Reset

- Pulsador de Encendido/Cambio de pantalla LCD:** De montaje externo como la pantalla LCD. Conectado entre el pin 7 de entrada/salida digital de Arduino, y +5V, para interrumpir el proceso del software, y realizar el encendido o cambio de pantalla en el caso de que esta ya lo estuviese. Conviene realizar un circuito de forma que cuando el pulsador no esté activado, el pin 7 esté puesto a masa con una resistencia, para evitar falsas interrupciones por ruido (ver esquema de la figura 40).

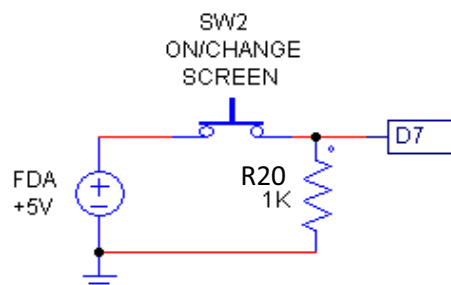


Figura 40: Esquema pulsador de encendido de pantalla LCD

- Conmutador de ON/OFF de sistema:** De montaje externo, conectado a la fase de alimentación del dispositivo (ver esquema de la figura 41).

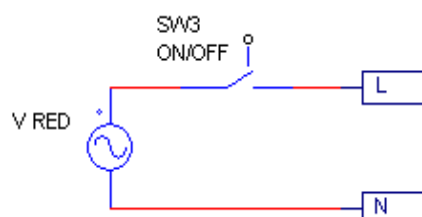


Figura 41: Esquema de conmutador de encendido

Se añaden dos LEDs como testigo luminoso de encendido, uno integrado en la PCB (LED1), y otro externo (LED2), a colocar por el usuario (ver esquema de la figura 42).

$$R18 = R19 = \frac{(5V - 2V)}{0,02A} = 150\Omega$$

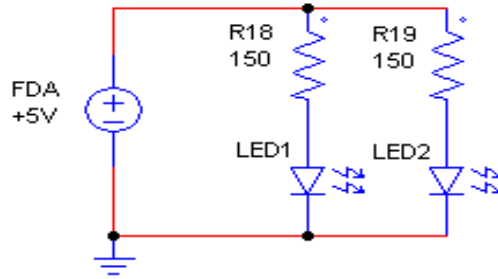


Figura 42: Esquema de LEDs

4. Prueba de funcionamiento y correcciones

Se realiza una prueba práctica del funcionamiento de los circuitos de acondicionamiento diseñados. No tiene sentido comprobar nada más, debido a la simplicidad de algunos bloques, y a la prefabricación de otros.

Para el montaje se utilizan varias placas de prueba, debido a la poca maniobrabilidad del transformador y la fuente de alimentación. Simplemente sirve para verificar la forma de onda de salida de ambos circuitos, con medida de tensión de red, y una corriente de unos 10A; no es el momento de realizar ajustes y/o calibraciones.

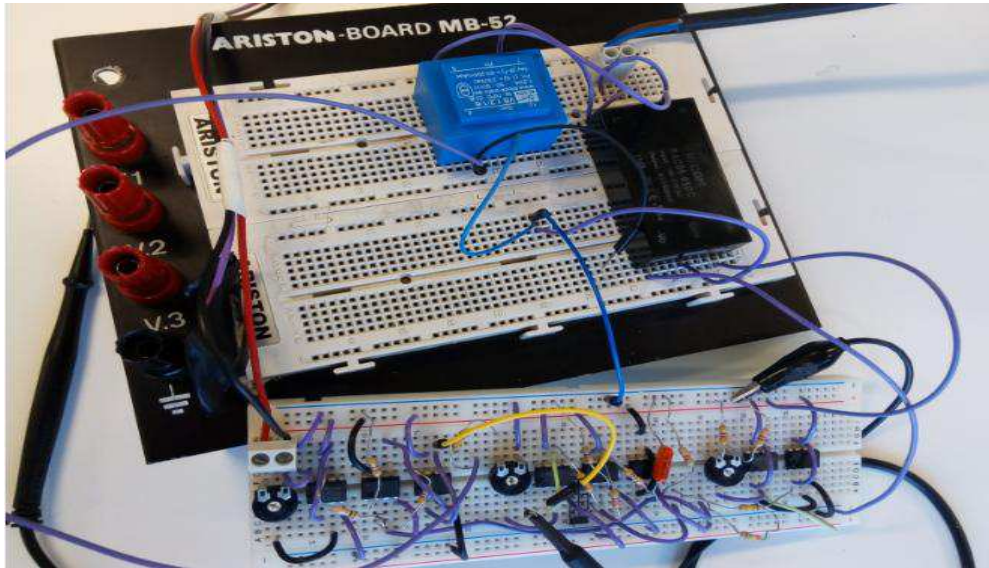


Figura 43: Fotografía de montaje en placa de pruebas

Como se puede observar en la foto del osciloscopio (ver figura 44), se consigue la forma de onda deseada para la medida de tensión de red, pero no consigue la amplitud con la que se había diseñado.

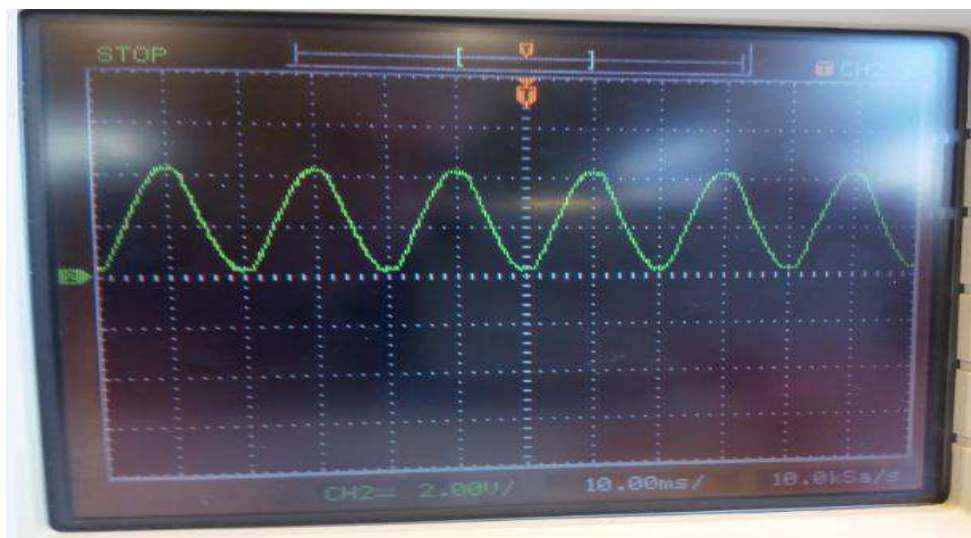


Figura 44: Fotografía de imagen de osciloscopio

Ahora se va a enumerar una serie de problemas que han surgido en el montaje, con sus correspondientes soluciones o cambios en el diseño.

4.1. Tensión de salida en los amplificadores operacionales

En los amplificadores operacionales, debido a su no idealidad; alimentados entre +5V y -5V, la salida en saturación nunca llega hasta la tensión de alimentación. En este caso hasta 4,2V como máximo. Este problema hace replantear el escalado de las señales de salida, así como el rango de salida.

Como los divisores de tensión tienen un margen de ajuste, se calibran los potenciómetros para que la máxima señal de salida sea de 4Vpp y centrada en un eje horizontal de 2V, por lo que el nuevo rango de salida es [0~4Vpp].

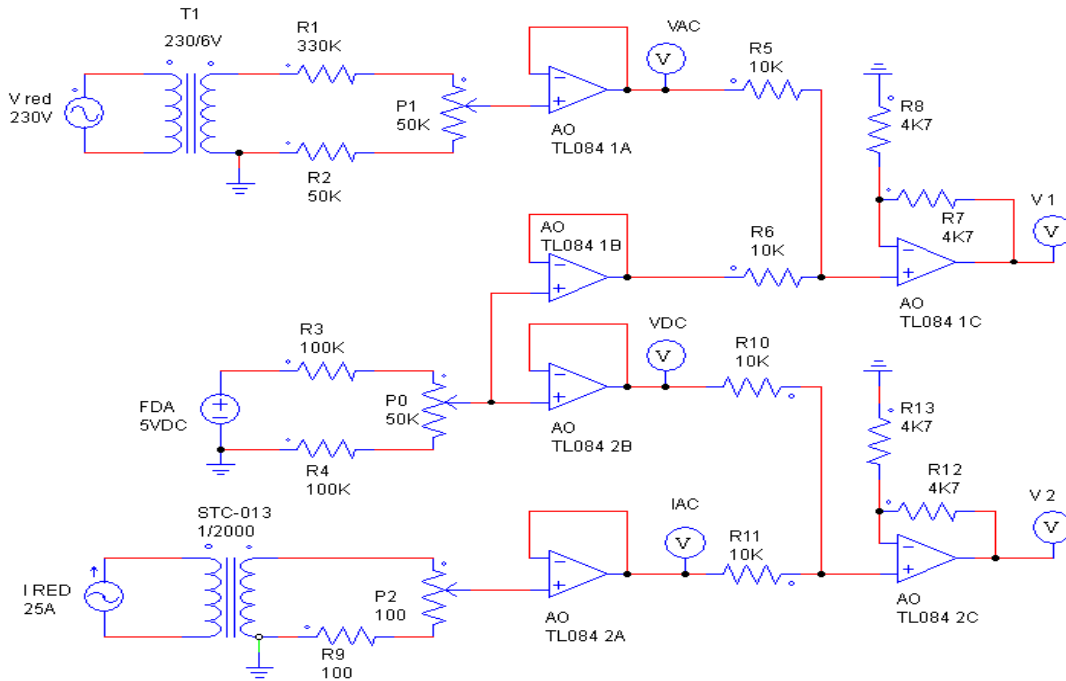


Figura 45: Esquema de circuitos de acondicionamiento rediseñado

A continuación se simula el circuito de la figura 45 para verificar el ajuste de las señales (ver gráfico de la figura 46). Para ver las señales con más claridad, a las de corriente se les suma un offset de desfase.

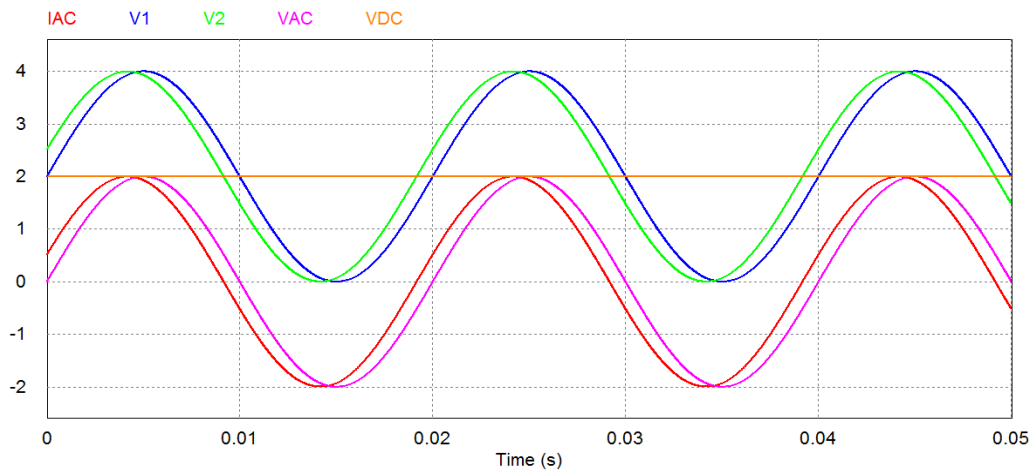


Figura 46: Simulación PSIM

4.2. Ruido acoplado a la señal medida

Se observa en la imagen de la figura 47, fotografiada del osciloscopio, una señal de alta frecuencia (6,66MHz), con un periodo de 150ns; acoplada a las señales medidas y una vez acondicionadas, en el tramo final del circuito.

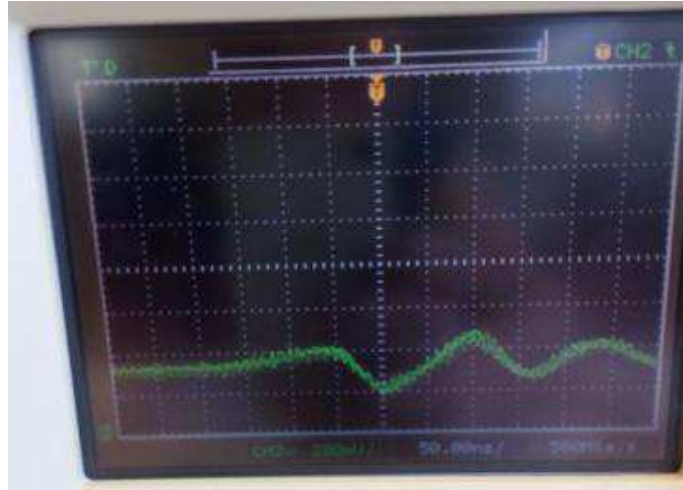


Figura 47: Fotografía de imagen de osciloscopio (perturbacion)

Puesto que es una señal de muy alta frecuencia, comparada con la señal portadora de 50 Hz; es posible atenuarla con un filtro RC paso bajo.

Se implementan dos filtros RC paso bajo iguales, con una frecuencia de corte de 100 Hz. Cada uno de ellos para cada circuito de acondicionamiento, con un seguidor de línea, como etapa final de ambos circuitos de acondicionamiento. Además, se acopla una resistencia en serie al filtro paso bajo, solo para considerarla en el diseño de la PCB, por si se da el caso de que sea necesario limitar corrientes. En caso de que no sea necesaria, se puenteará (ver esquema de la figura 48).

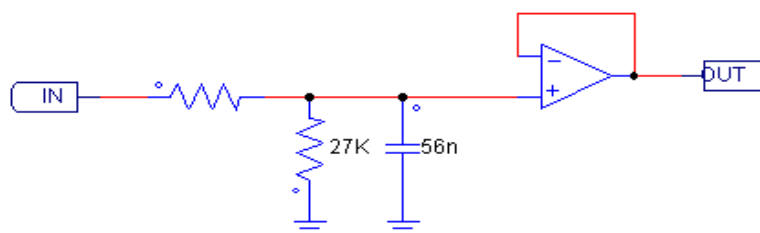


Figura 48: Esquema de filtro RC paso bajo

$$f_{CORTE} = \frac{1}{\omega * R * C} \quad C = 56nF \quad f_{CORTE} = 100Hz$$

$$I_{C_{MAX}} = \frac{4V}{X_C} = 4V * 2 * \pi * 50Hz * 56F * 10^{-9} = 70\mu A$$

$$R = \frac{1}{\omega * C * f_{CORTE}} = \frac{1}{2 * \pi * 100Hz * 56F * 10^{-9}} = 28,42K\Omega$$

Tras el montaje del filtro RC, se analiza la señal de salida con el osciloscopio, y se observa que el ruido no se atenúa del todo. En la fase de programación del software se verá hasta qué punto influyen estas perturbaciones.

4.3. Tensión en el secundario del transformador

En el transformador encapsulado, se supone que la tensión nominal en el secundario es de 6V eficaces, pero según el voltímetro, esta es de 7,7V eficaces. Este aumento de tensión provoca un desajuste en el divisor de tensión, con lo cual, se vuelve a calcular el valor de las resistencias, con el mismo criterio que antes. La imagen 49 muestra el esquema del divisor de tensión rediseñado.

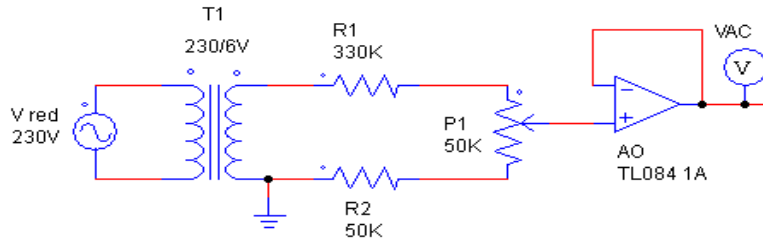


Figura 49: Esquema divisor de tensión de señal de tensión alterna rediseñado

$$I_{SECUNDARIO} = \frac{7,7V}{430K\Omega} = 17,9\mu A \quad S_{SECUNDARIO} = 7,7V * 17,9\mu A = 0,14mVA$$

$$VAC_{MIN} = 7,7V * \sqrt{2} * \frac{50K\Omega}{430K\Omega} = 1,27Vp \quad VAC_{MAX} = 7,7V * \sqrt{2} * \frac{100K\Omega}{430K\Omega} = 2,53Vp$$

4.4. Diseño final

Corregidos los problemas ocasionados por la fase de prueba práctica, el diseño final del circuito de acondicionamiento a desarrollar en la PCB es el que se muestra en la figura 50.

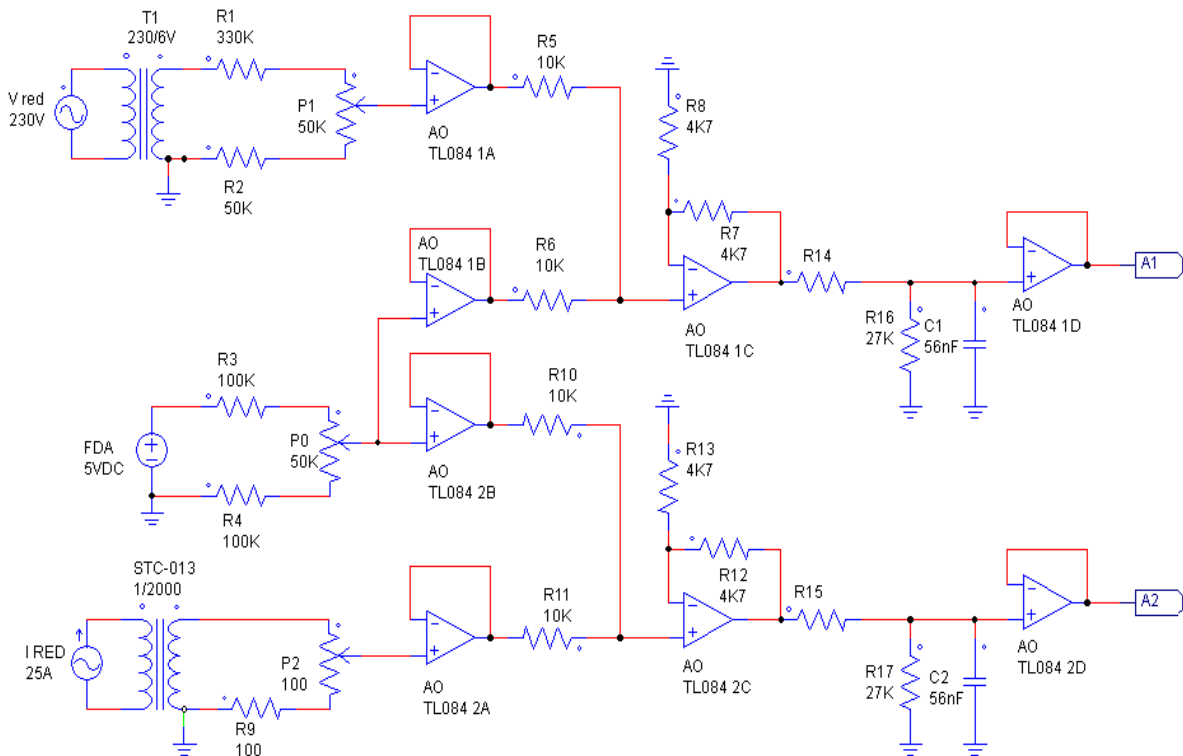


Figura 50: Esquema de diseño final de circuitos de acondicionamiento

5. Diseño de la PCB

El diseño de la PCB se desarrolla mediante “Design Spark”. Es una herramienta informática gratuita creada por “RS Components” que facilita el diseño de placas electrónicas, y lo convierte a archivos de formato GERBER. Este tipo de formato es el que utiliza la fresadora CNC con la que se va a fabricar la PCB.

En primer lugar, hay que enumerar una lista de los componentes y módulos que se van a integrar en la PCB; puesto que ni se integran todos los componentes enunciados hasta ahora, ni se enuncian todos los componentes necesarios para el dispositivo. Con lo cual se hará mención de los componentes que no se han tenido en cuenta hasta ahora, así como de los ya enunciados, como recordatorio.

Design Spark contiene en sus librerías información sobre las dimensiones y geometría de la mayoría de componentes genéricos. Una vez completada la lista de componentes, hay que verificar la existencia de todos los componentes en las librerías de Design Spark, ya que es indispensable en el diseño de la PCB contar con las características geométricas de cada componente. En caso de que no exista alguno de los componentes en las librerías, habrá que diseñarlo.

Con todos los componentes a disposición se procederá al diseño del diagrama esquemático, para su posterior volcado al diseño final de la PCB.

5.1. Componentes de PCB

- **Transformador y fuente de alimentación:** Como se comentó en apartados anteriores, los dos componentes son de montaje en PCB con la idea de integrarlos en ella.



Figura 51: Transformador PCB 230/6V



Figura 52: Fuente de alimentación RECOM PCB

- **Módulos de Arduino y RTC:** No se integrarán por montaje mediante soldadura pero si por conectores de pines, con lo cual es necesario considerarlos en el diseño debido a las pistas requeridas para las conexiones de datos y alimentación.



Figura 53: Arduino YUN

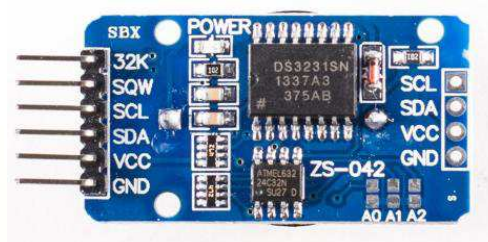


Figura 54: módulo RTC

- **Resistencias:** Las resistencias montadas serán de 1/8W de potencia, con un 1% de tolerancia, en busca de obtener la máxima precisión.



Figura 55: Resistencia 1/8W 1% de tolerancia

- **Potenciómetros:** Los potenciómetros serán de 3 pines, con ajuste giratorio de una vuelta.



Figura 56: Potenciómetro giratorio

- **Condensadores:** Los condensadores de los filtros de paso bajo serán de poliéster, debido a que los cerámicos tienen una mayor absorción y producen un efecto de memoria en la señal, con lo cual para montajes de filtros se desaconseja.

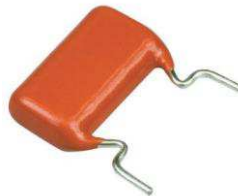


Figura 57: Condensador de poliéster

- **Circuito integrado TL084:** Puesto que en el circuito de acondicionamiento se utilizan 8 AO; para el diseño de la PCB se agrupan en 2 circuitos integrados (TL084), para montaje en PCB con orificio pasante, de 4 amplificadores operacionales cada uno. De esta manera se reduce el espacio necesario, el número de pistas, y la distancia de las mismas.



Figura 58: TL084

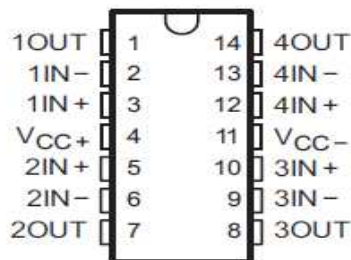


Figura 59: TL084 pinout

- **Zócalos de 14 pines:** En lugar de soldar a la PCB los circuitos integrados TL084, se sueldan zócalos de 7x2 pines en los que montar dichos integrados. Además de la ventaja que aporta en la sustitución de los integrados debido a no estar soldados, se evitan ser sometidos a la temperatura del proceso de soldadura y correr el riesgo de dañar el componente.



Figura 60: Zócalo 14 pines

- **Porta fusibles:** Para proteger los circuitos, se montan 3 porta fusibles para calibre de 5x20mm, de montaje en PCB con orificio pasante. Uno de ellos conectado en la entrada de la fase de alimentación; y los otros dos, en las salidas de tensión de +5V y -5V, de la fuente de alimentación.



Figura 61: Portafusibles

- **Fusibles:** En los porta fusibles montados en la PCB se instalan fusibles del tamaño 5x20mm, para proteger el dispositivo contra sobre intensidades y cortocircuitos.



Figura 62: Fusible 5x20mm

- **Pulsador:** Para la función de Reset en la PCB, un pulsador de 6x6mm de montaje por orificio pasante de 4 pines, de estado normalmente abierto.



Figura 63: Pulsador PCB

- **LED:** Testigo de encendido integrado en la PCB, diodo LED de color verde con diámetro de 5mm.



Figura 64: LED 5mm verde

- **Regletas de conexión y conectores macho-hembra:** Para las conexiones externas y los montajes superficiales de los módulos de Arduino y RTC. De montaje en PCB con orificio pasante.



Figura 65: Clema

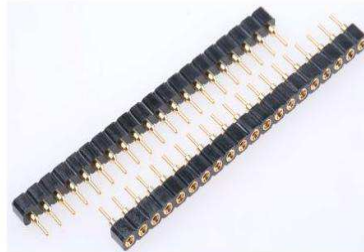


Figura 66: Conectores hembra



Figura 67: Conectores macho

- **Separadores y tornillería de nylon:** Para aportar sujeción mecánica a los módulos RTC y Arduino, además de separar la PCB de una base de apoyo. Formado por nylon, material no conductor, para evitar cortocircuitos o anomalías eléctricas no deseados.



Figura 68: Tornillería de nylon

5.2. Diseño de componentes en Design Spark

Como era de esperar, los componentes no tan comunes, no están disponibles en las librerías de Design Spark, por lo que este apartado queda destinado a la edición de componentes.

Para ello creamos una librería nueva, llamada “TFG”, en la que contener estos componentes específicos. La edición de nuevos componentes está basada en 3 partes: símbolo esquemático, símbolo PCB (footprint o huella), y vista 3D.

El símbolo PCB es básico, ya que es la vista de planta de las dimensiones reales del componente, con su forma geométrica y su distribución de pines de conexión. Para realizar esta parte con precisión absoluta, se consultan las referencias del datasheet del componente (ver anexo A.8).

En este símbolo quedará detallada la silueta del componente, la colocación y distribución de los pads con sus dimensiones, y formas geométricas, teniendo en cuenta el diámetro de los pines del componente para su montaje.

El símbolo esquemático no es indispensable, pero si es recomendable editarlo si se realiza el diagrama esquemático como es este caso. Tan solo es una representación del componente para interpretarlo en un esquema, por lo que lo editaremos a nuestra voluntad.

El único símbolo del que se va a prescindir es el de la vista 3D, ya que solo es útil para obtener una vista previa 3D de la fabricación y montaje de la PCB con todos sus componentes.

Se editan los símbolos esquemáticos (imágenes de la izquierda), y PCBs (imágenes de la derecha), de los componentes no encontrados en las librerías:

- **Porta fusible:** Aunque en la parte de diagrama esquemático lo que se ve es el fusible, lo importante son las dimensiones en planta del porta fusibles.

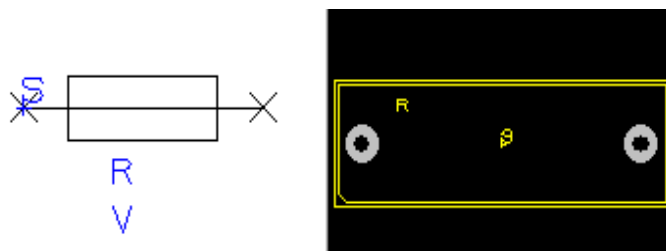


Figura 69: Símbolo esquemático y footprint de portafusibles

- **Fuente de alimentación RECOM:**

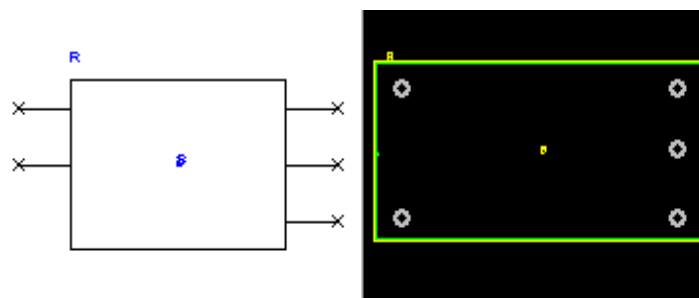


Figura 70: Símbolo esquemático y footprint de fuente de alimentación RECOM

- Transformador de 230/6V.

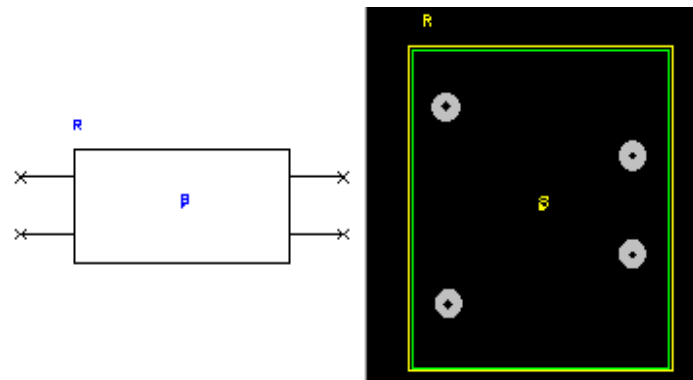


Figura 71: Símbolo esquemático y footprint de transformador PCB

- Regleta de 2 hilos.

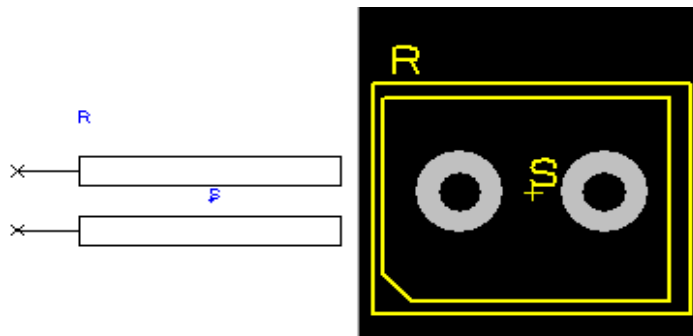


Figura 72: Símbolo esquemático y footprint de clema

- Conectores lineales de 6, 8 y 10 pines; y conector de 2x3 pines.

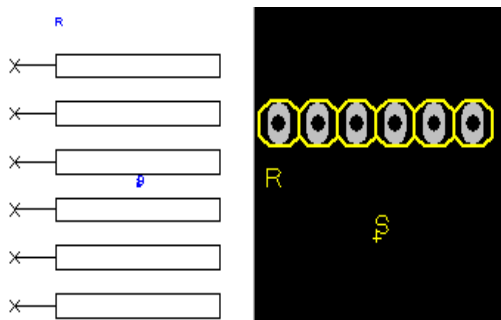


Figura 73: Esquemático y footprint de conector 6

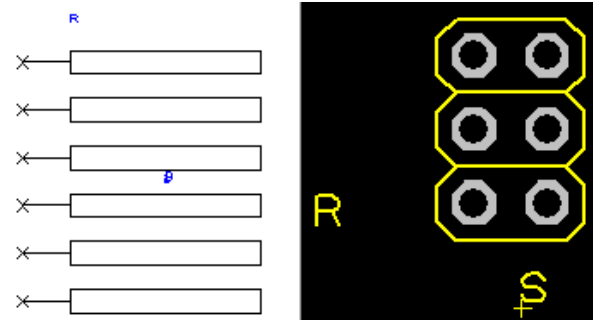


Figura 74: Esquemático y footprint de conector 2x3

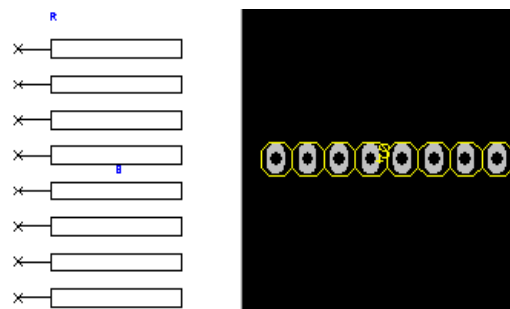


Figura 75: Esquemático y footprint de conector 8

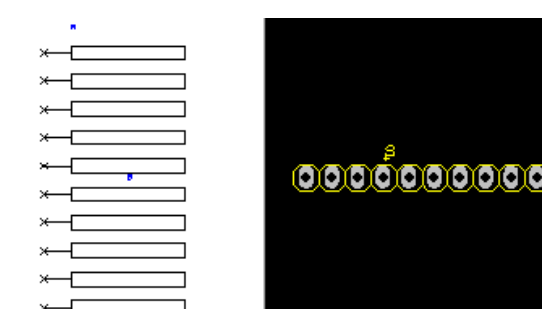


Figura 76: Esquemático y footprint de conector 10

En los casos de los módulos de Arduino y RTC, en vez de editarlos como componentes, se diseña una plantilla a medida en formato PCB de Design Spark. En estas plantillas se detalla la silueta del módulo, la distribución e identificación de conectores y pines, y los taladros para el montaje de tornillería de nylon.

Estas plantillas servirán para situar con precisión en la PCB los conectores de cada uno de los módulos. También servirán como referencia del espacio ocupado por cada uno de ellos, para conseguir una distribución uniforme de los componentes y evitar el contacto entre ellos.

- Plantilla de módulo de Arduino:** Vista en planta del Arduino (cara arriba), aunque en la PCB se montará cara abajo. A pesar de descargarla de la página oficial de Arduino, como una plantilla “Shield”, se verifica con el calibre todas las medidas entre conectores.

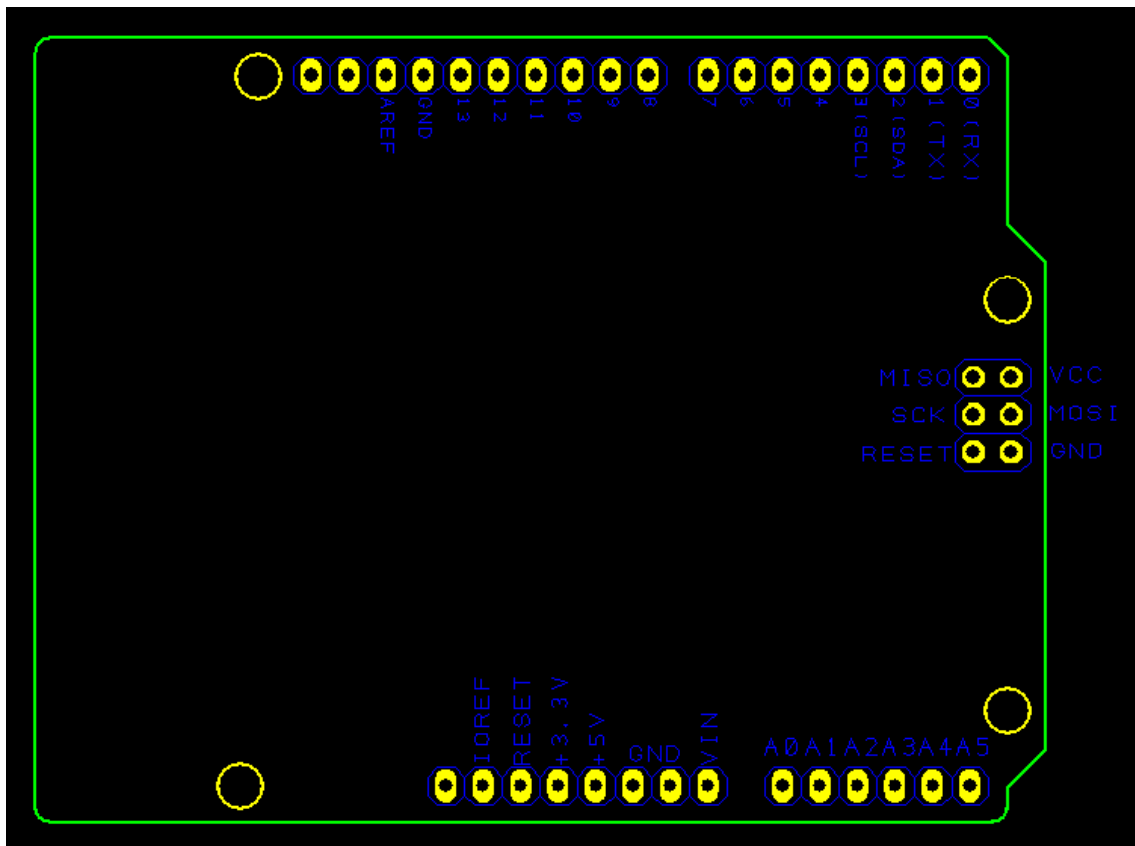


Figura 77: Símbolo PCB plantilla Arduino shield

- Plantilla de módulo de RTC:** Al no disponer de datasheet del módulo, se miden las distancias del perímetro y del centrado del conector con calibre.

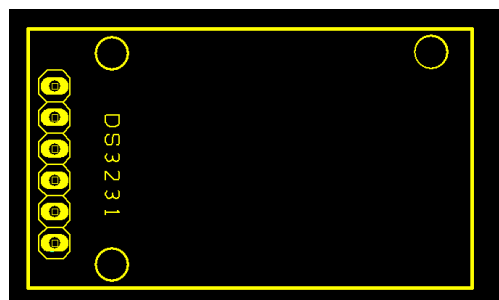


Figura 78: Símbolo PCB módulo RTC

5.3. Especificaciones del diseño de la PCB

Se establecen una serie de pautas para el diseño de la PCB:

- Todos los componentes son de montaje por orificio pasante (DIP), con lo cual, todos los puntos de soldadura serán en pads taladrados, y todos serán de forma circular.
- La PCB se diseñará a 2 capas: Todos los componentes se montarán en la capa superior para soldar sus pines por la capa inferior. En la medida de lo posible, las pistas de unión entre pads se localizarán en la capa inferior.
- Debido a limitaciones técnicas de la fresadora CNC con la que se va a realizar el proceso de fabricación; la anchura mínima de pistas, taladros, y separación mínima entre pista/pad y pista/pad debe ser de 0,5mm.
- Para no disponer de muchas medidas de anchura de pista; se asigna 0,5mm a las pistas de señal, y 1mm a las pistas de alimentación. En referencia al cálculo realizado por Design Spark para las secciones (ver figura 79), los valores asignados son suficientes para la corriente máxima suministrada por la fuente. Además se evitarán los ángulos de 90 grados o más, así como distancias excesivas.

The image shows a software interface for calculating track width. It is divided into two sections: 'Inputs' and 'Results'.
In the 'Inputs' section:
- 'Track Layer' has radio buttons for 'External' (selected) and 'Internal'.
- 'Track Thickness' is set to 0.035 mm.
- 'Current' is set to 0.6 Amps.
- 'Temperature Rise' is set to 10 C.
- 'Ambient Temperature' is set to 20 C.
- 'Track Length' is set to 1000 mm.
- 'Copper Resistivity' is set to 1.7e-006 Ohm-cm, with a 'Use Default' button next to it.
In the 'Results' section:
- 'Required Track Width' is calculated as 0.148485 mm.

Figura 79: Calculadora Design Spark de anchura de pista

- Se evitará el uso de vías, así como los cruces de pistas entre componentes. En caso de necesidad de vías, se colocarán en puntos accesibles para desempeñar la función de punto de test, ya sean pistas de señal o de alimentación.
- Se doblarán los conectores de montaje de Arduino, es decir, de cada uno de ellos la PCB integra 2. Uno para conectarse con Arduino, y otro para futuras ampliaciones o funciones extras que se quieran añadir, ya que la mayoría de pines analógicos y digitales se quedan sin uso.
- Se colocarán taladros en las esquinas de la PCB para atornillarla a una base, así como en las zonas donde se montan los módulos de Arduino y RTC, para sujetarlos de forma consistente mediante separadores y tornillería de nylon.
- Para reducir la caída de tensión en pistas, se añadirán planos de tensión y de masa en cada una de las láminas de cobre.

5.4. Diseño final de la PCB

Con todos los componentes ya disponibles en las librerías, se construye el diagrama esquemático de los componentes que integran la PCB. Ver anexo A7.1 diagrama esquemático PCB. Esta parte no es necesaria del todo, pero si recomendable, ya que en el diseño de la PCB, el programa es el que indica las conexiones necesarias mediante pistas o planos.

En la imagen 80 se observa la distribución de los componentes en la PCB, además de las plantillas de los módulos de Arduino y RTC; sobre una superficie rectangular de 150x110mm.

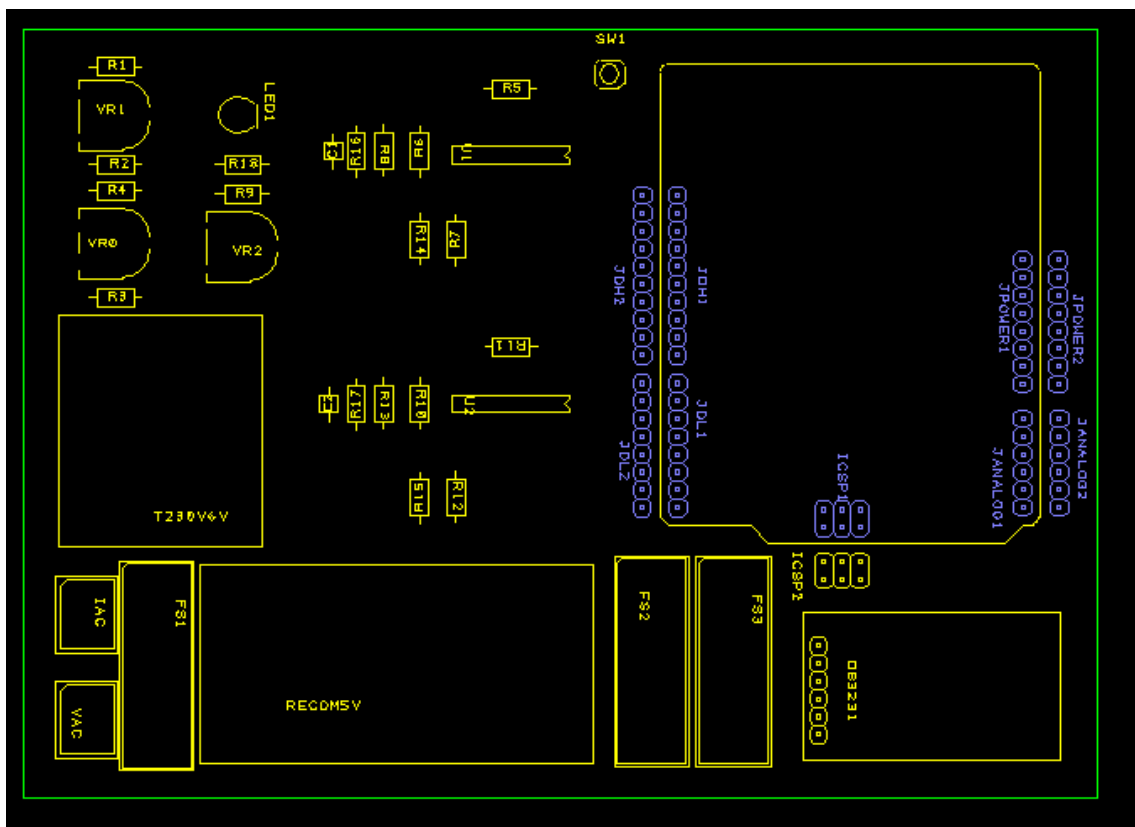


Figura 80: Diseño PCB Design Spark distribución de componentes

La plantilla de Arduino se ha situado de forma que los puertos de conexión del mismo (USB, micro USB y RJ-45), estén orientados a una zona accesible, es decir, hacia el exterior de la PCB.

Los conectores primarios de la zona cercana a la plantilla de Arduino (numerados con 1) son para el montaje del módulo de Arduino; los secundarios (numerados con 2), están conectados a los mismos pines que los primarios, con la posibilidad de conectar periféricos externos o futuras ampliaciones.

Se han situado las regletas de conexión de alimentación de red (VAC) y sensor de corriente (IAC), en un lugar accesible para la conexión.

Para más detalle ver documentación anexada de diagrama esquemático de PCB (A.7.1), diagrama esquemático completo (A.7.2), y archivos GERBER de fabricación.

El siguiente paso es enrutar todas las conexiones mediante pistas. Se enrutan todos los conectores primarios con secundarios, alimentaciones, circuitos de acondicionamiento, etc.

Se conectan las señales analógicas del circuito de acondicionamiento al conector JANALOG1, de forma que la señal de corriente continua sumada a las senoidales, la señal de tensión y la señal de corriente, quedan conectadas a los pines A0, A1, y A2, respectivamente.

Los pines del conector del módulo RTC se conectan a alimentación (+5V y GND) y al puerto I2C, compartido por los pines digitales 2 y 3 (SDA y SCL).

El SW1 se conecta entre los pines RESET y GND del conector JPOWER1; y el LED1 en serie a su resistencia R18, se alimenta entre +5V y GND.

Por motivos de diseño, se ha recurrido a la colocación de vías para la conexión de pistas en la cara superior con la inferior, que a la vez se utilizarán como puntos de test. La vía V10 une dos puntos de la señal analógica de continua (A0); la vía V11 une la señal analógica de tensión (A1); la vía V12 une una pista de extensión de masa (GND); y la vía V13 alimenta los integrados a -5V.

Por último, con todas las pistas terminadas, con las zonas libres se construyen dos planos. En la capa superior se forma un plano conectado a +5V, y en la capa inferior se forma otro conectado a masa (GND).

La imagen de la figura 81 corresponde al diseño final de la PCB. Los enrutados en color azul corresponden con la capa inferior de la PCB; y los enrutados verdes, con la capa superior de la misma. Para facilitar la identificación de las pistas y sus conexiones, se han deshabilitado los planos de tensión y masa de las capas superior e inferior.

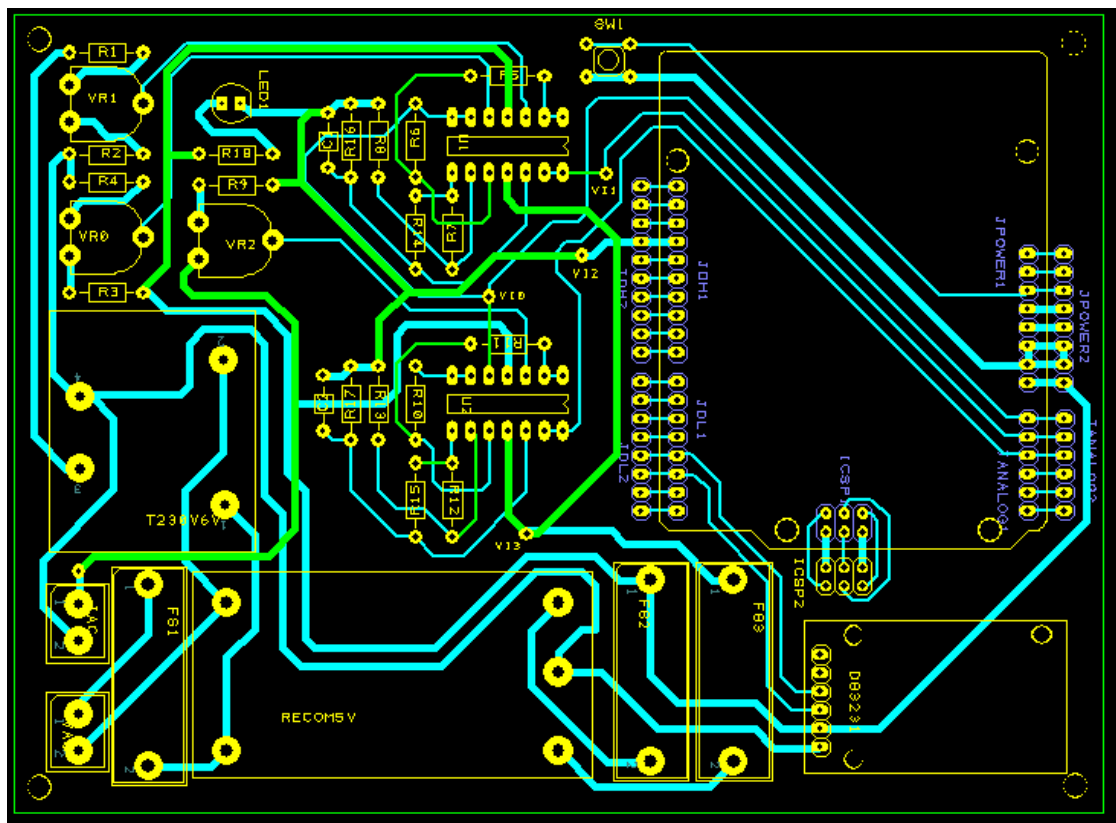


Figura 81: Diseño PCB Design Spark enrutado de pistas

El programa permite la opción de pre visualizar en vista 3-D el diseño de la PCB con todos sus componentes montados. Las imágenes de las figuras 82 y 83, muestran las vistas de las caras superior e inferior respectivamente. Hay dos errores de interpretación del programa; uno con los conectores de Arduino, ya que van montados en la capa superior de la PCB, pero soldados en el pad de la capa inferior; y otro, con los taladros para la sujeción mecánica.

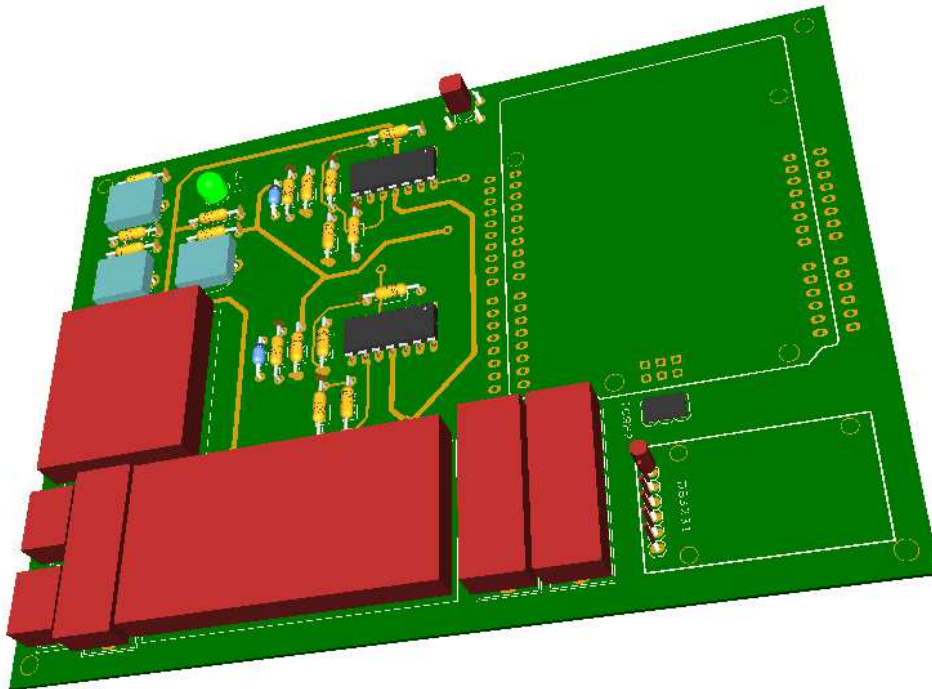


Figura 82: Diseño PCB vista 3D cara superior

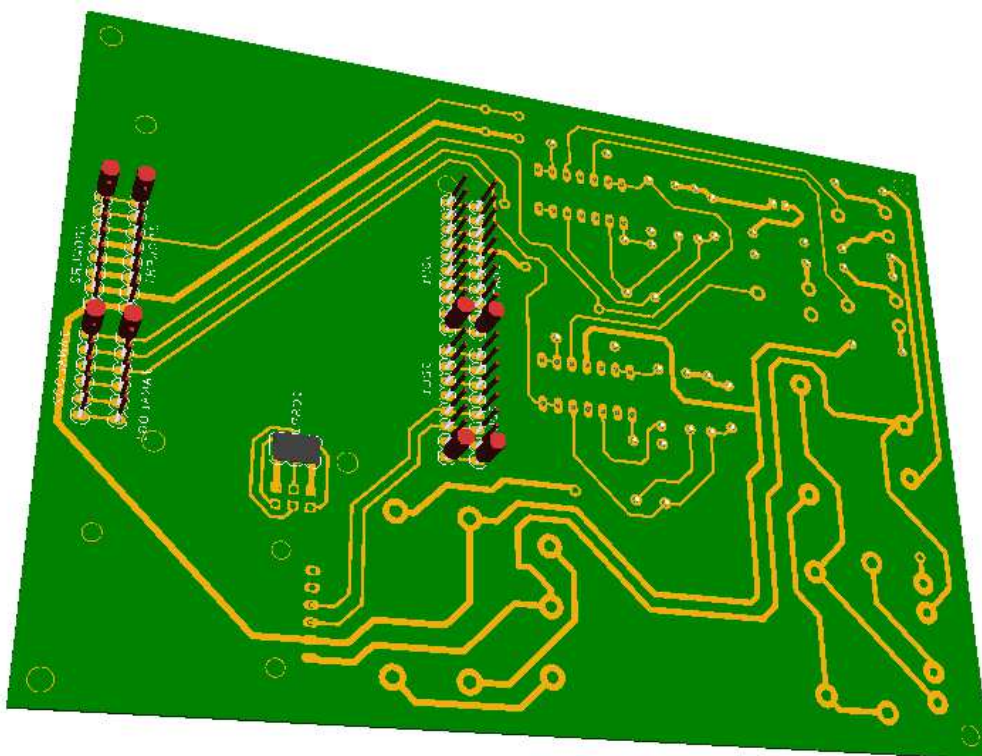


Figura 83: Diseño PCB vista 3D cara inferior

6. Fabricación y montaje de la PCB

Con el diseño de la PCB completado, se exporta a formato GERBER para la fabricación en fresadora CNC. La PCB se mecaniza a partir de una placa de fibra de vidrio, forrada por dos láminas de cobre, con un espesor de $35\mu\text{m}$. La imagen de la figura 84 corresponde con la capa superior de la PCB; la de la figura 85, con la capa inferior.

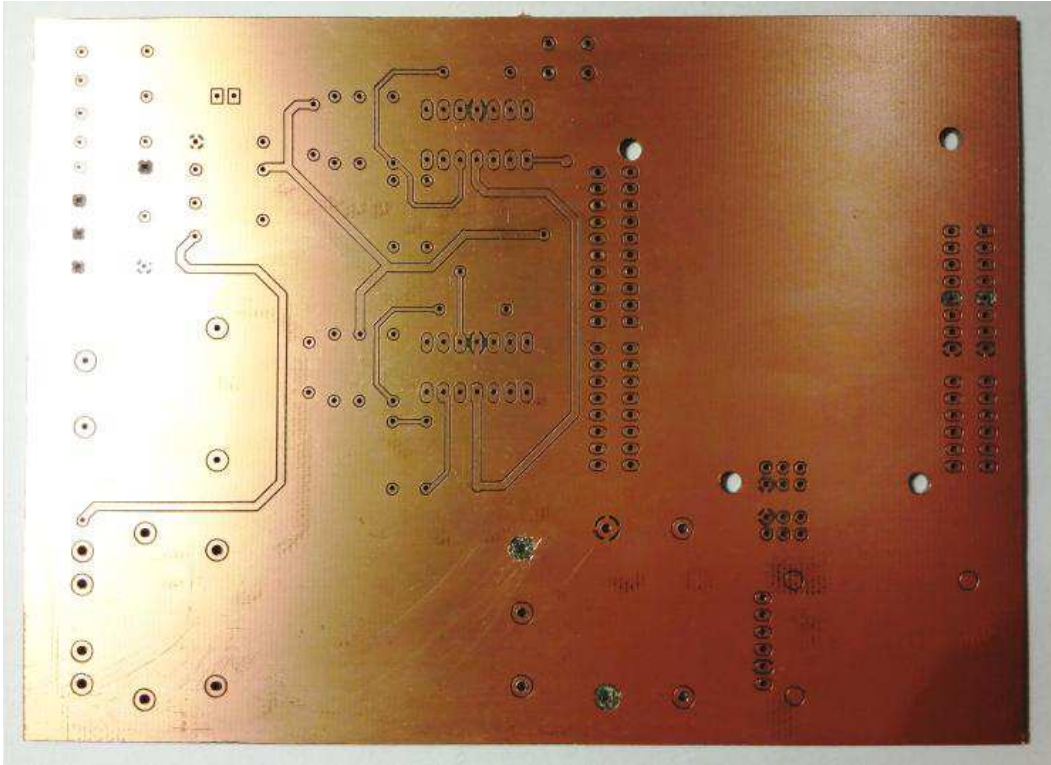


Figura 84: PCB vista superior planta

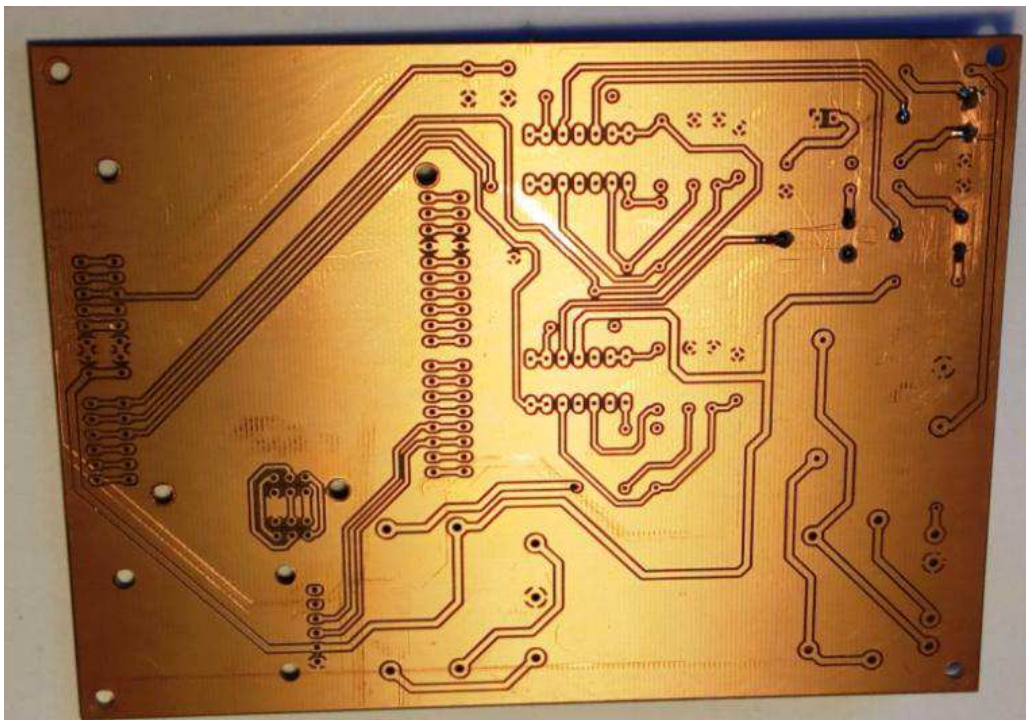


Figura 85: PCB vista inferior planta

En el proceso de fabricación se han producido errores que se han reparado manualmente. Los errores se pueden apreciar en la figura 84 de la página anterior, y comparar con la figura 85, ya que esta se hizo una vez solucionados los errores.

Uno de los errores es que la fresadora CNC no ha taladrado todos los puntos de sujeción; en concreto, los de las esquinas de la PCB, y los del módulo RTC. Con un taladro de columna, se perfora en los puntos donde debía haber taladrado la fresadora, con una broca de 1mm. Después se agrandan los agujeros aumentando el calibre de la broca paso a paso hasta el diámetro nominal de cada uno. Por precaución con la PCB, el aumento de diámetro se hace girando la broca a mano con suavidad, ya que apenas ofrece resistencia la fibra de vidrio.

Otro error cometido en la fabricación ha sido debido al diseño informático. Los pines de los potenciómetros no caben en los orificios de montaje. A pesar de haber escogido componentes genéricos, las medidas de pads no parecen estar bien dimensionadas en este componente de las librerías de Design Spark. Para solucionarlo se aumenta el diámetro del orificio, pasando brocas de mayor calibre, hasta el punto de eliminar casi el pad por completo, con que la soldadura se realizará sobre las pistas en algunos casos.

Solucionados los errores de fabricación, la PCB pasa a la fase de montaje. Puesto que todos los componentes son de montaje con orificio pasante, se integraran a la PCB por soldadura con estaño. Para el montaje se hace uso de un soldador de gas, un succionador, alicates, y estaño. En algunos casos, los puntos de soldadura están muy cerca entre sí, con lo cual conviene testear con ayuda de un polímetro, si en el punto soldado se han cortocircuitado puntos diferentes.

La imagen de la figura 86 muestra la cara superior de la PCB con todos los componentes, y los separadores de nylon montados, a falta de los módulos de Arduino y RTC.

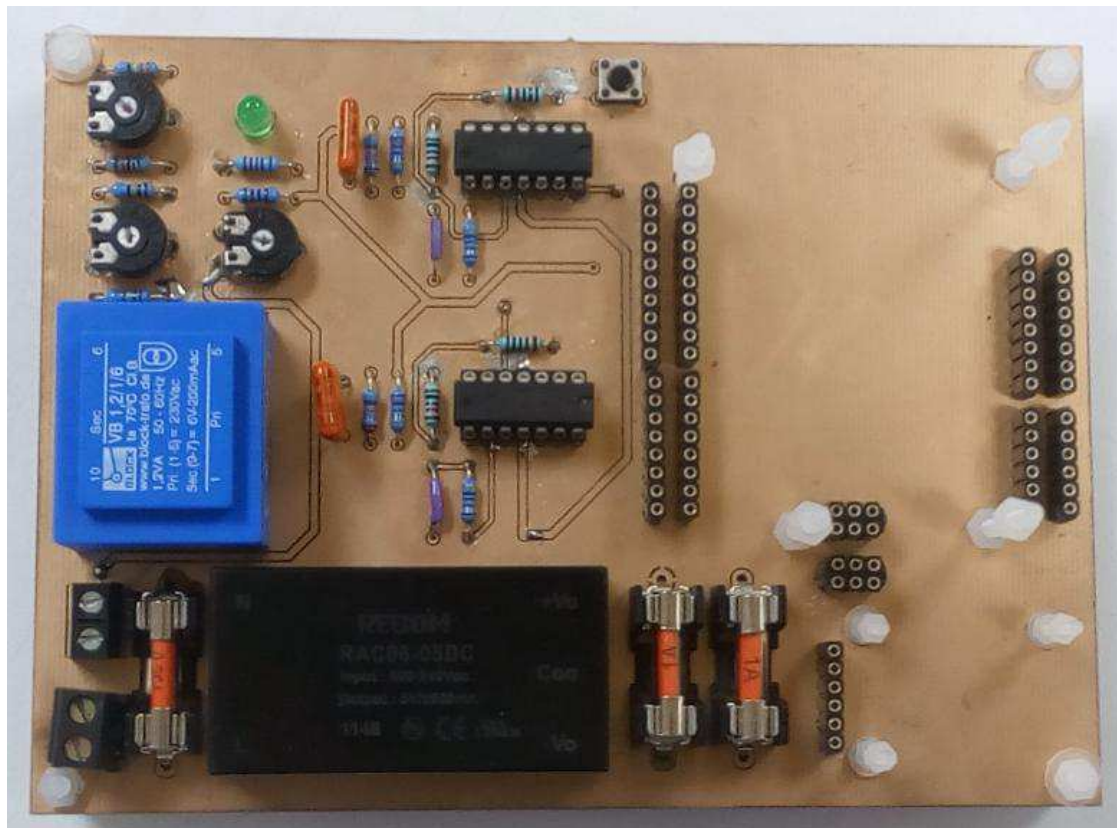


Figura 86: PCB con componentes vista superior

Las imágenes de las figuras 87 y 88, muestran el acabado final de la cara inferior de la PCB; y la cara superior, una vez montados los módulos de Arduino y RTC.

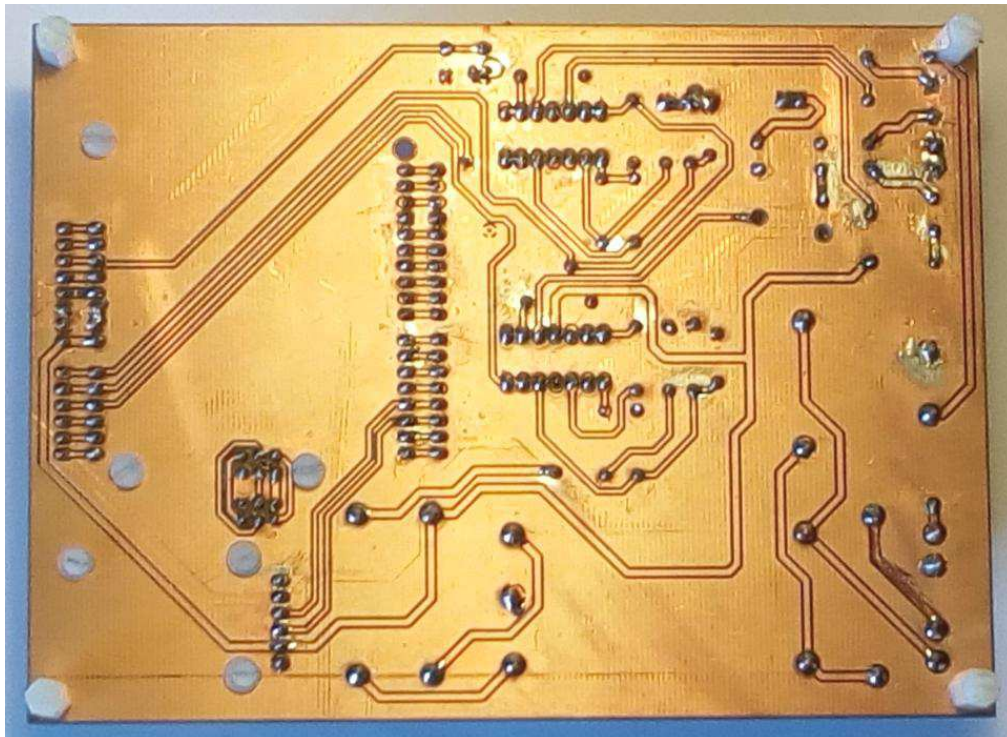


Figura 87: PCB con componentes vista inferior

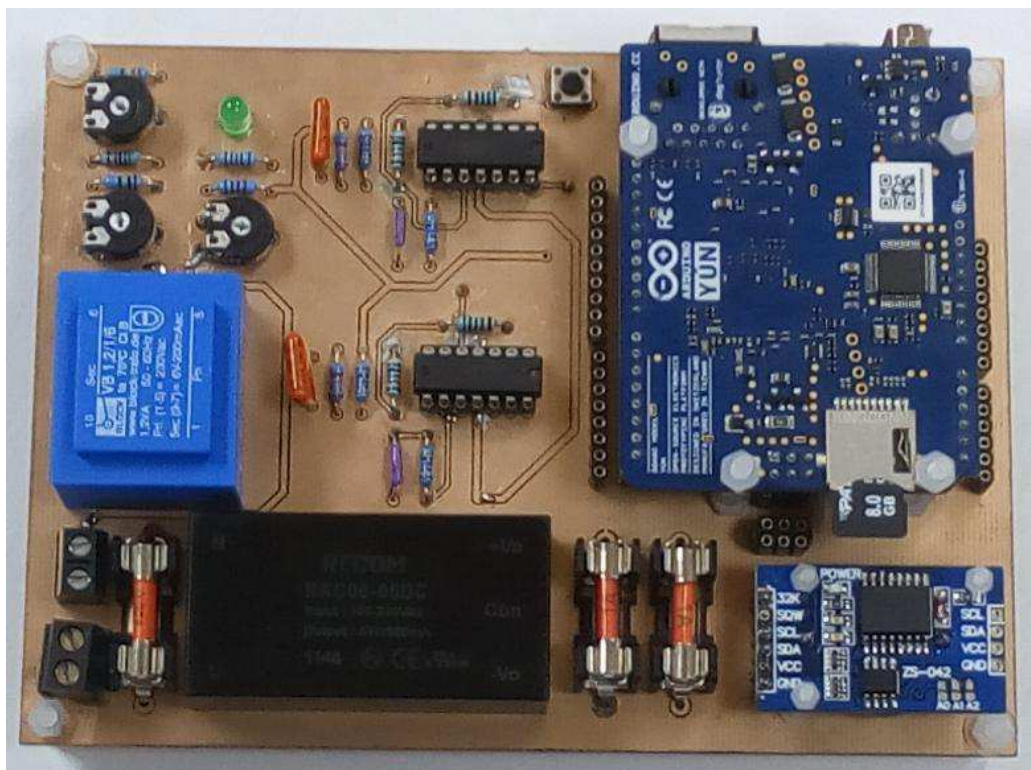


Figura 88: PCB completa

Se da por finalizado el diseño y montaje del hardware para proceder a su comprobación antes del desarrollo del software a implementar.

7. Primer encendido de Arduino y configuración

Una vez instalado el IDE de Arduino, siguiendo las recomendaciones de la web oficial (<http://www.arduino.cc>), además de los drivers necesarios para operar con la placa YUN, se conecta el Arduino YUN en solitario por el puerto micro USB al PC, y con el IDE de Arduino ejecutado, se comprueba su puerto de conexión para configurar el programa.

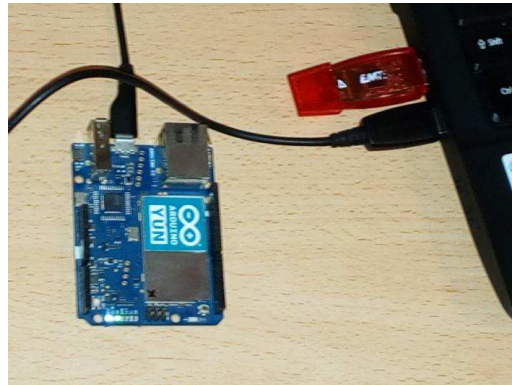


Figura 89: Arduino conectado por USB

Como indica la captura de la figura 90, desde la pestaña “Herramientas – Puertos” se asigna el puerto al que está conectado Arduino para establecer comunicación serie.

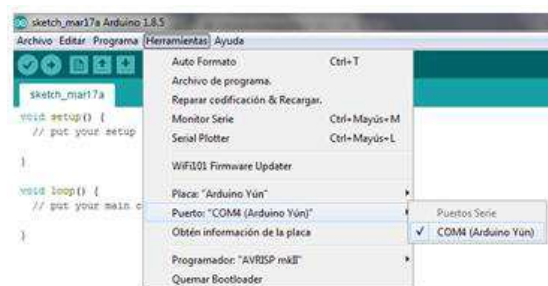


Figura 90: IDE Arduino

Ahora se va a configurar como cliente dentro de una red local privada. En primer lugar buscamos desde el PC su red WIFI y nos conectamos. Una vez conectado a la WIFI de Arduino YUN, con ayuda del IDE de Arduino, conocemos su IP local, en cuyo caso es la dirección 192.168.240.1.



Figura 91: Redes

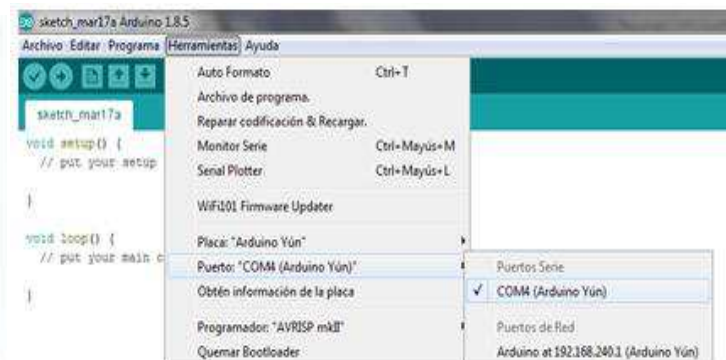


Figura 92: IDE Arduino

Identificada la IP local de nuestro Arduino YUN, accedemos a su panel de control web mediante cualquier navegador del PC. En la primera ventana nos pedirá una contraseña de acceso que por defecto es “Arduino”, más adelante la podremos cambiar.

Con la contraseña aceptada, la siguiente pantalla muestra la conexión WIFI actual, y la conexión inexistente por cable, como se puede ver en la captura de la figura 93.

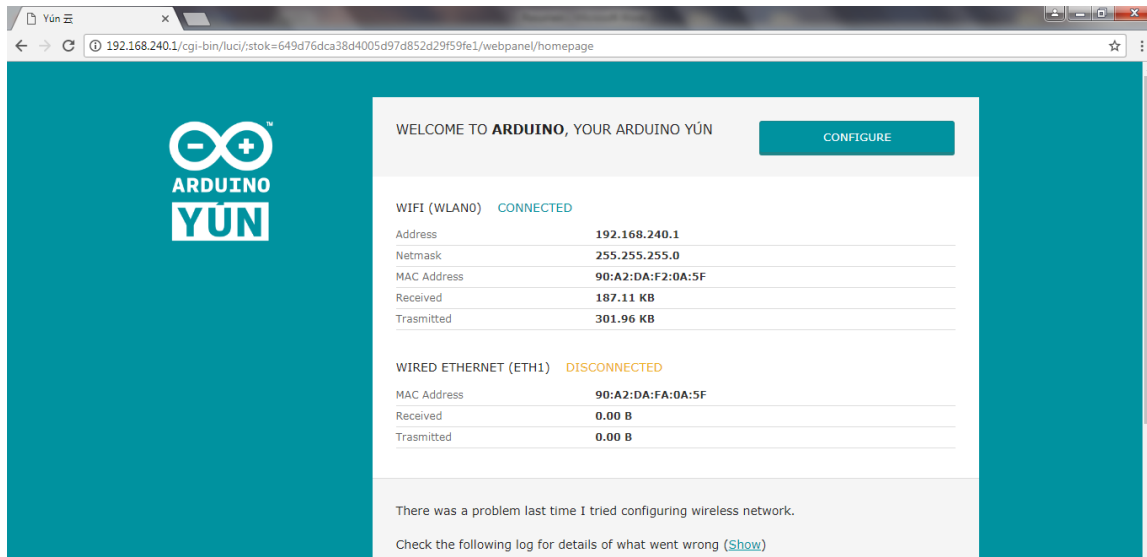


Figura 93: Configuración de Arduino YUN

Al entrar a “Configuración”, en el siguiente panel podemos conectar Arduino a nuestra red local de forma que podamos cargar el *sketch* y monitorizar datos por WIFI, sin necesidad de acceder a su propia red. Así cada vez que se encienda, automáticamente se conectará a nuestra red local. Para ello hay que introducir los datos y contraseña de nuestra red privada.

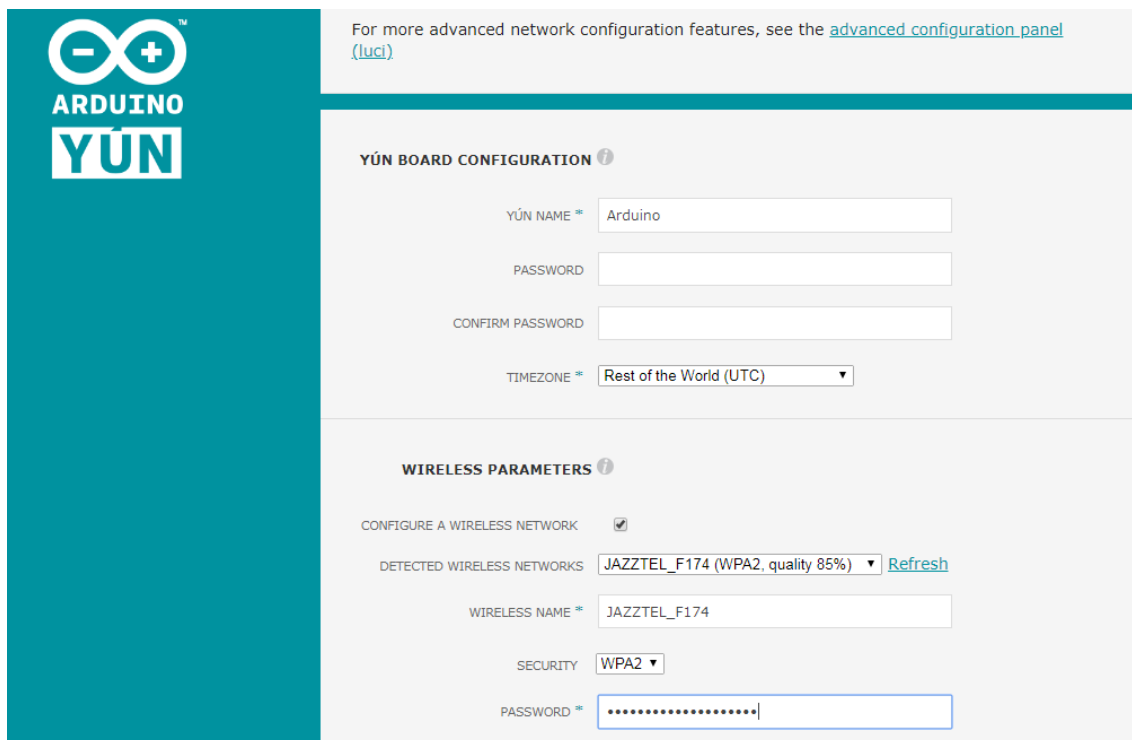


Figura 94: Configuración de Arduino YUN

Después de esperar a que se reinicie, comprobamos que al conectarnos a nuestra red local, tenemos la posibilidad de ver la dirección IP desde el IDE de Arduino.

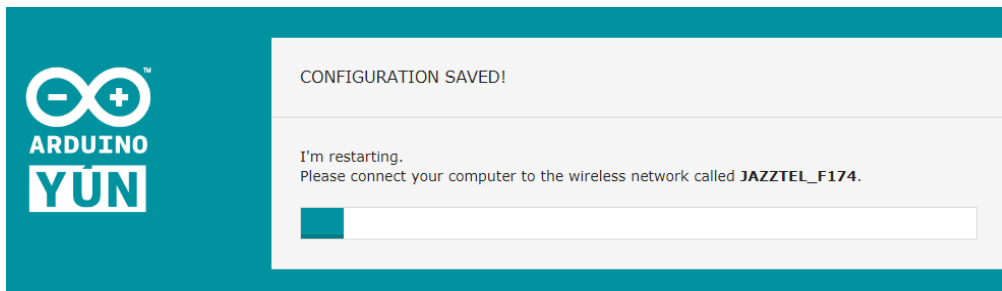


Figura 95: Configuración de Arduino YUN

En este caso, la IP de Arduino ha cambiado debido a que se ha unido a nuestra red. Ahora el acceso al panel de control web es mediante la nueva IP adquirida 192.168.1.131.

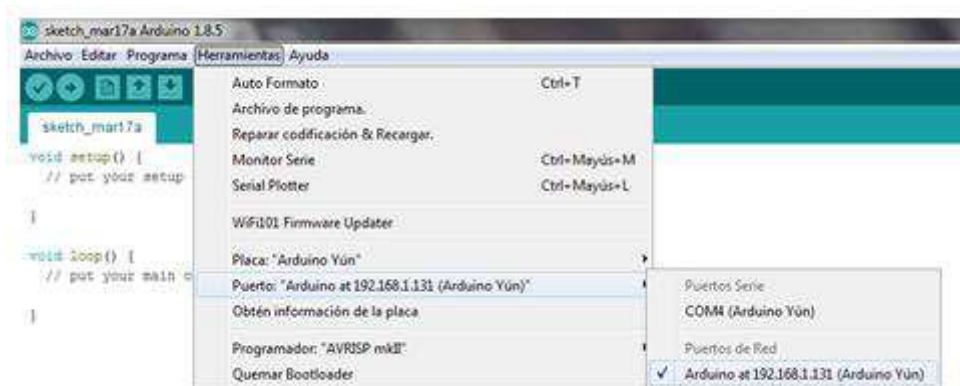


Figura 96: IDE Arduino

Como se puede observar en la captura de la figura 97, la dirección introducida en la barra superior del navegador coincide con la nueva IP asignada.

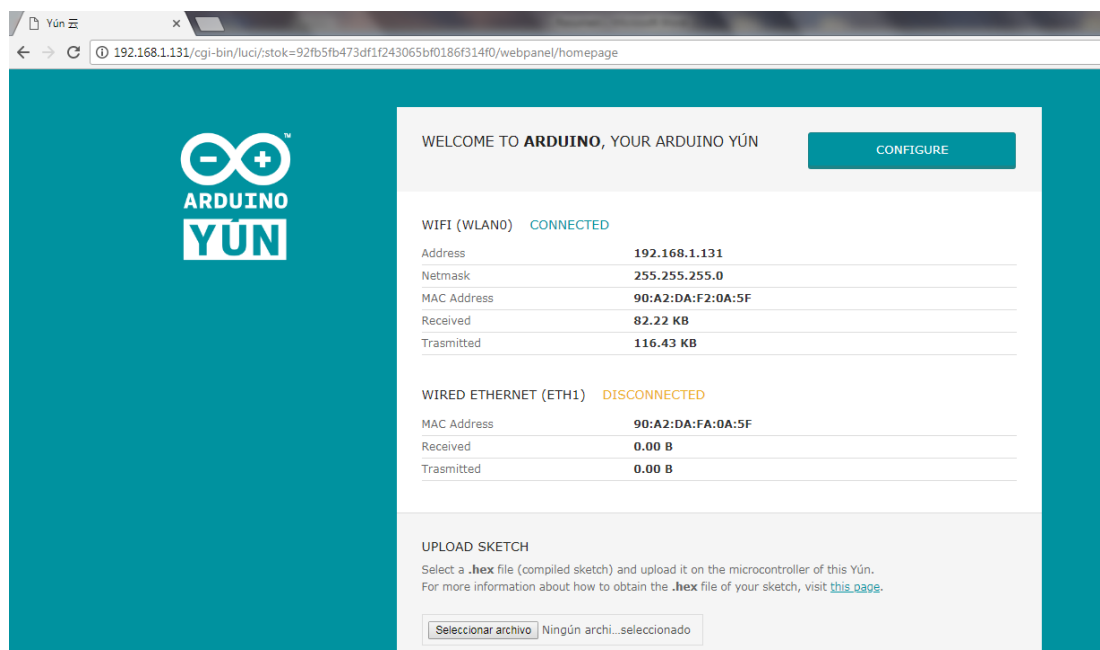


Figura 97: Configuración de Arduino YUN

A partir de ahora, cada vez que realicemos operaciones con Arduino montado en la PCB y conectado al PC por el puerto micro USB, habrá que tener cuidado con las conexiones de alimentación. Por la PCB llega una alimentación a Arduino de 5V, y a la vez el USB alimenta Arduino a una tensión teórica de 5V, pero en la práctica llega hasta 6V.

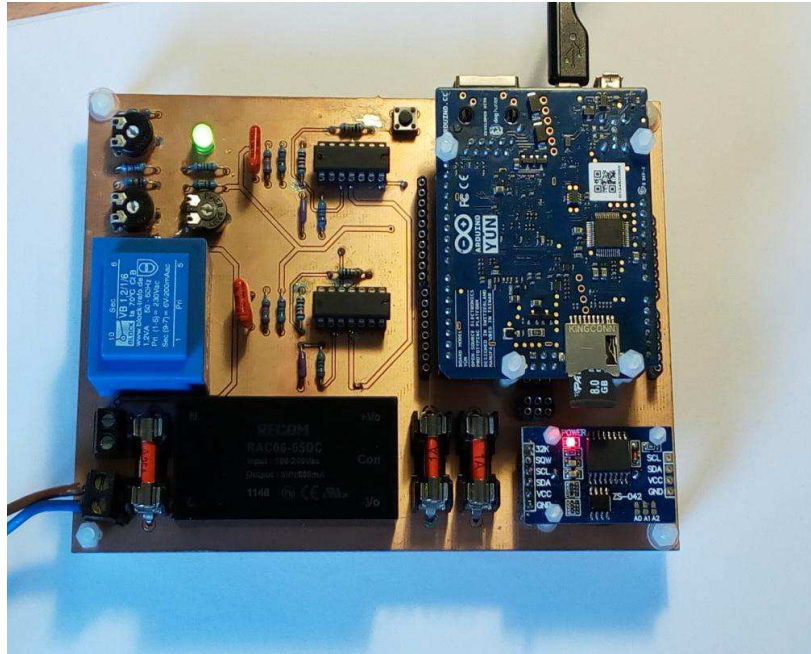


Figura 98: PCB conectada a la red y micro USB

Para evitar cortocircuitos o anomalías eléctricas, ya sea en la PCB del dispositivo, o en el propio PC, vamos a evitar esta situación. Si conectamos la PCB a la red, como está diseñada, y al montar Arduino en esta, eliminamos del conector macho, el pin VIN, aislamos los dos circuitos de continua. Los 5V de la fuente de alimentación alimentarían a la PCB y los módulos RTC y LCD, y los 6V del USB solo a Arduino. En la fotografía de la figura 99 se observa el pin eliminado.

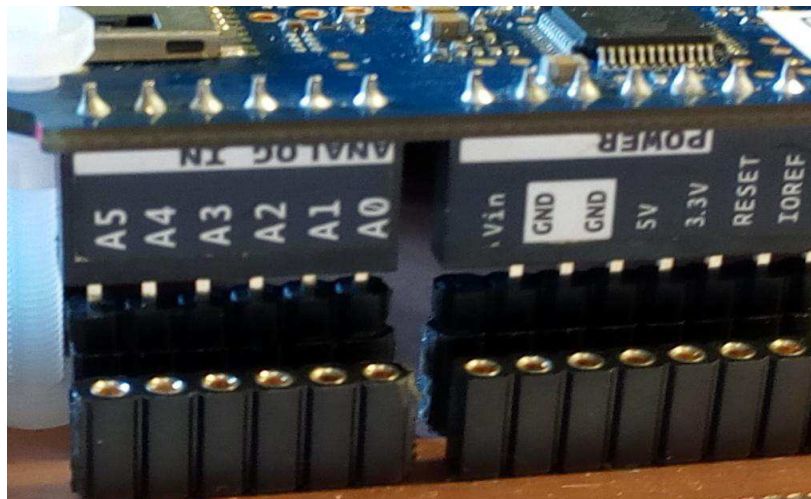


Figura 99: Arduino Power socket

Esta maniobra solo se realizará mientras sea necesaria la conexión de Arduino al PC. También se podría evitar la conexión con el uso del WIFI. El uso de WIFI funciona muy bien para cargar los *sketch*, pero para utilizar el Monitor serie del IDE de Arduino es lento y ha resultado problemático en pruebas ya realizadas, por lo que desestimamos la idea.

7.1. Identificación de direcciones de módulos en I2C

Como se van a comunicar los módulos de RTC y LCD mediante el mismo puerto I2C, es necesario conocer su direccionamiento para el control de ambos periféricos en el desarrollo del software del dispositivo.

La dirección debe ser única del periférico y no se asigna como tal, sino que ya pertenece al módulo, es algo así como la MAC. En algunos casos, estos módulos prefabricados cuentan con un jumper para elegir entre 2 direcciones diferentes en caso de utilizar módulos iguales conectados al mismo puerto.

Existe un *sketch* en la comunidad de Arduino como herramienta para identificar direcciones de módulos conectados al puerto I2C, que es con la que se va a trabajar. Ver anexo A.6.1 *sketch* I2C-SCANNER.

Para identificar cada módulo, se conectan al puerto I2C de forma individual, y se carga el *sketch* I2C-SCANNER al Arduino. Una vez compilado y cargado, abrimos el Monitor serie del IDE Arduino para leer la información proporcionada por la herramienta.

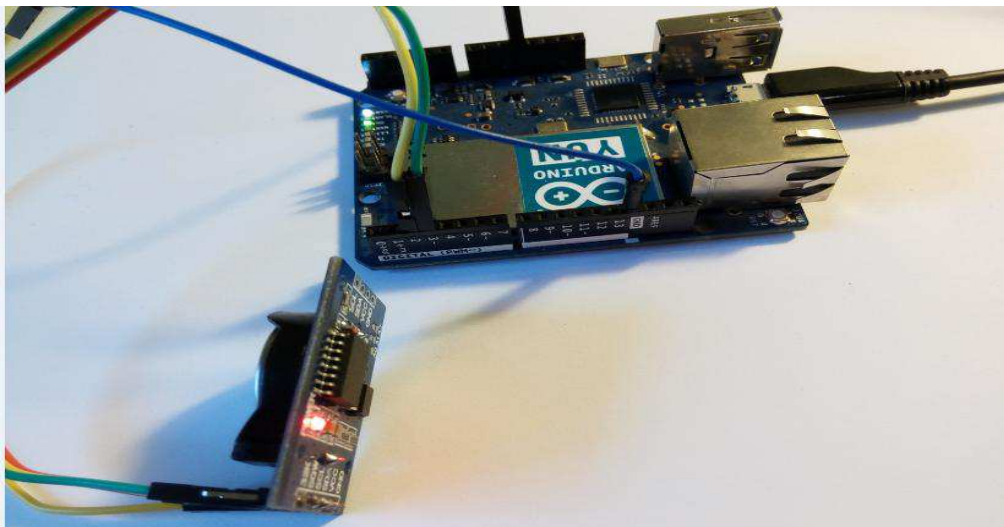


Figura 100: Conexión Arduino y RTC

Primero comprobamos el módulo RTC conectando el pin SDA con el pin digital 2, y el SCL con el pin digital 3, más 2 hilos de alimentación, como se ve en la imagen de la figura 100.

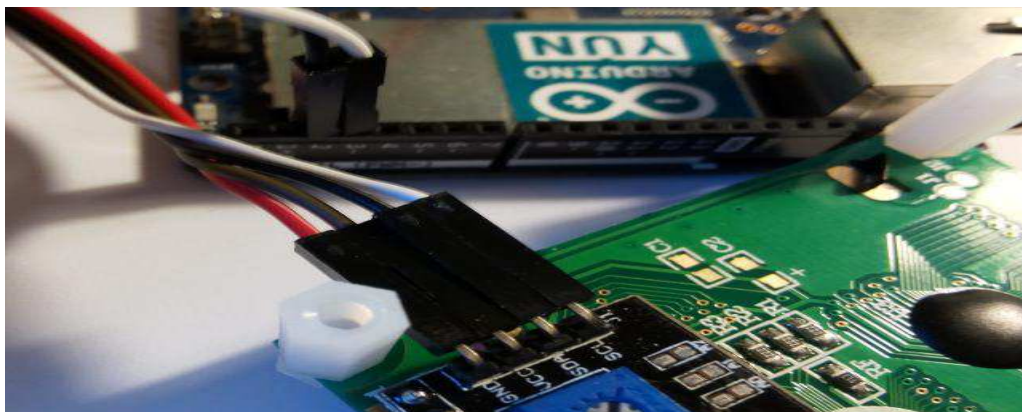
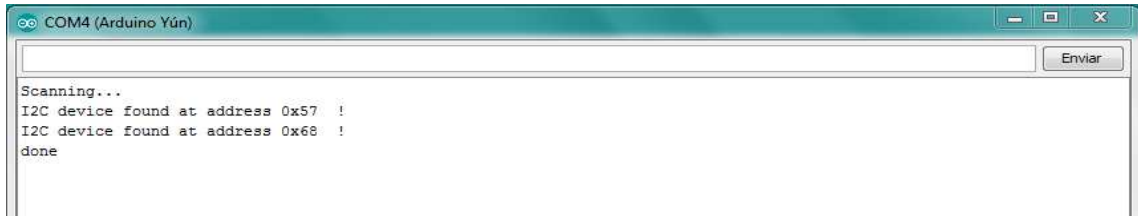


Figura 101: Conexión Arduino y pantalla LCD

Al abrir la pantalla del Monitor serie nos proporciona información de 2 direcciones. Debe ser porque el mismo módulo dispone de conexión a un sensor de temperatura, y esta información se envía por otra dirección diferente a la de fecha y hora del RTC (ver figura 102).

Consultando información de fabricantes de módulos RTC, a casi todos se les asigna la dirección 0x68, por lo que consideramos esta la dirección para instrucciones de lectura de datos del módulo, y descartamos la dirección 0x57.



```
COM4 (Arduino Yún)
Scanning...
I2C device found at address 0x57 !
I2C device found at address 0x68 !
done
```

Figura 102: Monitor serie

En la fotografía de la imagen 103 se puede observar la conexión del módulo LCD al puerto I2C de Arduino para su identificación.

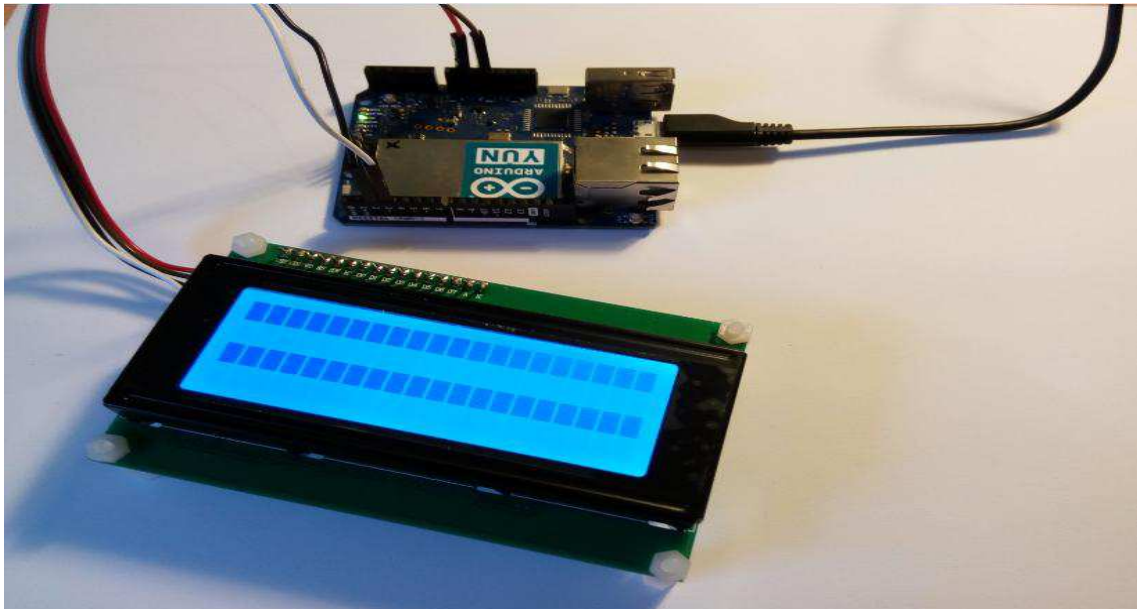


Figura 103: Conexión Arduino y pantalla LCD

Después de cargar el *sketch* I2C-SCANNER, abrimos la pantalla del Monitor serie para que nos indique si ha encontrado algún dispositivo conectado.



```
COM4 (Arduino Yún)
I2C Scanner
Scanning...
I2C device found at address 0x3F !
done
```

Figura 104: Monitor serie

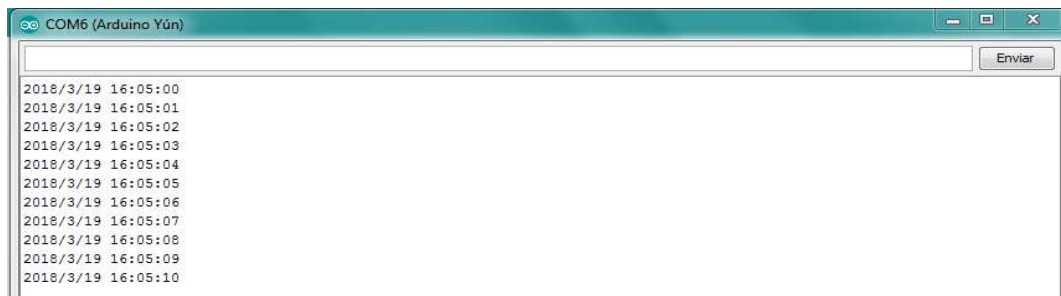
El módulo LCD con el que vamos a trabajar se comunica al I2C con la dirección 0x3F.

7.2. Ajuste de RTC

Para disponer de las funciones del módulo RTC es necesario incluir la librería “DS1307-DS3231” en el IDE de Arduino. Instalando la librería en el directorio “\Libraries\”, dispondremos de ejemplos explicados en el IDE para comprender el funcionamiento del módulo.

Por ahora nos limitaremos a poner en fecha y hora el RTC, y comprobar que al desconectar la alimentación y reconectar, mantiene la fecha y hora actualizadas gracias a la batería incorporada.

Para ello se carga el *sketch* de “Puesta_en_hora_3231RTC” a Arduino; ajustando los parámetros de fecha 19/03/18 y de hora 16:05:00. Tras desconectar la alimentación y reconectarla de nuevo, comprobamos el paso de los segundos abriendo la pantalla del monitor serie, como se ve en la figura 105. Ver *sketch* “Puesta_en_hora_3231RTC” en anexo A.6.2.



```
COM6 (Arduino Yún)
2018/3/19 16:05:00
2018/3/19 16:05:01
2018/3/19 16:05:02
2018/3/19 16:05:03
2018/3/19 16:05:04
2018/3/19 16:05:05
2018/3/19 16:05:06
2018/3/19 16:05:07
2018/3/19 16:05:08
2018/3/19 16:05:09
2018/3/19 16:05:10
```

Figura 105: Monitor serie

7.3. Ajuste de LCD

En el caso del módulo LCD solo se va a realizar una prueba de funcionamiento. Al igual que el módulo RTC, disponemos de una librería para facilitar la programación de instrucciones. La librería a incluir es la “LiquidCrystal_I2C”.

Se carga un *sketch* sencillo de los ejemplos de la librería, para comprobar la reproducción de caracteres en el LCD. Para su funcionamiento es necesario cambiar unos parámetros asignados por defecto. Uno de ellos es la dirección del dispositivo, que en nuestro caso es 0x3F; y los otros dos están relacionados con el número de columnas y filas de caracteres, que es de 20x4. Ver *sketch* de prueba “LCD” en anexo A.6.3.



Figura 106: Encendido pantalla LCD

7.4. Tarjeta MicroSD

Se utiliza como unidad de almacenamiento *flash* una tarjeta MicroSD, en la que almacenar los datos en formato de archivos de texto. En este caso se utiliza una de clase 10, con una capacidad de 8GB.



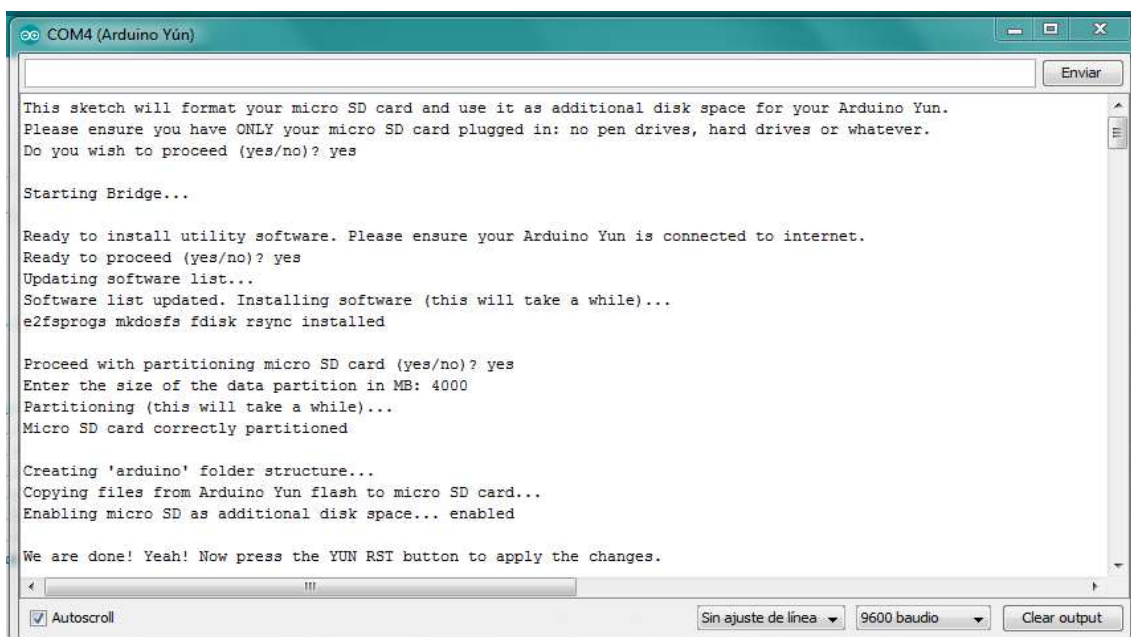
Figura 107: Tarjeta micro SD

Además del almacenamiento de datos de variables, la idea es aprovechar la tarjeta también como memoria extendida del sistema de Linux, para el almacenamiento de actualizaciones o paquetes del sistema.

Existe en la comunidad de Arduino, un *sketch* que realiza la función de particionar la tarjeta a las proporciones deseadas por el usuario. Una de las particiones es en formato Linux, y la otra en FAT32.

Para realizar correctamente el proceso es necesario disponer de conexión a internet, ya que durante el proceso se actualizan unos paquetes de software. En apartados anteriores ya ha sido conectado el YUN a la red local, por lo que no debería haber ningún problema.

Insertamos la tarjeta SD en el lector de Arduino y cargamos el *sketch* "YunDiskSpaceExpander". Ver anexo A.6.4 del *sketch* "YunDiskSpaceExpander". Desde el monitor serie seguimos las indicaciones del programa y asignamos 4GB a cada partición. La captura de la figura 108 muestra las indicaciones del *sketch* por el Monitor serie.

A screenshot of the Arduino IDE serial monitor window titled 'COM4 (Arduino Yún)'. The window shows the output of a sketch. The text in the monitor is as follows:

```
This sketch will format your micro SD card and use it as additional disk space for your Arduino Yun.
Please ensure you have ONLY your micro SD card plugged in: no pen drives, hard drives or whatever.
Do you wish to proceed (yes/no)? yes

Starting Bridge...

Ready to install utility software. Please ensure your Arduino Yun is connected to internet.
Ready to proceed (yes/no)? yes
Updating software list...
Software list updated. Installing software (this will take a while)...
e2fsprogs mkdosfs fdisk rsync installed

Proceed with partitioning micro SD card (yes/no)? yes
Enter the size of the data partition in MB: 4000
Partitioning (this will take a while)...
Micro SD card correctly partitioned

Creating 'arduino' folder structure...
Copying files from Arduino Yun flash to micro SD card...
Enabling micro SD as additional disk space... enabled

We are done! Yeah! Now press the YUN RST button to apply the changes.
```

The bottom of the window shows a scroll bar, an 'Autoscroll' checkbox which is checked, and dropdown menus for 'Sin ajuste de línea', '9600 baudio', and a 'Clear output' button.

Figura 108: Monitor serie

7.5. Ajustes de servidor web

Para monitorizar los datos medidos y calculados por el prototipo de forma remota, conectado a una IP pública, hay que configurar Arduino como un servidor web. Es necesario realizar ajustes en la configuración del Arduino y del router de la red privada, para que el router redirija las entradas y salidas del puerto 80 a la IP del Arduino.

En primer lugar, en la pantalla de configuración del Arduino YUN, los servicios REST se marcan como *open*, y así se habilita el envío de comandos por medio del navegador, para la comunicación entre cliente y servidor.

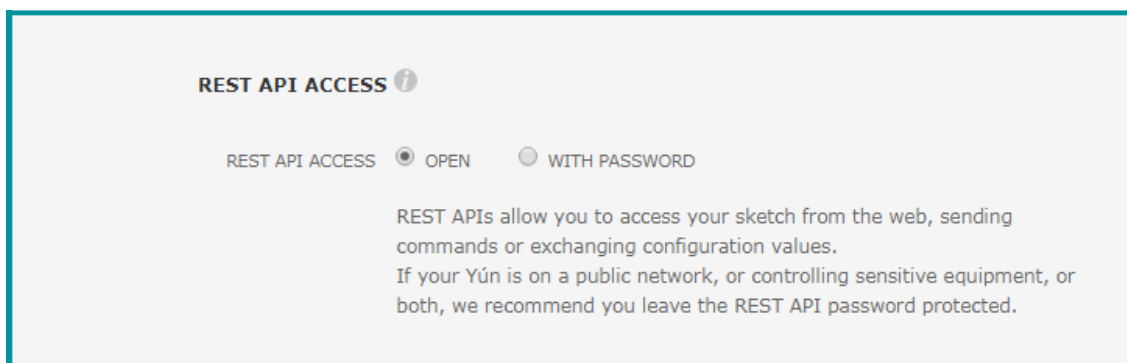


Figura 109: Configuración servicios REST

La misma pantalla principal de configuración muestra la dirección IP, y la dirección MAC. La MAC es una dirección que pertenece al hardware y que no se puede asignar, mientras que la IP es asignada por el router de la red privada. Cada vez que se reinicia el router asigna nuevas direcciones IP dentro de la red privada, por eso se les llama direcciones IP dinámicas.

| WIFI (WLAN0) CONNECTED | |
|---|--------------------------|
| Address | 192.168.1.131 |
| Netmask | 255.255.255.0 |
| MAC Address | 90:A2:DA:F1:14:BA |
| Received | 2.12 MB |
| Trasmitted | 713.79 KB |

Figura 110: Direcciones IP y MAC de Arduino

En el caso de un servidor web, es necesario que funcione siempre con una IP fija dentro de la red privada, para que el redireccionamiento sea siempre al mismo dispositivo con su IP asignada. Por lo tanto, se debe asignar al Arduino YUN una IP estática desde el panel de configuración del router.

Si la dirección IP del Arduino es la 192.168.1.131 como se observa en la figura 110; la dirección de red y la de acceso al router es la 192.168.1.1 dentro de la red privada. Se accede a la configuración del router con su dirección IP desde un navegador

La captura de la figura 111 muestra la pantalla de configuración del router. Una vez dentro del panel de configuración, con la pestaña “Advanced Setup / LAN”, se accede a un menú que permite la asignación de una IP estática a la MAC de un dispositivo.

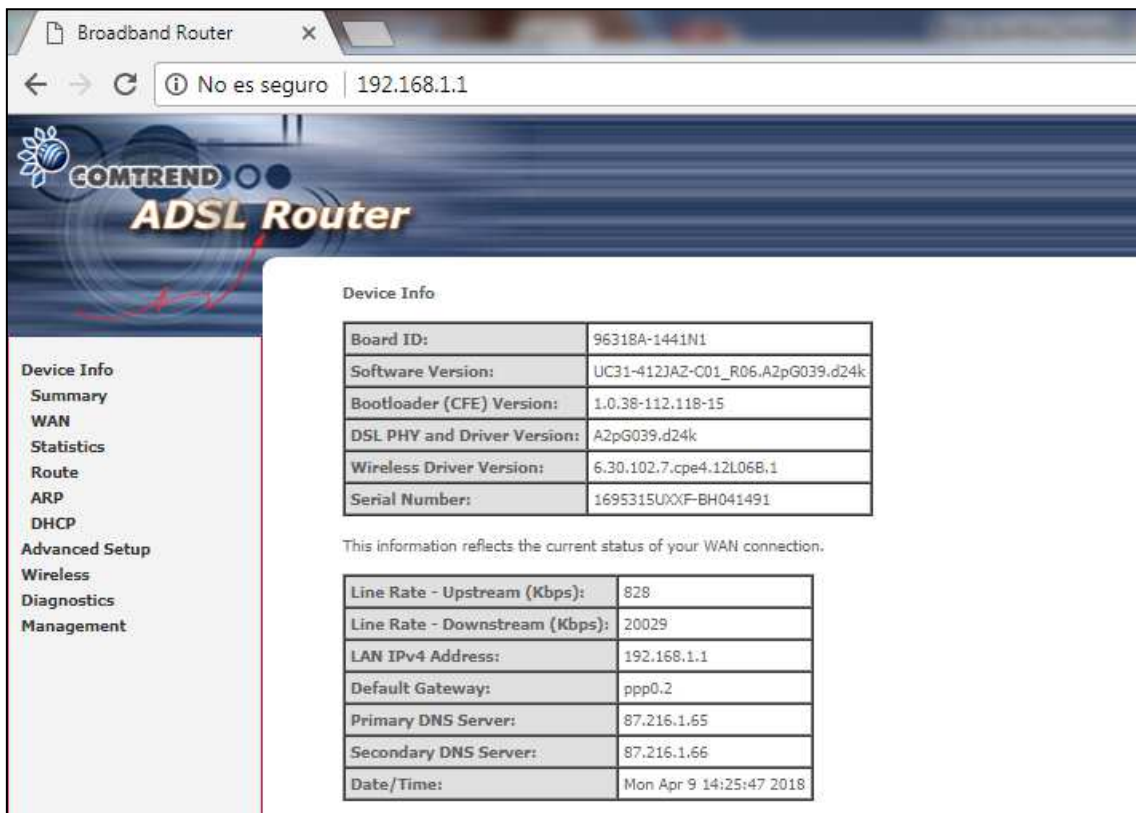


Figura 111: Pantalla de configuración del router

Con la misma IP que ya estaba asignada para el Arduino, se registra la MAC de Arduino vista en la figura 110 y se guardan los cambios. Dentro de la red privada, Arduino ya dispone de su IP estática.

La captura de la figura 112 muestra la asignación de la IP estática a la MAC de Arduino.

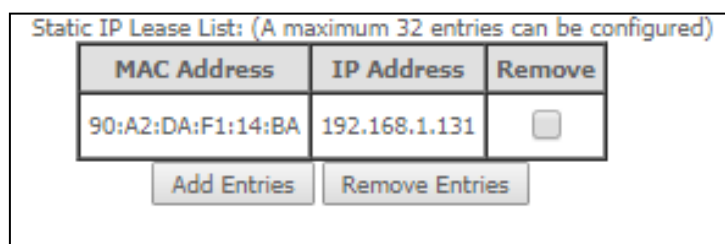


Figura 112: Asignación de IP estática

Otro ajuste necesario en el router es el redireccionamiento de los puertos 80 a la IP del Arduino. Desde el menú “Advanced Setup / NAT / Virtual servers” se configuran dichos puertos con la dirección IP asignada, tal y como se observa en la captura de la figura 113.

| External Port Start | External Port End | Protocol | Internal Port Start | Internal Port End |
|---------------------|-------------------|----------|---------------------|-------------------|
| 80 | 80 | TCP | 80 | 80 |

Figura 113: Configuración de puertos del router

La IP del router vista desde la red privada no es la misma que desde la red pública. Para el acceso desde la red pública es necesario conocer la dirección del router. Esta IP es dinámica, por lo que cada cierto tiempo cambia automáticamente, y es el único dispositivo visto desde la red pública, ya que los demás dispositivos conectados a él, forman una subred privada.

De momento apuntamos la IP pública del router para hacer una prueba de que el redireccionamiento funciona. En la pestaña “Device info / WAN” se obtiene la IP actual como se observa en la figura 114. En este momento la dirección IP pública es 95.20.254.110.

| Interface | Description | Type | VlanMuxId | Igmp | NAT | Firewall | Status | IPv4 Address |
|-----------|--------------|-------|-----------|----------|---------|----------|--------------|---------------|
| ppp0.1 | internet_atm | PPPoE | Disabled | Disabled | Enabled | Enabled | Disconnected | |
| ppp0.2 | internet_atm | PPPoE | 1074 | Disabled | Enabled | Enabled | Connected | 95.20.254.110 |

Figura 114: Información de direcciones IP del router

Conocida la dirección IP pública, desde otra red fuera del hogar, se accede a la IP mediante un navegador. Como se observa en la imagen de la figura 115, el redireccionamiento configurado en el router funciona y se accede directamente a la pantalla de configuración del Arduino YUN.

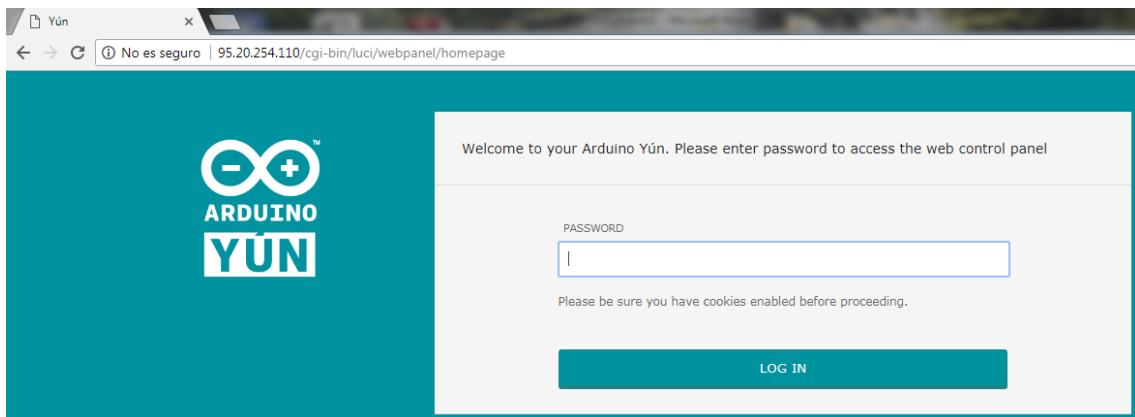


Figura 115: Acceso a Arduino desde IP pública

Este acceso solo funcionará durante un tiempo, hasta que cambie la IP del router. Una solución a este problema es convertir la IP en estática pagando una tarifa a la empresa operadora. Pero existen otro tipo de servicios gratuitos de empresas online en el que no es necesario convertir la IP en estática.

Por ejemplo, “ChangelP” que es una empresa online (www.ChangelP.com), da la opción de crear un dominio asignado a una dirección IP dinámica de forma gratuita. Esta IP dinámica asignada, se refresca automáticamente por medio de una comunicación establecida entre el router y el dominio creado. De esta forma, cada vez que la IP del router es cambiada, este la envía al dominio para que siempre conozca su dirección y mantenga la comunicación. Este servicio equivale a convertir una IP dinámica en estática, pero de forma virtual.

Para solicitar este servicio requiere crear una cuenta en la página web. Una vez registrados, se crea un dominio al que se le asigna la IP dinámica actual del router. El dominio creado para el acceso al prototipo es “*www.analizador_tfg_barbachano.dynamic-dns.net*”

Domain: analizador_tfg_barbachano.dynamic-dns.net

Total Records:3
Records Updated:3

Select All | Cancel All

| Hostname | Type | Value |
|----------|------|---------------|
| @ | A | 95.20.254.110 |
| ftp | A | 95.20.254.110 |
| www | A | 95.20.254.110 |

Figura 116: IP dinámica asignada al dominio en “ChangelP”

Se comprueba el funcionamiento del dominio accediendo desde una red externa, y como se observa en la captura de la figura 117, el dominio direcciona hasta la página de configuración del Arduino.

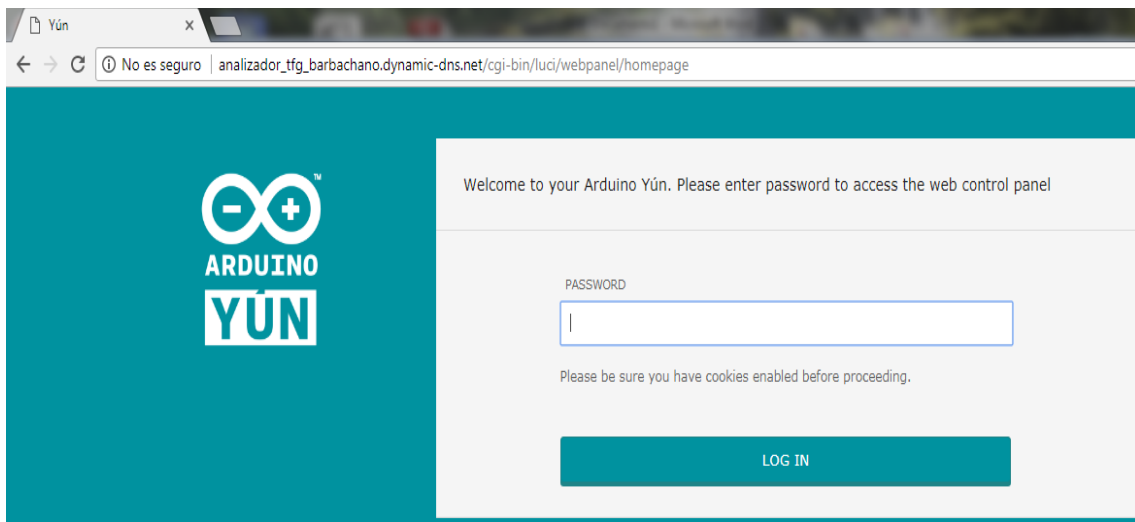


Figura 117: Acceso a Arduino desde el dominio

8. Calibración y monitorización de señales

Debido a los acondicionamientos de las señales muestreadas, las señales han sido reducidas en amplitud. Para calcular su valor original, debemos conocer su curva de calibración y estimar la ecuación de la recta.

Se supone que tratándose de transformadores de tensión y corriente, y el método de acondicionamiento implementado, esa curva de calibración debería ser una recta, debido a que teóricamente la señal de salida es lineal respecto a la señal de entrada. Pero en la práctica conviene hacer el análisis, ya que las no idealidades de los componentes pueden generar *offset* y desviaciones en la curva de calibración.

En la medida de la tensión, el valor a medir es prácticamente constante, por lo que con un simple factor de escalado nos sirve para calcular la amplitud de la señal medida. En cambio, la medida de corriente está comprendida en un rango de 0 a 25A, y según el *datasheet* del transformador de corriente tiene un 3% de no linealidad, por lo que conviene realizar la curva de calibración y obtener la ecuación característica.

Una vez calibradas las señales, se desarrollará un *sketch* con el que monitorizar las señales analógicas gráficamente desde la herramienta "Serial Plotter", como si se tratase de un osciloscopio; es decir, escalada en amplitud y tiempo con su natural forma de onda senoidal. Ver *sketch* "Analizador_de_redes_CALIBRACION" en anexo A.6.4.

En el *sketch* se utilizará la ecuación característica de la señal de corriente, y el factor de escalado de la señal de tensión, para representar a su escala original, en el "Serial Plotter", las señales muestreadas.

En el caso de la señal de corriente se analizará el desfase temporal entre la señal de salida de corriente respecto a la señal de salida de tensión, ya que es uno de los parámetros a medir. Debido a que los sensores son transformadores y están formados por devanados, estos provocan retraso en la corriente que circula por ellos, pudiendo perturbar la señal muestreada, y falsear la medida de desfase temporal.

Este retraso de corriente afecta sobre todo al circuito del transformador de corriente, ya que en este caso, la corriente medida se transforma a voltaje en el divisor de tensión, con lo cual, si la corriente estaba desfasada, la caída de tensión de la resistencia también lo estará.

8.1. Calibración y monitorización de señal de tensión

La señal de tensión a muestrear es la tensión de red, equivalente a los 230V eficaces, que una vez acondicionada, se convierte en una señal de 2Vp, con lo cual podemos calcular el factor de escalado para implementarlo en el software:

$$K_V = \frac{\text{Señal}_{ENTRADA}}{\text{Señal}_{SALIDA}} = \frac{230V * \sqrt{2}}{2V} = 115 * \sqrt{2}$$

Ajustado el factor de escalado en el *sketch*, alimentamos la PCB con la tensión de red y cargamos el programa a Arduino. Abrimos la pantalla del “Serial Plotter” y comprobamos la forma de onda de la señal muestreada y entre que niveles de amplitud oscila. Además nos sirve para comprobar la calidad de la señal interpretada por Arduino.

La imagen de la figura 118 corresponde a la interpretación de la señal muestreada por Arduino, escalada en amplitud, hasta los valores de pico de la tensión de red. Sin embargo, la escala de tiempos no está bien, ya que en 100ms debería contener 5 ciclos de señal senoidal de 50Hz. El error puede estar relacionado con la comunicación serie entre el PC y Arduino, y su velocidad de conexión.

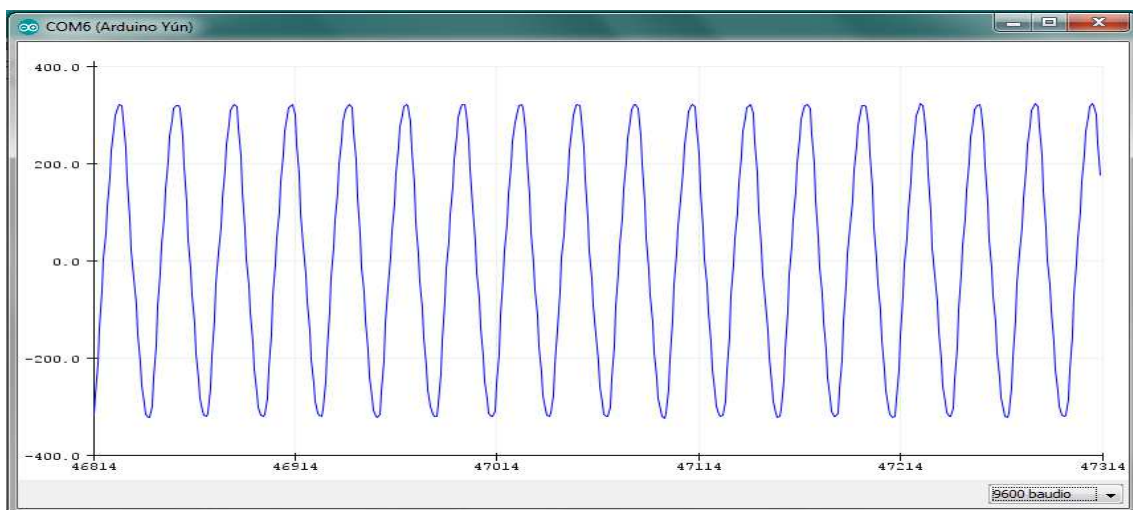


Figura 118: Serial plotter

8.2. Calibración y monitorización de señal de corriente

Para realizar la curva de calibración de la señal de corriente utilizaremos un banco de resistencias conectadas en paralelo, alimentadas por un autotransformador. Dicho autotransformador está alimentado a una tensión de línea de 400V y su protección rearmable limita la corriente a 20A, con lo cual, en la curva de calibración se tomarán muestras de corriente hasta 20A, el resto de valores se estimarán mediante la ecuación de la curva.



Figura 119: Banco de resistencias



Figura 120: Autotransformador

El sensor de corriente se coloca en la fase principal del secundario del autotransformador, seguido de una pinza amperimétrica con la que contrastar los valores de corriente medidos, como se muestra en la fotografía de la figura 121.



Figura 121: Resistencias en paralelo con autotransformador

Con el osciloscopio se comprueba la señal de salida del circuito de acondicionamiento de corriente pinchando la sonda en el pin de la señal analógica 2 de la PCB.

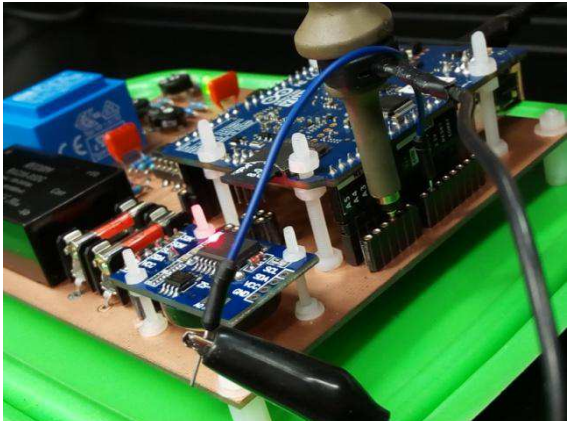


Figura 122: Sonda en señal de corriente



Figura 123: Monitorización de señales con osciloscopio

El barrido de muestras se realiza en pasos de 0,5A. Para ello se conectan en paralelo las 3 resistencias de menor valor del banco (55Ω cada una de ellas, en paralelo equivale a una resistencia de 18.33Ω), y en función de la medida indicada por la pinza amperimétrica, se va aumentando la tensión de salida del autotransformador. Si la salida es del 100% de tensión, equivalente a los 400V de tensión de línea, para la impedancia conectada, se consigue una corriente máxima de 21,82A.

Con los datos obtenidos de las muestras, se exportan a Excel para elaborar una tabla y un gráfico en el que se representa la señal de entrada (I) frente a la señal de salida (V). En el gráfico se calcula la ecuación característica de la curva y se compara con la curva obtenida de los resultados experimentales. Para ver los resultados obtenidos, ver tabla 5 en anexo A.2.

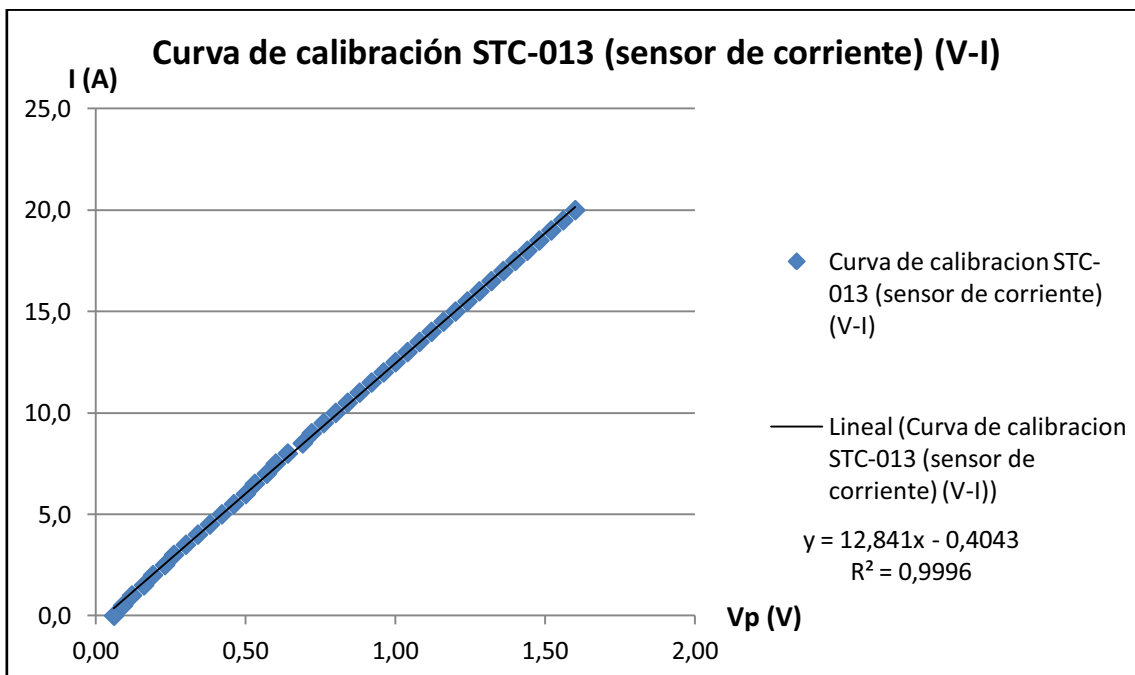


Figura 124: Gráfico de curva de calibración STC-013 (V-I)

Como se puede observar en el gráfico de la figura 124, la respuesta de salida se puede considerar lineal, ya que en las pequeñas desviaciones de los resultados experimentales, influye el error de los aparatos de medida, error humano, etc. Sí que será necesario corregir el *offset* en el software para conseguir mayor precisión en la medida de corriente.

Calculada la ecuación característica de la curva, introducimos el valor en el *sketch* de calibración y así repetir el proceso anterior de monitoreo de señal mediante “Serial Plotter”, solo que esta vez con la señal de corriente.

Las capturas de las figuras 125 y 126, corresponden a las medidas de corriente de 5A y 10A eficaces, respectivamente. Se puede comprobar que los valores de pico son por raíz de 2.

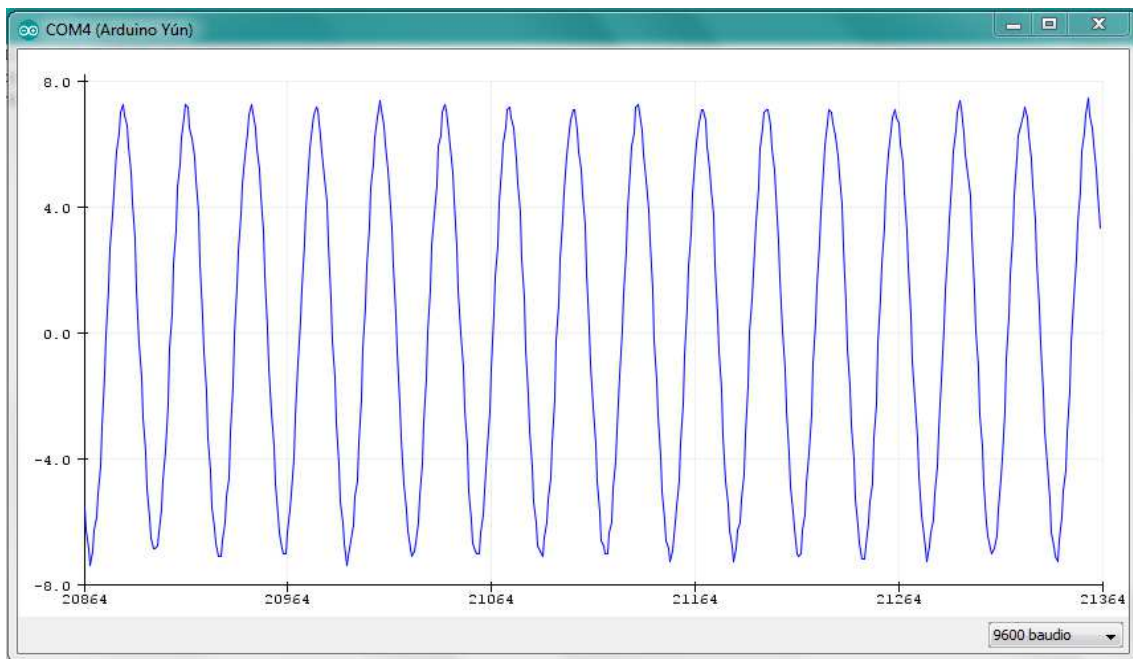


Figura 125: Serial plotter

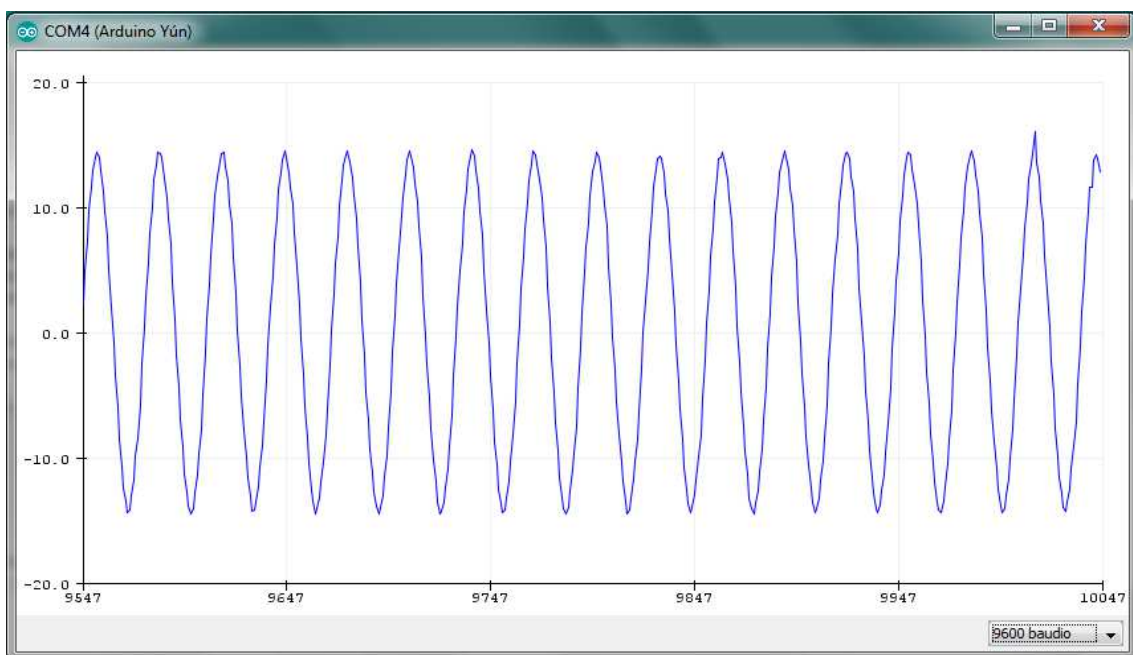


Figura 126: Serial plotter

8.3. Calibración del factor de potencia

Para el análisis de la curva de calibración referida al desfase entre las señales de corriente y tensión, utilizamos el mismo banco de resistencias alimentadas con el autotransformador. Siendo la resistencia teóricamente una carga totalmente resistiva, es decir, sin impedancias capacitivas ni inductivas parasitas, podemos deducir que el desfase entre las señales debe ser de 0° , lo que equivale a un factor de potencia igual a la unidad.

Conectando al osciloscopio cada señal a un canal diferente, se toman muestras del desfase en grados, a medida que se aumenta la corriente medida, mediante el aumento de tensión de salida del autotransformador. El osciloscopio cuenta con una herramienta que realiza el cálculo de desfase entre las dos señales.

La imagen de la figura 127 muestra las conexiones de las sondas en los pines analógicos de la PCB. La imagen de la figura 128 es la captura de las medidas del osciloscopio de tensión (señal amarilla), y corriente (señal azul).

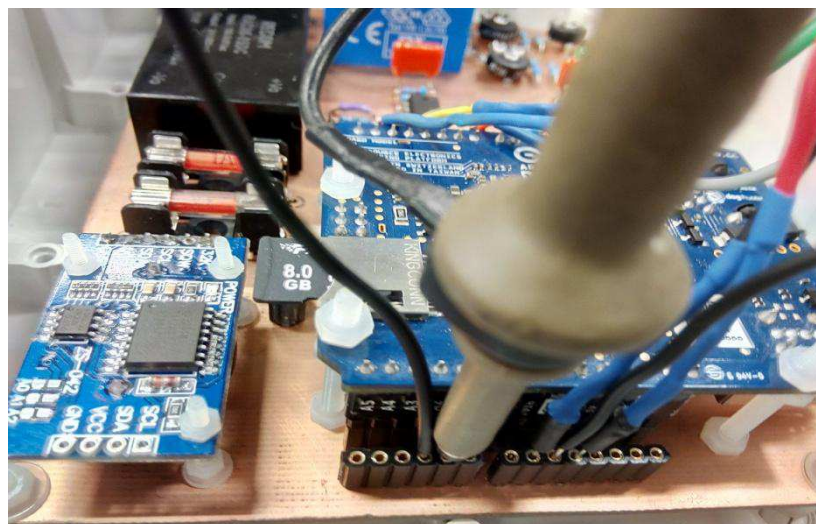


Figura 127: Fotografía de sonda conectada a PCB

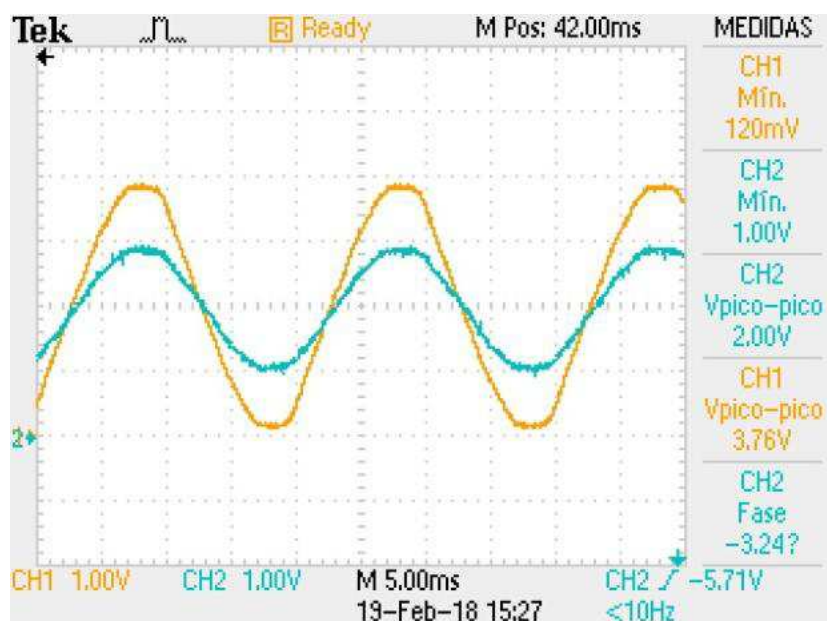


Figura 128: Captura de osciloscopio de tensión y corriente

Una vez tomadas las muestras, se exportan a Excel los resultados para elaborar una tabla y el gráfico de la figura 129 que representa el ángulo de desfase (en grados) frente a la corriente (en A), para estimar la ecuación característica de la recta. Ver los resultados obtenidos en las muestras en la tabla 6 del anexo A.3.

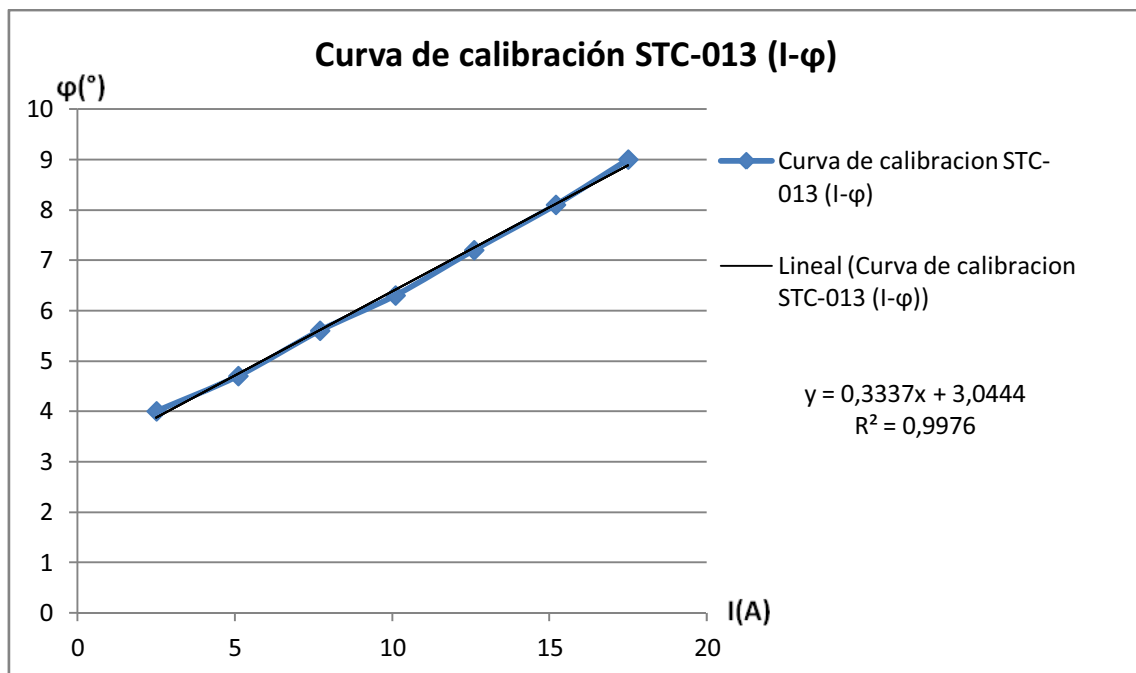


Figura 129: Gráfico de curva de calibración STC-013 (I-φ)

Por los resultados obtenidos en el ensayo, se deduce que era necesario realizar el análisis, ya que a medida que aumenta la corriente, el desfase es mayor. Conociendo la ecuación característica podremos reducir el error en el cálculo de factor de potencia, que es una de las medidas básicas.

La precisión con la que calculemos el factor de potencia, va a derivar en la precisión de la potencia activa, y por lo tanto en el cálculo de energía, ya que estas magnitudes son función del factor de potencia.

9. Desarrollo e implementación del software

El desarrollo del software del dispositivo se programa por medio del IDE Arduino. Para interpretar los pasos a seguir en la programación del mismo, se construyen unos diagramas de flujo, basados en bloques de subrutinas.

- **Cuerpo del programa:** Representa los bloques de las funciones principales a gran escala. Control de las llamadas a subrutinas e interrupciones.
- **Subrutina de muestreo de señales:** El primer bloque del programa consiste en analizar las señales muestreadas y calcular los valores instantáneos.
- **Subrutina de cálculos de potencia y energía:** En el segundo bloque se calculan parámetros en función de los datos instantáneos y del tiempo.
- **Subrutina de registro de datos:** Control de la lectura y escritura de datos acumulados.
- **Subrutina de monitorización y exportación de datos:** Escritura en la pantalla local de los datos calculados y acumulados, y exportación de los mismos a la nube.

9.1. Cuerpo del programa

El cuerpo del programa está compuesto por un bucle infinito en el que se hacen las llamadas a subrutinas en el orden indicado. Todas las subrutinas se realizan en cada ciclo de programa con el objetivo de refrescar los datos, excepto la subrutina de registro de datos. Debido al tiempo de programa que supone el acceso a la tarjeta SD, se accede a la subrutina una vez por minuto.

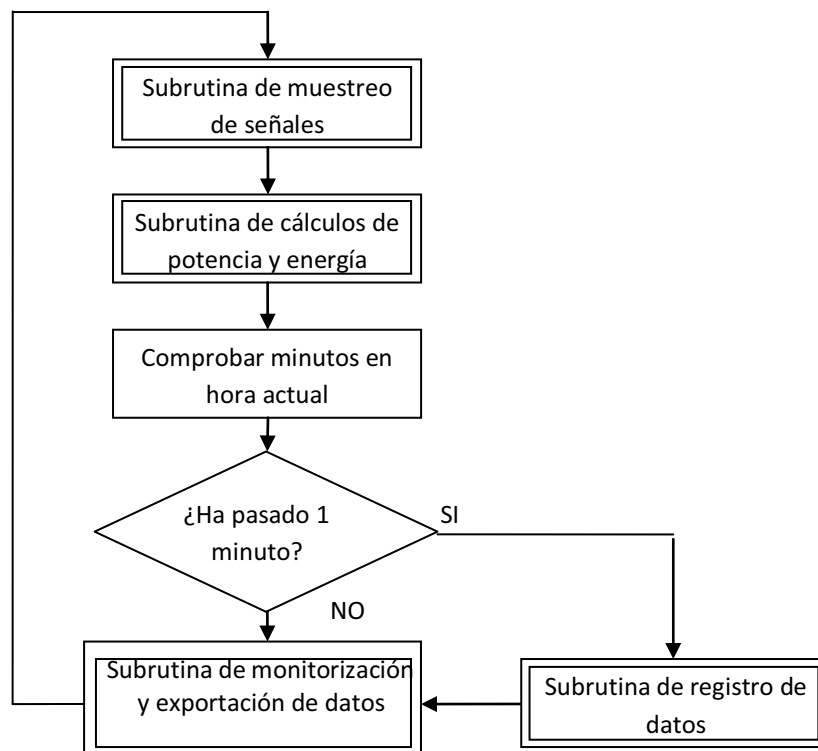


Figura 130: Diagrama de flujos cuerpo de programa

9.2. Subrutina de muestreo de señales

El bloque de muestreo de señales consiste en calcular el valor RMS de las señales analógicas de tensión y corriente muestreadas durante 5 ciclos (100ms), además del factor de potencia. Los valores calculados se almacenan en variables del programa, para operar en otras subrutinas.

Para el cálculo del valor RMS hay que tener en cuenta que debido a la frecuencia de muestreo, el software no interpreta la señal como una función continua senoidal, sino que lo interpreta como señales discretas cuantificadas según la resolución del convertidor analógico digital de Arduino, con lo cual se debe calcular como tal.

Siendo T, el valor tiempo del periodo para señales de función continua; y N, el número de muestras cuantificadas por el CAD para señales de función discreta:

$$V_{RMS}(continua) = \sqrt{\frac{1}{T} \left[\int_0^T [v(t)]^2 dt \right]} \quad I_{RMS}(continua) = \sqrt{\frac{1}{T} \left[\int_0^T [i(t)]^2 dt \right]}$$

$$V_{RMS}(discreta) = \sqrt{\frac{1}{N} \left[\sum_{i=1}^N v_i^2 \right]} \quad I_{RMS}(discreta) = \sqrt{\frac{1}{N} \left[\sum_{i=1}^N i_i^2 \right]}$$

En las imágenes de las figuras 131 y 132 se representa gráficamente la forma de onda senoidal como función continua (imagen izquierda), y como función discreta (imagen derecha), respectivamente. La imagen de la derecha coincide con la interpretación del CAD de Arduino de la señal real; solo que en este caso se ha representado con una frecuencia de muestreo de 1KHz y el CAD de Arduino muestrea a una frecuencia máxima de 10KHz.

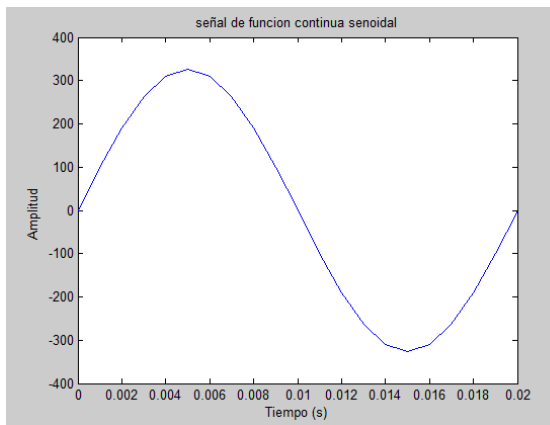


Figura 131: Representación gráfica continua Matlab

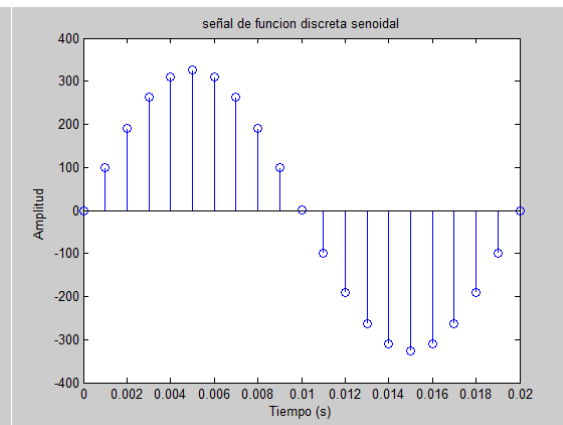


Figura 132: Representación gráfica discreta Matlab

Durante el muestreo de las señales, se inicia la cuenta de un temporizador en el flanco de bajada de la señal de tensión, justo en el paso por cero. El temporizador se detiene cuando la señal de corriente pasa también por cero en la transición del flanco de bajada. Con la diferencia temporal del paso por cero entre las dos señales, y conocida la frecuencia angular, se calcula la longitud en radianes, lo que equivale al ángulo de desfase (φ).

Siendo ω , la frecuencia angular en función de la frecuencia de la red (f):

$$\omega = 2 * \pi * f = \left[\frac{rad}{s} \right] \quad \varphi = t_{DEFASE} * \omega = [rad] \quad F.P. = \cos(\varphi)$$

En la gráfica de la figura 133 se muestra el ejemplo de corriente generada por una carga capacitiva, retrasada 15 ms respecto a la señal de tensión, lo que equivale a un desfase angular de 270° de retraso, o 90° de adelanto.

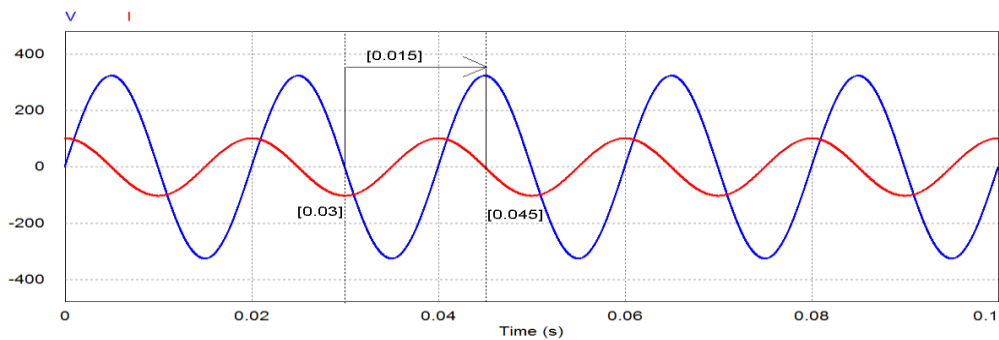


Figura 133: Simulación PSIM

El diagrama de flujos de la figura 134 representa los procesos a realizar por la subrutina de muestreo de señales.

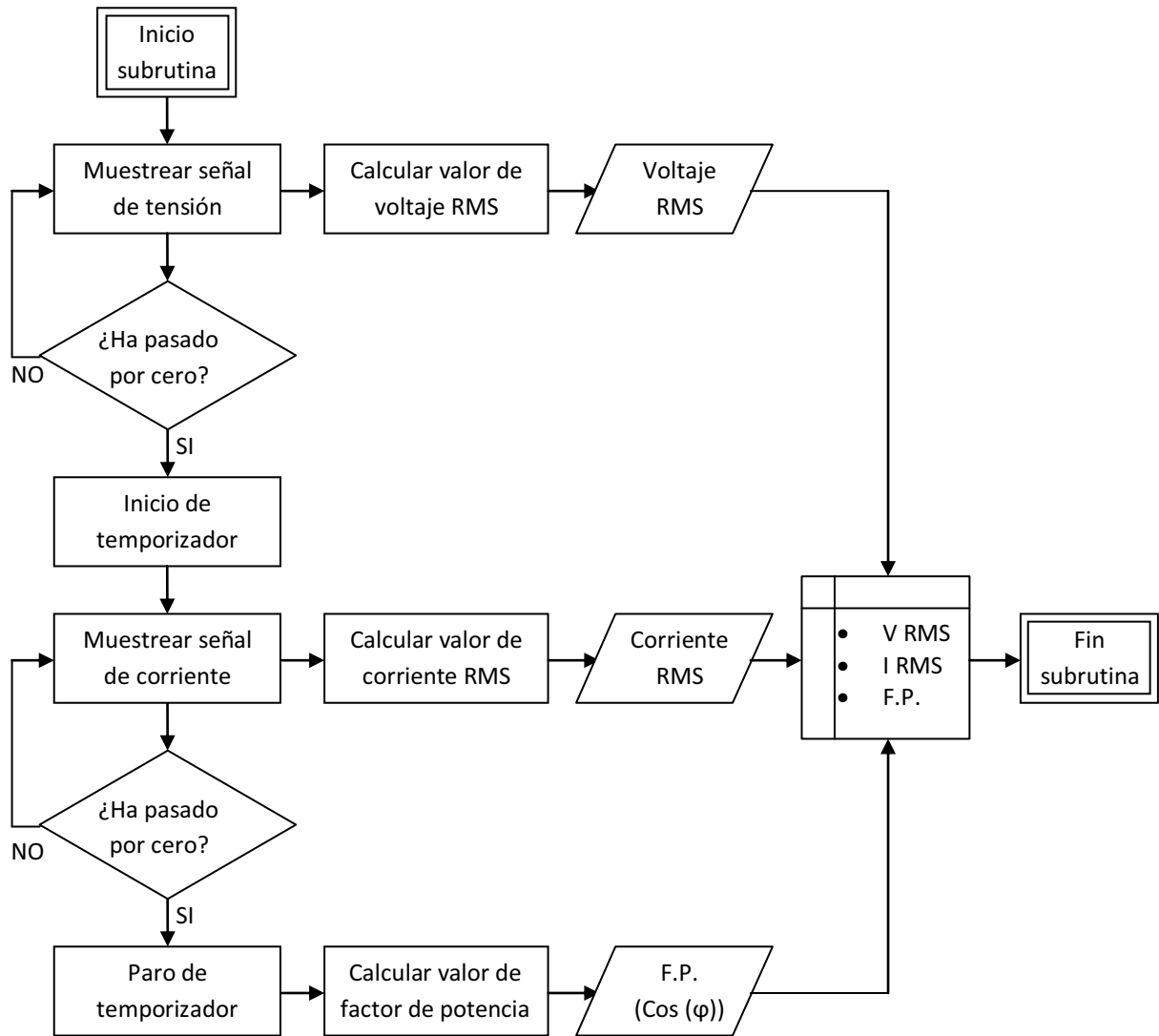


Figura 134: Diagrama de flujos de subrutina de muestreo de señales

9.3. Subrutina de cálculos de potencia y energía

Esta subrutina calcula los datos de potencia a partir de los datos guardados anteriormente en la memoria interna, de tensión, corriente y factor de potencia. Para el cálculo de energía, la subrutina calcula la energía consumida o generada en función del tiempo de duración del ciclo completo del programa.

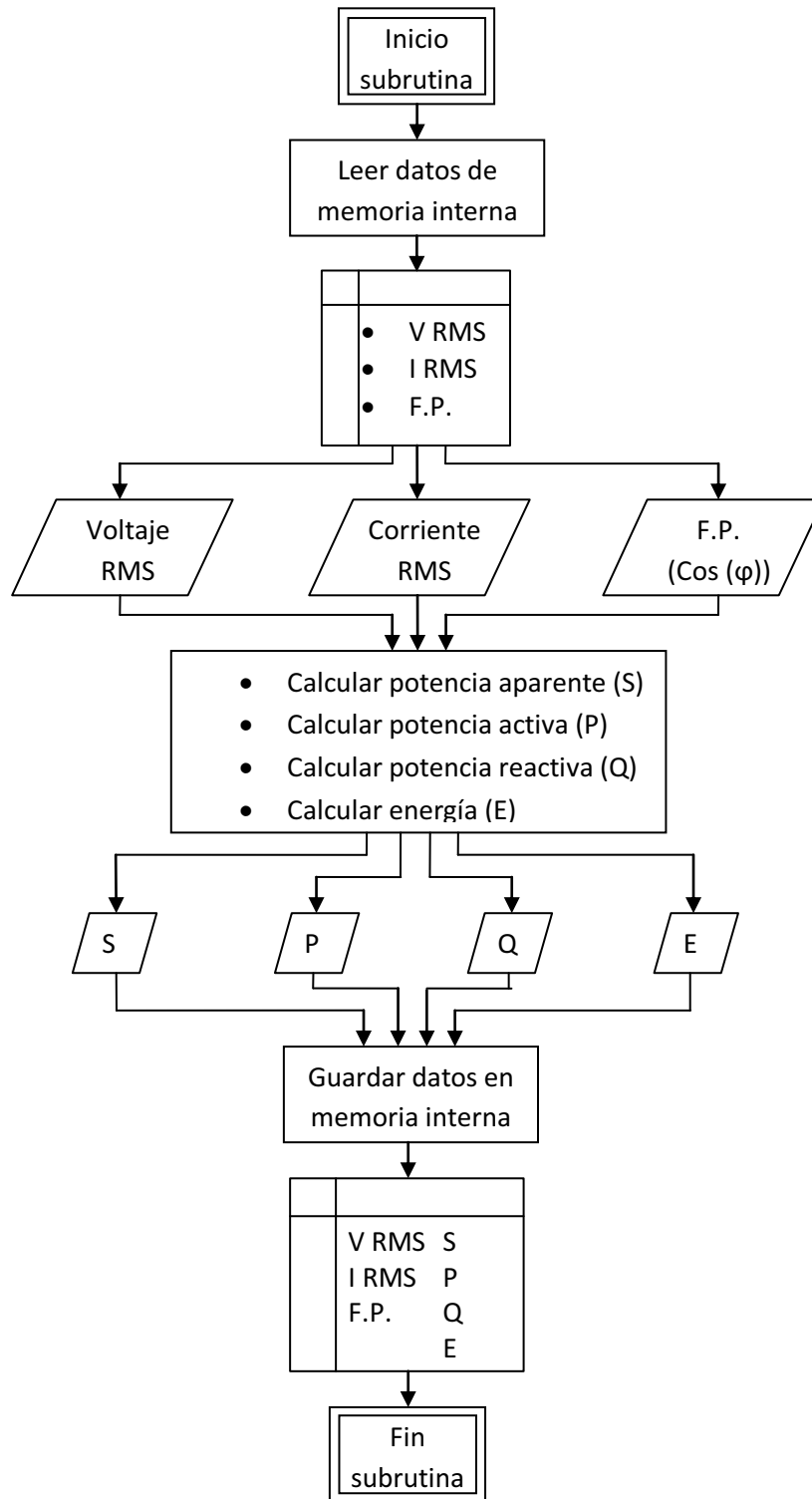


Figura 135: Diagrama de flujos de subrutina de cálculos de potencia y energía

9.4. Subrutina de registro de datos

La subrutina de registro de datos se realiza una vez por minuto. Consiste en un registro de datos en la tarjeta SD sobre un formato de archivos de texto. La energía consumida o generada durante ese minuto la añade al totalizador mensual de energía correspondiente. Además registra los picos máximos de corriente y potencia del mes actual; sobrescribiendo los datos instantáneos sobre los registrados, siempre que en la comparación sean superiores.

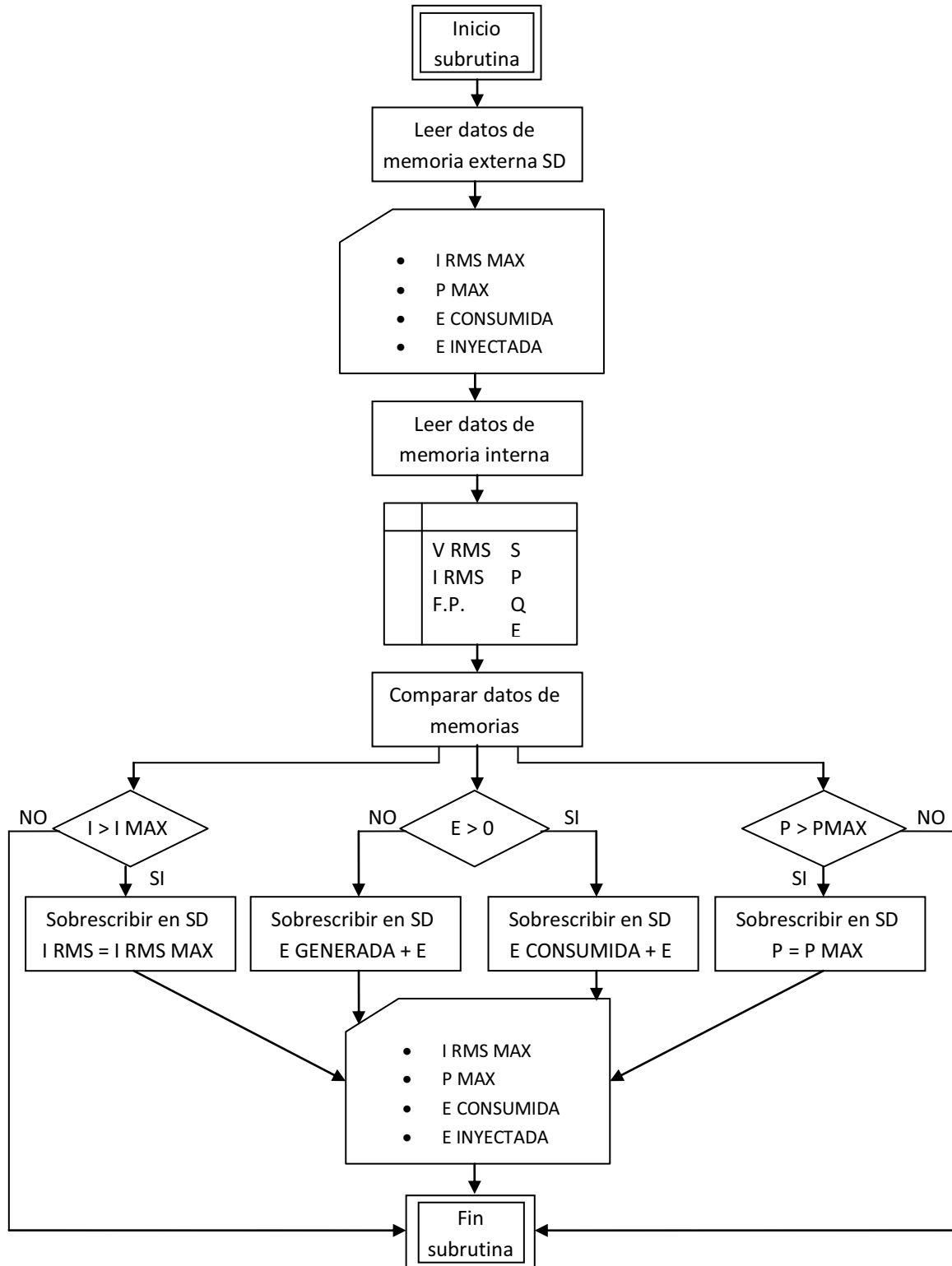


Figura 136: Diagrama de flujos de subrutina de registro de datos

9.5. Subrutina de monitorización y exportación de datos

La última subrutina se encarga de las operaciones de interfaz entre el usuario y el dispositivo. Escribe todos los datos instantáneos y acumulados en la pantalla local; y además los exporta a la nube haciendo la función de un servidor web.

Como no entran todos los datos a la vez en la pantalla LCD, en la práctica se mostrarán los datos por grupos, y mediante un pulsador se irá rotando el grupo de datos a visualizar.

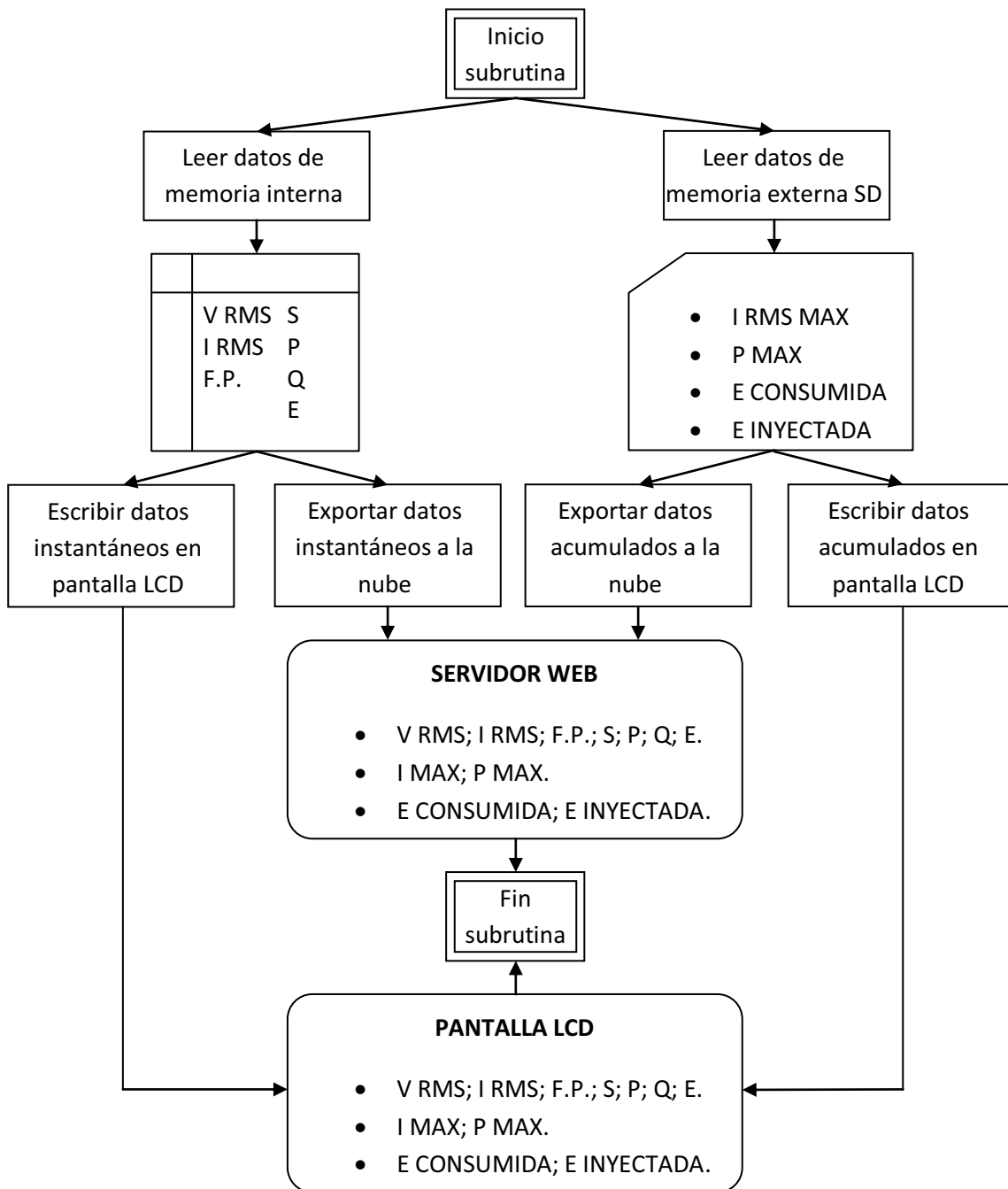


Figura 137: Diagrama de flujos de subrutina de monitorización y exportación de datos

10. Proyecto final

Tras finalizar la implementación del software, el prototipo ya está listo para el funcionamiento. Por razones estéticas y de protección del mismo, se realiza el montaje de la PCB con sus accesorios externos en una caja de registro de montaje superficial.

La caja del montaje (ver figura 138) es un modelo GEWISS 44207 con dimensiones de 190x140x70mm (largo, ancho, fondo):



Figura 138: Caja de registro de montaje superficial

En la fotografía de la figura 139 se observa como la PCB se monta en la base del fondo de la caja, con espacio suficiente para las conexiones laterales de alimentación y del sensor de corriente. Además se monta un pasa muros en un lateral de la caja para el acceso de los 4 hilos de dichas conexiones.



Figura 139: PCB montada en la caja de registro

Se instalan en la tapa de la caja la pantalla LCD, el pulsador de encendido y cambio de pantalla, y un LED de encendido del dispositivo. En la fotografía de la figura 140 se muestra la vista de la parte interna de la tapa.

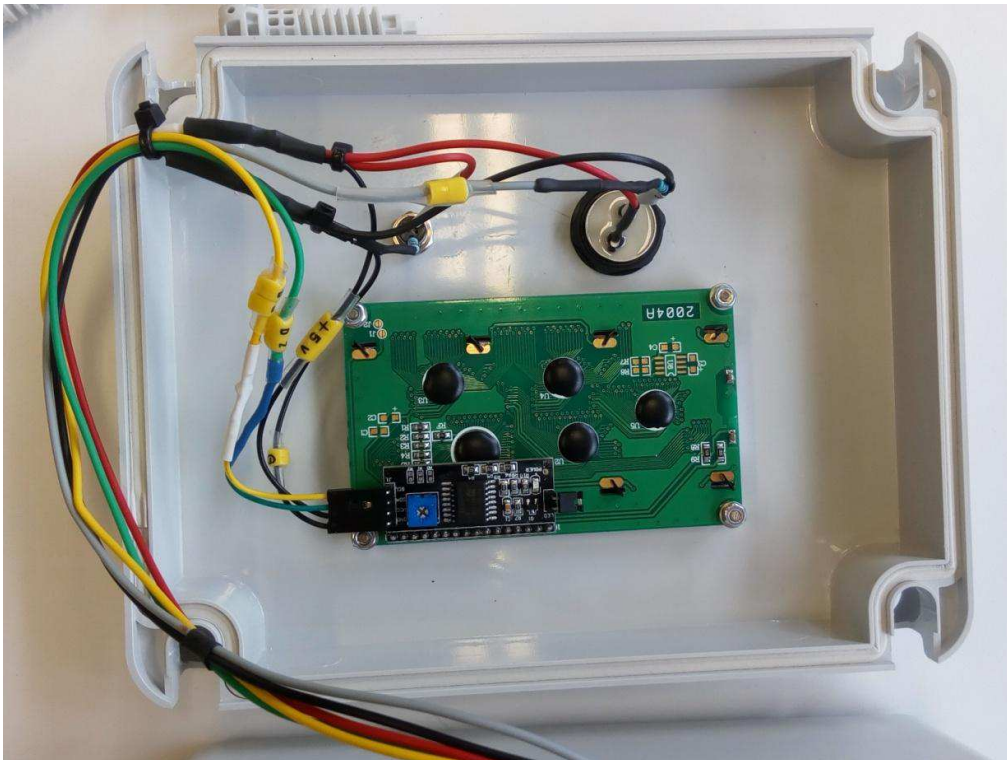


Figura 140: Vista interna de la tapa de la caja de registro

En la figura 141 se observa el acabado final de la parte interna.

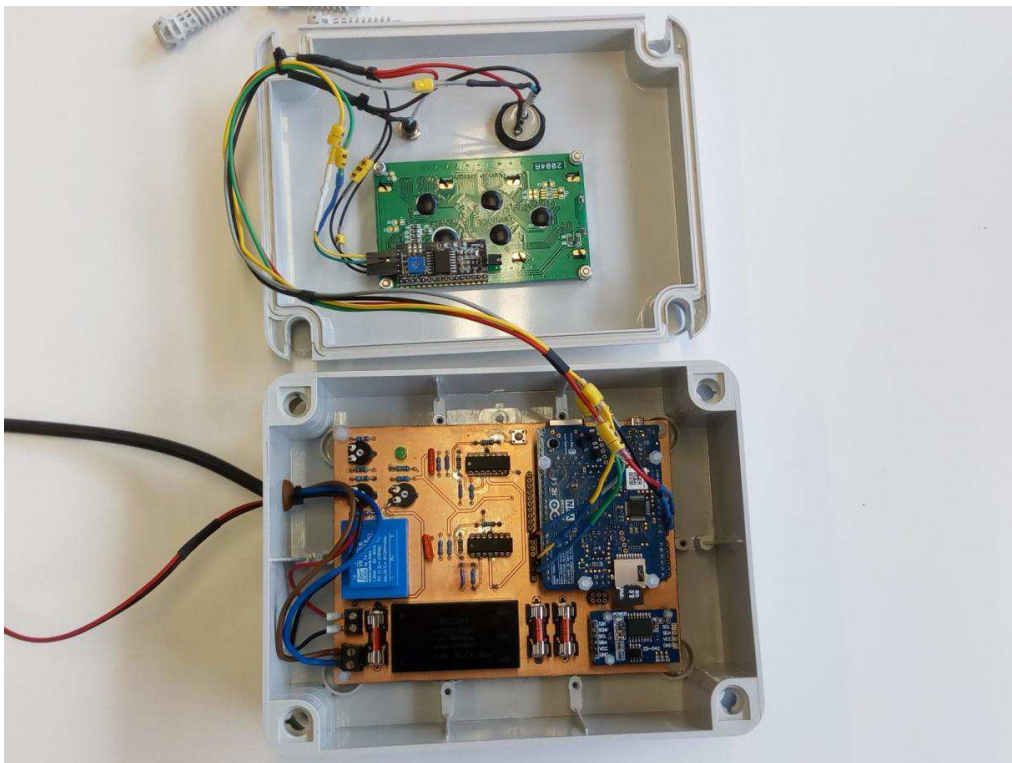


Figura 141: Vista interna del montaje

El acabado final externo que presenta el dispositivo, se muestra en la fotografía de la figura 142.

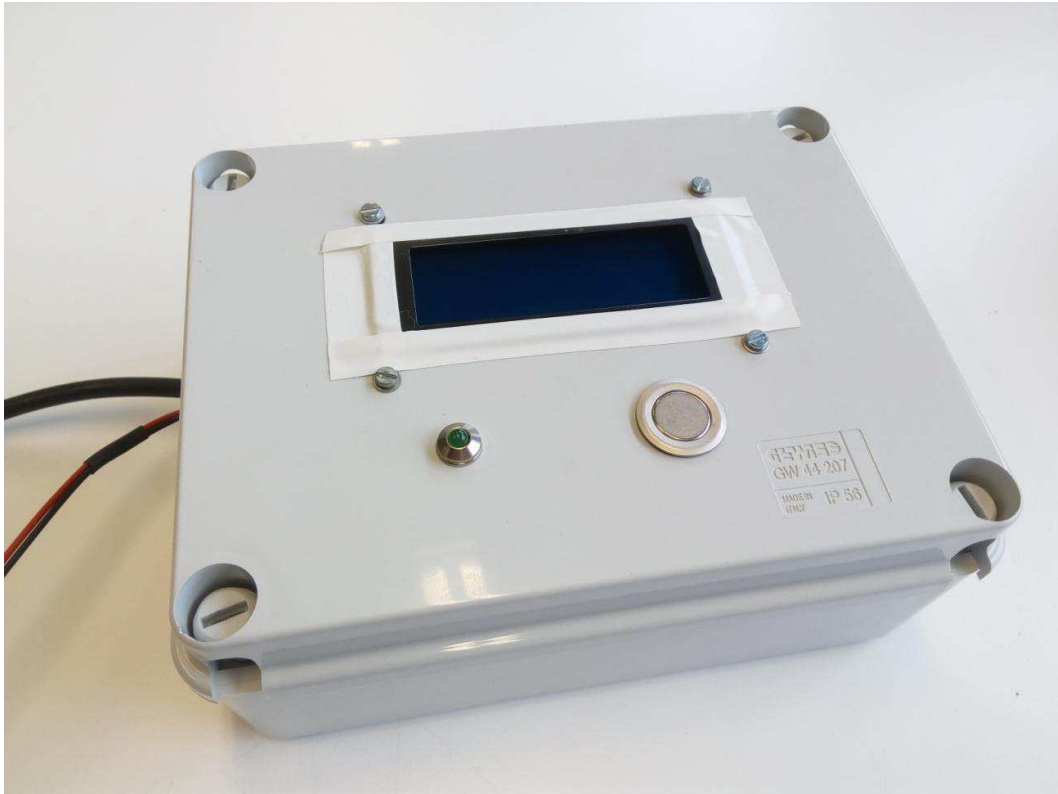


Figura 142: Vista exterior del montaje en la caja de registro

Ahora se procede al encendido del aparato, para comprobar la forma en que representa en pantalla los datos. Las dos primeras pantallas corresponden a representación de datos instantáneos. La primera muestra los valores de corriente, tensión y factor de potencia; y la segunda representa los valores del triángulo de potencias. Las figuras 143 y 144 corresponden a las dos pantallas de valores instantáneos.



Figura 143: Pantalla de valores instantáneos



Figura 144: Pantalla de valores instantáneos

La tercera pantalla (ver figura 145) muestra los valores acumulados en el mes actual; de corriente máxima, potencia máxima, y energía consumida.



Figura 145: Pantalla de máximos y acumulados

Las 4 siguientes pantallas (ver figuras 146, 147, 148 y 149) muestran los totalizadores de energía consumida y generada en todos los meses del año.



Figura 146: Pantalla de totalizador de consumo energético



Figura 147: Pantalla de totalizador de consumo energético



Figura 148: Pantalla de totalizador de inyección energética



Figura 149: Pantalla de totalizador de inyección energética

Por último, se ha implementado una pantalla de avisos. En caso de que se produzcan errores en la lectura o escritura de la tarjeta SD, debido al formato, estado defectuoso, o cualquier otra razón; esta pantalla mostrará el aviso, como aparece en la figura 150.

También muestra la fecha y hora del momento en el que se inició el dispositivo. Este dato sirve para verificar el correcto funcionamiento del dispositivo desde el momento en el que lo iniciamos por primera vez, ya que está programado para que se reinicie automáticamente en caso de defecto del software, y cada reinicio del dispositivo actualizaría la fecha y hora a la del último reinicio.

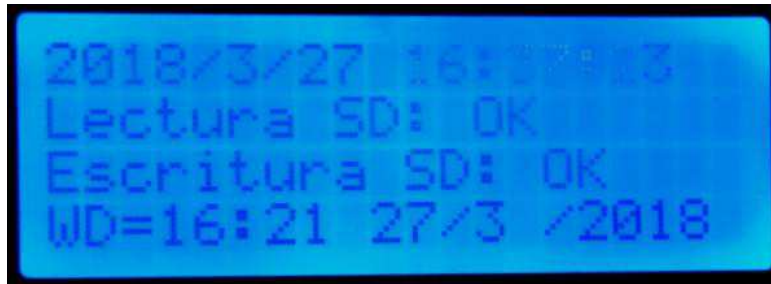


Figura 150: Pantalla de avisos

Se ha implementado la función del servidor web, con acceso a los datos, tanto desde la red pública como desde la red privada. Desde la red privada es suficiente con abrir desde el navegador la IP estática asignada a Arduino con los comandos añadidos de “/Arduino/YUN/” (192.168.1.131/Arduino/YUN/). Desde la red pública se puede acceder mediante el dominio y los comandos añadidos, ya que el dominio dirige hacia la página principal de configuración del Arduino YUN. La dirección desde la red pública es www.analizador_tfg_barbachano.dynamic-dns.net/arduino/YUN/.

La imagen de la figura 151 corresponde con el acceso a los datos mostrados por el prototipo desde una IP pública, y la imagen de la figura 152 corresponde al acceso desde la red privada.

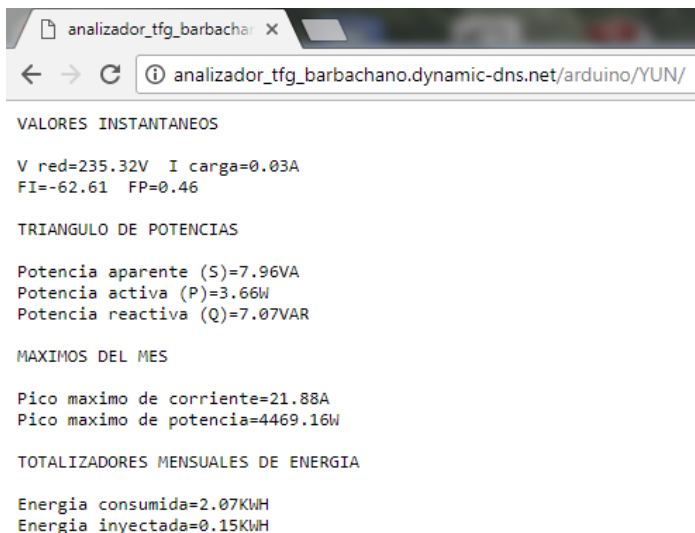


Figura 151: Acceso a los dato desde la red pública

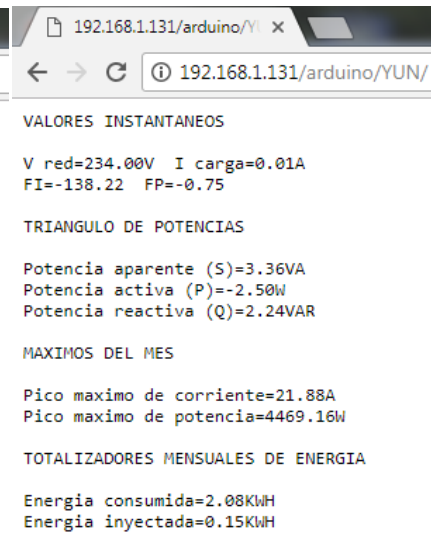


Figura 152: Acceso a los datos desde la red privada

11. Medida de error

Tratándose de un instrumento de medida, es indispensable conocer la precisión en el valor de la magnitud medida. Es necesario calcular el error de medición para conocer el grado de exactitud de la medida realizada en una magnitud.

Al igual que en el proceso de calibración del sensor de corriente, se utilizan el mismo banco de resistencias y autotransformador; con los que se generarán las variaciones de tensión y corriente a medir. Por cada medida realizada con el prototipo construido, se debe obtener otra medida más, en las mismas condiciones, mediante un dispositivo en el cual, se interpreten sus medidas como valor verdadero.

De esta manera, por cada muestra se obtiene el valor medido por el dispositivo construido y el valor verdadero, al que referenciar el valor medido. A partir de estos datos experimentales se calcula el error absoluto y el error relativo.

El error absoluto es la diferencia entre el valor medido y el valor verdadero, y se expresa en unidades de la propia magnitud (amperios, voltios,...). El error relativo es la relación entre el error absoluto y el valor verdadero, y se expresa en cociente o como porcentaje de dicho valor verdadero.

$$E_{ABSOLUTO} = Valor Verdadero - Valor Medido$$

$$E_{RELATIVO} = \frac{E_{ABSOLUTO}}{Valor Verdadero} * 100$$

Se toma como referencia de valor verdadero, el valor medido por un polímetro con pinza amperimétrica modelo PCE-PCM1 (ver ficha técnica en anexo). Este dispositivo mide todas las magnitudes de las que se van a calcular el error de medición. En este caso se utiliza el voltímetro, amperímetro, y medidor de ángulo de factor de potencia; cuyos errores de medida de tensión, corriente, y factor de potencia, son de 1.2%, 2%, y 0.02% respectivamente.



Figura 153: Polímetro PCE-PCM1

11.1. Error de medida en tensión

Para calcular el error en la medida de tensión, se realiza un barrido de muestras comprendido en un rango de 220V a 240V, ya que la tensión de la red se considera como un valor constante de 230V.

Con el polímetro conectado en paralelo a la salida del transformador, se ajusta este hasta obtener una tensión de 220V. Una vez ajustada la tensión de salida, se conecta a la alimentación del dispositivo construido. Partiendo de esta tensión de salida, se toman muestras de los valores medidos por los dos instrumentos, para diferentes niveles de tensión, hasta alcanzar los 240V.

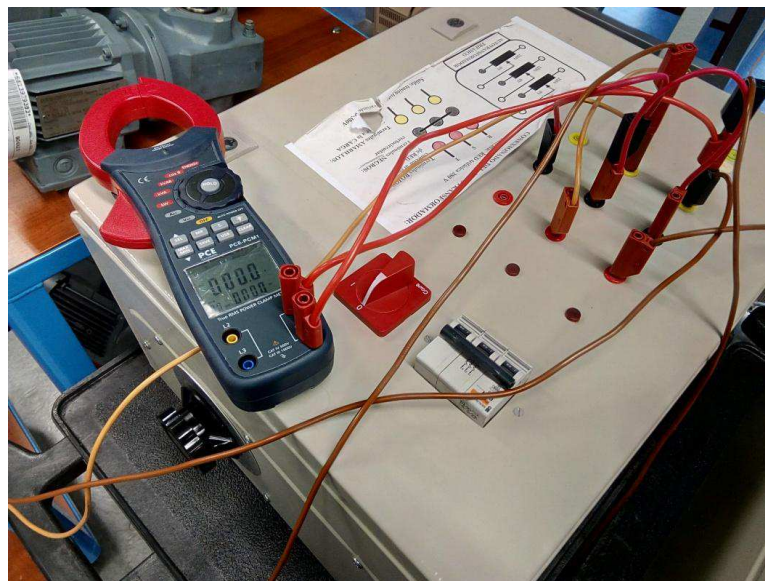


Figura 154: Polímetro conectado a autotransformador

Los datos experimentales medidos por la pinza PCE y el dispositivo se exportan a una tabla de Excel y se elaboran dos gráficos; uno del error absoluto frente al valor verdadero, y otro del error relativo frente al valor verdadero. Ver resultados en la tabla 7 del anexo A.4.1.

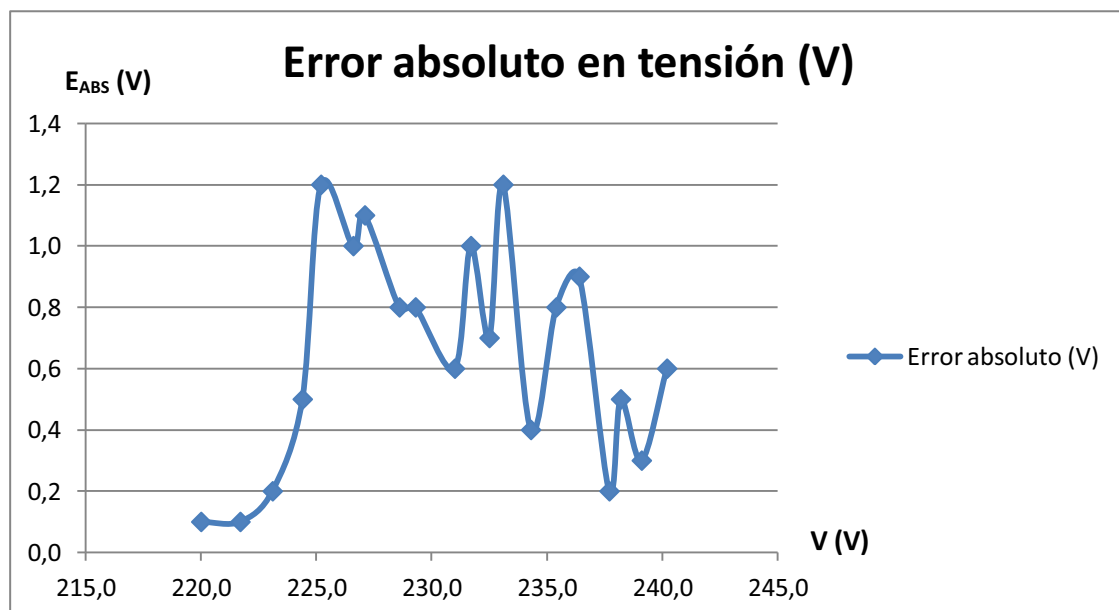


Figura 155: Gráfico de error absoluto en tensión

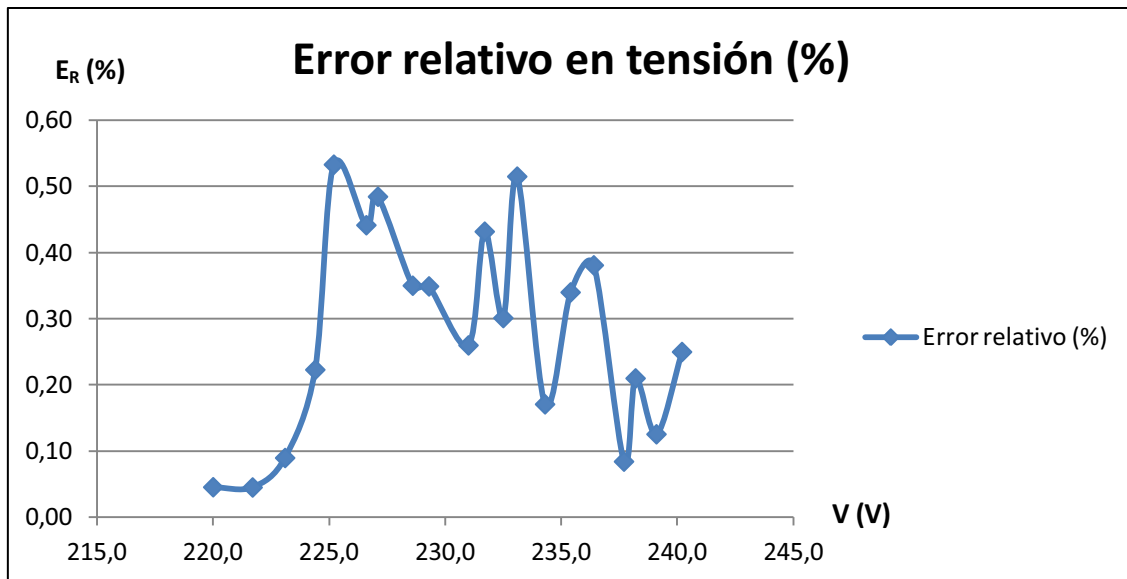


Figura 156: Gráfico de error relativo en tensión

Como se muestra en los gráficos de las figuras 155 y 156, la línea de error sufre una gran oscilación debido a que la medida de los dos instrumentos en este rango de tensiones no se estabiliza y oscila continuamente alrededor de 1V. A pesar de los problemas encontrados en la toma de muestras se continúa con el análisis.

Según los datos obtenidos, en el mismo punto de valor verdadero (225,2V), coinciden el máximo error absoluto, con un valor de 1,2V; y el máximo error relativo que equivale a 0,53%.

Teniendo en cuenta que la medida de valor verdadero se obtiene con un instrumento que tiene un error de 1,2% en la medida de tensión; un 0,53% de error sobre este, se puede considerar un valor aceptable.

11.2. Error de medida de corriente

En la medida de error de corriente, se alimenta el banco de resistencias con la salida del autotransformador ajustada a un valor constante de 230V, con el que alimentar también el dispositivo construido. Con todas las resistencias del banco conectadas en paralelo, se consiguen diferentes valores de corriente mediante combinaciones de las resistencias activadas.

La misma fase se mide con la pinza PCE y la pinza STC-013 del dispositivo, con la misma polaridad, y se realiza un barrido de muestras en un rango de 0 a 20A. Aunque el rango de medida del prototipo es de 0 a 25A, como se ha explicado en apartados anteriores, las protecciones del autotransformador limitan la corriente de salida en 20A, con lo cual no se superará este límite.

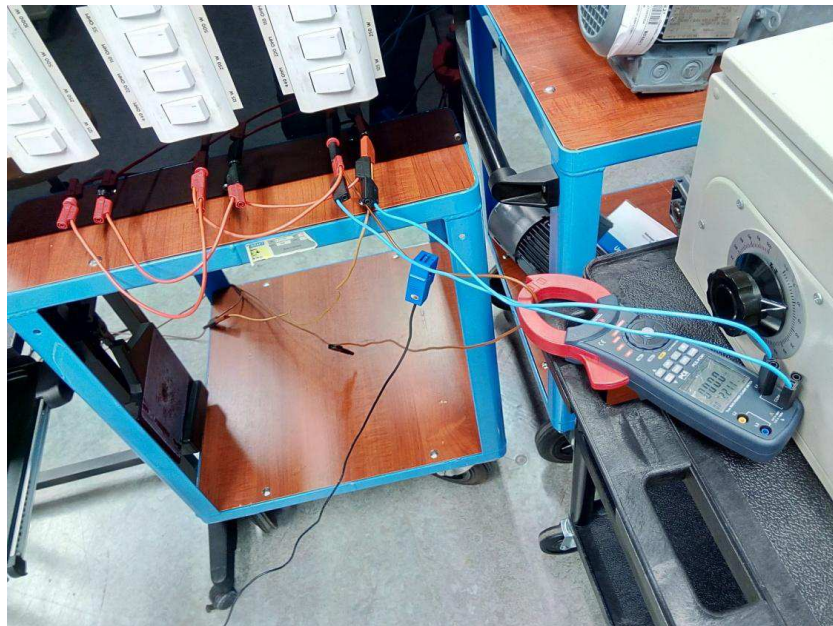


Figura 157: Conexión de las pinzas amperimétricas.

Al igual que en el caso anterior, se exportan los datos a una tabla de Excel, en la que se calcula por cada muestra realizada, sus errores absolutos y relativos, con los que construir dos gráficos. Ver los resultados obtenidos en la tabla 8 del anexo A.4.2.

El gráfico de la figura 158 representa el error absoluto en la medida de corriente frente al valor verdadero; y el gráfico de la figura 159 corresponde al error relativo en corriente frente al valor verdadero.

A diferencia de la medida de tensión, los valores medidos en cada muestra obtenida, son más estables, debido a que también son valores más pequeños que los medidos en tensión. Esto es una ventaja, ya que además de realizar la toma de muestras con más precisión; se reduce el error humano.

A pesar de dar mejores sensaciones la toma de muestras en la medida de corriente, como se muestra en los gráficos de las figuras 158 y 159, las líneas de error no mantienen una tendencia lineal.

En el gráfico de error absoluto en corriente, se observa un incremento de error en el último tramo del rango, a partir de los 18A aproximadamente.

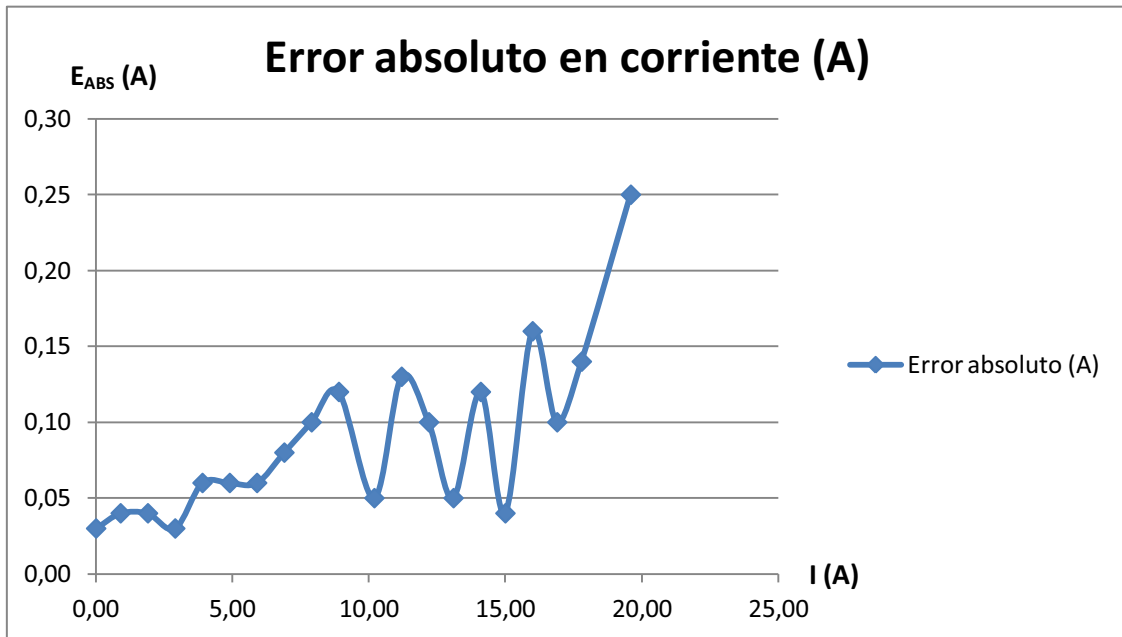


Figura 158: Gráfico de error absoluto en corriente

En el gráfico de error relativo en corriente, existen unos puntos críticos en las dos primeras muestras de 0A y 0,9A. El primero de ellos es una indeterminación y se ha eliminado del gráfico; y el segundo, siendo un error absoluto muy reducido, en relación al valor verdadero medido, corresponde a un error relativo alto comparado al resto de muestras.

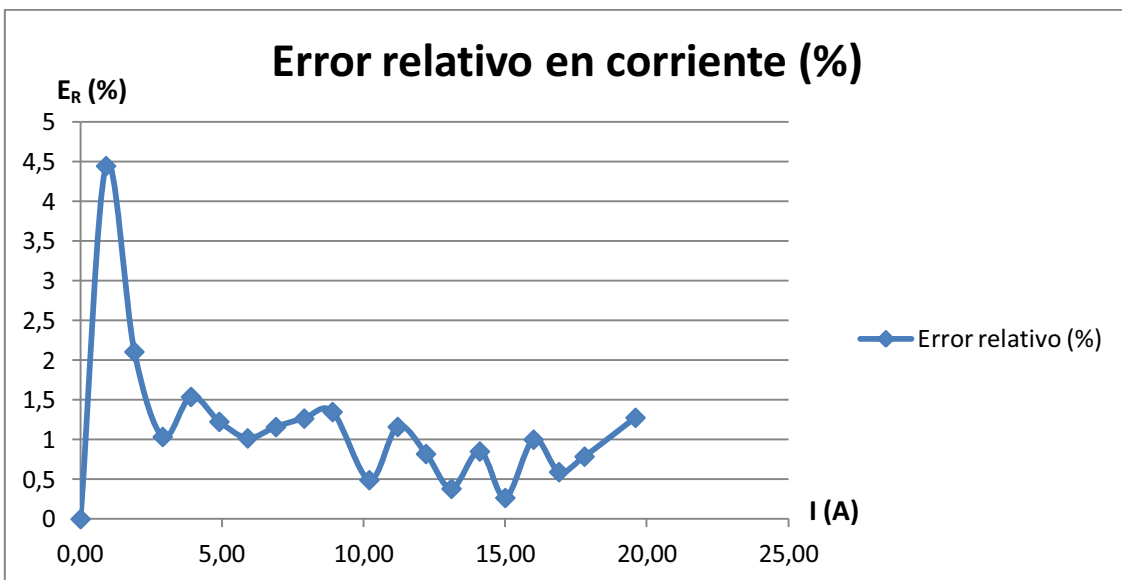


Figura 159: Gráfico de error relativo en corriente

Según los datos obtenidos, el error máximo absoluto es de 0,25A, y el máximo relativo es de 4,44%; aunque en el resto del rango de medida se reduce tanto que el pico máximo queda en 1,54%. Hay que tener en cuenta que la pinza con la que se mide el valor asignado como verdadero, tiene un error relativo en medida de corriente del 2%.

11.3. Error de medida de factor de potencia

Aunque el dispositivo es capaz de medir el desfase de 0° a 360° , correspondiente a los 4 cuadrantes; en esta prueba solo se realiza el barrido desde 0° hasta 90° , equivalentes al cuadrante de potencia activa consumida, y potencia reactiva generada.

Para generar los diferentes ángulos de desfase se utilizan cargas resistivas y capacitivas. Como carga resistiva se emplea el banco de resistencias, mientras que como cargas capacitivas se emplean condensadores de uso en corriente alterna con capacidades de 20mF y $7\mu\text{F}$, como los de las figuras 160 y 161.

Con todos los componentes conectados en paralelo y alimentados por el autotransformador, mediante combinaciones de resistencias activadas, se varía el ángulo de desfase entre tensión y corriente para la toma de muestras.



Figura 160: Condensador de 20mF



Figura 161: Condensador de 7μF

Para medir el factor de potencia con la pinza PCE, se conecta el voltímetro en paralelo a la salida del transformador al igual que el dispositivo; y la pinza PCE se conecta a la misma fase principal de alimentación de todos los componentes, al igual que la pinza del dispositivo. De esta manera se mide la misma fase de tensión y la misma fase de corriente.

Los datos obtenidos en cada muestra se exportan a una tabla de Excel para calcular el error absoluto y el error relativo de cada muestra; y se elabora un gráfico con cada error frente al valor verdadero medido por el polímetro. Ver resultados obtenidos en tabla 9 del anexo A.4.3.

Como se puede observar en los gráficos de las figura 162 y 163, la medida del factor de potencia es la medida menos precisa de las tres realizadas. Esto puede ser debido a las limitaciones del CAD de Arduino en cuanto a la frecuencia de muestreo y cuantificación de la señal, y a la propia velocidad del procesador de Arduino.

La lectura positiva de los errores calculados es que en condiciones normales de consumo de una vivienda, el factor de potencia puede estar entre 0,85 y 1, lo que equivale a un ángulo de desfase entre 0° y 32° . Este es el rango de medida de mayor precisión y con el que menos desviación se produce al realizar cálculos de potencia y energía.

En cambio desde los 70° hasta los 90°, el error obtenido se incrementa exageradamente. Es un rango de medida fuera de lo común para el consumo eléctrico de una vivienda, puesto que ningún electrodoméstico es una carga capacitiva; por lo tanto, esta imprecisión no resulta un inconveniente en el uso cotidiano del dispositivo.

Según el gráfico de error absoluto de la figura 162, el máximo error es de 10,8°, aunque en la zona normal de trabajo se sitúa en 4,4°.

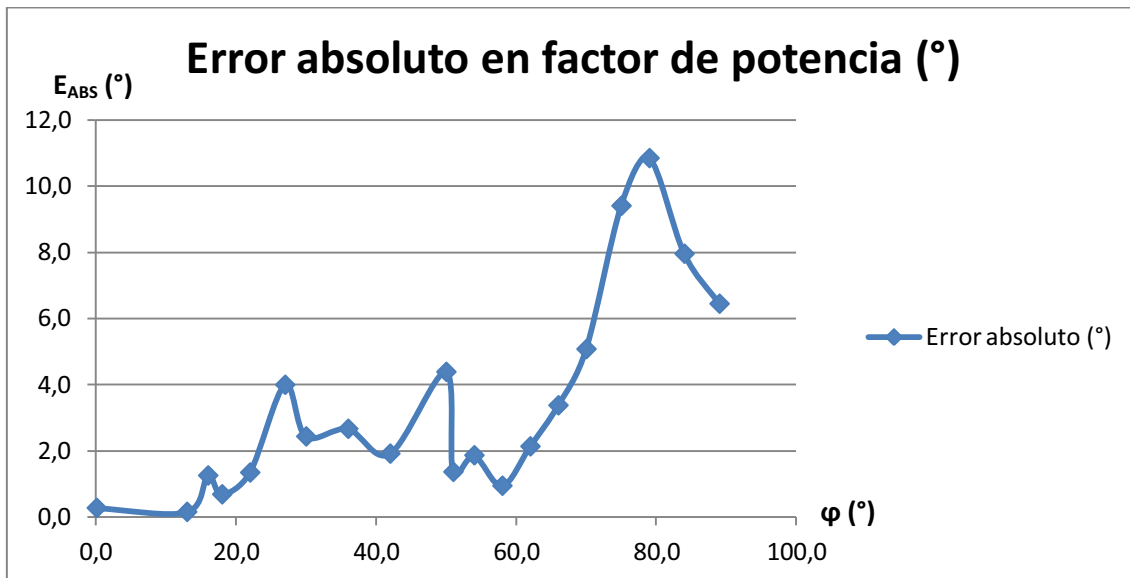


Figura 162: Gráfico de error absoluto en factor de potencia

En cuanto al máximo error relativo (ver figura 163) alcanza un valor de 14,78%, dentro de lo que se puede llamar la zona normal de trabajo. Aunque parezca un valor muy alto, en el mismo punto corresponde a un error absoluto de 4°.

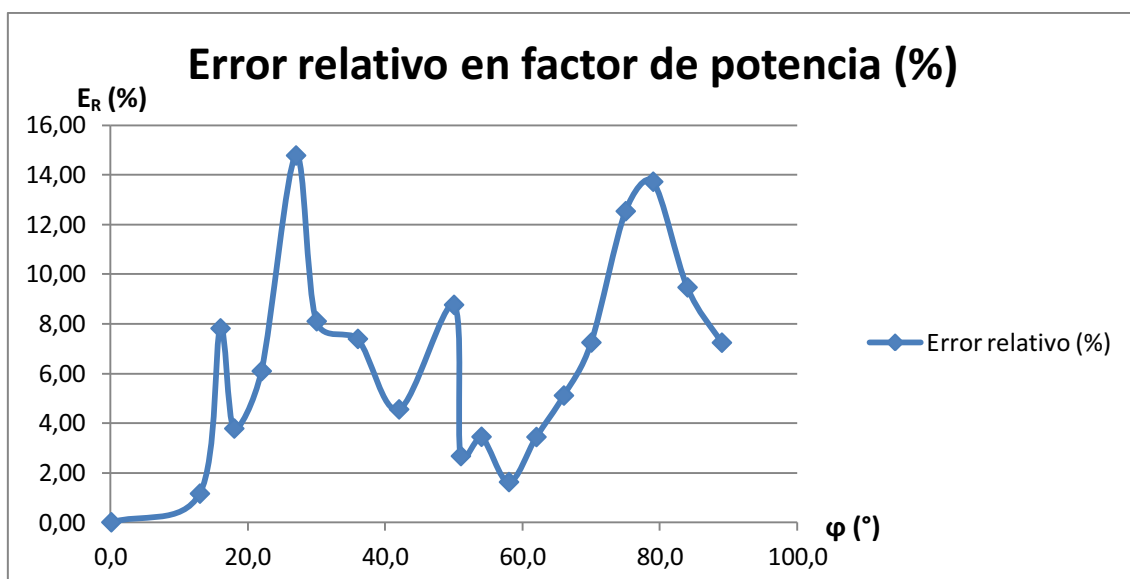


Figura 163: Gráfico de error relativo en factor de potencia

12. Conclusiones

A continuación se expondrán los objetivos logrados en cuanto a diseño y construcción durante la realización de este proyecto.

- Se ha realizado el diseño del hardware de un analizador de redes con el análisis de sus componentes y la interacción entre ellos.
- El diseño del esquema eléctrico de conexiones ha sido diseñado y desarrollado mediante el software Design Spark, para su fabricación en PCB.
- Se ha implementado el diseño en una PCB y se han hecho las labores de montaje, soldadura y comprobación de las mismas.
- Una vez completado el *hardware*, se ha configurado para la implementación del *firmware*.
- La programación del código en Arduino, permite la monitorización en la pantalla local y la exportación de datos a la nube. Además del registro en una memoria externa de picos máximos de corriente y tensión, y totalizadores de energía consumida y generada.
- Entre los objetivos alcanzados se encuentran la obtención de valores de medida directa (tensión, corriente, factor de potencia); y valores calculados a partir de las magnitudes medidas y el tiempo (potencia y energía).
- La configuración como servidor web ha sido implementada con éxito permitiendo la monitorización de datos en tiempo real desde un Smartphone o PC conectado a Internet.

En la tabla 4 se realiza una comparación entre el dispositivo construido y los dos productos de mercado analizados en el apartado de “Estado del arte” (capítulo 1.2.)

| | <i>Mirubee Mirubox V.2.</i> | <i>Efergy enqage solar kit</i> | <i>Analizador TFG</i> |
|---|--|---|------------------------------|
| <i>Medida de tensión</i> | SI | SI | SI |
| <i>Medida de corriente (Rango)</i> | SI (100A) | SI (95A) | SI (25A) |
| <i>Error de medida de tensión</i> | 1% | --% | 0,53% |
| <i>Error de medida de corriente</i> | 1% | --% | 1,5% |
| <i>Medida de consumo energético</i> | SI | SI | SI |
| <i>Medida de generación energética</i> | NO | SI | SI |
| <i>Conexión WIFI</i> | SI | SI | SI |
| <i>Exportación de datos a la nube</i> | SI | SI | SI |
| <i>Pantalla local</i> | NO | NO | SI |
| <i>Alimentación</i> | TENSION DE RED | PILAS | TENSION DE RED |
| <i>Precio</i> | 109,95 Euros | 129,99 Euros | 132,52 Euros |

Tabla 4: Comparación con productos del mercado

A la vista de la comparación en la tabla se obtienen las siguientes conclusiones finales:

- El prototipo diseñado dispone de al menos, las mismas funciones que el producto de mercado más completo analizado en este estudio.
- La implementación de la pantalla local para monitorizar los datos es una característica que lo diferencia del resto de productos.
- A pesar de que el rango de medida de corriente del prototipo es muy inferior a los otros dos productos; no resulta una característica importante teniendo en cuenta que el producto está orientado al uso particular de vivienda con un consumo eléctrico medio. El rango de medida de corriente impuesto en el prototipo (25A), es más que suficiente para medir los valores de corriente del consumo eléctrico medio de una vivienda.
- El grado de precisión en la medida del prototipo es similar a uno de los productos analizados; ya que otro de ellos ni si quiera lo indica en su ficha técnica. De cualquier manera, el producto que indica un error del 1%; tampoco especifica en su ficha técnica si se trata de un máximo como se ha supuesto en esta comparación, o se trata de un valor medio.
- Incluso construyendo el prototipo en hardware libre, el precio resulta algo más caro comparado al producto más completo de los analizados. Aproximadamente, la mitad del presupuesto se debe al precio del Arduino YUN, que fue elegido por sus funciones como servidor web. En caso de prescindir de la función de exportar los datos a la nube, el precio final del prototipo se reduciría a la mitad.
- Otro componente que encarece considerablemente el dispositivo es la fuente de alimentación. La necesidad de alimentación negativa de los amplificadores operacionales obliga a instalar una fuente de modo común conmutada. Con otra tipología en los circuitos de acondicionamiento en la que sea innecesaria la alimentación negativa, se podría prescindir de esta fuente.

13. Líneas futuras

Una vez terminado el proyecto, desde el punto de vista de los objetivos principales, se plantean una serie de mejoras y ampliaciones como líneas a seguir en el futuro, partiendo del mismo prototipo o desde una nueva versión.

13.1. Ampliación de funciones

El proyecto tiene un gran margen de ampliación de funciones debido a que hasta ahora solo se utilizan 3 entradas analógicas de las 6 disponibles; y 10 pines digitales de los 13 disponibles.

- **Compensación del factor de potencia:** Con una batería de condensadores por escalones se podría corregir el factor de potencia. En función de la necesidad de potencia reactiva capacitiva, una subrutina puede controlar las alimentaciones de los escalones de la batería mediante relés, habilitados por las salidas digitales de Arduino.
- **Implementación de alarma de intrusión:** Como el dispositivo ya está configurado para exportar datos a la nube, solo es necesario un mecanismo para detectar el estado de la puerta principal de la vivienda (abierta o cerrada). Con un final de carrera o un sensor de posición conectado a una entrada digital se puede detectar la intrusión, y enviar un aviso a la nube.
- **Implementación de alarma de incendios:** Al igual que en el caso anterior, con un sensor de CO₂, conectado a una de las entradas analógicas disponibles; se puede detectar un incendio en el hogar y enviar el aviso a la nube.
- **Programación de una App para Smartphone:** Una aplicación para Smartphone aportaría comodidad al usuario del dispositivo en las operaciones de recibir y enviar información. Con la aplicación se puede visualizar los datos de una forma más sofisticada, y evitar el acceso a los mismos mediante un navegador web.

13.2. Mejoras técnicas

El apartado de mejoras del prototipo se centra principalmente en los objetivos de aumentar la precisión en las medidas, y economizar el producto sin que ello suponga una reducción de sus funciones.

- **Construcción de la PCB con componentes SMD:** La sustitución de los componentes de montaje de orificio pasante por los de montaje superficial, reduciría el tamaño de la PCB. Incluso se podría desarrollar una placa de tipo shield para montar en Arduino, con las mismas dimensiones de planta. Al reducir la superficie de la PCB, también se reduce el coste y el tiempo de fabricación de la misma.
- **Aumento de precisión en la medida de señales:** Una forma de aumentar la precisión en las medidas de las señales analógicas de tensión y corriente es aprovechando todo el rango de entrada del CAD de Arduino. En este diseño la amplitud máxima de las señales de entrada al CAD es de 4Vpp, debido a que los amplificadores operacionales alimentados a +5V y -5V, en saturación no consiguen mantener estos niveles de tensión en salida. Una solución es sustituir la fuente de alimentación actual por una capaz de abastecer unas tensiones de 6V o 7V aproximadamente, con los que alimentar los amplificadores operacionales de los circuitos de acondicionamiento. De esta forma se conseguiría mantener las salidas en 5V y aprovechar por completo el rango de entradas analógicas del CAD.
- **Sustituir fuente de alimentación:** El componente más caro de la PCB es la fuente de alimentación. Una forma de reducir el presupuesto del prototipo es sustituir la fuente de alimentación por otra que no suministre tensiones negativas, que resulte más económica. Para ello es necesario diseñar los circuitos de acondicionamiento con otra tipología en la que no sea necesario alimentar los amplificadores operacionales con tensiones negativas.

14. Referencias bibliográficas

- [1] *"Conceptos de eficiencia energética"*,
Disponible en:
http://www.efenergia.com/conceptos-eficiencia-energetica/#Eficiencia_Energetica
- [2] *"El ahorro y la eficiencia energética. No, no son lo mismo."*,
Disponible en:
<http://www.escueladeenergiaverde.com>
- [3] *"El autoconsumo fotovoltaico inyectado a la red"*,
Disponible en:
<http://www.sitiosolar.com>
- [4] *"Hardware libre"*,
Disponible en:
http://www.ecured.cu/Hardware_libre
- [5] *"¿Qué es Arduino?"*,
Disponible en:
<http://www.arduinohtics.weebly.com>
- [6] *"Arduino YUN: características y capacidades"*, por García González, Anthony,
Disponible en:
<http://www.panamahitek.com>
- [7] *"¿Qué es el IDE de Arduino?"*,
Disponible en:
<http://www.aprendiendoarduino.wordpress.com>
- [8] *"Qué es un transformador, tipos y funcionamiento"*
Disponible en:
<http://www.piensa3d.com>

ANEXOS

A.1. Cálculo de potencia total

Se realiza el cálculo de potencia por circuitos individuales, y con el sumatorio total de las mismas, justificar la elección de la fuente de alimentación en base a su potencia nominal.

- **Divisor de tensión de corriente continua** (referido al esquema de la figura 50)

$$P_{DT} = P_{R3} + P_{P0} + P_{R4} = \frac{V^2}{R_{TOTAL}} = \frac{5V^2}{250K\Omega} = 0,1mW$$

- **Bloques sumadores de circuitos de acondicionamiento** (referido al esquema de la figura 50)

$$P_{SUMADOR} = P_{R5} + P_{R6} + P_{R7} + P_{R8} + P_{R10} + P_{R11} + P_{R12} + P_{R13}$$

$$P_{R5} = P_{R6} = P_{R10} = P_{R11} = \frac{V_{MAX}^2}{R} = \frac{5V^2}{10K\Omega} = 2,5mW$$

$$P_{R7} = P_{R8} = P_{R12} = P_{R13} = \frac{V_{MAX}^2}{R} = \frac{5V^2}{4K7\Omega} = 5,32mW$$

$$P_{MAX\ SUMADOR} = (2,5mW * 4) + (5,32mW * 4) = 31,28mW$$

- **Filtros paso bajo** (referido al esquema de la figura 50)

$$P_{FILTROS} = P_{R16} + P_{R17} = \frac{V_{MAX}^2}{R} * 2 = \frac{5V^2}{27K\Omega} * 2 = 1,85mW$$

- **Circuitos de LEDs** (referido al esquema de la figura 42)

$$I_{LED} = \frac{5V-2V}{150\Omega} = 20mA \quad V_{LED} = 2V$$

$$P_{LEDS} = P_{R18} + P_{R19} + P_{LED1} + P_{LED2} = (R_{18} * I_{LED}^2) + (R_{19} * I_{LED}^2) + (V_{LED} * I_{LED} * 2)$$

$$P_{LEDS} = (150\Omega * 20mA^2) + (150\Omega * 20mA^2) + (2V * 20mA * 2) = 200mW$$

- **Circuito de pulsador de encendido y cambio de pantalla** (referido al esquema de la figura 40)

$$P_{R20} = \frac{V^2}{R_{20}} = \frac{5V^2}{1K\Omega} = 25mW$$

- **Consumo de los amplificadores operacionales** (ver anexo datasheet)

$$V_{CC}^+ = 5V \quad V_{CC}^- = -5V \quad I_{MAX\ AO} = 2,8mA$$

$$P_{AO} = (V_{CC}^+ - V_{CC}^-) * I_{MAX} * 8 = [5V - (-5V)] * 2,8mA * 8 = 224mW$$

- **Consumo del módulo RTC**

$$P_{RTC} = I_{RTC} * V = 0,3mA * 5V = 1,5mW$$

- **Consumo del módulo de pantalla LCD**

$$P_{LCD} = I_{LCD} * V = 100mA * 5V = 500mW$$

- **Consumo del Arduino YUN**

$$P_{YUN} = I_{YUN} * V = 200mA * 5V = 1W$$

- **Sumatorio de potencias**

$$P_{TOTAL} = P_{DT} + P_{MAX\ SUMADOR} + P_{FILTROS} + P_{LEDS} + P_{R20} + P_{AO} + P_{RTC} + P_{LCD} + P_{YUN}$$

$$P_{TOTAL} = (0,1 + 31,28 + 1,85 + 200 + 25 + 224 + 1,5 + 500 + 1000)mW = 1,98W$$

A.2. Curvas de calibración sensor de corriente STC-013

En la tabla 5 se muestran los valores de tensión medidos con el osciloscopio en la salida del circuito de acondicionamiento de señal de corriente. Cada valor de muestra obtenido está relacionado con un valor de corriente medido en la entrada del circuito. Con los datos experimentales se construye el gráfico de la figura 124.

| <i>Curva de calibración STC-013 (sensor de corriente) (V-I)</i> | | | | |
|---|-------|----------|---------|--------|
| Nº muestra | I (A) | Vpp (mv) | Vp (mv) | Vp (v) |
| 1 | 0,0 | 120 | 60 | 0,06 |
| 2 | 0,5 | 180 | 90 | 0,09 |
| 3 | 1,0 | 240 | 120 | 0,12 |
| 4 | 1,5 | 320 | 160 | 0,16 |
| 5 | 2,0 | 380 | 190 | 0,19 |
| 6 | 2,5 | 460 | 230 | 0,23 |
| 7 | 3,0 | 520 | 260 | 0,26 |
| 8 | 3,5 | 600 | 300 | 0,30 |
| 9 | 4,0 | 680 | 340 | 0,34 |
| 10 | 4,5 | 760 | 380 | 0,38 |
| 11 | 5,0 | 840 | 420 | 0,42 |
| 12 | 5,5 | 920 | 460 | 0,46 |
| 13 | 6,0 | 1000 | 500 | 0,50 |
| 14 | 6,5 | 1060 | 530 | 0,53 |
| 15 | 7,0 | 1140 | 570 | 0,57 |
| 16 | 7,5 | 1200 | 600 | 0,60 |
| 17 | 8,0 | 1280 | 640 | 0,64 |
| 18 | 8,5 | 1380 | 690 | 0,69 |
| 19 | 9,0 | 1440 | 720 | 0,72 |
| 20 | 9,5 | 1520 | 760 | 0,76 |
| 21 | 10,0 | 1600 | 800 | 0,80 |
| 22 | 10,5 | 1680 | 840 | 0,84 |
| 23 | 11,0 | 1760 | 880 | 0,88 |
| 24 | 11,5 | 1840 | 920 | 0,92 |
| 25 | 12,0 | 1920 | 960 | 0,96 |
| 26 | 12,5 | 2000 | 1000 | 1,00 |
| 27 | 13,0 | 2080 | 1040 | 1,04 |
| 28 | 13,5 | 2160 | 1080 | 1,08 |
| 29 | 14,0 | 2240 | 1120 | 1,12 |
| 30 | 14,5 | 2320 | 1160 | 1,16 |
| 31 | 15,0 | 2400 | 1200 | 1,20 |
| 32 | 15,5 | 2480 | 1240 | 1,24 |
| 33 | 16,0 | 2560 | 1280 | 1,28 |
| 34 | 16,5 | 2640 | 1320 | 1,32 |
| 35 | 17,0 | 2720 | 1360 | 1,36 |
| 36 | 17,5 | 2800 | 1400 | 1,40 |
| 37 | 18,0 | 2880 | 1440 | 1,44 |
| 38 | 18,5 | 2960 | 1480 | 1,48 |
| 39 | 19,0 | 3040 | 1520 | 1,52 |
| 40 | 19,5 | 3120 | 1560 | 1,56 |
| 41 | 20,0 | 3200 | 1600 | 1,60 |

Tabla 5: Curva de calibración STC-013 (V-I)

A.3. Curva de calibración de factor de potencia

En la tabla se muestran los valores de ángulo de desfase en grados, de la señal de corriente respecto a la de tensión. A medida que aumenta la corriente, aumenta el ángulo de desfase.

Estos datos son obtenidos por el cálculo que realiza el osciloscopio al muestrear por canales separados las señales de tensión y corriente, a la salida del circuito de acondicionamiento, es decir, a la entrada del convertidor analógico digital del Arduino.

| Curva de calibración STC-013 (I- φ) | | |
|--|-------|---------------|
| Nº muestra | I (A) | φ (°) |
| 1 | 2,5 | 4 |
| 2 | 5,1 | 4,7 |
| 3 | 7,7 | 5,6 |
| 4 | 10,1 | 6,3 |
| 5 | 12,6 | 7,2 |
| 6 | 15,2 | 8,1 |
| 7 | 17,5 | 9 |

Tabla 6: Curva de calibración STC-013 (I- φ)

A.4. Medida de error absoluto y relativo

A.4.1. Medida de error de tensión

| Medida de error de tensión | | | | |
|-----------------------------------|----------------------------|-------------------------|---------------------------|---------------------------|
| Nº muestra | Valor verdadero (V) | Valor medido (V) | Error absoluto (V) | Error relativo (%) |
| 1 | 220,0 | 220,1 | 0,1 | 0,05 |
| 2 | 221,7 | 221,6 | 0,1 | 0,05 |
| 3 | 223,1 | 222,9 | 0,2 | 0,09 |
| 4 | 224,4 | 224,9 | 0,5 | 0,22 |
| 5 | 225,2 | 226,4 | 1,2 | 0,53 |
| 6 | 226,6 | 227,6 | 1,0 | 0,44 |
| 7 | 227,1 | 228,2 | 1,1 | 0,48 |
| 8 | 228,6 | 229,4 | 0,8 | 0,35 |
| 9 | 229,3 | 230,1 | 0,8 | 0,35 |
| 10 | 231,0 | 231,6 | 0,6 | 0,26 |
| 11 | 231,7 | 232,7 | 1,0 | 0,43 |
| 12 | 232,5 | 233,2 | 0,7 | 0,30 |
| 13 | 233,1 | 234,3 | 1,2 | 0,51 |
| 14 | 234,3 | 234,7 | 0,4 | 0,17 |
| 15 | 235,4 | 236,2 | 0,8 | 0,34 |
| 16 | 236,4 | 237,3 | 0,9 | 0,38 |
| 17 | 237,7 | 237,5 | 0,2 | 0,08 |
| 18 | 238,2 | 238,7 | 0,5 | 0,21 |
| 19 | 239,1 | 239,4 | 0,3 | 0,13 |
| 20 | 240,2 | 240,8 | 0,6 | 0,25 |

Tabla 7: Medida de error de tensión

A.4.2. Medida de error de corriente

| Medida de error de corriente | | | | |
|-------------------------------------|----------------------------|-------------------------|---------------------------|---------------------------|
| Nº muestra | Valor verdadero (A) | Valor medido (A) | Error absoluto (A) | Error relativo (%) |
| 1 | 0,00 | 0,03 | 0,03 | -- |
| 2 | 0,90 | 0,86 | 0,04 | 4,44 |
| 3 | 1,90 | 1,86 | 0,04 | 2,11 |
| 4 | 2,90 | 2,87 | 0,03 | 1,03 |
| 5 | 3,90 | 3,84 | 0,06 | 1,54 |
| 6 | 4,90 | 4,84 | 0,06 | 1,22 |
| 7 | 5,90 | 5,84 | 0,06 | 1,02 |
| 8 | 6,90 | 6,82 | 0,08 | 1,16 |
| 9 | 7,90 | 7,80 | 0,10 | 1,27 |
| 10 | 8,90 | 8,78 | 0,12 | 1,35 |
| 11 | 10,20 | 10,15 | 0,05 | 0,49 |
| 12 | 11,20 | 11,07 | 0,13 | 1,16 |
| 13 | 12,20 | 12,10 | 0,10 | 0,82 |
| 14 | 13,10 | 13,05 | 0,05 | 0,38 |
| 15 | 14,10 | 13,98 | 0,12 | 0,85 |
| 16 | 15,00 | 14,96 | 0,04 | 0,27 |
| 17 | 16,00 | 15,84 | 0,16 | 1,00 |
| 18 | 16,90 | 16,80 | 0,10 | 0,59 |
| 19 | 17,80 | 17,66 | 0,14 | 0,79 |
| 20 | 19,60 | 19,35 | 0,25 | 1,28 |

Tabla 8: Medida de error de corriente

A.4.3. Medida de error de factor de potencia

| Medida de error de factor de potencia | | | | |
|--|----------------------------|-------------------------|---------------------------|---------------------------|
| Nº muestra | Valor verdadero (°) | Valor medido (°) | Error absoluto (°) | Error relativo (%) |
| 1 | 89,0 | 95,44 | 6,44 | 7,24 |
| 2 | 84,0 | 76,05 | 7,95 | 9,46 |
| 3 | 79,0 | 68,16 | 10,84 | 13,72 |
| 4 | 75,0 | 65,60 | 9,40 | 12,53 |
| 5 | 70,0 | 64,93 | 5,07 | 7,24 |
| 6 | 66,0 | 62,63 | 3,37 | 5,11 |
| 7 | 62,0 | 59,87 | 2,13 | 3,44 |
| 8 | 58,0 | 58,94 | 0,94 | 1,62 |
| 9 | 54,0 | 55,86 | 1,86 | 3,44 |
| 10 | 51,0 | 52,36 | 1,36 | 2,67 |
| 11 | 50,0 | 54,38 | 4,38 | 8,76 |
| 12 | 42,0 | 43,91 | 1,91 | 4,55 |
| 13 | 36,0 | 38,66 | 2,66 | 7,39 |
| 14 | 30,0 | 32,43 | 2,43 | 8,10 |
| 15 | 27,0 | 30,99 | 3,99 | 14,78 |
| 16 | 22,0 | 23,34 | 1,34 | 6,09 |
| 17 | 18,0 | 18,68 | 0,68 | 3,78 |
| 18 | 16,0 | 17,25 | 1,25 | 7,81 |
| 19 | 13,0 | 12,85 | 0,15 | 1,15 |
| 20 | 0,1 | -0,17 | 0,27 | -- |

Tabla 9: Medida de error de factor de potencia

A.5. BOM LIST (presupuesto)

| BOM LIST | | | | | |
|--------------------------|-----------------------------|------------------|-----------------|----------------------|---------------------|
| Orden | Denominación | Código RS | Unidades | Precio/unidad | Precio total |
| 1 | Resistencia 1/8w 1% 100Ω | 892-8666 | 1 | 0,013 | 0,013 |
| 2 | Resistencia 1/8w 1% 150Ω | 845-8078 | 2 | 0,075 | 0,150 |
| 3 | Resistencia 1/8w 1% 1KΩ | 830-6757 | 1 | 0,720 | 0,720 |
| 4 | Resistencia 1/8w 1% 4K7Ω | 845-7959 | 4 | 0,095 | 0,380 |
| 5 | Resistencia 1/8w 1% 10KΩ | 830-7453 | 4 | 0,092 | 0,368 |
| 6 | Resistencia 1/8w 1% 27KΩ | 845-8268 | 2 | 0,075 | 0,150 |
| 7 | Resistencia 1/8w 1% 50KΩ | 893-2719 | 1 | 0,013 | 0,013 |
| 8 | Resistencia 1/8w 1% 100KΩ | 830-7652 | 2 | 0,072 | 0,144 |
| 9 | Resistencia 1/8w 1% 330KΩ | 851-4734 | 1 | 0,068 | 0,068 |
| 10 | Potenciómetro 100Ω | 473-499 | 1 | 0,392 | 0,392 |
| 11 | Potenciómetro 50KΩ | 896-7178 | 2 | 0,604 | 1,208 |
| 12 | Condensador poliéster 56nF | 115-269 | 2 | 0,368 | 0,736 |
| 13 | Zócalo 14 pines | 813-121 | 2 | 0,520 | 1,040 |
| 14 | LED 5mm | 228-6004 | 2 | 0,200 | 0,400 |
| 15 | Pulsador PCB | -- | 1 | 0,290 | 0,290 |
| 16 | Pulsador panel | 111-5808 | 1 | 4,560 | 4,560 |
| 17 | Fusible 5x20mm 0.6A | 878-3680 | 2 | 0,441 | 0,882 |
| 18 | Fusible 5x20mm 0.1A | 610-9850 | 1 | 0,312 | 0,312 |
| 19 | Portafusibles 5x20mm PCB | 176-9047 | 3 | 0,443 | 1,329 |
| 20 | Transformador 1.2VA 230/6V | 732-0531 | 1 | 3,690 | 3,690 |
| 21 | RECOM 6w | 166-6716 | 1 | 21,322 | 21,322 |
| 22 | Conector macho 2.54mm | 113-7911 | 3 | 0,296 | 0,888 |
| 23 | Conector hembra 2.54mm | 674-1536 | 3 | 0,502 | 1,506 |
| 24 | Clema 2 terminales | 193-0586 | 2 | 0,584 | 1,168 |
| 25 | Sensor de corriente STC-013 | -- | 1 | 6,220 | 6,220 |
| 26 | C.I. TL084 TEXAS | 121-8679 | 2 | 0,334 | 0,668 |
| 27 | Módulo RTC DS3231 QUIMAT | -- | 1 | 7,090 | 7,090 |
| 28 | Pantalla LCD 20x4 QUIMAT | -- | 1 | 10,990 | 10,990 |
| 29 | Arduino YUN | -- | 1 | 60,000 | 60,000 |
| 30 | Tarjeta SD 8GB clase 10 | -- | 1 | 5,820 | 5,820 |
| 31 | Tornillería nylon | -- | -- | 0,000 | 0,000 |
| 32 | Cable 0.5mm2 | -- | -- | 0,000 | 0,000 |
| 33 | Fundas termo retractiles | -- | -- | 0,000 | 0,000 |
| Presupuesto final | | | | | 132,517 |

Tabla 10: BOM LIST (presupuesto)

A.6. Códigos de programación en IDE Arduino

A.6.1. *Sketch* "I2C_SCANNER"

Este *sketch* sirve para identificar la dirección de un periférico conectado al Puerto I2C. Tras cargar el *sketch*, abrir la pantalla de *Monitor Serie* para ver la información recibida.

```
1 // -----
2 // i2c_scanner
3 //
4 // Version 1
5 //   This program (or code that looks like it)
6 //   can be found in many places.
7 //   For example on the Arduino.cc forum.
8 //   The original author is not know.
9 // Version 2, Juni 2012, Using Arduino 1.0.1
10 //   Adapted to be as simple as possible by Arduino.cc user Krodal
11 // Version 3, Feb 26 2013
12 //   V3 by louarnold
13 // Version 4, March 3, 2013, Using Arduino 1.0.3
14 //   by Arduino.cc user Krodal.
15 //   Changes by louarnold removed.
16 //   Scanning addresses changed from 0...127 to 1...119,
17 //   according to the i2c scanner by Nick Gammon
18 //   http://www.gammon.com.au/forum/?id=10896
19 // Version 5, March 28, 2013
20 //   As version 4, but address scans now to 127.
21 //   A sensor seems to use address 120.
22 // Version 6, November 27, 2015.
23 //   Added waiting for the Leonardo serial communication.
24 //
25 //
26 // This sketch tests the standard 7-bit addresses
27 // Devices with higher bit address might not be seen properly.
28 //
29
30 #include <Wire.h>
31
32
33 void setup()
34 {
35   Wire.begin();
36
37   Serial.begin(9600);
38   while (!Serial);           // Leonardo: wait for serial monitor
39   Serial.println("\nI2C Scanner");
40 }
41
```

```
42
43 void loop()
44 {
45     byte error, address;
46     int nDevices;
47
48     Serial.println("Scanning...");
49
50     nDevices = 0;
51     for(address = 1; address < 127; address++ )
52     {
53         // The i2c_scanner uses the return value of
54         // the Write.endTransmission to see if
55         // a device did acknowledge to the address.
56         Wire.beginTransmission(address);
57         error = Wire.endTransmission();
58
59         if (error == 0)
60         {
61             Serial.print("I2C device found at address 0x");
62             if (address<16)
63                 Serial.print("0");
64             Serial.print(address,HEX);
65             Serial.println(" !");
66
67             nDevices++;
68         }
69         else if (error==4)
70         {
71             Serial.print("Unknown error at address 0x");
72             if (address<16)
73                 Serial.print("0");
74             Serial.println(address,HEX);
75         }
76     }
77     if (nDevices == 0)
78         Serial.println("No I2C devices found\n");
79     else
80         Serial.println("done\n");
81
82     delay(5000);           // wait 5 seconds for next scan
83 }
```

A.6.2. Sketch “Puesta_en_hora_3231RTC”

El siguiente *sketch* es un ejemplo de la librería *RTC* instalada, en el que se ha modificado la hora actual para la puesta en hora del reloj. Se pueden dejar comentadas las líneas de ajuste de fecha y hora, situadas en el *Void Setup*, y comprobar con el *Monitor Serie* el paso de los segundos.

```
1 // Programa de puesta en hora del modulo RTC
2
3 #include <Wire.h>
4 #include <RTC.h>
5
6 RTC rtc(DST_ON); // OR DST_OFF
7
8 void setup()
9 {
10     Serial.begin(9600);
11     while(!Serial);
12     delay(1000);
13
14 // Ajustar los valores de fecha en este parrafo
15     rtc.setYear(2018);
16     rtc.setMonth(3);
17     rtc.setDay(19);
18     rtc.setDate( 2018, 3, 19 );
19
20 // Ajustar los valores de hora en este parrafo
21     rtc.setHour(16);
22     rtc.setMinutes(5);
23     rtc.setSeconds(0);
24     rtc.setTime( 16, 05, 00 );
25
26     rtc.setDateTime( 2018, 3, 19, 16, 05, 00 ); // Año, mes, dia, hora, minuto, segundo
27 }
28
29 void loop()
30 {
31     Data Time = rtc.getData(); //Recoge la informacion en una variable
32     Serial.println( Time.toString() ); //y la muestra en el monitor serie cada segundo
33     delay(1000);
34 }
```

A.6.3. Sketch "LCD"

El *sketch* es un ejemplo de la librería instalada *LiquidCrystal_I2C*, en el que se ha modificado la dirección del periférico (0x3F), y el número de columnas y filas (20 y 4), en la línea 7. El objetivo del *sketch* es que sirva como prueba de funcionamiento de la pantalla LCD.

```
1 //YWRBOT
2 //Compatible with the Arduino IDE 1.0
3 //Library version:1.1
4 #include <Wire.h>
5 #include <LiquidCrystal_I2C.h>
6
7 LiquidCrystal_I2C lcd(0x3F,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line display
8
9 void setup()
10 {
11   lcd.init(); // initialize the lcd
12   lcd.init();
13   // Print a message to the LCD.
14   lcd.backlight();
15   lcd.setCursor(3,0);
16   lcd.print("Hello, world!");
17   lcd.setCursor(2,1);
18   lcd.print("Ywrobot Arduino!");
19   lcd.setCursor(0,2);
20   lcd.print("Arduino LCM IIC 2004");
21   lcd.setCursor(2,3);
22   lcd.print("Power By Ec-yuan!");
23 }
24
25
26 void loop()
27 {
28 }
```

A.6.4. Sketch “YunDiskSpaceExpander”

El siguiente sketch ha sido descargado de la página oficial de Arduino. Una vez cargado, se abre la pantalla del Monitor serie para seguir las instrucciones del programa en el proceso de asignación de espacio de las particiones y formato de las mismas.

```
1 /*
2   Yún Disk Space Expander
3
4   Requirements:
5   * micro SD card
6   * internet connection
7
8   This sketch configure the SD card to expand the disk space
9   of the Yún. Upload, open the Serial Monitor and follow
10  the interactive procedure.
11
12  Warning: your SD card will be formatted and you will lose
13  the files it contains. Be sure you have backed it up before
14  using it for expanding Yún's disk space.
15
16  created Apr 2014
17  by Federico Fissore & Federico Vanzati
18
19  This code is in the public domain.
20
21  http://arduino.cc/en/Tutorial/ExtendingYunDiskSpace
22 */
23
24 #include <Process.h>
25
26 #define DEBUG 0
27 #define SUCCESSFUL_EXIT_CODE 0
28
29 void setup() {
30   Serial.begin(115200);
31   while (!Serial);
32
33   Serial.print(F("This sketch will format your micro SD card and use it as
34   additional disk space for your Arduino Yun.\nPlease ensure you have ONLY
35   your micro SD card plugged in: no pen drives, hard drives or whatever.\nDo
36   you wish to proceed (yes/no)?"));
37
38   expectYesBeforeProceeding();
39
40   Serial.println(F("\nStarting Bridge..."));
41
42   Bridge.begin();
43 }
```

```
44  haltIfSDAlreadyOnOverlay();
45
46  haltIfInternalFlashIsFull();
47
48  haltIfSDCardIsNotPresent();
49
50  installSoftware();
51
52  partitionAndFormatSDCard();
53
54  createArduinoFolder();
55
56  copySystemFilesFromYunToSD();
57
58  enableExtRoot();
59
60  Serial.print(F("\nWe are done! Yeah! Now press the YUN RST button to apply the changes."));
61 }
62
63 void loop() {
64     // This turns the sketch into a YunSerialTerminal
65     if (Serial.available()) {
66         char c = (char)Serial.read();
67         Serial1.write(c);
68     }
69     if (Serial1.available()) {
70         char c = (char)Serial1.read();
71         Serial.write(c);
72     }
73 }
74
75 void halt() {
76     Serial.flush();
77     while (true);
78 }
79
80 void expectYesBeforeProceeding() {
81     Serial.flush();
82     while (!Serial.available());
83     String answer = Serial.readStringUntil('\n');
84     Serial.print(F(" "));
85     Serial.println(answer);
86     if (answer != "yes") {
87         Serial.println(F("\nGoodbye"));
88         halt();
89     }
90 }
91
```

```
92 int readPartitionSize() {
93     int partitionSize = 0;
94     while (!partitionSize)
95     {
96         Serial.print(F("Enter the size of the data partition in MB: "));
97         while (Serial.available() == 0);
98
99         String answer = Serial.readStringUntil('\n');
100        partitionSize = answer.toInt();
101        Serial.println(partitionSize);
102        if (!partitionSize)
103            Serial.println(F("Invalid input, retry"));
104    }
105    return partitionSize;
106 }
107
108 void debugProcess(Process p) {
109     #if DEBUG == 1
110     while (p.running());
111
112     while (p.available() > 0) {
113         char c = p.read();
114         Serial.print(c);
115     }
116     Serial.flush();
117     #endif
118 }
119
120 void haltIfSDAlreadyOnOverlay() {
121     Process grep;
122     grep.runShellCommand(F("mount | grep ^/dev/sda | grep 'on /overlay'"));
123     String output = grep.readString();
124     if (output != "") {
125         Serial.println(F("\nMicro SD card is already used as additional Arduino Yun
126 disk space.Nothing to do.));
127         halt();
128     }
129 }
130
131 void haltIfSDCardIsNotPresent() {
132     Process ls;
133     int exitCode = ls.runShellCommand("ls /mnt/sda1");
134     if (exitCode != 0) {
135         Serial.println(F("\nThe micro SD card is not available"));
136         halt();
137     }
138 }
139 }
```

```
140 void haltIfInternalFlashIsFull() {
141     Process awk;
142     awk.runShellCommand(F("df / | awk '/rootfs/ {print $4}'"));
143     int output = awk.parseInt();
144     if (output < 1000) {
145         Serial.println(F("\nYou don't have enough disk space to install the utility
146         software. You need to free at least 1MB of Flash memory.\nRetry!"));
147         halt();
148     }
149 }
150
151 void installSoftware() {
152     Serial.print(F("\nReady to install utility software. Please ensure your
153     Arduino Yun is connected to internet.\nReady to proceed (yes/no)?"));
154     expectYesBeforeProceeding();
155     Serial.println(F("Updating software list..."));
156     Process opkg;
157     // update the packages list
158     int exitCode = opkg.runShellCommand("opkg update");
159     // if the exitCode of the process is OK the package has been installed correctly
160     if (exitCode != SUCCESSFUL_EXIT_CODE) {
161         Serial.println(F("err. with opkg, check internet connection"));
162         debugProcess(opkg);
163         halt();
164     }
165     Serial.println(F("Software list updated. Installing software ..."));
166     // install the utility to format in EXT4
167     exitCode = opkg.runShellCommand(F("opkg install e2fsprogs mkdosfs fdisk rsync"));
168     if (exitCode != SUCCESSFUL_EXIT_CODE) {
169         Serial.println(F("err. installing e2fsprogs mkdosfs fdisk"));
170         debugProcess(opkg);
171         halt();
172     }
173     Serial.println(F("e2fsprogs mkdosfs fdisk rsync installed"));
174 }
175
176 void partitionAndFormatSDCard() {
177     Serial.print(F("\nProceed with partitioning micro SD card (yes/no)?"));
178     expectYesBeforeProceeding();
179
180     unmount();
181
182     Process format;
183
184     //clears partition table
185     format.runShellCommand("dd if=/dev/zero of=/dev/sda bs=4096 count=10");
186     debugProcess(format);
187
188     // create the first partition
189     int dataPartitionSize = readPartitionSize();
190
```



```
191 Serial.println(F("Partitioning (this will take a while)..."));
192 String firstPartition = "(echo n; echo p; echo 1; echo; echo +";
193 firstPartition += dataPartitionSize;
194 firstPartition += "M; echo w) | fdisk /dev/sda";
195 format.runShellCommand(firstPartition);
196 debugProcess(format);
197
198 unmount();
199
200 // create the second partition
201 format.runShellCommand(F("(echo n; echo p; echo 2; echo; echo; echo w) |
202 fdisk /dev/sda"));
203 debugProcess(format);
204
205 unmount();
206
207 // specify first partition is FAT32
208 format.runShellCommand(F("(echo t; echo 1; echo c; echo w) | fdisk /dev/sda"));
209
210 unmount();
211
212 delay(5000);
213
214 unmount();
215
216 // format the first partition to FAT32
217 int exitCode = format.runShellCommand(F("mkfs.vfat /dev/sda1"));
218 debugProcess(format);
219 if (exitCode != SUCCESSFUL_EXIT_CODE) {
220     Serial.println(F("\nerr. formatting to FAT32"));
221     halt();
222 }
223 delay(100);
224
225 // format the second partition to Linux EXT4
226 exitCode = format.runShellCommand(F("mkfs.ext4 /dev/sda2"));
227 debugProcess(format);
228 if (exitCode != SUCCESSFUL_EXIT_CODE) {
229     Serial.println(F("\nerr. formatting to EXT4"));
230     halt();
231 }
232
233 Serial.println(F("Micro SD card correctly partitioned"));
234 }
235
236 void createArduinoFolder() {
237     Serial.print(F("\nCreating 'arduino' folder structure..."));
238     Process folder;
239     folder.runShellCommand(F("mkdir -p /mnt/sda1"));
240     folder.runShellCommand(F("mount /dev/sda1 /mnt/sda1"));
241     folder.runShellCommand(F("mkdir -p /mnt/sda1/arduino/www"));
242     unmount();
243 }
```

```
244
245 void copySystemFilesFromYunToSD() {
246     Serial.print(F("\nCopying files from Arduino Yun flash to micro SD card..."));
247     Process copy;
248
249     copy.runShellCommand(F("mkdir -p /mnt/sda2"));
250     copy.runShellCommand(F("mount /dev/sda2 /mnt/sda2"));
251     copy.runShellCommand(F("rsync -a --exclude=/mnt/ --exclude=/www/sd /overlay/
252 /mnt/sda2/"));
253
254     unmount();
255 }
256
257 void unmount() {
258     Process format;
259     format.runShellCommand(F("umount /dev/sda?"));
260     debugProcess(format);
261     format.runShellCommand(F("rm -rf /mnt/sda?"));
262     debugProcess(format);
263 }
264
265 void enableExtRoot() {
266     Serial.print(F("\nEnabling micro SD as additional disk space... "));
267
268     Process fstab;
269
270     fstab.runShellCommand(F("uci add fstab mount"));
271     fstab.runShellCommand(F("uci set fstab.@mount[0].target=/overlay"));
272     fstab.runShellCommand(F("uci set fstab.@mount[0].device=/dev/sda2"));
273     fstab.runShellCommand(F("uci set fstab.@mount[0].fstype=ext4"));
274     fstab.runShellCommand(F("uci set fstab.@mount[0].enabled=1"));
275     fstab.runShellCommand(F("uci set fstab.@mount[0].enabled_fsck=0"));
276     fstab.runShellCommand(F("uci set fstab.@mount[0].options=rw, sync, noatime, nodiratime"));
277     fstab.runShellCommand(F("uci commit"));
278
279     Serial.println(F("enabled"));
280 }
```

A.6.5. Sketch “Analizador_de_redes_CALIBRACION”

```

1 /*
2
3 CALIBRACION ANALIZADOR DE REDES. TFG. INGENIERIA ELECTRICA Y ELECTRONICA
4 *****
5 Autor: BARBACHANO CIRION, Angel. Fecha: Abril 2018
6
7 Este sketch ha sido desarrollado para la calibracion y monitorizacion de las señales
8 analogicas muestreadas por el analizador de redes; para comprobar la interpretacion
9 de las formas de onda por el CAD, asi como eliminar los offsets y ajustar la salida con
10 las curvas de calibracion.
11
12 */
13
14 #include <Arduino.h>
15 #include <Console.h>
16 #include <Bridge.h>
17 // #include <avr/wdt.h>
18
19 #define sq(x) ((x)*(x))
20
21 static float VESCALE=((5.0*115.0*sqrt(2.0))); //Escalado de tension
22 static float IESCALE=((5.0*12.88)*sqrt(2.0)); //Escalado de corriente
23 static float IOFFSET=(0.404*5.0)*sqrt(2.0); // Y=12,88*x-0.404
24 static float OMEGA=(2.0*PI*50.0); //Velocidad angular a 50 hz
25 static float DEGRAD=(2.0*PI/360.0); //Conversion Grados-Radianes
26 static float RADDEG=(360.0/(2.0*PI)); //Conversion Radianes-Grados
27
28 unsigned long int TZEROV; //Instante de pico maximo tension
29 unsigned long int TZEROI; //Instante de pico maximo corriente
30
31 float VAC; //Valores de V e I RMS, y F.P.
32 float IAC;
33 float COSFI;
34
35 void setup()
36 {
37 Bridge.begin();
38 Serial.begin(9600);
39 Serial.println("inicializando");
40 }
41
42 //El cuerpo del programa esta hecho para realizar subrutinas por partes:
43 //por un lado monitorizar las formas de ondasenoidales con la subrutina Senoidal,
44 // y por otro lado comprobar los valores eficaces de V e I, y el F.P.
45
46 //Comentar las subrutinas que no interesen en el momento.
47
48 //En especial la subrutina de monitorizar senoidales, si se ejecuta con las demas
49 //se reduce mucho el tiempo de muestreo de señales y puede perderse calidad,
50 //la idea de la subrutina es ver la onda lo mejor posible.
51

```

```

52 void loop()
53 {
54   Senoidal();
55   Veficaz();
56   Ieficaz();
57   FP();
58 }
59
60
61 //-----CALCULO DE V RMS-----//
62
63 void Veficaz()
64 {
65   int Valor0=analogRead(A0);      //Valor binario de la componente DC
66   int Valor1=analogRead(A1);      //Valor binario de la senoidal de tension
67   float SUMATORIOV;              //Suma de cuadrados de las muestras escaladas
68   int N=0;                        //Numero de muestras del sumatorio
69   unsigned long int TIME=millis(); //Tiempo de inicio del bucle
70
71   while((millis()-TIME)<=100)     //En el bucle se toman muestras durante 5 ciclos
72   {
73     Valor1 = analogRead(A1);
74     SUMATORIOV=SUMATORIOV + sq(((Valor1-Valor0)*VESCALE/1023));
75     N=N+1;
76   }
77
78   VAC=sqrt((SUMATORIOV/N));       //calcula la raiz del sumatorio entre muestras
79   Serial.println(VAC);
80 }
81
82
83 //-----CALCULO DE I RMS-----//
84 //Misma estructura que el calculo de tension con la analogica A2
85
86 void Ieficaz()
87 {
88   int Valor0=analogRead(A0);      //Valor binario de la componente DC
89   int Valor2=analogRead(A2);      //Valor binario de la senoidal de corriente
90   float SUMATORIOI;
91   int N=0;
92   unsigned long int TIME=millis();
93
94   while((millis()-TIME)<=100)
95   {
96     Valor2 = analogRead(A2);
97     SUMATORIOI=SUMATORIOI + sq(((Valor2-Valor0)*IESCALE/1023)-(IOFFSET));
98     N=N+1;
99   }
100
101   IAC=sqrt((SUMATORIOI/N));
102   Serial.println(IAC);
103 }
104
105

```

```
106 //----CALCULO DE FP-----//
107
108 void FP()
109 {
110     boolean TONI=false;
111     boolean TONV=false;
112     boolean DECR=false;
113     boolean PICO=false;
114     unsigned long int WAIT;
115     unsigned long int FLANCO=0;
116     int DIFERENCIA;
117     int Valor0=analogRead(A0);
118     int Valor1=analogRead(A0);
119     int Valor2=analogRead(A0);
120     WAIT=millis();
121
122 //El siguiente bucle copia el tiempo de la señal de tension en flanco de bajada en 0
123 //Primero comprueba que esta en flanco de subida, despues comprueba que ha pasado el
124 //pico maximo y despues espera al 0.
125
126 while (TONV==false)
127     {
128         if ((analogRead(A1)>Valor1) && (DECR==false))
129             {
130                 Valor1=analogRead(A1);
131                 while ((analogRead(A1)>Valor1) && (DECR==false)) {PICO=true;}
132             }
133
134         Valor1=analogRead(A0);
135         if ((PICO==true) && (DECR==false))
136             {
137                 DECR=true;
138                 FLANCO=millis();
139             }
140
141         DIFERENCIA=(analogRead(A1)-analogRead(A0));
142         while ((DIFERENCIA<=15) && (DIFERENCIA>=-15) && (DECR==true) && (millis()-FLANCO<5))
143             {
144                 TZEROV=micros();
145                 TONV=true;
146                 break;
147             }
148
149         if ((millis()-FLANCO>5) && (DECR==true))
150             {
151                 DECR=false;
152                 PICO=false;
153                 Valor1=analogRead(A0);
154             }
155
156         if (millis()-WAIT>150){break;}
157     }
158
```

```

159  DECR=false;
160  PICO=false;
161  FLANCO=0;
162  WAIT=millis();
163  Valor2=analogRead(A0);
164
165  //El bucle funciona igual que el de tension pero con señal de corriente
166
167  while (TONI==false)
168  {
169      if ((analogRead(A2)>Valor2) && (DECR==false))
170      {
171          Valor2=analogRead(A2);
172          while ((analogRead(A2)>Valor2) && (DECR==false)) {PICO=true;}
173      }
174
175      Valor2=analogRead(A0);
176      if ((PICO==true) && (DECR==false))
177      {
178          DECR=true;
179          FLANCO=millis();
180      }
181
182      DIFERENCIA=(analogRead(A2)-analogRead(A0));
183      while ((DIFERENCIA<=0) && (DIFERENCIA>=-15) && (DECR==true) && (millis()-FLANCO<5))
184      {
185          TZEROI=micros();
186          TONI=true;
187          break;
188      }
189
190      if ((millis()-FLANCO>5) && (DECR==true))
191      {
192          DECR=false;
193          PICO=false;
194          Valor2=analogRead(A0);
195      }
196
197      if ((millis()-WAIT>150)) {break; }
198  }
199
200  //Calculo de desfase en radianes y comprenderlo en el dominio de [-180;180]
201
202  float FI;
203
204  if ((TONV==true) && (TONI==true))
205  {
206      FI=( (TZEROI-TZEROV) * (OMEGA) * (RADDEG) /1000000.0);
207      for (; (FI>180.0);) {FI=FI-360.0;}
208      FI=( (-FI) + (FPESCALE*IAC) +FPOFFSET);
209  }
210
211  COSFI=cos (FI*DEGRAD);
212  Serial.println (FI);
213  Serial.println (COSFI);
214 }

```

```
215 |
216 |
217 | //----MONITORIZAR SENOIDALES-----//
218 |
219 | void Senoidal()
220 | {
221 |     int Valor0=analogRead(A0); //Valor binario de continua
222 |     int Valor1=analogRead(A1); //Valor binario de VAC
223 |     int Valor2=analogRead(A2); //Valor binario de IAC
224 |
225 |     float VDC=((Valor0*5.0)/1023.0);
226 |     float VACseno=(Valor1-Valor0)*(VESCALE/1023.0);
227 |     float IACseno((((Valor2-Valor0)*((IESCALE)/1023.0))-(IOFFSET));
228 |
229 |     Serial.println(VDC);
230 |     Serial.println(VACseno);
231 |     Serial.println(IACseno);
232 | }
```

A.6.6. Sketch “Analizador_de_redes_FIRMWARE”

Este es el sketch para cargar a Arduino en el cual se implementan las subrutinas de todas las funciones a realizar por el prototipo. Una vez cargado el sketch, el dispositivo ya puede funcionar de forma autónoma.

```

1  /*
2
3  FIRMWARE ANALIZADOR DE REDES. TFG.  INGENIERIA ELECTRICA Y ELECTRONICA
4  ****
5  Autor: BARBACHANO CIRION, Angel.   Fecha: Abril 2018
6
7  */
8
9
10 #include <BridgeServer.h>
11 #include <YunServer.h>
12 #include <BridgeClient.h>
13 #include <YunClient.h>
14 #include <Bridge.h>
15 #include <Console.h>
16 #include <FileIO.h>
17 #include <LiquidCrystal_I2C.h>
18 #include <Arduino.h>
19 #include <Wire.h>
20 #include <RTC.h>
21 #include <math.h>
22 #include <avr/wdt.h>
23
24 #define sq(x) ((x)*(x))
25
26 RTC rtc(DST_ON);
27
28 BridgeServer SERVIDOR;
29
30 LiquidCrystal_I2C lcd(0x3F,20,4);
31
32
33 //-----DECLARACION DE VARIABLES GLOBALES-----//
34
35 static float VESCALE=((5.0*115.0*sqrt(2.0))); //Escalado de tension
36 static float IESCALE=((5.0*12.88*sqrt(2.0))); //Escalado de corriente
37 static float IOFFSET=((0.18)); // Y=12,88*x-0.404
38 static float FPESCALE=0.333;
39 static float FPOFFSET=3.044;
40 static float OMEGA=(2.0*PI*50.0); //Velocidad angular a 50 hz
41 static float DEGRAD=(2.0*PI/360.0); //Conversion Grados-Radianes
42 static float RADDEG=(360.0/(2.0*PI)); //Conversion Radianes-Grados
43
44 unsigned long int TZEROV; //Valor en us en instante de pico de V
45 unsigned long int TZEROI; //Valor de us en instante de pico de I
46 unsigned long int TONLCD=0;

```



```

47
48 byte MINUTOSD;
49
50 volatile byte SCREEN=1;
51
52 boolean FAILRSD=false;           //Flag de error de lectura SD
53 boolean FAILWSD=false;           //Flag de error de escritura SD
54 boolean FLAGINT=false;           //Flag de interrupcion
55 boolean FLAGSD=false;            //Flag de habilitacion de SD
56 boolean FAILCLIENT=false;        //Flag de error de conexion de cliente
57
58 float FI;                         //Angulo en grados de -180 a 180
59 float VAC;                         //Modulo de tension eficaz instantanea en V
60 float IAC;                         //Modulo de corriente eficaz instantanea en A
61 float P;
62 float WHC=0.00;
63 float WHI=0.00;
64 float E=0;                         //Energia temporal en watios*hora  w*h=kw*h/1000
65 float IMAX=0.00;
66 float PMAX=0.00;
67
68 char* RUTA[]={""};
69
70 Data TIME;                         //Cadenas de valores de fecha y hora
71 Data TIMEWD;
72
73
74 //////////////////////////////////////////////////--SUBROUTINA DE INICIO--////////////////////////////////////
75
76 void setup()
77 {
78   wdt_disable();
79   Bridge.begin();
80   Serial.begin(9600);
81   SERVIDOR.listenOnLocalhost();
82   SERVIDOR.begin();
83   FileSystem.begin();
84   lcd.init();
85   pinMode(7, INPUT);
86   attachInterrupt(4, Change_Screen, RISING);
87   TIME = rtc.getData();
88   MINUTOSD = TIME.minutes;
89   wdt_enable(WDTO_8S);
90   File WATCHDOG=FileSystem.open("/mnt/sd/TFG/WD.txt", FILE_WRITE);
91   WATCHDOG.print(TIME.toString());
92   WATCHDOG.close();
93   TIMEWD = rtc.getData();

```

```

94 // if (!FileSystem.exists("/mnt/sd/TFG/")) {FileSystem.mkdir("/mnt/sd/TFG/");}
95 // if (!FileSystem.exists("/mnt/sd/TFG/ENE.txt"))
96 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/ENE.txt", FILE_WRITE); NEWFILE.close();}
97 // if (!FileSystem.exists("/mnt/sd/TFG/FEB.txt"))
98 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/FEB.txt", FILE_WRITE); NEWFILE.close();}
99 // if (!FileSystem.exists("/mnt/sd/TFG/MAR.txt"))
100 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/MAR.txt", FILE_WRITE); NEWFILE.close();}
101 // if (!FileSystem.exists("/mnt/sd/TFG/ABR.txt"))
102 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/ABR.txt", FILE_WRITE); NEWFILE.close();}
103 // if (!FileSystem.exists("/mnt/sd/TFG/MAY.txt"))
104 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/MAY.txt", FILE_WRITE); NEWFILE.close();}
105 // if (!FileSystem.exists("/mnt/sd/TFG/JUN.txt"))
106 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/JUN.txt", FILE_WRITE); NEWFILE.close();}
107 // if (!FileSystem.exists("/mnt/sd/TFG/JUL.txt"))
108 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/JUL.txt", FILE_WRITE); NEWFILE.close();}
109 // if (!FileSystem.exists("/mnt/sd/TFG/AGO.txt"))
110 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/AGO.txt", FILE_WRITE); NEWFILE.close();}
111 // if (!FileSystem.exists("/mnt/sd/TFG/SEP.txt"))
112 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/SEP.txt", FILE_WRITE); NEWFILE.close();}
113 // if (!FileSystem.exists("/mnt/sd/TFG/OCT.txt"))
114 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/OCT.txt", FILE_WRITE); NEWFILE.close();}
115 // if (!FileSystem.exists("/mnt/sd/TFG/NOV.txt"))
116 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/NOV.txt", FILE_WRITE); NEWFILE.close();}
117 // if (!FileSystem.exists("/mnt/sd/TFG/DIC.txt"))
118 //     {File NEWFILE=FileSystem.open("/mnt/sd/TFG/DIC.txt", FILE_WRITE); NEWFILE.close();}
119 }
120
121
122 //////////////////////////////////////////////////--CUERPO DEL PROGRAMA--////////////////////////////////////
123
124 void loop()
125 {
126   wdt_reset();
127   V_Eficaz();
128   I_Eficaz();
129   Calculo_PHASE();
130   Calculo_ENERGIA();
131   Registro_SD();
132   Valores_LCD();
133   Servidor_Web();
134 }
135
136
137 //////////////////////////////////////////////////--SUBROUTINA DE INTERRUPCION--////////////////////////////////////
138
139 void Change_Screen()
140 {
141   if ((SCREEN>=0) && (SCREEN<8) && (FLAGINT==false))
142   {
143     SCREEN++;
144     TONLCD=millis();
145     FLAGINT=true;
146   }
147   else if ((SCREEN>=8) && (FLAGINT==false))
148   {
149     SCREEN=1;
150     TONLCD=millis();
151     FLAGINT=true;
152   }
153 }

```

```

154
155
156 //-----ESPACIO PARA SUBROUTINAS Y FUNCIONES-----//
157 //-----
158
159 //-----CALCULO DE MODULO DE TENSION EFICAZ-----//
160
161 void V_Eficaz()
162 {
163     int Valor0=analogRead(A0); //Valor binario de la componente DC sumada a las senoidales
164     int Valor1=analogRead(A1); //Valor binario de la senoidal de tension
165     float SUMATORIOV; //Suma de cuadrados de las muestras escaladas
166     int N=0; //Numero de muestras del sumatorio
167     unsigned long int TIME=millis(); //Tiempo de inicio del bucle
168
169     while((millis()-TIME)<=300) //bucle de 300ms, correspondientes a 15 ciclos de senoidal,
170     {
171         Valor1 = analogRead(A1);
172         SUMATORIOV=SUMATORIOV + sq((Valor1-Valor0)*VESCALE/1023);
173         N=N+1;
174     }
175     VAC=sqrt((SUMATORIOV/N)); //Calculo de valor RMS
176 }
177
178
179 //-----CALCULO DE MODULO DE CORRIENTE EFICAZ-----//
180
181 void I_Eficaz()
182 {
183     int Valor0=analogRead(A0); //Valor binario de la componente DC sumada a las senoidales
184     int Valor2=analogRead(A2); //Valor binario de la senoidal de corriente
185     float SUMATORIOI;
186     int N=0;
187     unsigned long int TIME=millis();
188
189     while((millis()-TIME)<=300)
190     {
191         Valor2 = analogRead(A2);
192         SUMATORIOI=SUMATORIOI + sq(((Valor2-Valor0)*IESCALE/1023));
193         N=N+1;
194     }
195     IAC=abs(sqrt((SUMATORIOI/N))-IOFFSET); //Calculo de valor RMS
196 }
197

```

```

198
199 //////////////////////////////////////////////////--CALCULO DE DESFASE--////////////////////////////////////
200 //Interpretando que el fasor de tension esta en el eje x, la corriente adelantada
201 //estaria en angulo positivo de 0 a 180 y la corriente en retraso de 0 a -180 en grados.
202
203 void Calculo_PHASE()
204 {
205     boolean TONI=false;
206     boolean TONV=false;
207     boolean DECR=false;
208     boolean PICO=false;
209     unsigned long int WAIT;
210     unsigned long int FLANCO=0;
211     int DIFERENCIA;
212     int Valor0=analogRead(A0);
213     int Valor1=analogRead(A0);
214     int Valor2=analogRead(A0);
215     WAIT=millis();
216
217     while (TONV==false)
218     {
219         if ((analogRead(A1)>Valor1) && (DECR==false))
220         {
221             Valor1=analogRead(A1);
222             while ((analogRead(A1)>Valor1) && (DECR==false)) {PICO=true; }
223         }
224
225         Valor1=analogRead(A0);
226         if ((PICO==true) && (DECR==false))
227         {
228             DECR=true;
229             FLANCO=millis();
230         }
231
232         DIFERENCIA=(analogRead(A1)-analogRead(A0));
233         while ((DIFERENCIA<=15) && (DIFERENCIA>=-15) && (DECR==true) && (millis()-FLANCO<5))
234         {
235             TZEROV=micros();
236             TONV=true;
237             break;
238         }
239
240         if ((millis()-FLANCO>5) && (DECR==true))
241         {
242             DECR=false;
243             PICO=false;
244             Valor1=analogRead(A0);
245         }
246
247         if (millis()-WAIT>150) {break;}
248     }
249
250     DECR=false;
251     PICO=false;
252     FLANCO=0;
253     WAIT=millis();
254     Valor2=analogRead(A0);
255

```

```

256 while (TONI==false)
257 {
258     if ((analogRead(A2)>Valor2) && (DECR==false))
259     {
260         Valor2=analogRead(A2);
261         while ((analogRead(A2)>Valor2) && (DECR==false)) {PICO=true;}
262     }
263
264     Valor2=analogRead(A0);
265     if ((PICO==true) && (DECR==false))
266     {
267         DECR=true;
268         FLANCO=millis();
269     }
270
271     DIFERENCIA=(analogRead(A2)-analogRead(A0));
272     while ((DIFERENCIA<=0) && (DIFERENCIA>=-15) && (DECR==true) && (millis()-FLANCO<5))
273     {
274         TZEROI=micros();
275         TONI=true;
276         break;
277     }
278
279     if ((millis()-FLANCO>5) && (DECR==true))
280     {
281         DECR=false;
282         PICO=false;
283         Valor2=analogRead(A0);
284     }
285
286     if ((millis()-WAIT>150)) {break;}
287 }
288
289 if ((TONV==true) && (TONI==true))
290 {
291     FI=((TZEROI-TZEROV)*(OMEGA)*(RADDEG)/1000000.0); //Calculo de desfase en radianes
292     for (; (FI>180.0);) {FI=FI-360.0;} //Valores entre -180 y 180 grados
293     FI=(-FI)+(FPESCALE*IAC)+FPOFFSET; //funciones trigonometricas siempre en radianes
294 }
295 }
296
297
298 //////////////////////////////////--CALCULO DE ENERGIA--////////////////////////////////////
299
300 void Calculo_ENERGIA()
301 {
302     Data NOW=rtc.getData();
303     for (int i=0; i<1;)
304     {
305         if(NOW.seconds!=TIME.seconds){i++;}
306         delay(100);
307         NOW=rtc.getData();
308     }
309
310     if((NOW.seconds)<(TIME.seconds)) {NOW.seconds=NOW.seconds + 60;}
311
312     P=(VAC*IAC*(cos(FI*DEGRAD)));
313     E=E+((P)/3600.00f)*(NOW.seconds-TIME.seconds);
314     TIME = rtc.getData();
315 }

```

```
316
317
318 //-----REGISTRO DE ENERGIA Y MAXIMOS MENSUALES EN TARJETA SD-----//
319
320 void Registro_SD()
321 {
322     if (TIME.minutes!=MINUTOSD)
323     {
324         MINUTOSD=TIME.minutes;
325         FLAGSD=true;
326         byte SECUENCIA=1;
327         String CADENA="";
328         char A;
329         byte NUMMES=TIME.month;
330         Serial.println(NUMMES);
331
332         switch (NUMMES)
333         {
334             case 1: RUTA[0]={"/mnt/sd/TFG/ENE.txt"} ; break;
335             case 2: RUTA[0]={"/mnt/sd/TFG/FEB.txt"} ; break;
336             case 3: RUTA[0]={"/mnt/sd/TFG/MAR.txt"} ; break;
337             case 4: RUTA[0]={"/mnt/sd/TFG/ABR.txt"} ; break;
338             case 5: RUTA[0]={"/mnt/sd/TFG/MAY.txt"} ; break;
339             case 6: RUTA[0]={"/mnt/sd/TFG/JUN.txt"} ; break;
340             case 7: RUTA[0]={"/mnt/sd/TFG/JUL.txt"} ; break;
341             case 8: RUTA[0]={"/mnt/sd/TFG/AGO.txt"} ; break;
342             case 9: RUTA[0]={"/mnt/sd/TFG/SEP.txt"} ; break;
343             case 10: RUTA[0]={"/mnt/sd/TFG/OCT.txt"} ; break;
344             case 11: RUTA[0]={"/mnt/sd/TFG/NOV.txt"} ; break;
345             case 12: RUTA[0]={"/mnt/sd/TFG/DIC.txt"} ; break;
346         }
347
348         File REGR = FileSystem.open(RUTA[0], FILE_READ);
349         if (REGR)
350         {
351             FAILRSD=false;
352             REGR.seek(0);
353
354             while (REGR.available())
355             {
356                 A=REGR.read();
357                 if (A!=44) {CADENA=CADENA+A;}
358
359                 else if (A==44)
360                 {
361                     switch (SECUENCIA)
362                     {
363                         case 1: WHC=CADENA.toFloat(); CADENA=""; SECUENCIA=2; break;
364                         case 2: WHI=CADENA.toFloat(); CADENA=""; SECUENCIA=3; break;
365                         case 3: IMAX=CADENA.toFloat(); CADENA=""; SECUENCIA=4; break;
366                         case 4: PMAX=CADENA.toFloat(); CADENA=""; SECUENCIA=1; break;
367                     }
368                 }
369             }
370
371             REGR.close();
372
```

```
373     if (E>=0) {WHC=WHC+E;}
374     if (E<0) {WHI=WHI+E;}
375     if (P<P) {P=P;}
376     if (I<I) {I=I;}
377
378     E=0.00;
379     File REGW = FileSystem.open(RUTA[0], FILE_WRITE);
380
381     if (REGW)
382     {
383     FAILWSD=false;
384     REGW.seek(0);
385     REGW.print(WHC);
386     REGW.print(",");
387     REGW.print(WHI);
388     REGW.print(",");
389     REGW.print(IMAX);
390     REGW.print(",");
391     REGW.print(P);
392     REGW.print(",");
393     REGW.close();
394     }
395
396     else if (!REGW) {FAILWSD=true;}
397     }
398     else if (!REGR) {FAILRSD=true;}
399     }
400 }
401
402
403 //////////////////////////////////////////////////--SUBROUTINA DE PANTALLAS EN LCD--////////////////////////////////////
404
405 void Valores_LCD()
406 {
407     if ((millis()-TONLCD)>30000)
408     {
409     SCREEN=0;
410     lcd.noDisplay();
411     lcd.noBacklight();
412     }
413
414     if ((millis()-TONLCD)>50) {FLAGINT=false;}
415
416
```

```
417 //-----PANTALLA 1-----
418
419     if ((SCREEN==1) && (FLAGINT==false))
420     {
421         lcd.display();
422         lcd.clear();
423         lcd.backlight();
424         lcd.setCursor(0,0); lcd.print("VALORES INSTANTANEOS");
425         lcd.setCursor(0,1); lcd.print("V red=");
426         lcd.setCursor(6,1); lcd.print(VAC);
427         lcd.setCursor(11,1); lcd.print("V");
428         lcd.setCursor(0,2); lcd.print("I carga=");
429         lcd.setCursor(8,2); lcd.print(IAC);
430         lcd.setCursor(13,2); lcd.print("A");
431         lcd.setCursor(0,3); lcd.print("FI=");
432         lcd.setCursor(3,3); lcd.print(FI);
433         lcd.setCursor(11,3); lcd.print("FP=");
434         lcd.setCursor(14,3); lcd.print(cos(FI*DEGRAD));
435     }
436
437
438 //-----PANTALLA 2-----
439
440     if ((SCREEN==2) && (FLAGINT==false))
441     {
442         lcd.display();
443         lcd.clear();
444         lcd.backlight();
445         lcd.setCursor(0,0); lcd.print("VALORES INSTANTANEOS");
446         lcd.setCursor(0,1); lcd.print("S=");
447         lcd.setCursor(2,1); lcd.print(IAC*VAC);
448         lcd.setCursor(9,1); lcd.print("VA");
449         lcd.setCursor(12,1); lcd.print("FP=");
450         lcd.setCursor(15,1); lcd.print(cos(FI*DEGRAD));
451         lcd.setCursor(0,2); lcd.print("P=");
452         lcd.setCursor(2,2); lcd.print(P);
453         lcd.setCursor(9,2); lcd.print("W");
454         lcd.setCursor(0,3); lcd.print("Q=");
455         lcd.setCursor(2,3); lcd.print(sqrt(sq(VAC*IAC)-sq(P)));
456         lcd.setCursor(9,3);
457         if (FI<0) {lcd.print("VAR (L)");}
458         else if (FI>0) {lcd.print("VAR (C)");}
459         else if (FI==0) {lcd.print("VAR");}
460     }
```



```
461
462
463 //-----PANTALLA 3-----
464
465     if ((SCREEN==3) && (FLAGINT==false))
466     {
467         lcd.display();
468         lcd.clear();
469         lcd.backlight();
470         lcd.setCursor(0,0); lcd.print("MAXIMOS Y ACUMULADOS");
471         lcd.setCursor(0,1); lcd.print("Imax=");
472         lcd.setCursor(5,1); lcd.print(IMAX);
473         lcd.setCursor(10,1);lcd.print("A");
474         lcd.setCursor(0,2);lcd.print("Pmax=");
475         lcd.setCursor(6,2); lcd.print(PMAX/1000.00f);
476         lcd.setCursor(10,2);lcd.print("KW");
477         lcd.setCursor(0,3); lcd.print("Econsumida=");
478         lcd.setCursor(12,3);lcd.print(WHC/1000.00f);
479         lcd.setCursor(17,3);lcd.print("KWH");
480     }
481
482
483 //-----LECTURA DE DATOS EN SD PARA PANTALLAS 4, 5, 6 Y 7-----
484
485     if ((SCREEN<=7) && (SCREEN>=4) && (FLAGINT==false))
486     {
487         byte SECUENCIA=1;
488         String CADENA="";
489         char A;
490         float CONSUMO=0.00;
491         float INYECTADO=0.00;
492         float CENE=0.00; float CFEB=0.00; float CMAR=0.00;
493         float CABR=0.00; float CMAY=0.00; float CJUN=0.00;
494         float CJUL=0.00; float CAGO=0.00; float CSEP=0.00;
495         float COCT=0.00; float CNOV=0.00; float CDIC=0.00;
496         float YENE=0.00; float YFEB=0.00; float YMAR=0.00;
497         float YABR=0.00; float YMAY=0.00; float YJUN=0.00;
498         float YJUL=0.00; float YAGO=0.00; float YSEP=0.00;
499         float YOCT=0.00; float YNOV=0.00; float YDIC=0.00;
500
501         if (FLAGSD==true)
502         {
503             FLAGSD==false;
504             for (int i=1; i<13; i++)
505             {
506                 boolean READ=false;
```

```
507     switch (i)
508     {
509         case 1: RUTA[0]={"/mnt/sd/TFG/ENE.txt"} ; break;
510         case 2: RUTA[0]={"/mnt/sd/TFG/FEB.txt"} ; break;
511         case 3: RUTA[0]={"/mnt/sd/TFG/MAR.txt"} ; break;
512         case 4: RUTA[0]={"/mnt/sd/TFG/ABR.txt"} ; break;
513         case 5: RUTA[0]={"/mnt/sd/TFG/MAY.txt"} ; break;
514         case 6: RUTA[0]={"/mnt/sd/TFG/JUN.txt"} ; break;
515         case 7: RUTA[0]={"/mnt/sd/TFG/JUL.txt"} ; break;
516         case 8: RUTA[0]={"/mnt/sd/TFG/AGO.txt"} ; break;
517         case 9: RUTA[0]={"/mnt/sd/TFG/SEP.txt"} ; break;
518         case 10: RUTA[0]={"/mnt/sd/TFG/OCT.txt"} ; break;
519         case 11: RUTA[0]={"/mnt/sd/TFG/NOV.txt"} ; break;
520         case 12: RUTA[0]={"/mnt/sd/TFG/DIC.txt"} ; break;
521     }
522     File REGR = FileSystem.open(RUTA[0], FILE_READ);
523     if (REGR)
524     {
525         FAILRSD=false;
526         REGR.seek(0);
527
528         while ((REGR.available()) && (READ==false))
529         {
530             A=REGR.read();
531             if (A!=44) {CADENA=CADENA+A;}
532             else if (A==44)
533             {
534                 switch (SECUENCIA)
535                 {
536                     case 1: CONSUMO=abs(CADENA.toFloat());
537                         CADENA="";
538
539                         switch (i)
540                         {
541                             case 1: CENE=(CONSUMO/1000.00f); break;
542                             case 2: CFEB=(CONSUMO/1000.00f); break;
543                             case 3: CMAR=(CONSUMO/1000.00f); break;
544                             case 4: CABR=(CONSUMO/1000.00f); break;
545                             case 5: CMAY=(CONSUMO/1000.00f); break;
546                             case 6: CJUN=(CONSUMO/1000.00f); break;
547                             case 7: CJUL=(CONSUMO/1000.00f); break;
548                             case 8: CAGO=(CONSUMO/1000.00f); break;
549                             case 9: CSEP=(CONSUMO/1000.00f); break;
550                             case 10: COCT=(CONSUMO/1000.00f); break;
551                             case 11: CNOV=(CONSUMO/1000.00f); break;
552                             case 12: CDIC=(CONSUMO/1000.00f); break;
553                         }
554                         SECUENCIA=2;
555                         break;
556
557                     case 2: INYECTADO=abs(CADENA.toFloat());
558                         CADENA="";
559                         Serial.println(i);
```

```
559         switch (i)
560         {
561             case 1: YENE=(INYECTADO/1000.00f); break;
562             case 2: YFEB=(INYECTADO/1000.00f); break;
563             case 3: YMAR=(INYECTADO/1000.00f); break;
564             case 4: YABR=(INYECTADO/1000.00f); break;
565             case 5: YMAY=(INYECTADO/1000.00f); break;
566             case 6: YJUN=(INYECTADO/1000.00f); break;
567             case 7: YJUL=(INYECTADO/1000.00f); break;
568             case 8: YAGO=(INYECTADO/1000.00f); break;
569             case 9: YSEP=(INYECTADO/1000.00f); break;
570             case 10: YOCT=(INYECTADO/1000.00f); break;
571             case 11: YNOV=(INYECTADO/1000.00f); break;
572             case 12: YDIC=(INYECTADO/1000.00f); break;
573         }
574         SECUENCIA=1;
575         READ=true;
576         break;
577     }
578 }
579 }
580 REGR.close();
581 }
582 else if (!REGR) {FAILRSD=true;}
583 }
584 }
585
586
587 //-----PANTALLA 4-----
588
589 if (SCREEN==4)
590 {
591     lcd.display();
592     lcd.clear();
593     lcd.backlight();
594     lcd.setCursor(0,0); lcd.print("CONSUMIDO (KWH)");
595     lcd.setCursor(0,1); lcd.print("ENE=");
596     lcd.setCursor(4,1); lcd.print(CENE);
597     lcd.setCursor(10,1);lcd.print("ABR=");
598     lcd.setCursor(14,1);lcd.print(CABR);
599     lcd.setCursor(0,2); lcd.print("FEB=");
600     lcd.setCursor(4,2); lcd.print(CFEB);
601     lcd.setCursor(10,2);lcd.print("MAY=");
602     lcd.setCursor(14,2);lcd.print(CMAY);
603     lcd.setCursor(0,3); lcd.print("MAR=");
604     lcd.setCursor(4,3); lcd.print(CMAR);
605     lcd.setCursor(10,3);lcd.print("JUN=");
606     lcd.setCursor(14,3);lcd.print(CJUN);
607 }
608
```

```
609
610 //-----PANTALLA 5-----
611
612     if (SCREEN==5)
613     {
614         lcd.display();
615         lcd.clear();
616         lcd.backlight();
617         lcd.setCursor(0,0); lcd.print("CONSUMIDO (KWH)");
618         lcd.setCursor(0,1); lcd.print("JUL=");
619         lcd.setCursor(4,1); lcd.print(CJUL);
620         lcd.setCursor(10,1);lcd.print("OCT=");
621         lcd.setCursor(14,1);lcd.print(COCT);
622         lcd.setCursor(0,2); lcd.print("AGO=");
623         lcd.setCursor(4,2); lcd.print(CAGO);
624         lcd.setCursor(10,2);lcd.print("NOV=");
625         lcd.setCursor(14,2);lcd.print(CNOV);
626         lcd.setCursor(0,3); lcd.print("SEP=");
627         lcd.setCursor(4,3); lcd.print(CSEP);
628         lcd.setCursor(10,3);lcd.print("DIC=");
629         lcd.setCursor(14,3);lcd.print(CDIC);
630     }
631
632
633 //-----PANTALLA 6-----
634
635     if (SCREEN==6)
636     {
637         lcd.display();
638         lcd.clear();
639         lcd.backlight();
640         lcd.setCursor(0,0); lcd.print("INYECTADO (KWH)");
641         lcd.setCursor(0,1); lcd.print("ENE=");
642         lcd.setCursor(4,1); lcd.print(YENE);
643         lcd.setCursor(10,1);lcd.print("ABR=");
644         lcd.setCursor(14,1);lcd.print(YABR);
645         lcd.setCursor(0,2); lcd.print("FEB=");
646         lcd.setCursor(4,2); lcd.print(YFEB);
647         lcd.setCursor(10,2);lcd.print("MAY=");
648         lcd.setCursor(14,2);lcd.print(YMAY);
649         lcd.setCursor(0,3); lcd.print("MAR=");
650         lcd.setCursor(4,3); lcd.print(YMAR);
651         lcd.setCursor(10,3);lcd.print("JUN=");
652         lcd.setCursor(14,3);lcd.print(YJUN);
653     }
654
```

```
655
656 //-----PANTALLA 7-----
657
658     if (SCREEN==7)
659     {
660         lcd.display();
661         lcd.clear();
662         lcd.backlight();
663         lcd.setCursor(0,0); lcd.print("INYECTADO (KWH)");
664         lcd.setCursor(0,1); lcd.print("JUL=");
665         lcd.setCursor(4,1); lcd.print(YJUL);
666         lcd.setCursor(10,1); lcd.print("OCT=");
667         lcd.setCursor(14,1); lcd.print(YOCT);
668         lcd.setCursor(0,2); lcd.print("AGO=");
669         lcd.setCursor(4,2); lcd.print(YAGO);
670         lcd.setCursor(10,2); lcd.print("NOV=");
671         lcd.setCursor(14,2); lcd.print(YNOV);
672         lcd.setCursor(0,3); lcd.print("SEP=");
673         lcd.setCursor(4,3); lcd.print(YSEP);
674         lcd.setCursor(10,3); lcd.print("DIC=");
675         lcd.setCursor(14,3); lcd.print(YDIC);
676     }
677 }
678
679
680 //-----PANTALLA 8-----
681
682     if ((SCREEN==8) && (FLAGINT==false))
683     {
684         lcd.display();
685         lcd.clear();
686         lcd.setCursor(0,0); lcd.print(TIME.toString());
687         lcd.setCursor(0,1);
688         if (FAILRSD==false) {lcd.print("Lectura SD: OK");}
689         else if (FAILRSD==true) {lcd.print("Lectura SD: Error");}
690         lcd.setCursor(0,2);
691         if (FAILWSD==false) {lcd.print("Escritura SD: OK");}
692         else if (FAILWSD==true) {lcd.print("Escritura SD: Error");}
693         lcd.setCursor(0,3); lcd.print("WD=");
694         lcd.setCursor(3,3); lcd.print(TIMEWD.hour24h);
695         lcd.setCursor(5,3); lcd.print(":");
696         lcd.setCursor(6,3); lcd.print(TIMEWD.minutes);
697         lcd.setCursor(9,3); lcd.print(TIMEWD.day);
698         lcd.setCursor(11,3); lcd.print("/");
699         lcd.setCursor(12,3); lcd.print(TIMEWD.month);
700         lcd.setCursor(14,3); lcd.print("/");
701         lcd.setCursor(15,3); lcd.print(TIMEWD.year);
702     }
703 }
704
```

```
705
706 //-----SUBROUTINA DE EXPORTACION DE DATOS AL SERVIDOR-----//
707
708 void Servidor_Web()
709 {
710     BridgeClient CLIENTE = SERVIDOR.accept();
711     if (CLIENTE)
712     {
713         String PETICION = CLIENTE.readStringUntil('\r');
714         if (PETICION=="YUN")
715         {
716             CLIENTE.println("VALORES INSTANTANEOS");
717             CLIENTE.println(" ");
718             CLIENTE.print("V red="); CLIENTE.print(VAC); CLIENTE.print("V");
719             CLIENTE.print(" ");
720             CLIENTE.print("I carga="); CLIENTE.print(IAC); CLIENTE.println("A");
721             CLIENTE.print("FI="); CLIENTE.print(FI); CLIENTE.print(" ");
722             CLIENTE.print("FP="); CLIENTE.println(cos(FI*DEGRAD));
723             CLIENTE.println(" ");
724             CLIENTE.println("TRIANGULO DE POTENCIAS");
725             CLIENTE.println(" ");
726             CLIENTE.print("Potencia aparente (S)="); CLIENTE.print(VAC*IAC); CLIENTE.println("VA");
727             CLIENTE.print("Potencia activa (P)="); CLIENTE.print(P); CLIENTE.println("W");
728             CLIENTE.print("Potencia reactiva (Q)="); CLIENTE.print(sqrt(sq(VAC*IAC)-sq(P)));
729             CLIENTE.println("VAR");
730             CLIENTE.println(" ");
731             CLIENTE.println("MAXIMOS DEL MES");
732             CLIENTE.println(" ");
733             CLIENTE.print("Pico maximo de corriente="); CLIENTE.print(IMAX); CLIENTE.println("A");
734             CLIENTE.print("Pico maximo de potencia="); CLIENTE.print(PMAX); CLIENTE.println("W");
735             CLIENTE.println(" ");
736             CLIENTE.println("TOTALIZADORES MENSUALES DE ENERGIA");
737             CLIENTE.println(" ");
738             CLIENTE.print("Energia consumida="); CLIENTE.print(WHC/1000.00); CLIENTE.println("KWH");
739             CLIENTE.print("Energia inyectada="); CLIENTE.print(abs(WHI/1000.00)); CLIENTE.println("KWH");
740         }
741         CLIENTE.stop();
742         FAILCLIENT=false;
743     }
744
745     if (!CLIENTE) {FAILCLIENT=true;}
746 }
747
```

A.7. Planos

A.7.1. Diagrama esquemático PCB

A.7.2. Diagrama funcional

A.8. Datasheets

A.8.1. Efergy engage solar kit

A.8.2. Mirubee Mirubox V.2.

A.8.3. Polímetro PCE-PCM1

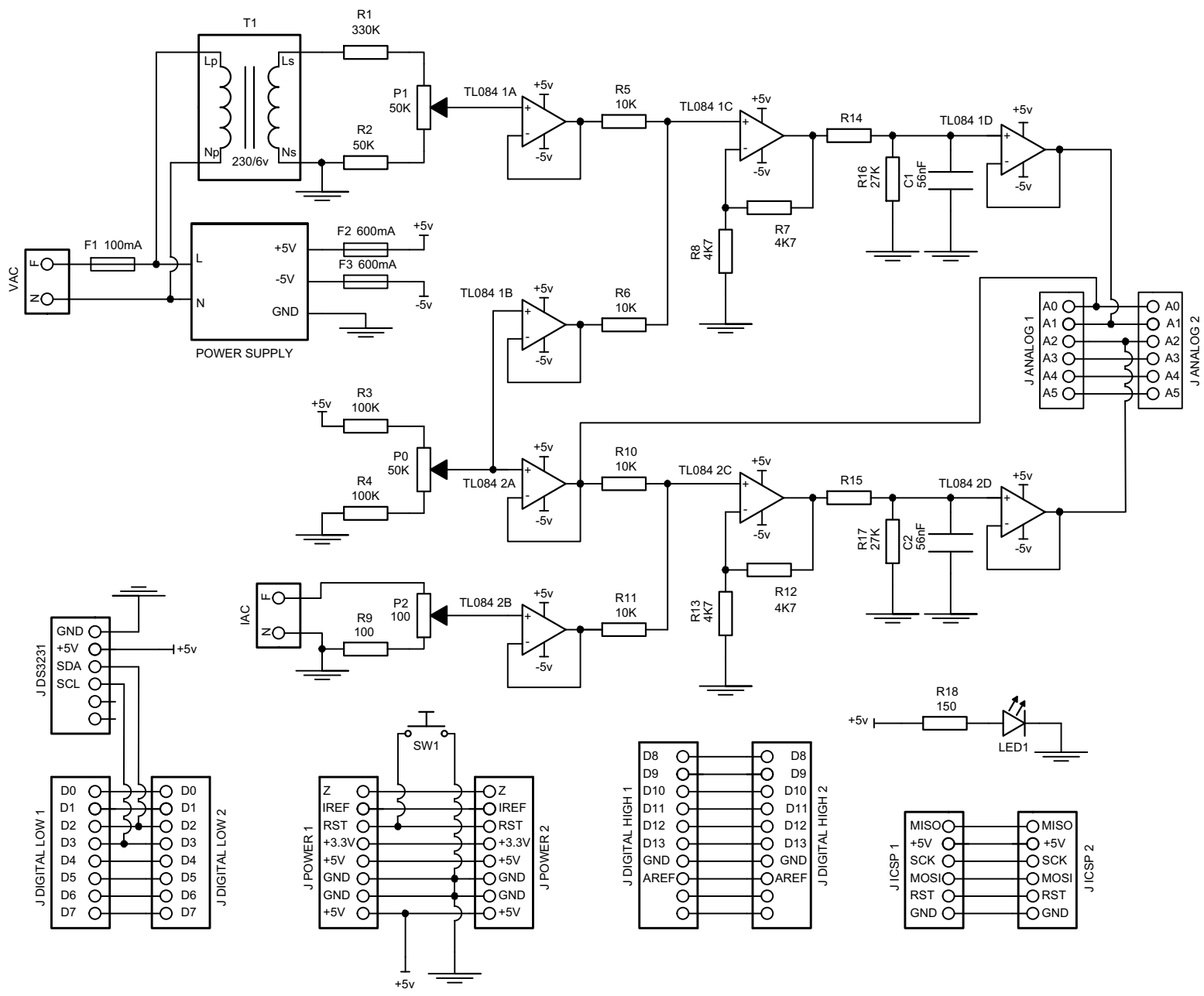
A.8.4. Transformador Block 230/6V 1,2VA


A.8.5. RECOM RAC06-05DC 6W

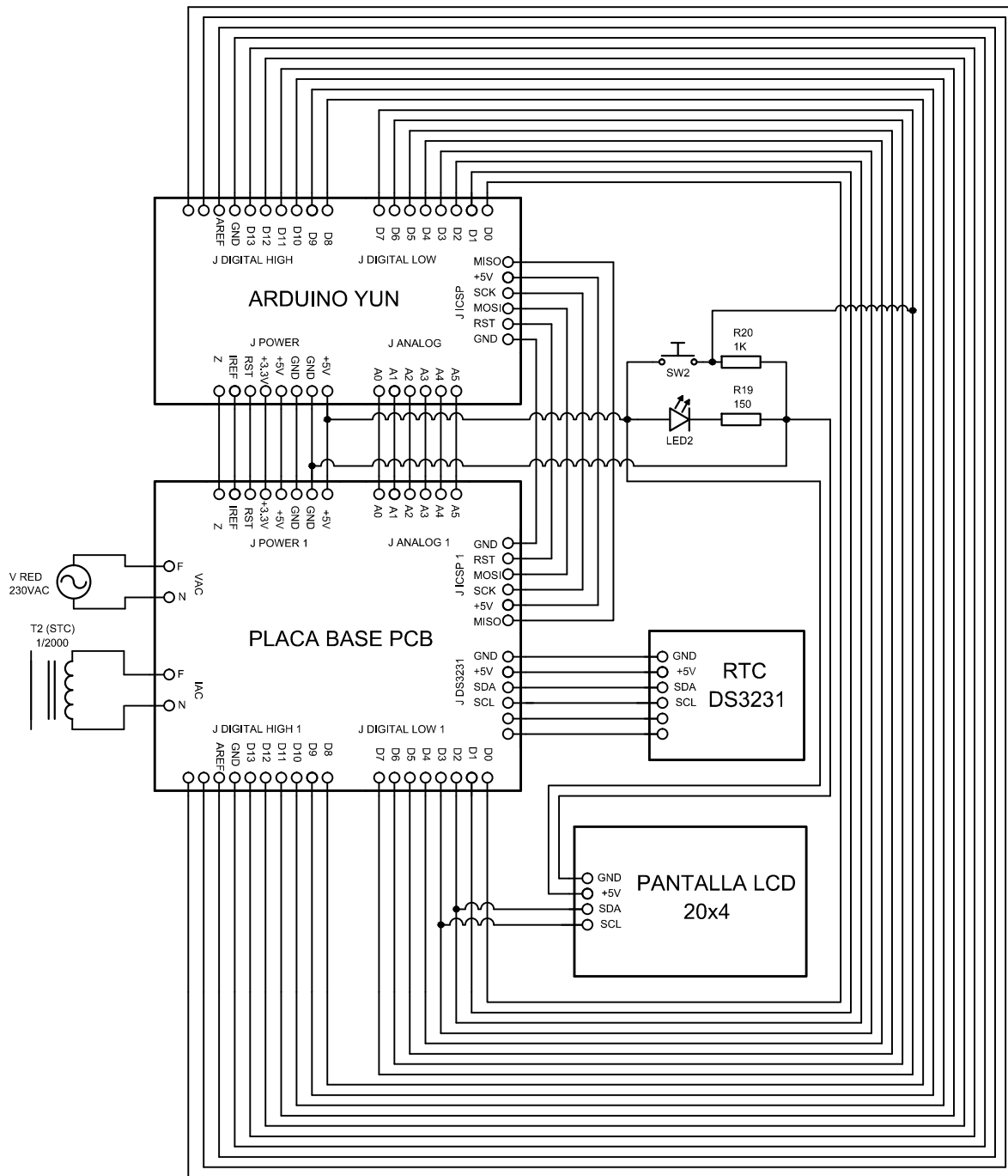
A.8.6. Pinza STC-013

A.8.7. Portafusibles 5x20 OGN

A.8.8. TEXAS Operational Amplifier TL08XX JFET



| | | |
|---|---|-----------|
|  Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa | DEPARTAMENTO DE ING. DEPARTAMENTO DE INGENIERIA ELECTRONICA | |
| | E.T.S.I.I.T. GRADO INGENIERIA ELECTRONICA | |
| IMPLEMENTACION DE UN ANALIZADOR DE REDES CON ARDUINO | | |
| PROYECTO: BARBACHANO CIRION, ANGEL | | |
| REALIZADO: BARBACHANO CIRION, ANGEL | | |
| FIRMA: | | |
| PLANO: | ESCALA: | N° PLANO: |
| 20/04/2018 | 20/04/2018 | A.7.1. |



| | | | | |
|---|--|---|---------------------|--|
|  Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa | E.T.S.I.I.T. | DEPARTAMENTO: DEPARTAMENTO DE ING. ELECTRICA ELECTRONICA | | |
| | GRADO INGENIERIA ELECTRICA ELECTRONICA | REALIZADO: BARBACHANO CIRION, ANGEL | | |
| PROYECTO: IMPLEMENTACION DE UN ANALIZADOR DE REDES CON ARDUINO | | FIRMA: | | |
| PLANO: DIAGRAMA ESQUEMÁTICO | FECHA: 20/04/2018 | ESCALA: | Nº PLANO: A.7.2. | |



Pinza amperimétrica medidora de potencia PCE-PCM1

pinza amperimetrica con memoria interna, puerto USB y software / medición de potencia para 3 fases / medidor de energía / TRMS

La pinza amperimétrica medidora de potencia PCE-PCM1 es un medidor de uso múltiple para determinar la potencia absorbida así como el consumo de energía. Además de la medición de corriente alterna y tensión, esta pinza amperimétrica medidora de potencia puede determinar la potencia absorbida actual en redes monofásicas o trifásicas, así como el consumo de energía (kWh) en redes monofásicas. También se le indica el ángulo de desfase y el factor de potencia en la gran pantalla con iluminación de fondo. Esta pinza amperimétrica medidora de potencia le permite así una valoración rápida y sencilla de la potencia absorbida y del consumo de energía de máquinas e instalaciones. Las dimensiones compactas y la carcasa robusta le permiten un uso móvil al técnico o ingeniero in situ. En la memoria interna de la pinza amperimétrica medidora de potencia puede almacenar valores individuales que posteriormente puede recuperar para su análisis. También integra un puerto USB para la transmisión en tiempo real de los valores actuales al PC, donde se le presentan como tabla o gráfico. [Aquí](#) encontrará una visión general de todas las pinzas amperimétricas de PCE Instruments. Si tiene alguna pregunta sobre la pinza amperimétrica medidora de potencia, consulte la siguiente ficha técnica o póngase en contacto con nosotros en el número de teléfono 902 044 604 para España, para Latinoamérica e internacional +34 967 543 695 o en el número +56 2 29381530 para Chile. Nuestros técnicos e ingenieros le asesorarán con mucho gusto sobre este pinza amperimetrica medidora de potencia y sobre cualquier producto de nuestros [sistemas de regulación y control](#), [medidores](#) o [balanzas](#) PCE Ibérica S.L.



Pinza amperimétrica medidora de potencia PCE-PCM1

PCE Ibérica S.L. | Mayor 53 – Bajo | 02500 Tobarra (Albacete)
Tel: 902 044 604 | Fax: +34 967 543 542 | Email: info@pce-iberica.es
<http://www.pce-iberica.es/>



- Medición de potencia:
 - Potencia activa (kW)
 - Potencia aparente (kVA)
 - Potencia reactiva (kVAR)
 - Factor de potencia ($\cos \varphi$)
 - Ángulo de fase
 - Contador de energía activa (kWh)
- Selección de rango automático
- Memoria de valores
- Tensión (TRMS)
- Corriente (TRMS)
- Frecuencia
- Registrador de datos de 99 valores visualizables en pantalla
- Valores mínimo, máximo y Data Hold

Especificaciones técnicas

| | |
|---------------------------------------|--|
| Tensión (AC/TRMS, 50 ... 200 Hz) | 100 / 400 / 750 V; $\pm(1,2 \% + 5$ dígitos) |
| Corriente (AC/TRMS, 50 ... 200 Hz) | 40 / 100 / 400 / 1000 A / $\pm (2,0 \% + 5$ dígitos) |
| Medición de potencia activa | 4 ... 750 kW, $\pm (3,0 \% + 5$ dígitos) |
| Medición de potencia aparente | 4 ... 750 kVA, $\pm (3,0 \% + 5$ dígitos) |
| Medición de potencia reactiva | 4 ... 750 kVAR, $\pm (3,0 \% + 5$ dígitos) |
| Energía activa (contador kWh) | 1 ... 9999 kWh, $\pm (3,0 \% + 2$ dígitos) |
| Factor de potencia ($\cos \varphi$) | 0,3 ... 1 inductivo y capacitivo / $\pm (0,02 \% + 2$ dígitos) |
| a > 10 A y >45 V | |
| Ángulo de fase | 0 ... 90°, $\pm 2,0^\circ$ |
| Frecuencia | 50 ... 200 Hz; $\pm (0,5 \% + 5$ dígitos) |
| Memoria interna | 99 valores de medición |
| Pantalla | 9999 valores con gráfico de barra e iluminación de fondo |
| Alimentación | batería de 9 V |
| Dimensiones (ancho x profundo x alto) | 105 x 47 x 294 mm |
| Peso | 495 g |
| Categoría de sobretensión | CAT IV 600 V / CAT III 1000 V |



Contenido del envío

- 1 x pinza amperimétrica medidora de potencia PCE-PCM1,
 - 4 x conectores de prueba,
 - 4 x pinzas de cocodrilo,
 - 2 x pinzas de medición,
 - 1 x cable interfaz USB,
 - 1 x software para PC,
 - 1 x batería de 9 V,
 - 1 x bolsa de transporte
- instrucciones de uso



Accesorios opcionales disponibles

- Calibración / certificación ISO

La calibración de laboratorio ISO incluido un certificado para que mantenga su control interno de calidad ISO. En el certificado aparece también el nombre de la persona que lo pide o los datos de su empresa.



Aquí encuentra usted una visión general de [todos los medidores](#) que le ofrece PCE Instruments.



engage + STXtra sensor

Submetering and Microgeneration

efergy®



Instantly see the cost of using energy in your home



View energy use on our web portal or smartphone app



Usage history per day, week, month or year



Monitor multiple circuits and compare their consumption



See when and how you consumed electricity



Download your daily, weekly, monthly or average data

engage + STXtra sensor

Submetering and Microgeneration

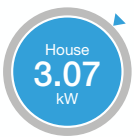


DESCRIPTION

With the **engage + STXtra sensor** you will be able to view your home electricity consumption and an additional electric circuit in real-time from anywhere and at any time.



KEY FEATURES



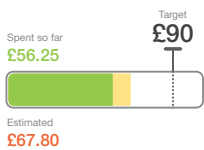
Real-time usage

See your home and additional electric circuit consumption in real-time.



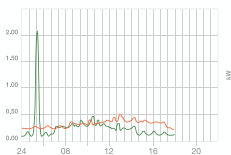
Budget

Set your monthly budget and see how close you are to meeting it.



Cost so far

View what you have spent so far today, this month and from a date of your choice.



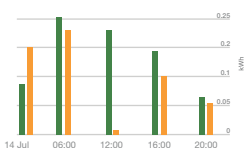
Compare

Check through hours, days or months to compare your energy consumption.



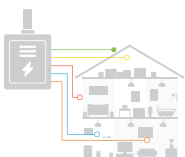
Download report

Download your daily, weekly, and monthly energy data report onto your PC/Mac.



Usage history

Check the amount of energy consumed per day, week, month or year.

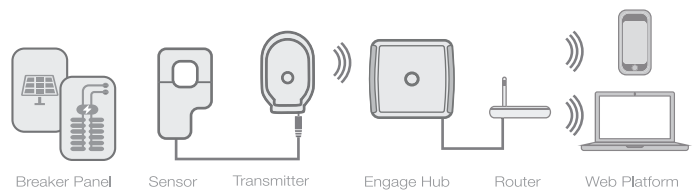


Add-on options

Add more sensors to monitor individual circuits of the home or to monitor micro-generation.

HOW IT WORKS

One sensor is clipped onto the live feed cable at your breaker panel and another on the cable coming from the solar inverter. These sensors are then connected to the efergy transmitters. The transmitters wirelessly send real time data to the hub. The engage hub is connected to your internet router via Ethernet cable. The web platform or app will then show you how much energy you are using in the same units that are on your bill.



TECHNICAL DETAILS

| | |
|----------------|------------|
| Model Name | engage hub |
| Model Number: | HH-2.0 |
| Frequency: | 433.5MHz |
| Dimensions | 84x80x27mm |
| Certifications | |

AVAILABLE FOR



IN THE BOX

- 1 x engage hub | 2 x Transmitter | 2 x Sensor | 1 x Cable Ethernet | 1 x AC/DC Adaptor | 1 x Manual

La MIRUBOX es un medidor eléctrico para líneas monofásicas. La caja permite conectar hasta 3 sensores de corriente de núcleo partido de hasta 100A. La MIRUBOX es una solución orientada al segmento doméstico y pequeño comercio, que permite conocer desde un mismo cuadro eléctrico el consumo individualizado de los principales electrodomésticos o aparatos conectados como la iluminación, el Aire Acondicionado o la producción de las placas fotovoltaicas.

El dispositivo utiliza el circuito de medida de tensión para auto-alimentarse, cuenta con tres sensores de corriente que permiten determinar el consumo general y dos circuitos auxiliares; a través de conexión WiFi envía y registra los datos en la nube para su posterior análisis y gestión mediante la plataforma de mirubee. La conexión a la red WIFI local en la MIRUBOX se realiza a través de la APP gratuita (iOS / Android). La plataforma WEB permite visualizar y analizar la información generada por los dispositivos.

Características técnicas

| | | |
|--------------------------------------|---|---|
| Circuito de alimentación | Tipo conexión | Monofásico |
| | Rango tensión | 85...265 Vac |
| | Frecuencia | 50 - 60 Hz |
| | Consumo | 20 mA |
| Circuito de medida | Tensión nominal | 85...265 Vp-n |
| | Corriente nominal | 100 A |
| Clase de precisión | Tensión | 1% |
| | Corriente | 1% |
| Comunicaciones | Tipo | Wi-Fi (IEEE 802.11) |
| | Protocolo | HTTP, Modbus/TCP, XML |
| | Rango frecuencia | 2,405 - 2,480 GHz |
| | Encriptación | AES128 |
| | Certificación | FCC(USA), IC(CANADA), ETSI(EUROPA) |
| Características constructivas | Material envolvente | Auto-extinguible UNE 21031 90 °C |
| | Peso | 180 g |
| | Grado de protección | IP 20 |
| | Temperatura de trabajo | -25...+45 °C |
| Condiciones ambientales | Humedad (sin condensación) | 5 ... 95% (sin condensación) |
| | Altitud máxima | 2000 m |
| | Seguridad | IEC 61010-1:2001 Protección al choque eléctrico Por doble aislamiento Clase II |
| Normas | UNE-EN 61010-2-030:2011, UNE-EN 61326-1:2006, EN 301 489-17 V2.2.1 | |





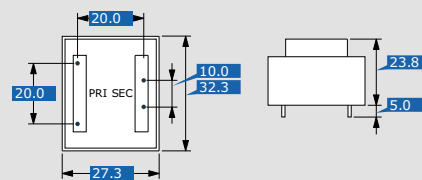
VB 1,2/1/6

Short circuit proof PCB transformer

- Maximum power from minimum size
- Vacuum resin moulded (self extinguishing to UL 94 V-0)

Short circuit proof safety isolating transformer to IEC 61558-2-6, DIN EN 61558-2-6, VDE 0570 part 2-6

*Transformer to IEC 61558-2-1, DIN EN 61558-2-1, VDE 0570 part 2-1 (without VDE-Approval)



Dimensions in mm



VB 1,2/1/6

Short circuit proof PCB transformer



| | |
|--------------------------------|---------------------------------|
| Input data | |
| Input voltage | 230 V |
| Frequency range | 50 - 60 Hz |
| No-load loss (typ.) | 1.0 W |
| Output data | |
| Output voltage | 6 V |
| No-load voltage (ca. x factor) | 1.31 |
| Efficiency | 57 % |
| Power | 1.20 VA |
| Standards and security | |
| Approvals | C-UL-US, ENEC 10 (VDE) |
| Short circuit strength | inherently short-circuit proof |
| Protection index | IP 00 |
| Safety class (prepared) | II |
| Ambient temperature max. | 70 °C |
| Class of Insulation System | VDE=B, UL=105 |
| Test voltage | 5000 V, 50 Hz |
| Mechanical data | |
| Type | encapsulated |
| Core type | EI 30/12,5 |
| Terminals | Pins for printed circuit boards |
| Pin (∅) | 0.8 mm |
| Weight | 0.08 kg |
| Order numbers | |
| Order Number | VB 1,2/1/6 |

BLOCK Transformatoren-Elektronik GmbH

Max-Planck-Straße 36-46

27283 Verden

Germany

Phone +49 4231 678-0

Fax +49 4231 678-177

Features

Regulated Converters

- Very Compact AC-DC Power Supply
- 6 Watt PCB Mount Package
- Single and Dual Output Versions
- Universal Input Voltage Range
- Class II Power Supply with 3kVAC Isolation
- Low Output Ripple and Noise
- Short Circuit, Overload, Overtemp Protected
- Low Standby Power Consumption
- UL/cUL Certified, CE Marked

Selection Guide

| Part Number | Input Range (VAC) | Output Voltage (VDC) | Output Current (mA) | Efficiency** (%) | Max Capacitive Load** |
|--------------|-------------------|----------------------|---------------------|------------------|-----------------------|
| RAC06-3.3SC* | 80-264 | 3.3 | 1500 | 70 | 2500µF |
| RAC06-05SC* | 80-264 | 5 | 1200 | 75 | 2200µF |
| RAC06-09SC* | 80-264 | 9 | 667 | 77 | 1200µF |
| RAC06-12SC* | 80-264 | 12 | 500 | 78 | 680µF |
| RAC06-15SC* | 80-264 | 15 | 400 | 79 | 450µF |
| RAC06-24SC* | 80-264 | 24 | 250 | 79 | 220µF |
| RAC06-05DC* | 80-264 | ±5 | ±600 | 75 | ±1000µF |
| RAC06-12DC* | 80-264 | ±12 | ±250 | 78 | ±330µF |
| RAC06-15DC* | 80-264 | ±15 | ±200 | 79 | ±220µF |

* add suffix /W for wired version

** measured @ 115VAC/60Hz with full load

Specifications

(Ta=25°C and after warm up time, measured at 115VAC/60Hz and 230VAC/50Hz)

| | | |
|--|---|-------------------|
| Input Voltage Range (with derating) | 80-264VAC or 115-370VDC | |
| Rated Power | 6 Watts max. | |
| Input Frequency Range (for AC Input) | 47-440Hz | |
| Input Current (full load) | 115/230VAC | 120mA / 78mA typ. |
| No Load Power Consumption | 250mW max. | |
| Inrush Current (<2ms) | 115/230VAC | 30A/60A max. |
| Holdup Time (Full Load) | 115VAC | 10ms typ. |
| Leakage Current | 0.85mA max. | |
| Output Voltage Accuracy (Full load) | ±2% max. | |
| Line Voltage Regulation (low line, high line at full load) | ±0.3% typ. | |
| Load Voltage Regulation (5% to 100% full load) | ±0.5% typ. | |
| Output Ripple and Noise (20MHz BW) | 3.3V Output | 120mVp-p max. |
| (with 100nF across each output) | All Others | 150mVp-p max. |
| Operating Frequency (full load) | 132kHz typ. | |
| Minimum Load = 0% | specifications valid for 5% minimum load only | |
| RMS Isolation Voltage (input to output) | 3kVAC / 1minute | |
| Isolation Resistance (input to output) | 1 GΩ min. | |
| Isolation Capacitance (input to output) | 1000pF typ. | |
| Short Circuit Protection | Hiccup, Automatic Restart | |
| Overload Protection | 115VAC/230VAC | 115% ~ 145% |
| Output Overvoltage Protection | Zener Diode Clamp | |
| (set at 4.8V - 5.4V for 3.3V and others 110% - 135% of its Nominal Output Voltage) | | |
| Operating Temperature Range | (free air convection, without derating) | -25°C to +55°C |
| | (free air convection, with derating) | -25°C to +75°C |
| Storage Temperature Range | -40°C to +100°C | |

POWERLINE

AC/DC-Converter

with 3 year Warranty

RECOM

6 Watt Single/Dual Output



E224736

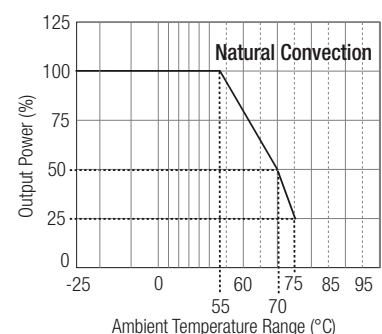
EN-60950-1 Certified
UL-60950-1 Certified

RAC06-C

Description

Compact, low cost, high efficiency, universal input switching AC/DC power module for PCB or wired mounting with single or dual outputs. CE marked and UL/cUL Certified.

Derating Graph (Ambient Temperature)



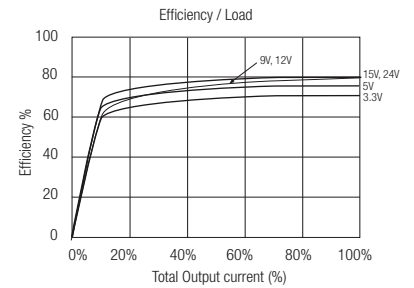
Please Read Application Notes

www.recom-electronic.com

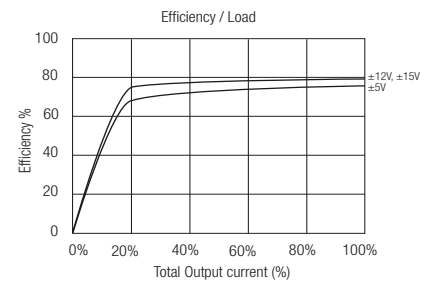
Typical Characteristics

| | | |
|---------------------------------|--|--------------------------------------|
| Recommended External Input Fuse | 1.5A Slow Blow | |
| Humidity | 95% RH max. | |
| Case Material | Plastic Case filled with Epoxy (UL94V-0) | |
| Package Weight | Single/Dual | 35g/38g |
| Packing Quantity | Single Output | 10 pcs |
| | Dual Output | 9 pcs |
| | /W Version | 1 pc |
| Emissions | CE | EN 55022: 2006 + A1: 2007 / Class B |
| | EMC | EN 55024:1998 + A1:2001 + A2:2003 |
| | Harmonics | EN 61000-3-2:2006 / Class A |
| | Flicker | EN 61000-3-3:1995 + A1:2001 +A2:2005 |
| Immunity | ESD | IEC 61000-4-2 / Criterion B |
| | RS | IEC 61000-4-3 / Criterion A |
| | EFT | IEC 61000-4-4 / Criterion B |
| | Surge | IEC 61000-4-5 / Criterion B |
| | CS | IEC 61000-4-6 / Criterion A |
| | PMF | IEC 61000-4-8 / Criterion A |
| | Voltage Variations | IEC 61000-4-11 / Criteria B + C |
| | MTBF (+25°C) | using MIL-HDBK-217F |
| (+65°C) | | >200 x 10 ³ hours |
| Certifications: | | |
| UL General Safety | Report: E224736 | UL-60950-1, 2nd Edition |
| cUL | Report: E224736 | C22.2 No. 60950-1-07, 2nd Edition |
| EN General Safety | Report: SPCLVD1105038 | EN-60950-1, 2nd Edition |
| CE | | EN55022 Class B |

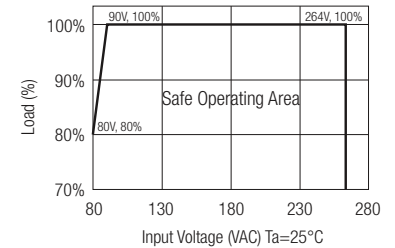
Single Output



Dual Output

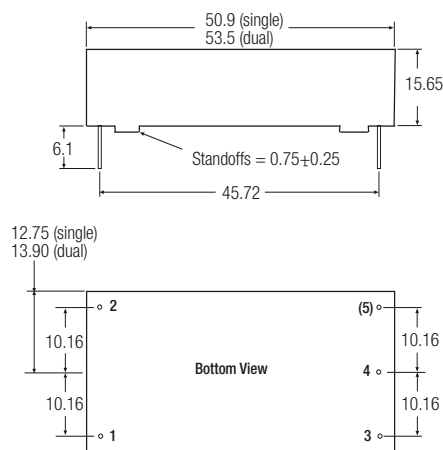


Input Voltage Derating Graph

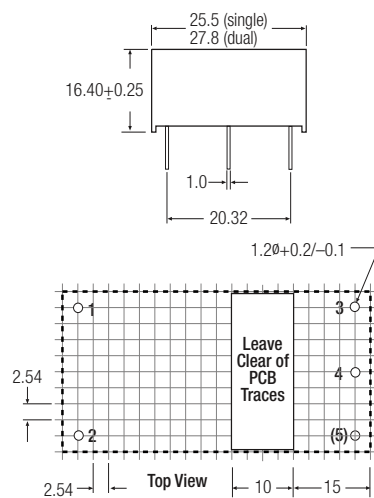
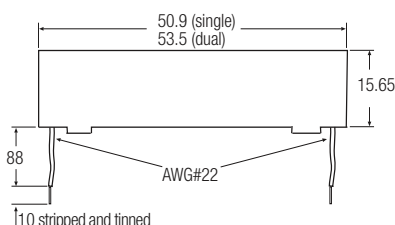


Package Style and Pinning

PCB Mounting Version (Standard)



Wired Version (Suffix /W)



Recommended Footprint Details



Pin Connections

| Pin # | Single Out | Dual Out |
|-----------|------------|------------|
| 1 | VAC in (N) | VAC in (N) |
| 2 | VAC in (L) | VAC in (L) |
| 3 | +VDC out | +VDC out |
| 4 | -VDC out | Com |
| 5 | No Pin | -VDC out |
| Tolerance | ± 0.5 mm | |

Wire Connections

| Wire | AWG#22 | Single Out | Dual Out |
|--------|--------|------------|------------|
| Blue | 1015 | VAC in (N) | VAC in (N) |
| Brown | 1015 | VAC in (L) | VAC in (L) |
| Red | 1007 | +VDC out | +VDC out |
| Black | 1007 | -VDC out | Com |
| Orange | 1007 | | -VDC out |

SPECIFICATION

Customer Title : XiDi Technology

Product Name: Split-core current

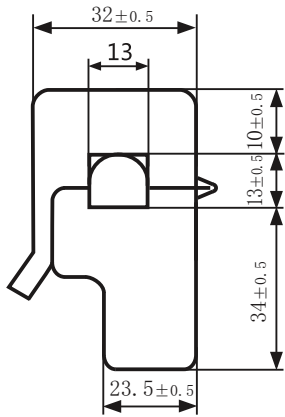
Manufacture Model : SCT-013-000

transformer

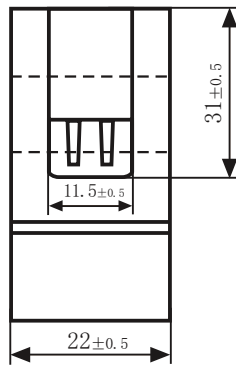
Charateristics: open size:13mm×13mm
 1m leading wire
 Core material:Ferrite
 Fire resistance property:in accordance with
 UL 94-V0
 Dielectric strength: 1000V AC/1min 5mA
 (between shell and output)



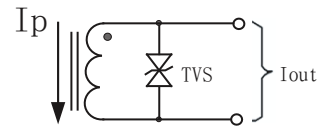
Outline size diagram:(in mm)



Front View



Side View



TVS: Transient-voltage Suppressor

Current output type

Schematic Diagram

Typical table of technical parameters:

| | | | |
|---------------|------------------|------------------|--|
| input current | output voltage | non-linearity | build-in sampling resistance (R _L) |
| 0-100A | 0-50mV | ±3% | Ω |
| turn ratio | resistance grade | work temperature | dielectric strength(between shell and output) |
| 100A:0.05A | Grade B | -25°C~+70°C | 1000V AC/1min 5mA |

Customer Sign:

Beijing YaoHuadechang Electronic Co.,Ltd

Phone: 0355-7929499-803

Cell: 13693334514

Contact Name: Engineer Chen

Approve Sign:Chenjianping

2011-7-26

Fuseholder Open Design, 5 x 20 mm, THT, var. Covers, IEC 60335-1



500 VAC · 4 W/16 A (VDE) · 500 V · 16 A (UL/CSA)

See below:

[Approvals and Compliances](#)

Description

- For appliances in unattended use
- Glow wire tests acc. to IEC 60695-2-12 and -13
- Low profile fuse block with insulation walls


Applications

- Household appliances

Weblinks

[pdf datasheet](#), [html-datasheet](#), [General Product Information](#), [Packaging details](#), [Distributor-Stock-Check](#), [Accessories](#), [Detailed request for product](#), [Microsite](#)

Technical Data

| | |
|------------------------------|---|
| Shock-Safe Category | PC1 |
| Fuse-Link | 5 x 20mm |
| Mounting | PCB |
| Attachment | Screw, rivet or THT |
| Terminal | Solder THT |
| Rated Voltage | 500VAC (VDE), 500V (UL/CSA) |
| Rated current | 10/16A (VDE), 16A (UL/CSA) |
| Rated Power Acceptance IEC | 4 W / 16 A @ Ta 23°C 4 W / 10A with black cover 2.5 W / 10 A with transparent cover, see derating curves |
| Degree of Protection | IP 20 (with cover) |
| Protection Class | Suitable for appliances with protection class I acc. to IEC 61140 |
| Admissible Ambient Air Temp. | -40°C to 85°C |
| Climatic Category | 40/085/21 acc. to IEC 60068-1 |
| Material: Socket | see variants |
| Material: Cover | Thermoplastic UL 94V-0 |
| Material: Terminals | Tin-Plated Copper Alloy |
| Unit Weight | 1.7 g |
| Storage Conditions | 0°C to 60°C, max. 70% r.h. |
| Product Marking |  , Type, Rated Voltage, Rated current, Power Rating, Dielectric strength |

| | |
|------------------------------|--|
| Soldering Methods | Wave Soldering Profile |
| Solderability | 245°C / 3 sec acc. to IEC 60068-2-20, Test Ta, method 1 |
| Resistance to Soldering Heat | 260°C / 10 sec acc. to IEC 60068-2-20, Test Tb, method 1 |
| Contact Resistance | ≤ 10mΩ at 20 mV acc. to IEC 60127-6 |
| Dielectric Strength | > 3kV between Life parts (50Hz; 1 min) |
| Impulse Withstand Voltage | > 4kV between Life parts |
| Insulation Resistance | ≥ 10MΩ (500VDC; 1 min) |
| Overvoltage Category | III acc. to IEC 60664-1 |
| Pollution Degree | 3 acc. to IEC 60664-1 |




Approvals and Compliances

Detailed information on product approvals, code requirements, usage instructions and detailed test conditions can be looked up in [Details about Approvals](#)

Approvals




The approval mark is used by the testing authorities to certify compliance with the safety requirements placed on electronic products.

Approval Reference Type: OGN

| Approval Logo | Certificates | Certification Body | Description |
|---|-------------------------------|--------------------|----------------------------------|
|  | VDE Approvals | VDE | VDE Certificate Number: 40001042 |
|  | VDE Approvals | VDE | VDE Certificate Number: 40045404 |
|  | UL Approvals | UL | UL File Number: E39328 |



Product standards

Product standards that are referenced

| Organization | Design | Standard | Description |
|--|-----------------------|----------------------|--|
|  | Designed according to | IEC 60127-6 | Miniature fuses. Part 6. Fuse-holders for miniature fuse-links |
|  | Designed according to | UL 4248-1 | Fuseholder general requirements |
|  | Designed according to | CSA C22.2 no. 4248.1 | Fuseholder general requirements |





Application standards

Application standards where the product can be used

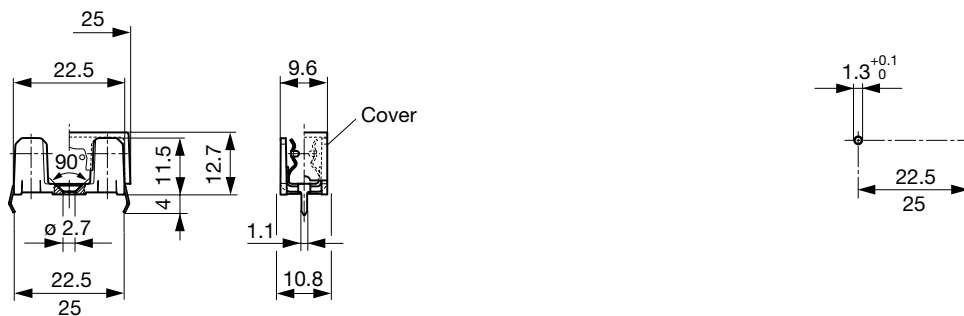
| Organization | Design | Standard | Description |
|--|--------------------------------|--------------|---|
|  | Designed for applications acc. | IEC/UL 60950 | IEC 60950-1 includes the basic requirements for the safety of information technology equipment. |
|  | Designed for applications acc. | IEC 60335-1 | Safety of electrical appliances for household and similar purposes. Meets the requirements for appliances in unattended use. This includes the enhanced requirements of glow wire tests acc. to IEC 60695-2-12 and -13. |

Compliances

The product complies with following Guide Lines

| Identification | Details | Initiator | Description |
|--|--|-------------|---|
|  | CE declaration of conformity | SCHURTER AG | The CE marking declares that the product complies with the applicable requirements laid down in the harmonisation of Community legislation on its affixing in accordance with EU Regulation 765/2008. |
|  | RoHS | SCHURTER AG | EU Directive RoHS 2011/65/EU |
|  | China RoHS | SCHURTER AG | The law SJ / T 11363-2006 (China RoHS) has been in force since 1 March 2007. It is similar to the EU directive RoHS. |
|  | REACH | SCHURTER AG | On 1 June 2007, Regulation (EC) No 1907/2006 on the Registration, Evaluation, Authorization and Restriction of Chemicals 1 (abbreviated as "REACH") entered into force. |

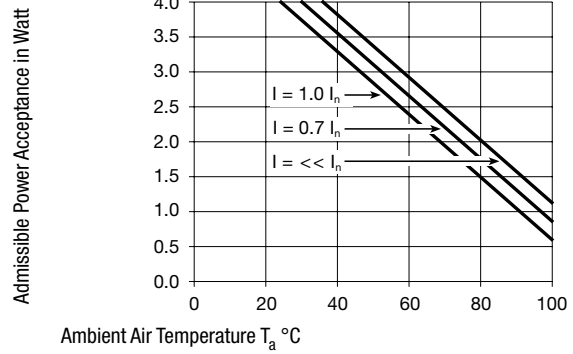
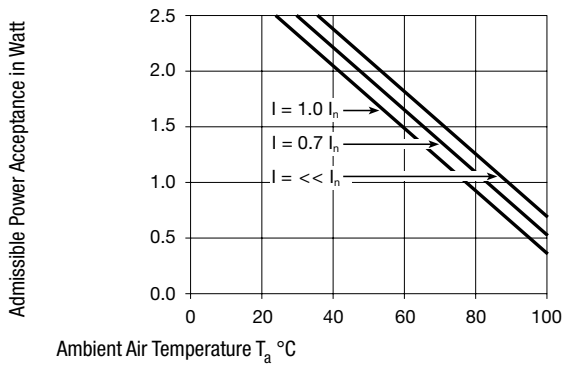
Dimension [mm]



Drilling diagram

Derating Curves

With Transparent Cover



All Variants

| Holder | Pin distance | Colour Holder | Material | Remark | Order Number |
|--------|--------------|---------------|---------------|--------|--------------|
| ● | 22.5 | black | Thermoplastic | - | 0031.8201 |
| ● | 22.5 | black | Thermoplastic | - | 0031.8201.G |
| ● | 25 | black | Thermoplastic | - | 0031.8211 |
| ● | 25 | black | Thermoplastic | - | 0031.8211.G |

Most Popular.

Availability for all products can be searched real-time: <https://www.schurter.com/en/Stock-Check/Stock-Check-SCHURTER>

The fuseholder is suitable for use in equipment according to IEC 60335-1.

| Packaging Unit | xxxx.xxxx xxxx.xxxx.G | Bulk 128 x 91 x 60 mm (100 pcs.) Bulk 210 x 200 x 100 mm (1000 pcs.) |
|----------------|--------------------------|---|
|----------------|--------------------------|---|

Accessories

Description



Covers for OGN, OGN-SMD
 Cover for Holder OGN, OGN-SMD

| | |
|--|--|
| | |
| | |

...



Adapter to OGN, OGN-SMD
 Fuse Carriage with Handle for OGN, OGN-SMD

| | |
|-------|-----------|
| black | 0853.1201 |
| white | 0853.1202 |

TL08xx JFET-Input Operational Amplifiers

1 Features

- Low Power Consumption: 1.4 mA/ch Typical
- Wide Common-Mode and Differential Voltage Ranges
- Low Input Bias Current: 30 pA Typical
- Low Input Offset Current: 5 pA Typical
- Output Short-Circuit Protection
- Low Total Harmonic Distortion: 0.003% Typical
- High Input Impedance: JFET Input Stage
- Latch-Up-Free Operation
- High Slew Rate: 13 V/μs Typical
- Common-Mode Input Voltage Range Includes V_{CC+}

2 Applications

- Tablets
- White goods
- Personal electronics
- Computers

3 Description

The TL08xx JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient.

Device Information⁽¹⁾

| PART NUMBER | PACKAGE | BODY SIZE (NOM) |
|-------------|------------|--------------------|
| TL084xD | SOIC (14) | 8.65 mm × 3.91 mm |
| TL08xxFK | LCCC (20) | 8.89 mm × 8.89 mm |
| TL084xJ | CDIP (14) | 19.56 mm × 6.92 mm |
| TL084xN | PDIP (14) | 19.3 mm × 6.35 mm |
| TL084xNS | SO (14) | 10.3 mm × 5.3 mm |
| TL084xPW | TSSOP (14) | 5.0 mm × 4.4 mm |

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Schematic Symbol

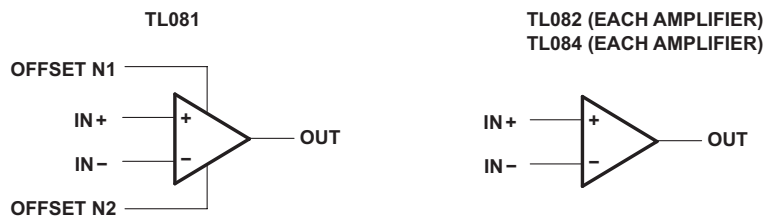


Table of Contents

| | | | |
|--|-----------|--|-----------|
| 1 Features | 1 | 8.2 Functional Block Diagram | 14 |
| 2 Applications | 1 | 8.3 Feature Description | 14 |
| 3 Description | 1 | 8.4 Device Functional Modes | 14 |
| 4 Revision History | 2 | 9 Applications and Implementation | 15 |
| 5 Pin Configuration and Functions | 3 | 9.1 Application Information | 15 |
| 6 Specifications | 5 | 9.2 Typical Applications | 15 |
| 6.1 Absolute Maximum Ratings | 5 | 9.3 System Examples | 16 |
| 6.2 ESD Ratings | 5 | 10 Power Supply Recommendations | 18 |
| 6.3 Recommended Operating Conditions | 5 | 11 Layout | 18 |
| 6.4 Thermal Information | 6 | 11.1 Layout Guidelines | 18 |
| 6.5 Electrical Characteristics for TL08xC, TL08xxC, and TL08xI | 6 | 11.2 Layout Examples | 19 |
| 6.6 Electrical Characteristics for TL08xM and TL084x | 7 | 12 Device and Documentation Support | 20 |
| 6.7 Operating Characteristics | 7 | 12.1 Documentation Support | 20 |
| 6.8 Dissipation Rating Table | 8 | 12.2 Related Links | 20 |
| 6.9 Typical Characteristics | 9 | 12.3 Community Resources | 20 |
| 7 Parameter Measurement Information | 13 | 12.4 Trademarks | 20 |
| 8 Detailed Description | 14 | 12.5 Electrostatic Discharge Caution | 20 |
| 8.1 Overview | 14 | 12.6 Glossary | 20 |
| | | 13 Mechanical, Packaging, and Orderable Information | 20 |

4 Revision History

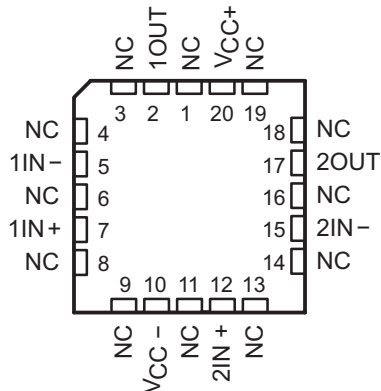
NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

| Changes from Revision H (January 2014) to Revision I | Page |
|---|------|
| • Added <i>Pin Configuration and Functions</i> section, <i>Storage Conditions</i> table, <i>ESD Ratings</i> table, <i>Feature Description</i> section, <i>Device Functional Modes</i> , <i>Application and Implementation</i> section, <i>Power Supply Recommendations</i> section, <i>Layout</i> section, <i>Device and Documentation Support</i> section, and <i>Mechanical, Packaging, and Orderable Information</i> section | 1 |
| • Added <i>Applications</i> | 1 |
| • Moved <i>Typical Characteristics</i> into <i>Specifications</i> section. | 9 |

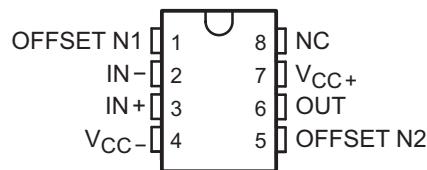
| Changes from Revision G (September 2004) to Revision H | Page |
|---|------|
| • Updated document to new TI data sheet format - no specification changes. | 1 |
| • Deleted <i>Ordering Information</i> table. | 1 |

5 Pin Configuration and Functions

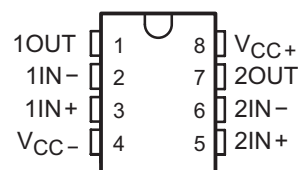
TL082 FK Package
20-Pin LCCC
Top View



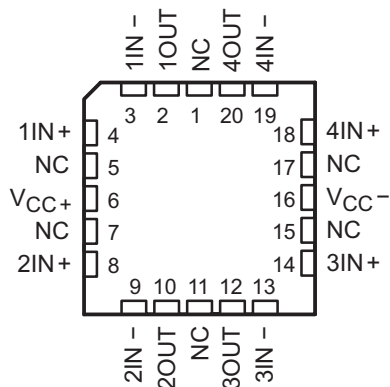
TL081 and TL081x D, P, and PS Package
8-Pin SOIC, PDIP, and SO
Top View



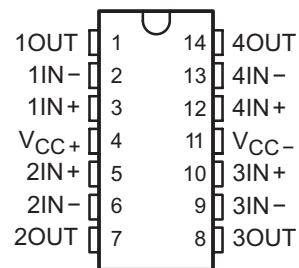
TL082 and TL082x D, JG, P, PS and PW Package
8-Pin SOIC, CDIP, PDIP, SO, and TSSOP
Top View



TL084 FK Package
20-Pin LCCC
Top View



TL084 and TL084x D, J, N, NS and PW Package
14-Pin SOIC, CDIP, PDIP, SO, and TSSOP
Top View



Pin Functions

| NAME | PIN | | | | | I/O | DESCRIPTION |
|------|----------------|-----------------------------|-------|-----------------------------|----|-----|----------------|
| | TL081 | TL082 | TL084 | | | | |
| | SOIC, PDIP, SO | SOIC, CDIP, PDIP, SO, TSSOP | LCCC | SOIC, CDIP, PDIP, SO, TSSOP | | | |
| 1IN- | — | 2 | 5 | 2 | 3 | I | Negative input |
| 1IN+ | — | 3 | 7 | 3 | 4 | I | Positive input |
| 1OUT | — | 1 | 2 | 1 | 2 | O | Output |
| 2IN- | — | 6 | 15 | 6 | 9 | I | Negative input |
| 2IN+ | — | 5 | 12 | 5 | 8 | I | Positive input |
| 2OUT | — | 7 | 17 | 7 | 10 | O | Output |
| 3IN- | — | — | — | 9 | 13 | I | Negative input |
| 3IN+ | — | — | — | 10 | 14 | I | Positive input |
| 3OUT | — | — | — | 8 | 12 | O | Output |
| 4IN- | — | — | — | 13 | 19 | I | Negative input |
| 4IN+ | — | — | — | 12 | 18 | I | Positive input |
| 4OUT | — | — | — | 14 | 20 | O | Output |

Pin Functions (continued)

| NAME | PIN | | | | | I/O | DESCRIPTION |
|------------------|----------------|-----------------------------|------|-----------------------------|------|-----|-------------------------|
| | TL081 | TL082 | | TL084 | | | |
| | SOIC, PDIP, SO | SOIC, CDIP, PDIP, SO, TSSOP | LCCC | SOIC, CDIP, PDIP, SO, TSSOP | LCCC | | |
| IN- | 2 | — | — | — | — | I | Negative input |
| IN+ | 3 | — | — | — | — | I | Positive input |
| NC | 8 | — | 1 | — | — | — | Do not connect |
| | | | 3 | | | | |
| | | | 4 | | | | |
| | | | 6 | | | | |
| | | | 8 | | | | |
| | | | 9 | | | | |
| | | | 11 | | | | |
| | | | 13 | | | | |
| | | | 14 | | | | |
| 16 | | | | | | | |
| 18 | | | | | | | |
| 1 | — | — | — | — | — | — | Input offset adjustment |
| 5 | — | — | — | — | — | — | Input offset adjustment |
| 6 | — | — | — | — | — | O | Output |
| V _{CC-} | 4 | 4 | 10 | 11 | 16 | — | Power supply |
| V _{CC+} | 7 | 8 | 20 | 4 | 6 | — | Power supply |

6 Specifications

6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)⁽¹⁾

| | | | MIN | MAX | UNIT |
|-----------|--|------------------------------|--|-----|------|
| V_{CC+} | Supply voltage ⁽²⁾ | | | 18 | V |
| V_{CC-} | | | | -18 | |
| V_{ID} | Differential input voltage ⁽³⁾ | | | ±30 | V |
| V_I | Input voltage ⁽²⁾⁽⁴⁾ | | | ±15 | V |
| | Duration of output short circuit ⁽⁵⁾ | | Unlimited | | |
| | Continuous total power dissipation | | See Dissipation Rating Table | | |
| T_A | Operating free-air temperature | TL08_C TL08_AC TL08_BC | 0 | 70 | °C |
| | | TL08_I | -40 | 85 | |
| | | TL084Q | -40 | 125 | |
| | | TL08_M | -55 | 125 | |
| | Operating virtual junction temperature | | | 150 | °C |
| T_C | Case temperature for 60 seconds | FK package | TL08_M | 260 | °C |
| | Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds | J or JG package | TL08_M | 300 | °C |
| T_{stg} | Storage temperature | | -65 | 150 | °C |

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions* is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values, except differential voltages, are with respect to the midpoint between V_{CC+} and V_{CC-} .
- (3) Differential voltages are at $IN+$, with respect to $IN-$.
- (4) The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15 V, whichever is less.
- (5) The output may be shorted to ground or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.

6.2 ESD Ratings

| | | | VALUE | UNIT |
|-------------|-------------------------|--|-------|------|
| $V_{(ESD)}$ | Electrostatic discharge | Human body model (HBM), per ANSI/ESDA/JEDEC JS-001 ⁽¹⁾ | 1000 | V |
| | | Charged-device model (CDM), per JEDEC specification JESD22-C101 ⁽²⁾ | 1500 | |

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

| | | | MIN | MAX | UNIT |
|-----------|---------------------|--------|---------------|---------------|------|
| V_{CC+} | Supply voltage | | 5 | 15 | V |
| V_{CC-} | Supply voltage | | -5 | -15 | V |
| V_{CM} | Common-mode voltage | | $V_{CC-} + 4$ | $V_{CC+} - 4$ | V |
| T_A | Ambient temperature | TL08xM | -55 | 125 | °C |
| | | TL08xQ | -40 | 125 | |
| | | TL08xl | -40 | 85 | |
| | | TL08xC | 0 | 70 | |

6.4 Thermal Information

| THERMAL METRIC ⁽¹⁾ | TL08xx | | | | | | | | UNIT |
|---|----------|---------|----------|---------|------------------|------------------|------------|---------|------|
| | D (SOIC) | | N (PDIP) | NS (SO) | P (PDIP) | PS (SO) | PW (TSSOP) | | |
| | 8 PINS | 14 PINS | 14 PINS | 14 PINS | {PIN COUNT} PINS | {PIN COUNT} PINS | 8 PINS | 14 PINS | |
| R _{θJA} Junction-to-ambient thermal resistance ⁽²⁾⁽³⁾ | 97 | 86 | 76 | 80 | 85 | 95 | 149 | 113 | °C/W |

- (1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, [SPRA953](#).
- (2) Maximum power dissipation is a function of T_{J(max)}, R_{θJA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is P_D = (T_{J(max)} – T_A) / R_{θJA}. Operating at the absolute maximum T_J of 150°C can affect reliability.
- (3) The package thermal impedance is calculated in accordance with JESD 51-7.

6.5 Electrical Characteristics for TL08xC, TL08xxC, and TL08xl

V_{CC±} = ±15 V (unless otherwise noted)

| PARAMETER | TEST CONDITIONS | T _A ⁽¹⁾ | TL081C, TL082C, TL084C | | | TL081AC, TL082AC, TL084AC | | | TL081BC, TL082BC, TL084BC | | | TL081I, TL082I, TL084I | | | UNIT |
|--|---|-------------------------------|------------------------|------------------|-----|---------------------------|------------------|-----|---------------------------|------------------|-----|------------------------|------------------|-------|------|
| | | | MIN | TYP | MAX | MIN | TYP | MAX | MIN | TYP | MAX | MIN | TYP | MAX | |
| V _{IO} Input offset voltage | V _O = 0, R _S = 50 Ω | 25°C | | 3 | 15 | | 3 | 6 | | 2 | 3 | | 3 | 6 | mV |
| | | Full range | | | 20 | | | 7.5 | | | 5 | | | 9 | |
| α _{VIO} Temperature coefficient of input offset voltage | V _O = 0, R _S = 50 Ω | Full range | | 18 | | | 18 | | | 18 | | | 18 | μV/°C | |
| I _{IO} Input offset current ⁽²⁾ | V _O = 0 | 25°C | | 5 | 200 | | 5 | 100 | | 5 | 100 | | 5 | 100 | pA |
| | | Full range | | | 2 | | | 2 | | | 2 | | | 10 | nA |
| I _{IB} Input bias current ⁽²⁾ | V _O = 0 | 25°C | | 30 | 400 | | 30 | 200 | | 30 | 200 | | 30 | 200 | pA |
| | | Full range | | | 10 | | | 7 | | | 7 | | | 20 | nA |
| V _{ICR} Common-mode input voltage range | | 25°C | ±11 | -12 to 15 | | ±11 | -12 to 15 | | ±11 | -12 to 15 | | ±11 | -12 to 15 | V | |
| V _{OM} Maximum peak output voltage swing | R _L = 10 kΩ | 25°C | ±12 | ±13.5 | | ±12 | ±13.5 | | ±12 | ±13.5 | | ±12 | ±13.5 | V | |
| | R _L ≥ 10 kΩ | | ±12 | | | ±12 | | | ±12 | | | ±12 | | | |
| | R _L ≥ 2 kΩ | Full range | ±10 | ±12 | | ±10 | ±12 | | ±10 | ±12 | | ±10 | ±12 | | |
| A _{VD} Large-signal differential voltage amplification | V _O = ±10 V, R _L ≥ 2 kΩ | 25°C | 25 | 200 | | 50 | 200 | | 50 | 200 | | 50 | 200 | V/mV | |
| | | Full range | 15 | | | 15 | | | 25 | | | 25 | | | |
| B ₁ Unity-gain bandwidth | | 25°C | | 3 | | | 3 | | | 3 | | | 3 | MHz | |
| r _i Input resistance | | 25°C | | 10 ¹² | | | 10 ¹² | | | 10 ¹² | | | 10 ¹² | Ω | |
| CMRR Common-mode rejection ratio | V _{IC} = V _{ICRmin} , V _O = 0, R _S = 50 Ω | 25°C | 70 | 86 | | 75 | 86 | | 75 | 86 | | 75 | 86 | dB | |
| k _{SVR} Supply-voltage rejection ratio (ΔV _{CC±} /ΔV _{IO}) | V _{CC} = ±15 V to ±9 V, V _O = 0, R _S = 50 Ω | 25°C | 70 | 86 | | 80 | 86 | | 80 | 86 | | 80 | 86 | dB | |

- (1) All characteristics are measured under open-loop conditions with zero common-mode voltage, unless otherwise specified. Full range for T_A is 0°C to 70°C for TL08_C, TL08_AC, TL08_BC and –40°C to 85°C for TL08_I.
- (2) Input bias currents of an FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive, as shown in [Figure 13](#). Pulse techniques must be used that maintain the junction temperature as close to the ambient temperature as possible.

Electrical Characteristics for TL08xC, TL08xxC, and TL08xl (continued)

 $V_{CC\pm} = \pm 15\text{ V}$ (unless otherwise noted)

| PARAMETER | TEST CONDITIONS | $T_A^{(1)}$ | TL081C, TL082C, TL084C | | | TL081AC, TL082AC, TL084AC | | | TL081BC, TL082BC, TL084BC | | | TL081I, TL082I, TL084I | | | UNIT |
|-----------------|--|-------------|------------------------|-----|-----|---------------------------|-----|-----|---------------------------|-----|-----|------------------------|-----|-----|------|
| | | | MIN | TYP | MAX | MIN | TYP | MAX | MIN | TYP | MAX | MIN | TYP | MAX | |
| I_{CC} | Supply current (each amplifier) $V_O = 0$, No load | 25°C | | 1.4 | 2.8 | | 1.4 | 2.8 | | 1.4 | 2.8 | | 1.4 | 2.8 | mA |
| V_{O1}/V_{O2} | Crosstalk attenuation $A_{VD} = 100$ | 25°C | | 120 | | | 120 | | | 120 | | | 120 | | dB |

6.6 Electrical Characteristics for TL08xM and TL084x

 $V_{CC\pm} = \pm 15\text{ V}$ (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | T_A | TL081M, TL082M | | | TL084Q, TL084M | | | UNIT |
|-----------------|---|------------|----------------|------------|-----|----------------|------------|-----|------------------------------|
| | | | MIN | TYP | MAX | MIN | TYP | MAX | |
| V_{IO} | Input offset voltage $V_O = 0$, $R_S = 50\ \Omega$ | 25°C | | 3 | 6 | | 3 | 9 | mV |
| | | Full range | | | 9 | | 15 | | |
| α_{VIO} | Temperature coefficient of input offset voltage $V_O = 0$, $R_S = 50\ \Omega$ | Full range | | 18 | | | 18 | | $\mu\text{V}/^\circ\text{C}$ |
| I_{IO} | Input offset current ⁽²⁾ $V_O = 0$ | 25°C | | 5 | 100 | | 5 | 100 | pA |
| | | 125°C | | | 20 | | 20 | nA | |
| I_{IB} | Input bias current ⁽²⁾ $V_O = 0$ | 25°C | | 30 | 200 | | 30 | 200 | pA |
| | | 125°C | | | 50 | | 50 | nA | |
| V_{ICR} | Common-mode input voltage range | 25°C | ± 11 | -12 to 15 | | ± 11 | -12 to 15 | | V |
| V_{OM} | Maximum peak output voltage swing $R_L = 10\ \text{k}\Omega$ $R_L \geq 10\ \text{k}\Omega$ $R_L \geq 2\ \text{k}\Omega$ | 25°C | ± 12 | ± 13.5 | | ± 12 | ± 13.5 | | V |
| | | Full range | ± 12 | | | ± 12 | | | |
| | | | ± 10 | ± 12 | | ± 10 | ± 12 | | |
| A_{VD} | Large-signal differential voltage amplification $V_O = \pm 10\ \text{V}$, $R_L \geq 2\ \text{k}\Omega$ | 25°C | | 25 | 200 | | 25 | 200 | V/mV |
| | | Full range | | 15 | | 15 | | | |
| B_1 | Unity-gain bandwidth | 25°C | | 3 | | | 3 | | MHz |
| r_i | Input resistance | 25°C | | 10^{12} | | | 10^{12} | | Ω |
| CMRR | Common-mode rejection ratio $V_{IC} = V_{ICRmin}$, $V_O = 0$, $R_S = 50\ \Omega$ | 25°C | | 80 | 86 | | 80 | 86 | dB |
| k_{SVR} | Supply-voltage rejection ratio ($\Delta V_{CC\pm}/\Delta V_{IO}$) $V_{CC} = \pm 15\ \text{V}$ to $\pm 9\ \text{V}$, $V_O = 0$, $R_S = 50\ \Omega$ | 25°C | | 80 | 86 | | 80 | 86 | dB |
| I_{CC} | Supply current (each amplifier) $V_O = 0$, No load | 25°C | | 1.4 | 2.8 | | 1.4 | 2.8 | mA |
| V_{O1}/V_{O2} | Crosstalk attenuation $A_{VD} = 100$ | 25°C | | 120 | | | 120 | | dB |

(1) All characteristics are measured under open-loop conditions, with zero common-mode input voltage, unless otherwise specified.

(2) Input bias currents of a FET-input operational amplifier are normal junction reverse currents, which are temperature sensitive, as shown in Figure 13. Pulse techniques must be used that maintain the junction temperatures as close to the ambient temperature as possible.

6.7 Operating Characteristics

 $V_{CC\pm} = \pm 15\ \text{V}$, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

| PARAMETER | TEST CONDITIONS | MIN | TYP | MAX | UNIT |
|-----------|--|------------------|-----|-----|------------------|
| SR | $V_I = 10\ \text{V}$, $R_L = 2\ \text{k}\Omega$, $C_L = 100\ \text{pF}$, See Figure 19 | 8 ⁽¹⁾ | 13 | | V/ μs |
| | $V_I = 10\ \text{V}$, $R_L = 2\ \text{k}\Omega$, $C_L = 100\ \text{pF}$, $T_A = -55^\circ\text{C}$ to 125°C , See Figure 19 | 5 ⁽¹⁾ | | | |

(1) On products compliant to MIL-PRF-38535, this parameter is not production tested.

Operating Characteristics (continued)

$V_{CC\pm} = \pm 15\text{ V}$, $T_A = 25^\circ\text{C}$ (unless otherwise noted)

| PARAMETER | | TEST CONDITIONS | | MIN | TYP | MAX | UNIT |
|-----------|--------------------------------|---|--------------------------------------|-----|-----|--------|------------------------------|
| t_r | Rise-time | $V_I = 20\text{ V}$, $R_L = 2\text{ k}\Omega$, $C_L = 100\text{ pF}$, See Figure 19 | | | | 0.05 | μs |
| | overshoot factor | | | | | | |
| V_n | Equivalent input noise voltage | $R_S = 20\ \Omega$ | $f = 1\text{ kHz}$ | | | 18 | $\text{nV}/\sqrt{\text{Hz}}$ |
| | | | $f = 10\text{ Hz to } 10\text{ kHz}$ | | | 4 | μV |
| I_n | Equivalent input noise current | $R_S = 20\ \Omega$, | $f = 1\text{ kHz}$ | | | 0.01 | $\text{pA}/\sqrt{\text{Hz}}$ |
| THD | Total harmonic distortion | $V_{\text{rms}} = 6\text{ V}$, $A_{\text{VD}} = 1$, $R_S \leq 1\text{ k}\Omega$, $R_L \geq 2\text{ k}\Omega$, $f = 1\text{ kHz}$, | | | | 0.003% | |

6.8 Dissipation Rating Table

| PACKAGE | $T_A \leq 25^\circ\text{C}$ POWER RATING | DERATING FACTOR | DERATE ABOVE T_A | $T_A = 70^\circ\text{C}$ POWER RATING | $T_A = 85^\circ\text{C}$ POWER RATING | $T_A = 125^\circ\text{C}$ POWER RATING |
|------------|---|---------------------------|-----------------------|--|--|---|
| D (14 pin) | 680 mW | 7.6 mW/ $^\circ\text{C}$ | 60 $^\circ\text{C}$ | 604 mW | 490 mW | 186 mW |
| FK | 680 mW | 11.0 mW/ $^\circ\text{C}$ | 88 $^\circ\text{C}$ | 680 mW | 680 mW | 273 mW |
| J | 680 mW | 11.0 mW/ $^\circ\text{C}$ | 88 $^\circ\text{C}$ | 680 mW | 680 mW | 273 mW |
| JG | 680 mW | 8.4 mW/ $^\circ\text{C}$ | 69 $^\circ\text{C}$ | 672 mW | 546 mW | 210 mW |

6.9 Typical Characteristics

Data at high and low temperatures are applicable only within the rated operating free-air temperature ranges of the various devices. The Figure numbers referenced in the following graphs are located in [Parameter Measurement Information](#).

Table 1. Table of Graphs

| | | Figure |
|----------|---|--|
| V_{OM} | Maximum peak output voltage | versus Frequency versus Free-air temperature versus Load resistance versus Supply voltage |
| | | Figure 1, Figure 2, Figure 3 Figure 4 Figure 5 Figure 6 |
| A_{VD} | Large-signal differential voltage amplification | versus Free-air temperature versus Load resistance |
| | Differential voltage amplification | versus Frequency with feed-forward compensation |
| P_D | Total power dissipation | versus Free-air temperature |
| I_{CC} | Supply current | versus Free-air temperature versus Supply voltage |
| | | Figure 11 Figure 12 |
| I_{IB} | Input bias current | versus Free-air temperature |
| | Large-signal pulse response | versus Time |
| V_O | Output voltage | versus Elapsed time |
| CMRR | Common-mode rejection ratio | versus Free-air temperature |
| V_n | Equivalent input noise voltage | versus Frequency |
| THD | Total harmonic distortion | versus Frequency |

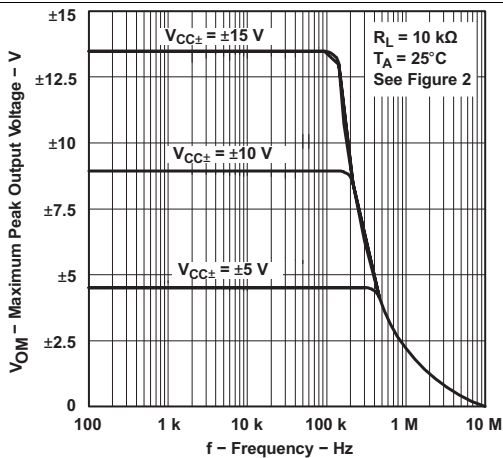


Figure 1. Maximum Peak Output Voltage vs Frequency

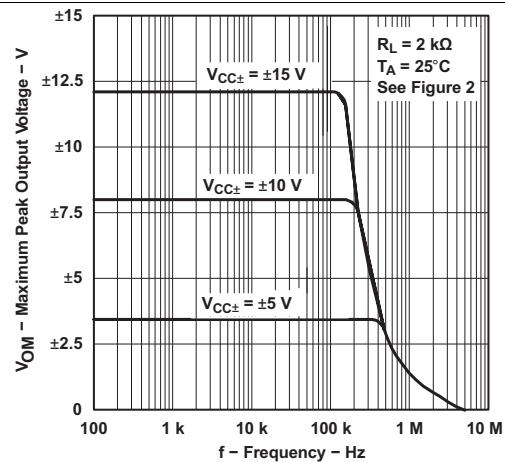


Figure 2. Maximum Peak Output Voltage vs Frequency

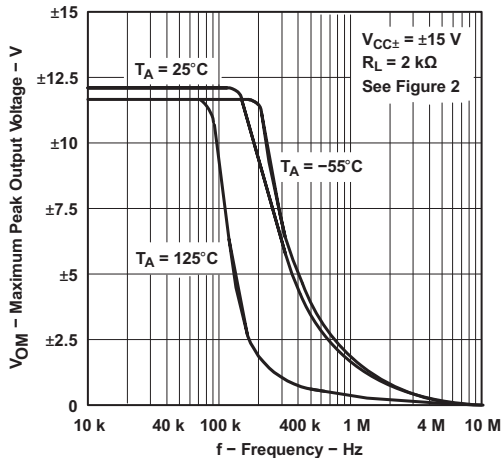


Figure 3. Maximum Peak Output Voltage vs Frequency

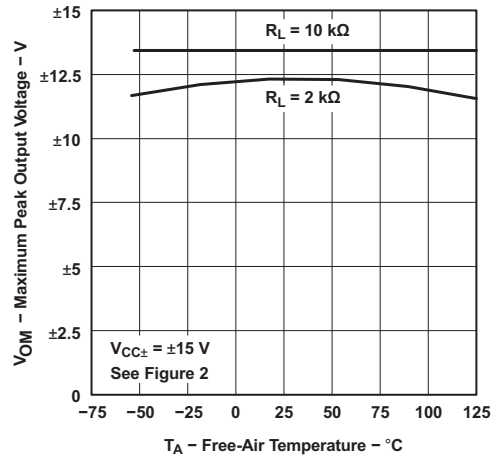


Figure 4. Maximum Peak Output Voltage vs Free-Air Temperature

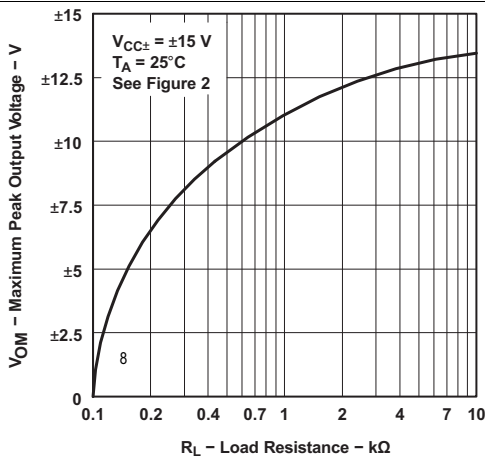


Figure 5. Maximum Peak Output Voltage vs Load Resistance

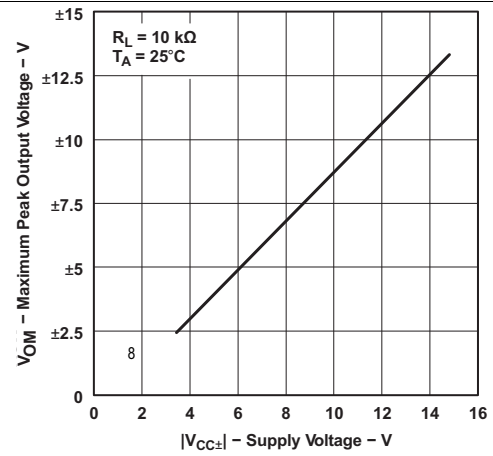


Figure 6. Maximum Peak Output Voltage vs Supply Voltage

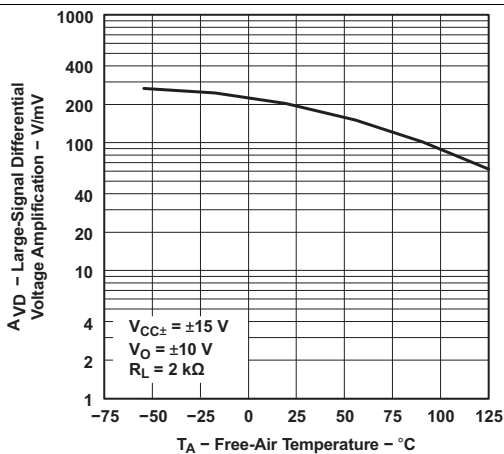


Figure 7. Large-Signal Differential Voltage Amplification vs Free-Air Temperature

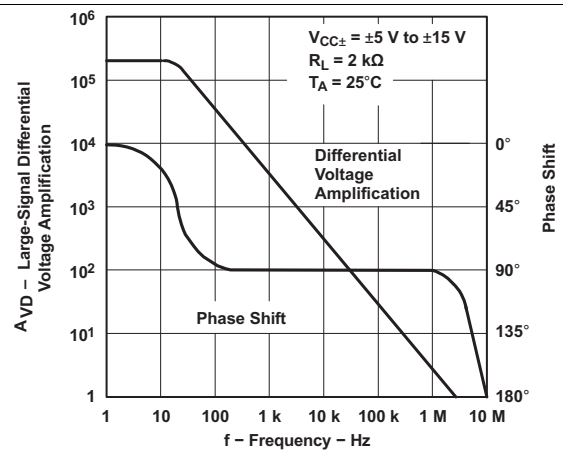


Figure 8. Large-Signal Differential Voltage Amplification and Phase Shift vs Frequency

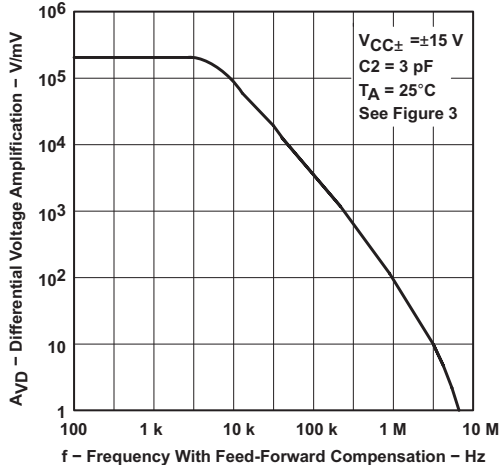


Figure 9. Differential Voltage Amplification vs Frequency with Feed-Forward Compensation

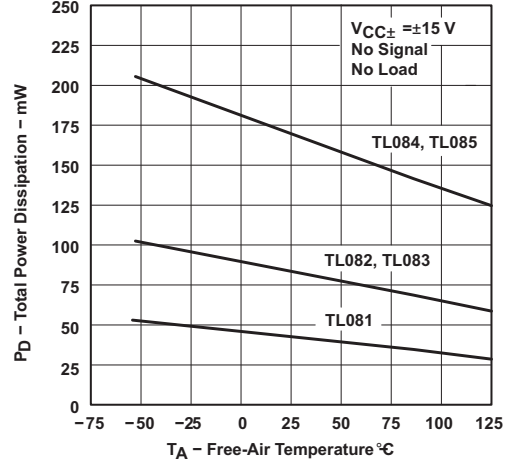


Figure 10. Total Power Dissipation vs Free-Air Temperature

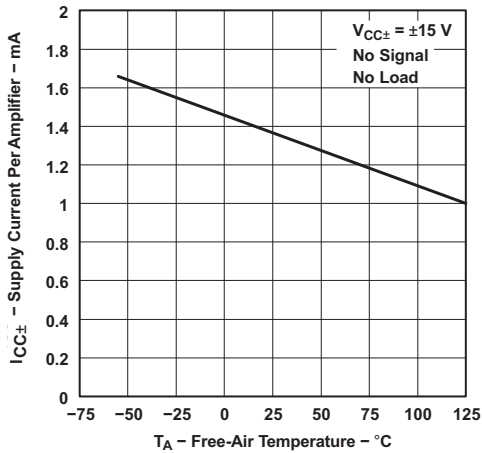


Figure 11. Supply Current per Amplifier vs Free-Air Temperature

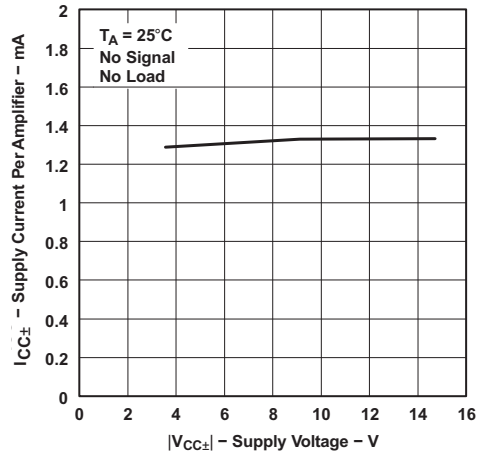


Figure 12. Supply Current per Amplifier vs Supply Voltage

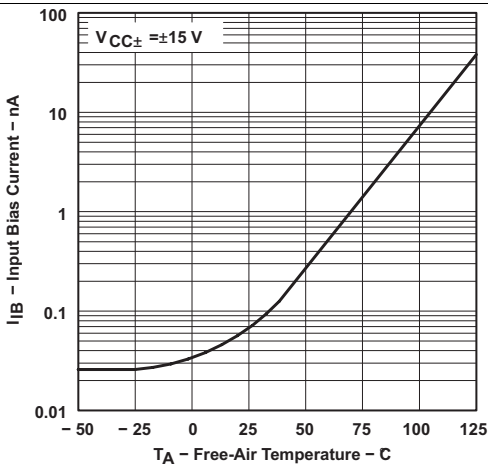


Figure 13. Input Bias Current vs Free-Air Temperature

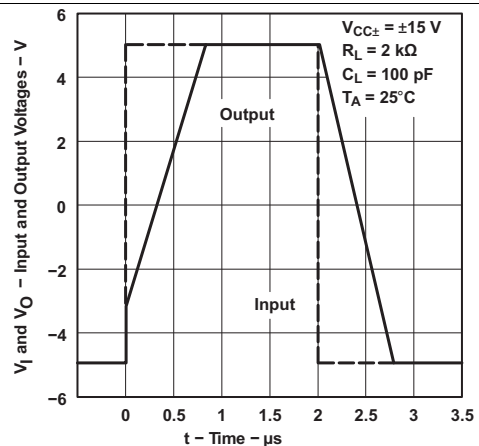


Figure 14. Voltage-Follower Large-Signal Pulse Response

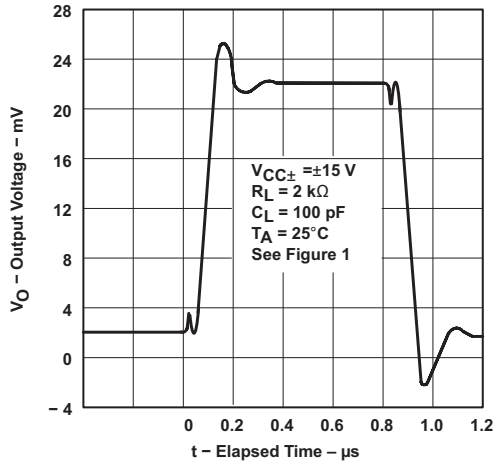


Figure 15. Output Voltage vs Elapsed Time

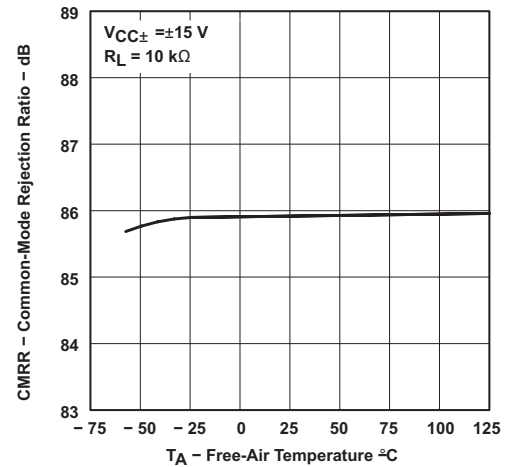


Figure 16. Common-Mode Rejection Ratio vs Free-Air Temperature

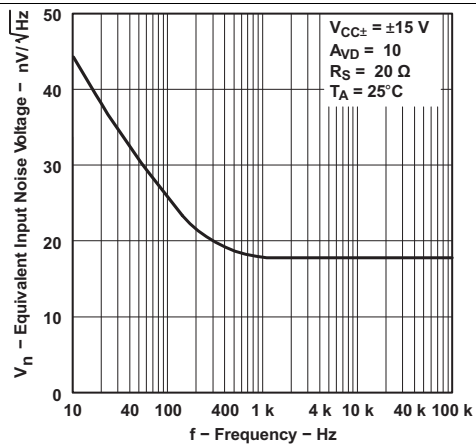


Figure 17. Equivalent Input Noise Voltage vs Frequency

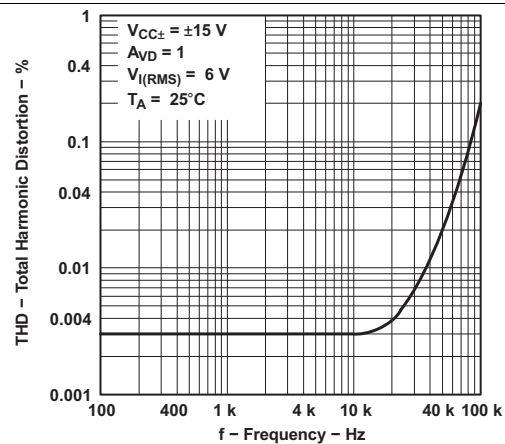


Figure 18. Total Harmonic Distortion vs Frequency

7 Parameter Measurement Information

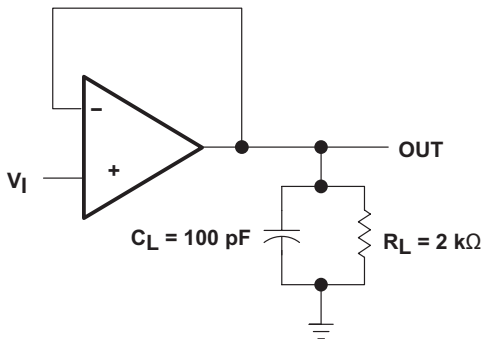


Figure 19. Test Figure 1

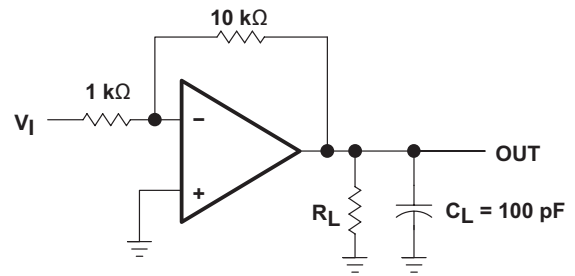


Figure 20. Test Figure 2

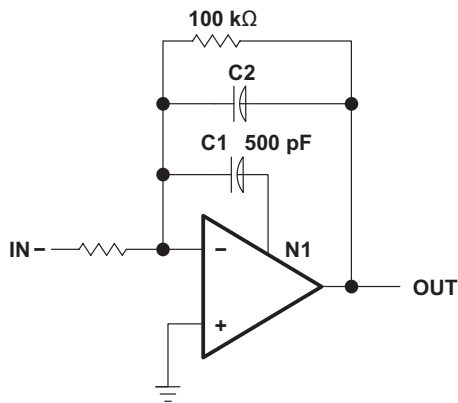


Figure 21. Test Figure 3

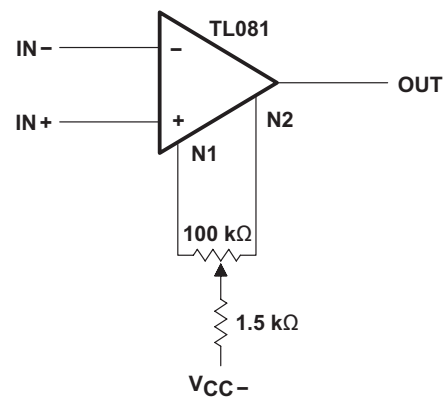


Figure 22. Test Figure 4

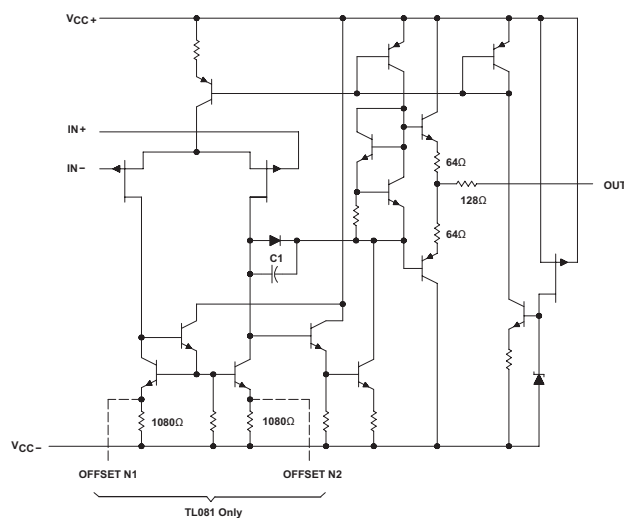
8 Detailed Description

8.1 Overview

The TL08xx JFET-input operational amplifier family is designed to offer a wider selection than any previously developed operational amplifier family. Each of these JFET-input operational amplifiers incorporates well-matched, high-voltage JFET and bipolar transistors in a monolithic integrated circuit. The devices feature high slew rates, low input bias and offset currents, and low offset-voltage temperature coefficient. Offset adjustment and external compensation options are available within the TL08xx family.

The C-suffix devices are characterized for operation from 0°C to 70°C. The I-suffix devices are characterized for operation from –40°C to 85°C. The Q-suffix devices are characterized for operation from –40°C to +125°C. The M-suffix devices are characterized for operation over the full military temperature range of –55°C to +125°C.

8.2 Functional Block Diagram



8.3 Feature Description

8.3.1 Total Harmonic Distortion

Harmonic distortions to an audio signal are created by electronic components in a circuit. Total harmonic distortion (THD) is a measure of harmonic distortions accumulated by a signal in an audio system. These devices have a very low THD of 0.003% meaning that the TL08x devices will add little harmonic distortion when used in audio signal applications.

8.3.2 Slew Rate

The slew rate is the rate at which an operational amplifier can change its output when there is a change on the input. These devices have a 13-V/μs slew rate.

8.4 Device Functional Modes

These devices are powered on when the supply is connected. This device can be operated as a single-supply operational amplifier or dual-supply amplifier depending on the application.

9 Applications and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

The TL08x series of operational amplifiers can be used in countless applications. The few applications in this section show principles used in all applications of these parts.

9.2 Typical Applications

9.2.1 Inverting Amplifier Application

A typical application for an operational amplifier is an inverting amplifier. This amplifier takes a positive voltage on the input, and makes it a negative voltage of the same magnitude. In the same manner, it also makes negative voltages positive.

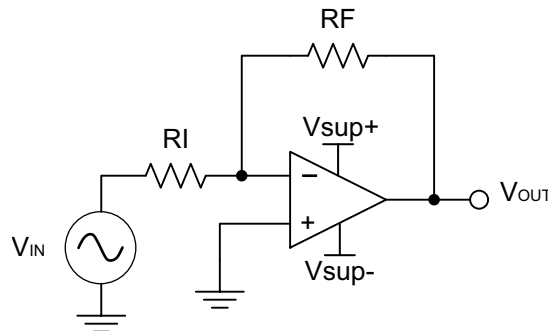


Figure 23. Schematic for Inverting Amplifier Application

9.2.1.1 Design Requirements

The supply voltage must be chosen such that it is larger than the input voltage range and output range. For instance, this application will scale a signal of ± 0.5 V to ± 1.8 V. Setting the supply at ± 12 V is sufficient to accommodate this application.

9.2.1.2 Detailed Design Procedure

Determine the gain required by the inverting amplifier:

$$A_v = \frac{V_{OUT}}{V_{IN}} \quad (1)$$

$$A_v = \frac{1.8}{-0.5} = -3.6 \quad (2)$$

Once the desired gain is determined, choose a value for R_I or R_F . Choosing a value in the $k\Omega$ range is desirable because the amplifier circuit will use currents in the milliamp range. This ensures the part will not draw too much current. This example will choose 10 $k\Omega$ for R_I which means 36 $k\Omega$ will be used for R_F . This was determined by Equation 3.

$$A_v = -\frac{R_F}{R_I} \quad (3)$$

Typical Applications (continued)

9.2.1.3 Application Curve

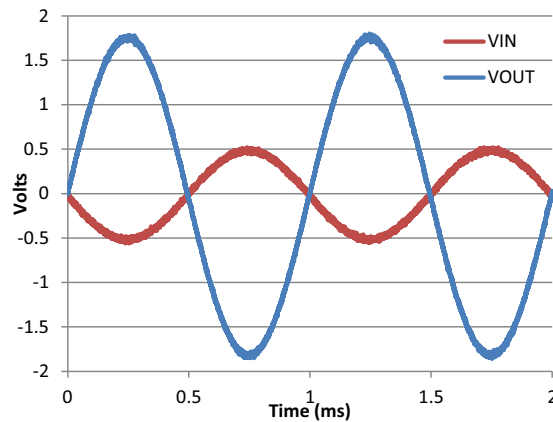


Figure 24. Input and output voltages of the inverting amplifier

9.3 System Examples

9.3.1 General Applications

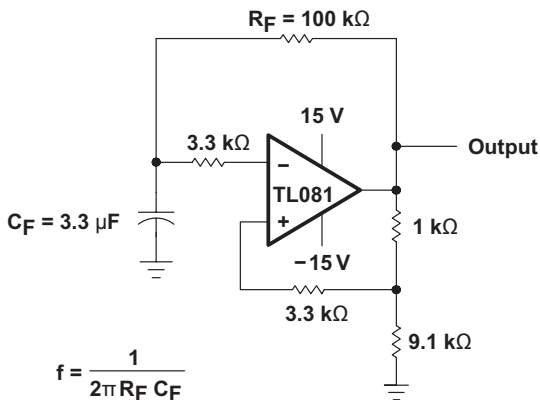


Figure 25. 0.5-Hz Square-Wave Oscillator

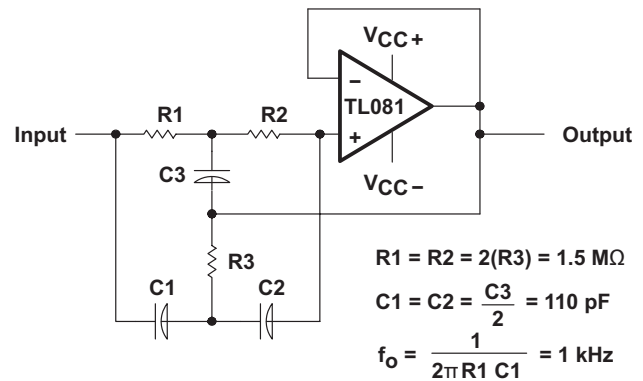


Figure 26. High-Q Notch Filter

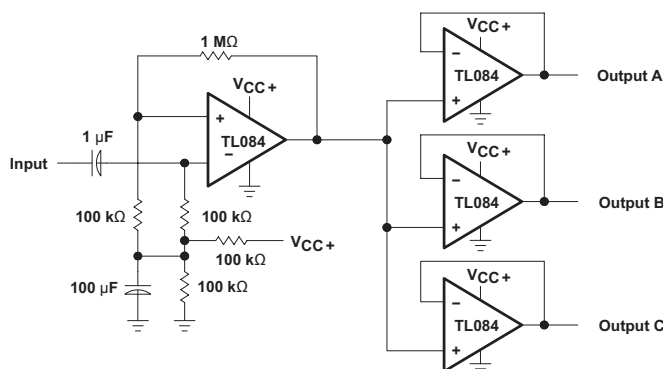


Figure 27. Audio-Distribution Amplifier

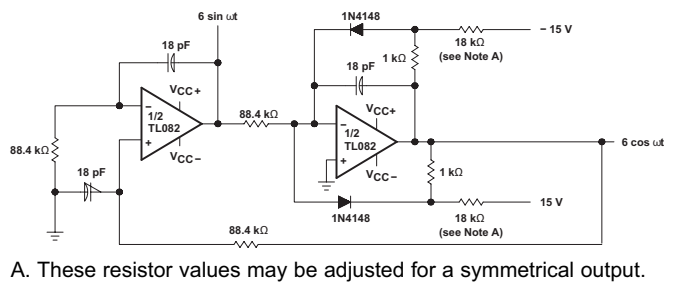
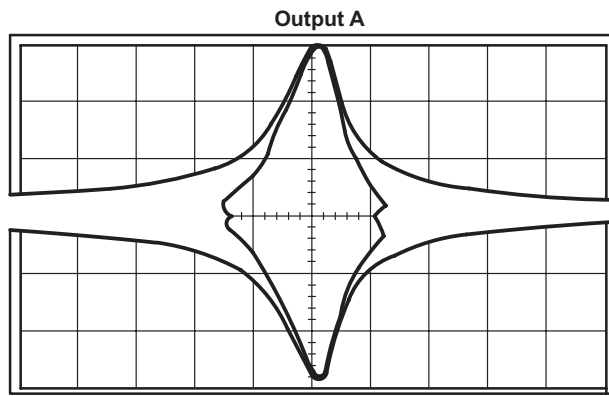
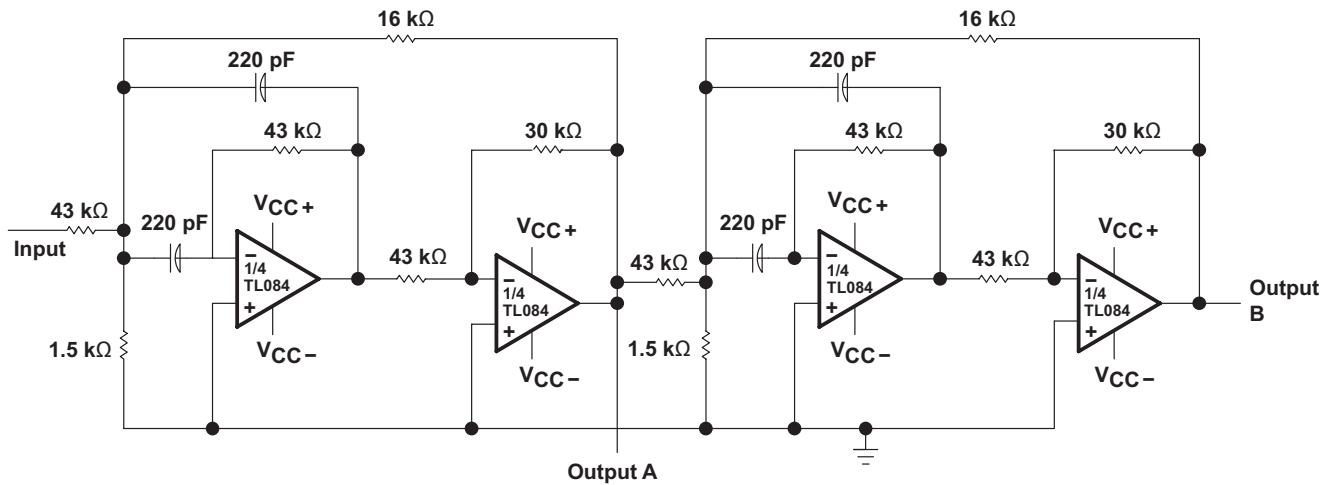
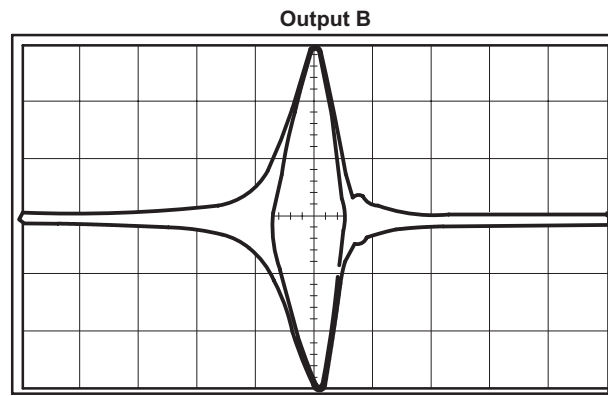


Figure 28. 100-kHz Quadrature Oscillator

System Examples (continued)



2 kHz/div
Second-Order Bandpass Filter
 $f_o = 100 \text{ kHz}$, $Q = 30$, GAIN = 4



2 kHz/div
Cascaded Bandpass Filter
 $f_o = 100 \text{ kHz}$, $Q = 69$, GAIN = 16

Figure 29. Positive-Feedback Bandpass Filter