

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

IMPLEMENTACIÓN DE UN CONTADOR Y CLASIFICADOR DE MONEDAS AUTOMÁTICO

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES

Kevin Miguel Landázuri Muñoz

kevin.landazuri@epn.edu.ec

DIRECTOR: ING. LEANDRO ANTONIO PAZMIÑO ORTIZ, MSC.

leandro.pazmino@epn.edu.ec

CODIRECTOR: ING. MÓNICA DE LOURDES VINUEZA RHOR, MSC.

monica.vinueza@epn.edu.ec

Quito, mayo 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr Landázuri Muñoz Kevin Miguel como requerimiento parcial a la obtención del título de TECNÓLOGO EN ELECTRÓNICA Y TELECOMUNICACIONES, bajo nuestra supervisión:

Ing. Leandro Pazmiño, MSc.

DIRECTOR DEL PROYECTO

Ing. Mónica Vinuesa, MSc.

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Kevin Miguel Landázuri Muñoz con CI: 1726620022 declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.



**Kevin Miguel Landázuri
Muñoz**

DEDICATORIA

A mi familia, en especial a mis padres por haberme forjado como la persona que soy hoy en día. Muchos logros han sido gracias a su apoyo, siempre han estado ahí para apoyarme y corregirme, siendo un pilar fundamental en mi vida en conjunto con mis hermanos y hermana. Todos ellos me han dado todo lo que soy, mis valores, mis principios, mi perseverancia y empeño.

AGRADECIMIENTO

En primer lugar, a Dios ya que me ha dado fortaleza, sabiduría y una gran familia. Sabiendo que soy su hijo y que puedo vivir confiado en Él, nunca me ha dejado solo y nunca lo hará. A mi familia nuevamente por su apoyo incondicional y por qué siempre han estado conmigo a pesar de mis errores.

De igual manera al Ingeniero Leandro Pazmiño y a la ESFOT por brindarme sus conocimientos y su paciencia para formarme profesionalmente.

ÍNDICE DE CONTENIDOS

CERTIFICACIÓN	I
DECLARACIÓN	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDOS	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	XII
RESUMEN.....	XIII
<i>ABSTRACT</i>	XIV
1 INTRODUCCIÓN.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Fundamentos.....	2
Placas de desarrollo	2
Módulo RTC DS1307.....	3
Módulos GPRS/GSM.....	4
Sensores detectores infrarrojos	7
2 METODOLOGÍA.....	9
2.1 Descripción de la metodología usada	9
3 RESULTADOS Y DISCUSIÓN	11
3.1 Identificación y selección de los requerimientos necesarios para implementar el proyecto	11
Selección del método de clasificación de las monedas.....	11
Selección del método de conteo y envío de datos del prototipo.....	16
Selección de la plataforma de desarrollo de <i>software</i>	16
Selección del <i>hardware</i>	17
Selección del sensor de conteo de monedas.....	18

Interfaz de usuario	18
Selección del regulador de voltaje	20
Selección del dispositivo de fecha y hora.....	21
Selección del módulo GPRS/GSM.....	21
Selección de las fuentes de alimentación	23
3.2 Diseño de la máquina contadora	25
Selección del material para la barra clasificadora	25
Colocación y dimensionamiento de las medidas para cada moneda en la barra clasificadora.....	26
Diseño del separador de monedas	29
Diseño del contenedor de monedas.....	29
Diseño del distribuidor de monedas	29
Diseño del panel de control	30
Diseño de los circuitos electrónicos	30
Diseño de los circuitos de simulación	34
3.3 Desarrollo del <i>software</i> a emplearse en el microcontrolador.....	37
Esquemas del código de programación	37
Programación del código en Arduino IDE	41
Compilación del código de programación	47
3.4 Implementación del prototipo de la máquina contadora de monedas.....	48
Separador de monedas	48
Barra clasificadora de monedas.....	48
Base del prototipo.....	50
Implementación de la barra clasificadora en la base del prototipo	50
Barra de plástico.....	51
Contenedores de monedas.....	52
Fabricación del diseño de circuitos electrónicos PCB	53
Fabricación de los circuitos electrónicos en baquelita.....	56
Soldado de los elementos electrónicos	59

Instalación de los sensores detectores	60
Implementación de los elementos electrónicos en panel de control	61
Implementación de los circuitos electrónicos y módulos	61
Instalación del distribuidor de monedas	65
Implementación y finalización el prototipo	67
3.5 Realización de pruebas de funcionamiento de la máquina y del envío de notificaciones al usuario final.....	69
Pruebas de funcionamiento del <i>software</i>	69
Pruebas del conteo de monedas	74
Comprobación de la llegada de la información final al usuario.....	75
3.6 Costo del proyecto.....	76
4 CONCLUSIONES Y RECOMENDACIONES	78
4.1 Conclusiones	78
4.2 Recomendaciones	79
5 REFERENCIAS BIBLIOGRÁFICAS.....	82
ANEXOS.....	87
Anexo: CÓDIGO DE PROGRAMACIÓN.....	i
Anexo: MANUAL DE MANTENIMIENTO	xxviii
Anexo: MANUAL DE USO	xxxii

ÍNDICE DE FIGURAS

Figura 1.1 Placa de desarrollo	2
Figura 1.2 Módulo RTC DS1307	3
Figura 1.3 Configuración de pines del Módulo RTC DS1307	3
Figura 1.4 Módulo IOT-GA6	4
Figura 1.5 Distribución de pines del módulo IOT-GA6.....	5
Figura 1.6 Antena PCB y antena helicoidal	5
Figura 1.7 Distribución de pines	7
Figura 1.8 Módulo sensor infrarrojo TCRT5000	8
Figura 1.9 Sensor CNY70	8
Figura 3.1 Moneda de 10 centavos estadounidense	11
Figura 3.2 Moneda de 10 centavos nacional	12
Figura 3.3 Moneda de 1 centavo estadounidense.....	12
Figura 3.4 Moneda de 1 centavo nacional.....	12
Figura 3.5 Moneda de 5 centavos estadounidense	13
Figura 3.6 Moneda de 5 centavos nacional.....	13
Figura 3.7 Moneda de 25 centavos estadounidense	14
Figura 3.8 Moneda de 25 centavos nacional.....	14
Figura 3.9 Moneda de 1 dólar estadounidense	14
Figura 3.10 Moneda de 50 centavos estadounidense	15
Figura 3.11 Moneda de 50 centavos nacional.....	15
Figura 3.12 Pantalla LCD 16x4	19
Figura 3.13 Regulador de voltaje LM7805.....	21
Figura 3.14 Fuente de alimentación principal.....	24
Figura 3.15 Fuente de alimentación para módulo GPRS/GSM	25
Figura 3.16 Tubo de aluminio.....	26
Figura 3.17 Medida del espacio inferior.....	26
Figura 3.18 Medida de la moneda de 10 centavos.....	27
Figura 3.19 Medida de la moneda de 1 centavo.....	27
Figura 3.20 Medida de la moneda de 5 centavos.....	27
Figura 3.21 Medida de la moneda de 25 centavos.....	28
Figura 3.22 Medida de la moneda de 1 dólar	28
Figura 3.23 Medida de la moneda de 50 centavos.....	28
Figura 3.24 Diseño y medidas del separador de monedas.....	29
Figura 3.25 Soporte de camino	29

Figura 3.26	Diseño del panel de control.....	30
Figura 3.27	Diseño del circuito del regulador de voltaje	31
Figura 3.28	Diseño PCB del circuito del regulador de voltaje	31
Figura 3.29	Diseño en <i>Proteus</i> de placa de control de pulsadores y <i>led</i> indicador.	32
Figura 3.30	Circuito PCB de placa de control de pulsadores y <i>led</i> indicador	32
Figura 3.31	Diseño en <i>Proteus</i> de placa de control de pantalla LCD.....	33
Figura 3.32	Circuito PCB de placa de control de pantalla LCD	34
Figura 3.33	Diseño de simulación de pantalla LCD.....	35
Figura 3.34	Diseño de circuito simulador de selección de modo	35
Figura 3.35	Diseño de circuito simulador de opciones	36
Figura 3.36	Diseño de circuito simulador de <i>led</i> indicador	36
Figura 3.37	Circuito emulador de sensor detector de monedas	36
Figura 3.38	Diagrama de flujo general	37
Figura 3.39	Diagrama de flujo de OPCIONES.....	38
Figura 3.40	Diagrama de flujo de PROCESO.....	40
Figura 3.41	Librerías	41
Figura 3.42	Variables para el conteo de monedas	43
Figura 3.43	Variables para anti-rebote	43
Figura 3.44	Variables para la función <i>millis</i>	43
Figura 3.45	Variables de lectura digital	44
Figura 3.46	Distribución y configuración de pines de conexión	44
Figura 3.47	Configuración de <i>Setup</i>	45
Figura 3.48	Subrutina INTRO.....	46
Figura 3.49	Estructura del código de programación para la lectura y conteo de cada monedas.....	47
Figura 3.50	Subrutina VERIFICAR.....	47
Figura 3.51	Separador de monedas.....	48
Figura 3.52	Barra clasificadora de monedas	48
Figura 3.53	Implementación de separadores a la barra clasificadora.....	49
Figura 3.54	Barra clasificadora de monedas finalizada	49
Figura 3.55	Barra clasificadora complementada con piezas de madera.....	50
Figura 3.56	Base del prototipo	50
Figura 3.57	Sostenedores de la barra calificadora	51
Figura 3.58	Implementación de la barra clasificadora a la base del prototipo.....	51
Figura 3.59	Resultado final de la implementación de la barra clasificadora con la base del prototipo	51

Figura 3.60 Barra de plástico implementada	52
Figura 3.61 Composición de piezas	52
Figura 3.62 Pieza deslizadora de monedas	53
Figura 3.63 Contenedores de monedas	53
Figura 3.64 Exportación de circuitos PCB en pdf	54
Figura 3.65 Configuración de impresión de pistas conductoras	54
Figura 3.66 Configuración de impresión de señalización de elementos	55
Figura 3.67 Impresión de pistas conductoras	55
Figura 3.68 Impresión de señalización	56
Figura 3.69 Implementación de la impresión PCB sobre baquelita.....	56
Figura 3.70 Planchado de pistas sobre baquelita	57
Figura 3.71 Enfriamiento de baquelita.....	57
Figura 3.72 Proceso de quemado de baquelita	58
Figura 3.73 Pistas de la placa PCB.....	58
Figura 3.74 Señalización de elementos electrónicos.....	59
Figura 3.75 Circuito en baquelita perforada	59
Figura 3.76 Soldadura de elementos electrónicos.....	60
Figura 3.77 Placa PCB en base de madera	60
Figura 3.78 Instalación de sensores detectores de monedas.....	61
Figura 3.79 Conexión de cables a elementos electrónicos del panel de control	61
Figura 3.80 Placa de control de pulsadores, <i>switch</i> y <i>led</i> indicador.....	62
Figura 3.81 Placa de control de pantalla LCD	62
Figura 3.82 Placa del módulo RTC	63
Figura 3.83 Placa de <i>switch</i> de encendido y apagado.....	63
Figura 3.84 Placa distribuidora de voltaje.....	64
Figura 3.85 Placa del módulo GPRS/GSM.....	65
Figura 3.86 Placa de Arduino Mega 2560	65
Figura 3.87 Ventilador de PC	66
Figura 3.88 Estructura conductora de monedas.....	66
Figura 3.89 Soportes del distribuidor de monedas	66
Figura 3.90 Implementación de orificio y piezas de conducción	67
Figura 3.91 Implementación de la barra para contenedores.....	67
Figura 3.92 Señalización del panel de control	68
Figura 3.93 Prototipo finalizado.....	68
Figura 3.94 Animación introductoria en simulación	69
Figura 3.95 Animación introductoria en prototipo	69

Figura 3.96	Formatos de modos de funcionamiento en simulación	70
Figura 3.97	Formatos de modos de funcionamiento en prototipo	70
Figura 3.98	Simulación del conteo de monedas	71
Figura 3.99	Proceso de conteo de monedas en el prototipo	71
Figura 3.100	Simulación de la opción GUARDAR	71
Figura 3.101	Opción GUARDAR en prototipo	72
Figura 3.102	Simulación de la opción ENVIAR	72
Figura 3.103	Opción ENVIAR en el prototipo	73
Figura 3.104	Simulación de la opción RESULTADOS	73
Figura 3.105	Opción RESULTADOS en prototipo	73
Figura 3.106	Simulación de la opción <i>RESET</i>	74
Figura 3.107	Opción <i>RESET</i> en prototipo	74
Figura 3.108	Monedas en el distribuidor	75
Figura 3.109	Monedas en los contenedores	75
Figura 3.110	Resultados del conteo de monedas	75
Figura 3.111	Notificación por SMS al usuario con fecha y hora	76

ÍNDICE DE TABLAS

Tabla 1.1	Descripción de pines del Módulo RTC DS1307	4
Tabla 1.2	Descripción de pines del Módulo IOT-GA6	6
Tabla 1.3	Descripción de la distribución de pines del módulo SIM800L.....	7
Tabla 3.1	Información de la moneda de 10 centavos.....	12
Tabla 3.2	Información de la moneda de 1 centavo	13
Tabla 3.3	Información de la moneda de 5 centavos	13
Tabla 3.4	Información de la moneda de 25 centavos.....	14
Tabla 3.5	Información de la moneda de 1 dólar.....	15
Tabla 3.6	Información de la moneda de 50 centavos.....	15
Tabla 3.7	Comparativa entre <i>Arduino</i> y <i>Raspberry Pi</i>	17
Tabla 3.8	Diferencias entre sensores infrarrojos.....	18
Tabla 3.9	Especificación de funciones de pines de la pantalla LCD 16x4.....	19
Tabla 3.10	Cuadro comparativo de módulos GPRS/GSM	22
Tabla 3.11	Consumo de corriente y voltaje de elementos electrónicos.....	23
Tabla 3.12	Lista de componentes del circuito del regulador de voltaje	31
Tabla 3.13	Lista de elementos de placa de control de pulsadores y <i>led</i> indicador.....	33
Tabla 3.14	Lista de componentes de placa de control de pantalla LCD.....	34
Tabla 3.15	Costo del proyecto	77

RESUMEN

Como primer punto se tiene la introducción del documento, en donde se plantea y justifica la creación del proyecto, también se encuentra información adicional sobre los objetivos del mismo. Además, se muestran los fundamentos teóricos los cuales abarcan conceptos sobre sensores, microcontroladores, módulos y demás, estos ayudan al sustento del desarrollo.

Como segundo punto se tiene la metodología del proyecto, en esta parte se describe el proceso que se realizó para cada uno de los objetivos específicos en los cuales se explica detalladamente la manera en la que se desarrolló el proyecto de inicio a fin.

Para el tercer punto se tienen los resultados y discusión, en esta parte se añade el desarrollo de los requerimientos que fueron necesarios para implementar el proyecto tanto en *software* como en *hardware*, también se incluye el diseño para cada área del proyecto y los circuitos que fueron requeridos. Además, se detalla el desarrollo del *software* y la implementación final del proyecto, por último, se añaden los resultados de las pruebas de funcionamiento.

Como punto final se tienen las conclusiones y recomendaciones, las mismas que fueron desarrolladas en base a la experiencia adquirida en la fabricación del proyecto. Adicionalmente se añaden las referencias bibliográficas con las que se realizó y fundamentó el marco teórico.

Finalmente, se incluye la documentación respectiva de los anexos, la cual comprende de: el código de programación y el manual de mantenimiento, con el objetivo de facilitar la comprensión y manejo del proyecto.

PALABRAS CLAVE: Arduino, sensores, contador de monedas, automatización.

ABSTRACT

The first point is the introduction, it explains the proposal and justification of the project, there is also additional information about the project objectives. Furthermore, the theoretical foundations are shown, it covers concepts about sensors, microcontrollers, modules and others, these will help to support the development.

The second point is the project methodology, this part describes the process carried out for each objective. This explains in detail how the project was developed from start to end.

For the third point contains the results and discussion, including the development of the requirements needed to implement the project parts in software and hardware. In addition, includes the design of each area of the project and the circuits required. Also it details the development of the software and the final implementation of the project, and finally, the results of the functional tests are showed.

As a final point, the conclusions and recommendations are included, they were developed based on the experience acquired in the manufacturing of the project. Additionally, the bibliographic references used for the theoretical framework development are added.

Finally, the documentation of the annexes is included, it covers: the code and the maintenance manual, in order to facilitate the understanding and management of the project.

KEYWORDS: *Arduino, sensors, coin counter, automation.*

1 INTRODUCCIÓN

Actualmente el conteo y clasificación de monedas se realiza de forma manual en la mayoría de establecimientos y empresas tales como: supermercados, bancos e incluso iglesias, siendo esto un proceso tedioso para los empleados. Además, existe la posibilidad de cometer errores de conteo puesto que los seres humanos tienden a equivocarse, por lo que esto podría generar un desfase económico en el informe de contabilidad de aquellas entidades.

La mayoría de las empresas siempre buscan la manera de mejorar la gestión de sus procesos, tratando de reducir y optimizar el tiempo, para así garantizar su productividad y competitividad en el mercado. Adicionalmente, con el fin de innovar implementan nueva maquinaria la cual ayuda automatizando los procesos de tareas repetitivas.

Adicionalmente, con la nueva modalidad de vida a causa de la pandemia generada por el SARS.CoV-2, la mayoría de superficies pueden ser consideradas una fuente de contagio, en este caso las monedas son fuente para tener una posibilidad muy alta de contagio debido a que estas pasan de persona en persona y pueden contener los microorganismos del virus ocasionando un contagio seguro en la persona que posea las monedas contaminadas [1]. Dadas estas razones se plantea el prototipo, con el fin de solucionar los inconvenientes mencionados.

1.1 Objetivo general

Implementar una máquina contadora y clasificadora de monedas automática que envíe una notificación a un número predeterminado.

1.2 Objetivos específicos

- Identificar los requerimientos necesarios para implementar el proyecto.
- Construir el *hardware* que permita la clasificación correcta de las monedas.
- Desarrollar el *software* a emplearse en el microcontrolador.
- Implementar el prototipo de la máquina contadora de monedas.
- Realizar pruebas de funcionamiento de la máquina y del envío de notificaciones al usuario final.

1.3 Fundamentos

Placas de desarrollo

- Arduino

Es una placa de desarrollo que funciona con el microprocesador ATMEGA 2560, la cual está conformada por diferentes elementos como son: pines de entrada y salida digitales, pines de entrada analógicos, regulador de 5 (V_{DC}) y 3.3 (V_{DC}), *jack* de alimentación DC, adaptador USB, pulsador de reseteo, *leds* indicadores, pines para conexión I2C, pines para transmisión y recepción de datos, entre muchas cosas más, en la Figura 1.1 se muestra la placa de desarrollo Arduino MEGA 2560 [2] [3].

Arduino MEGA 2560 emplea *software* libre, además contiene todos los recursos necesarios para dar soporte al microcontrolador y con esto desarrollar diferentes tipos de proyectos.

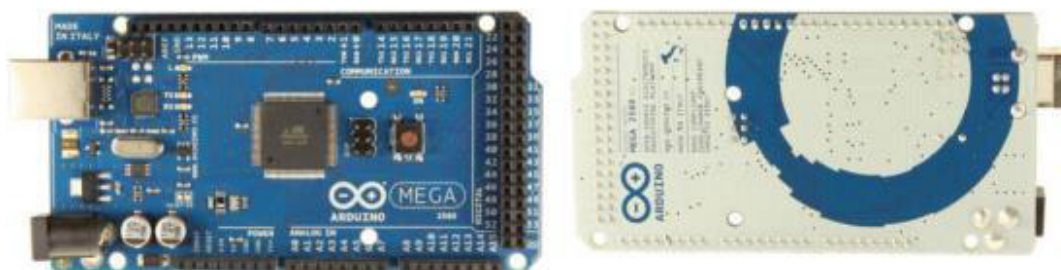


Figura 1.1 Placa de desarrollo [2]

- *Raspberry Pi*

Básicamente es un ordenador pequeño, tiene todos los recursos necesarios para cumplir con tareas básicas, además, puede ser programado y por consecuencia compila los programas que se ejecuten en el mismo [4]. Es de tamaño compacto y permite conectar varios tipos de accesorios dándole una versatilidad interesante, tiene como función principal enseñar informática [4]. Debido a que por medio de programas como *Python* o *Scratch* se aprende a programar [5].

Módulo RTC DS1307

Es un reloj en tiempo real/calendario, BCD (*Binary Codec Decimal* – Decimal Codificado en Binario) de baja potencia, que tiene como función llevar la cuenta de la fecha y la hora de manera autónoma, el módulo se muestra en la Figura 1.2. Además, se puede agregar una pila de litio en el módulo RTC DS1307 para que, en caso de que se pierda la fuente de alimentación, el módulo no desfase ni la fecha ni la hora [6].

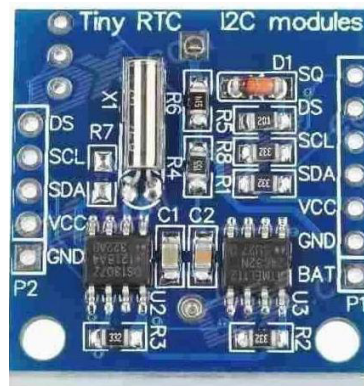


Figura 1.2 Módulo RTC DS1307 [7]

El módulo opera entre 3.3 (V_{DC}) y 5 (V_{DC}) positivos de alimentación, con la pila de litio puede llegar a funcionar aproximadamente por un periodo de 9 años. Este módulo tiene una interfaz I2C, dispone de dos circuitos integrados como se muestra en la Figura 1.2 cada uno cumple una función. En la Figura 1.3 se aprecia la distribución de pines y en la Tabla 1.1 con mayor detalle la función de cada pin [8].

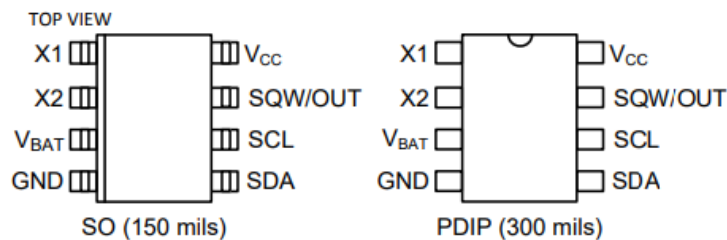


Figura 1.3 Configuración de pines del Módulo RTC DS1307 [9]

Tabla 1.1 Descripción de pines del Módulo RTC DS1307 [8]

Número de Pin	Nombre de Pin	Descripción
1,2	X1, X2	Oscilador de cristal, debe ir conectado a estos pines
3	V-Bat	Conecta el terminal positivo de la batería
4	GND	Conecta la parte negativa de la fuente de alimentación
5,6	SCL, SDA	Pines de comunicación I2C
7	SQW/OUT	Pin de salida para obtener frecuencias de onda cuadrada
8	V _{CC}	Conecta la parte positiva de la fuente de alimentación

Módulos GPRS/GSM

- Módulo IOT-GA6

El módulo IOT-GA6 es un *shield* en miniatura del módulo inalámbrico A6, el cual permite enviar y recibir SMS, datos, además de realizar y recibir llamadas de voz. Cuenta con 16 pines de conexión, comunicación UART, zócalo de conexión para la tarjeta SIM, tal como se muestra en la Figura 1.4 [10] [11].

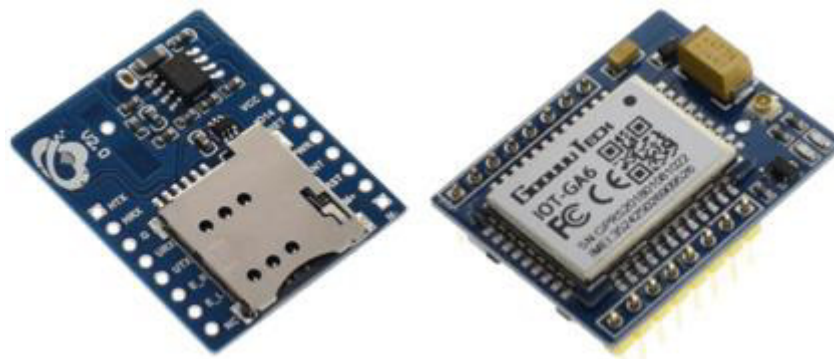


Figura 1.4 Módulo IOT-GA6 [10]

Una de las grandes ventajas de este módulo es su bajo consumo de corriente, se alimenta con 5 (V_{DC}) ya que contiene un *microchip* el cual cumple la función de regulador de voltaje y envía el voltaje necesario al *chip* [11].

Por otra parte, el módulo se puede controlar por medio de cualquier microcontrolador que cuente con comunicación UART (*Universal Asynchronous Receiver-Transmitter*) o en español (Receptor-Trasmisor Asíncrono Universal) [12]. Esto indica que podrá manejar comandos AT.

El módulo contiene dos *leds* indicadores tal como se muestra en la Figura 1.5, el primero es para indicar la alimentación y el segundo muestra el estado de conexión del módulo. De esta forma el segundo *led* oscila cada cierto tiempo, es decir, que si el *led* oscila constantemente quiere decir que el módulo aún no se ha podido conectar a la red, si el *led* oscila cada 3 (segundos) significa que se ha logrado conectar a la red [11].

Adicionalmente, en la Figura 1.5 se muestra la distribución de pines del módulo y en la Tabla 1.2 se indica a detalle la función de cada uno de los pines.

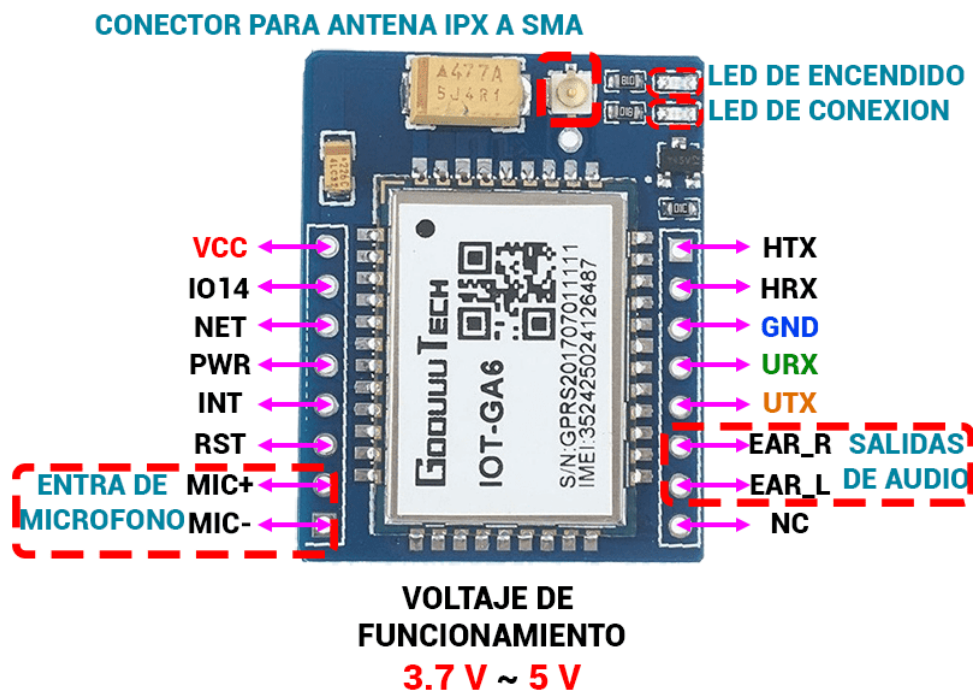


Figura 1.5 Distribución de pines del módulo IOT-GA6 [13]

No obstante, el módulo permite la conexión de una antena para facilitar la recepción de señal. Se tienen dos tipos de antena, helicoidal y PCB, las cuales se muestran en la Figura 1.6.

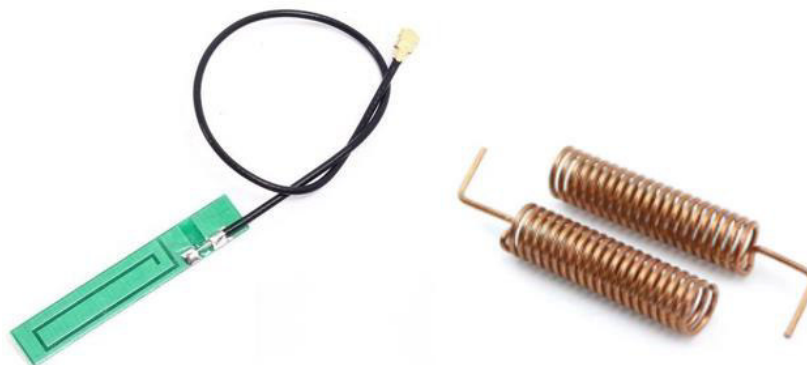


Figura 1.6 Antena PCB (izquierda) y antena helicoidal (derecha) [14] [15]

Tabla 1.2 Descripción de pines del Módulo IOT-GA6 [16]

Nombre de pin	Descripción
V _{cc}	Pin de alimentación positivo
IO14	Versión de <i>hardware</i> para GPIO14
NET	Pin de conexión para antena
PWR	Pin de botón de encendido
INT	Pin de control de modo de bajo consumo
RST	Pin de reseteo del módulo
MIC+, MIC-	Pines de conexión para micrófono
HTX	Pin transmisor de puerto serial de descarga
HRX	Pin receptor de puerto serial de descarga
GND	<i>Ground</i> , pin de alimentación negativo
URX	Pin receptor de comunicación UART
UTX	Pin transmisor de comunicación UART
EAR_R, EAR_L	Pines de conexión para auricular
NC	NC, V3 versión de <i>hardware</i> para GPIO3

- Módulo SIM800L

Es un módulo de telefonía celular en miniatura que permite la transmisión GPRS, envío y recepción de SMS, además de realizar y recibir llamadas de voz. También se controla por medio de cualquier microcontrolador que tenga una UART [17] [18].

El módulo funciona correctamente en un rango de alimentación de 3.7 (V_{DC}) a 4.4 (V_{DC}) con picos de corriente de 2 (A_{DC}). Tiene un *led*, el cual cumple la función de indicador, este *led* parpadea cada cierto tiempo y con ello se podrá saber el estado de la red de telefonía móvil a la cual se quiere conectar, posee 12 pines de conexión los cuales se muestran en la Figura 1.7 y su función en la Tabla 1.3 [19].

Se tiene los siguientes tiempos de oscilación del *led*:

- Oscilación cada 1 segundo: Indica que el módulo aún no se ha podido conectar a la red de telefonía móvil.
- Oscilación cada 2 (segundos): Indica que la conexión de datos GPRS que se solicitó está activa.
- Oscilación cada 3 (segundos): Indica que el módulo ha logrado conectarse a la red de telefonía móvil y que está listo para enviar y recibir datos o voz.



Figura 1.7 Distribución de pines [19]

Tabla 1.3 Descripción de la distribución de pines del módulo SIM800L [19]

Número de Pin	Nombre de Pin	Descripción
1	NET	Pin para soldar la antena helicoidal
2	Vcc	Pin de conexión positiva de la fuente de alimentación
3	RST	Pin de restablecimiento o reseteo
4	RXD	Pin RECEPTOR de comunicación serial
5	TXD	Pin TRANSMISOR de comunicación serial
6	GND	Pin de tierra
7,8	SPK-, SPK+	Pines para la conexión de un altavoz
9,10	MIC-, MIC+	Pines para la conexión de un micrófono
11	DTR	Pin de activación o desactivación del modo <i>sleep</i>
12	Ring	Pin de interrupción del módulo

Sensores detectores infrarrojos

- Módulo TCRT5000

Este es un sensor óptico reflexivo, posee una construcción compacta, donde una fuente de luz infrarroja y un detector de luz que consta de un fototransistor, están puestos en la misma dirección, de esta forma se puede detectar la presencia de objetos [20].

Posee también un circuito integrado LM393 el cual cumple con la función de comparador de voltaje. La precisión que tiene este módulo es una gran ventaja, además de su bajo y accesible precio en el mercado.

Se puede calibrar el sensor con un potenciómetro de precisión el cual se encuentra en la parte de atrás del módulo, en la Figura 1.8 se muestra al módulo TCRT5000, donde

también se observa que posee 4 pines de conexión los cuales son V_{CC} , GND, D0 y A0 [21] [22].

El módulo trabaja en un rango de 3.3 (V_{DC}) a 5 (V_{DC}), con un rango de detección de entre 0.2 (mm) a 15 (mm). Además, es capaz de generar una señal analógica y una señal digital a su salida [23] [24].

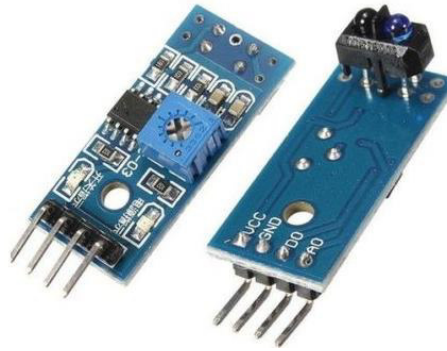


Figura 1.8 Módulo sensor infrarrojo TCRT5000 [25]

- CNY70

Comprende un sensor óptico de reflexión, tiene un diodo emisor de luz infrarroja y un fototransistor. Permite detectar objetos, incluso medir distancia, se muestra en la Figura 1.9. Posee las siguientes características [26] [27].

- Tipo de paquete: con terminales para soldar.
- Tipos de detector: fototransistor.
- Dimensiones: 7x7x6 (mm).
- Distancia de operación máxima: 5 (mm).
- Longitud de onda: 950 (nm).
- Corriente máxima de *led*: 50 (mA).
- Corriente máxima de transistor: 50 (mA).
- Voltaje de operación: 5 (V).



Figura 1.9 Sensor CNY70 [27]

2 METODOLOGÍA

2.1 Descripción de la metodología usada

Objetivo 1:

Para determinar los requerimientos se realizó una investigación sobre las diferentes áreas que abarcó el proyecto. En la parte del *hardware* se implementó un método mecánico usado en proyectos similares, el cual basa su funcionamiento acorde al diferente tamaño en área y perímetro de cada moneda. Para el *software* se analizó qué plataforma de desarrollo de *software* libre permitía la creación del código fuente y que fuese capaz de satisfacer todos los requerimientos y necesidades que tuvo el proyecto.

El prototipo debe enviar la información final de manera inalámbrica, por ende, se analizó algunas formas de hacerlo tales como GPRS, GSM, *Bluetooth*, entre otras. Se identificó el tipo de red que se adaptó de la mejor manera y que cumplió con las necesidades del prototipo.

Además, se investigaron los tipos de componentes electrónicos que se necesitaron, tales como sensores, reguladores de voltaje, *shields*, entre otros con el objetivo de obtener los componentes electrónicos adecuados y que se adaptaron de la mejor manera al proyecto.

Por último, se hizo una recopilación de todos los datos obtenidos de la investigación y se seleccionó el *hardware*, *software* y componentes electrónicos adecuados para la implementación del proyecto.

Objetivo 2:

En base a los requerimientos del prototipo se diseñó el *hardware* requerido para la implementación del prototipo de clasificación de monedas de manera mecánica, y además se seleccionaron los elementos electrónicos necesarios para el conteo de las monedas.

Para la parte del sistema de conteo electrónico se investigó qué tipo de sensores eran los más adecuados y que se adaptaban de la mejor manera, los cuales permiten hacer el conteo de las monedas de manera correcta y eficiente a través de la parte mecánica del prototipo.

Además, se seleccionó un dispositivo que permite enviar la información final del conteo de las monedas de manera inalámbrica a un usuario determinado. Por lo que se tomó

la opción de un módulo de telefonía móvil GPRS/GSM, que por medio de la red de telefonía móvil envía un mensaje de texto con la información correspondiente al usuario determinado.

Objetivo 3:

Se utilizó un sistema microprocesado que permitió tomar un algoritmo y codificarlo en una notación, de modo que pudo ser ejecutado por una computadora, es decir, que se utilizó un *software* de programación de alto nivel, también que contuvo todas las funciones que se necesitó ejecutar para la implementación del proyecto.

Se tomó en cuenta que el sistema microprocesado fuera capaz de ejecutar la parte de comunicaciones inalámbricas, y que pueda usar los diferentes tipos de redes inalámbricas, con el objetivo de que satisfaga el envío de la información final al usuario determinado.

Además, que contó con la cantidad de recursos necesarios tales como: librerías, adaptación de *shields*, entre otros. Es un *software* multiplataforma el cual puede trabajar en las principales plataformas (Mac, Windows) y logró satisfacer todas los requerimientos y necesidades del proyecto.

Objetivo 4:

Una vez que se diseñó el *hardware* adecuado y se realizó la programación del sistema microprocesado se realizó la implementación del prototipo para lo cual se construyó los circuitos electrónicos necesarios.

Se usó el método de construcción de circuitos electrónicos por baquelita o circuito impreso PCB (*Printed Circuit Board*), por lo que se utilizó el *software* ISIS *Proteus* que por medio de sus herramientas ARES (*Advanced Routing and Editing Software*) permitió la fabricación de las placas de circuito impreso que se necesitó. Además, se utilizó la baquelita del tipo perforada para los circuitos electrónicos relativamente fáciles de hacer o los que contenían pocos elementos electrónicos, con el objetivo de economizar.

Objetivo 5:

Se hicieron pruebas de validación del funcionamiento del dispositivo y se identificaron las fallas del mismo para solventarlas. Se verificó que la información llegue sin errores de manera inalámbrica por medio de la red de telefonía móvil. Y con ello se garantizó el funcionamiento correcto y adecuado del prototipo.

3 RESULTADOS Y DISCUSIÓN

En esta parte se explica la manera en la que se identificaron y escogieron los requerimientos del prototipo, los diseños e implementación de cada elemento del prototipo tanto de *software* como *hardware*, además el desarrollo del código de programación, finalmente las pruebas de funcionamiento del prototipo.

3.1 Identificación y selección de los requerimientos necesarios para implementar el proyecto

Selección del método de clasificación de las monedas

Para esto fue necesario realizar una investigación concreta sobre aspectos y características de cada moneda, el más importante conocer el diámetro de las mismas, ya que con este dato se diseñó y dimensionaron las medidas de la barra clasificadora de monedas.

Cabe mencionar que en Ecuador circulan dos tipos de monedas, las acuñadas en los Estados Unidos de América y las monedas nacionales, cada una con símbolos e identificativos propios de cada país, pero ambas con el mismo valor para cada denominación [28].

A continuación, se detalla los aspectos importantes de cada tipo de moneda ordenadas desde la más pequeña hasta la más grande en función del diámetro de las mismas. En la Figura 3.2 se muestra la moneda ecuatoriana de 10 centavos, en la Figura 3.1 la versión americana, las figuras seguirán este mismo esquema para las monedas de 1 centavo, 5 centavos, 25 centavos, 1 dólar y 50 centavos. Por lo que la información de las imágenes de las monedas incluye las figuras de la Figura 3.1 a la Figura 3.11, así como las tablas de la Tabla 3.1 a la Tabla 3.6.

- Moneda de 10 centavos



Figura 3.1 Moneda de 10 centavos estadounidense [29]



Figura 3.2 Moneda de 10 centavos nacional [30]

Tabla 3.1 Información de la moneda de 10 centavos [29]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cobre bañado en cuproníquel
Diámetro (mm)	17.91
Grosor (mm)	1.3
Peso (g)	2.27

- Moneda de 1 centavo



Figura 3.3 Moneda de 1 centavo estadounidense [31]



Figura 3.4 Moneda de 1 centavo nacional [30]

Tabla 3.2 Información de la moneda de 1 centavo [31]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cinc bañado en cobre
Diámetro (mm)	19.05
Grosor (mm)	1.55
Peso (g)	2.5

- Moneda de 5 centavos



Figura 3.5 Moneda de 5 centavos estadounidense [32]



Figura 3.6 Moneda de 5 centavos nacional [30]

Tabla 3.3 Información de la moneda de 5 centavos [32]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cuproníquel
Diámetro (mm)	21.21
Grosor (mm)	1.95
Peso (g)	5

- Moneda de 25 centavos



Figura 3.7 Moneda de 25 centavos estadounidense [33]



Figura 3.8 Moneda de 25 centavos nacional [30]

Tabla 3.4 Información de la moneda de 25 centavos [33]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cobre bañado en cuproníquel
Diámetro (mm)	26.26
Grosor (mm)	1.75
Peso (g)	5.67

- Moneda de 1 dólar



Figura 3.9 Moneda de 1 dólar estadounidense [34]

Tabla 3.5 Información de la moneda de 1 dólar [34]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cobre bañado en cuproníquel
Diámetro (mm)	25.26
Grosor (mm)	2
Peso (g)	8.2

- Moneda de 50 centavos



Figura 3.10 Moneda de 50 centavos estadounidense [35]



Figura 3.11 Moneda de 50 centavos nacional [30]

Tabla 3.6 Información de la moneda de 50 centavos [35]

Característica	Descripción
País de origen	Estados Unidos
Aleación	Cobre bañado en cuproníquel
Diámetro (mm)	30.61
Grosor (mm)	2.15
Peso (g)	11.34

Acorde a esta información se optó por un método de clasificación, el cual funcione en base al diámetro de cada moneda, por ende, se diseñó una barra clasificadora de monedas, el diseño e implementación de la misma se muestra más adelante.

Selección del método de conteo y envío de datos del prototipo

Uno de los requerimientos del prototipo aparte de la clasificación de las monedas, es el conteo de las mismas. Por lo tanto, es necesario el uso de un sensor el cual sea capaz de realizar esta función.

Al hacer la investigación pertinente, se llegó a la conclusión, que el mejor sensor es del tipo infrarrojo, debido a que en el diseño de la máquina contadora las monedas caerán por medio de un “camino” hacia los contenedores, entonces al poner un sensor del tipo infrarrojo se tendrá el haz de luz infrarroja atravesando el “camino”, por tanto, al momento de pasar la moneda el sensor la detectará y llevará a cabo el conteo.

Por otra parte, se requiere enviar la información del conteo de monedas de manera inalámbrica, por lo que por medio de la red de telefonía móvil se realizará el envío de un mensaje de texto con dicha información.

Tomando en cuenta estos requerimientos se optó por el uso de un módulo GPRS/GSM, en sí, se usará la red GSM debido a que el envío de la información del conteo de monedas se hará por medio de mensajes de texto. Además, al enviar la información por mensajes de texto se usará la red 2G por lo que se tendrá una mayor cobertura y no será necesaria una velocidad de transmisión alta para el envío de los datos. Por ende, 2G es la red ideal para este proceso.

Selección de la plataforma de desarrollo de *software*

En primer lugar, se tomó en consideración algunas plataformas de desarrollo principalmente las que son de *hardware* libre, en cuanto a los requerimientos del prototipo se necesita entradas y salidas digitales para la conexión de sensores, pulsadores, interruptores, *leds*, elementos electrónicos y *shields*. Asimismo, funciones de comunicación (Tx y Rx) para la conexión del módulo GSM y (SDA – SCL) para la conexión del módulo RTC.

Una de las plataformas es Arduino, la cual cumple con todos los requerimientos, pero también está *Raspberry Pi* la cual puede adaptarse de igual forma, en la Tabla 3.7 se observan las diferencias entre estas dos plataformas de desarrollo de *software*.

Tabla 3.7 Comparativa entre Arduino y *Raspberry Pi* [36]

Arduino	<i>Raspberry Pi</i>
Precio económico en el mercado	No tan económico a comparación de Arduino
Contiene un solo puerto USB <i>Type-B</i> hembra para conexión al PC	Tiene 4 puertos USB para conectar distintos dispositivos
Sus componentes y sensores funcionan de manera integrada	Se debe instalar librerías y <i>software</i> para interactuar con sensores y demás componentes
Utiliza un procesador de AVR	Utiliza procesadores ARM
Se conecta a internet por puerto RJ-45	Necesita <i>hardware</i> externo para conectarse a la red
Puede trabajar con batería	Es complejo hacerlo funcionar con batería.
Utiliza IDE Arduino y C/C++	Se recomienda utilizar <i>Python</i> , pero puede usar C, C++
Ejecuta un programa una y otra vez	Puede ejecutar varios programas simultáneamente

En comparación, Arduino ofrece más ventajas, costo menor en el mercado, mejores formas de conexión con elementos externos ya sea conexión a la red o conexión de *shields*, además se tiene conocimiento sobre el manejo del entorno de Arduino por lo que se puede programar sin problemas, adicionalmente posee una gran cantidad de foros y recursos en Internet creados por la comunidad, los cuales facilitan el aprendizaje y son de gran ayuda.

Por estas razones y en función de los requerimientos del prototipo se optó por la placa de desarrollo Arduino Mega 2560, debido a su gran capacidad de entradas y salidas digitales y también porque cuenta con las funciones de comunicación y los recursos antes mencionados.

Selección del *hardware*

Para esto se recopiló toda la información obtenida de la investigación, de esta forma se logró identificar los componentes electrónicos necesarios para la implementación del prototipo, los cuales se los detalla a continuación.

Selección del sensor de conteo de monedas

En términos generales se necesitaba un componente electrónico que sea capaz de detectar el paso de las monedas, concretamente un sensor, como consecuencia se eligió un sensor del tipo infrarrojo, se tuvieron varias opciones de las cuales se optó por el módulo TCRT500 y el sensor CNY70, como resultado se obtuvo la comparativa que se aprecia en la Tabla 3.8.

Tabla 3.8 Diferencias entre sensores infrarrojos [21] [27]

Módulo TCRT5000	Sensor CNY70
Posee circuito acondicionador	No tiene acondicionamiento
Permite regular la distancia de detección	Se debe crear un circuito para ajustar la distancia de detección
Tamaño relativamente grande en comparación del sensor CNY70	Tamaño diminuto
Es de gran precisión	No es tan preciso
Ofrece señal analógica y digital de salida	Solo ofrece señal digital de salida

En base a las características de cada uno, se optó por el módulo TCRT5000 siendo el que mejor se adaptó a los requerimientos y necesidades del prototipo, principalmente por la precisión que posee, además de la alternativa de calibrar el sensor por medio del potenciómetro de precisión que tiene integrado y por su accesible precio en el mercado.

Posteriormente se realizaron las pruebas de funcionamiento del módulo y cumplió de gran manera con su cometido. Adicionalmente, se aprovechó la salida del pin digital del módulo para que se envíe a la placa de desarrollo Arduino Mega 2560 y con ello cumplir con el conteo y detección de monedas.

Interfaz de usuario

Es importante destacar que el prototipo maneja datos almacenados en variables, por ende, se necesita visualizar los mismos, además que la máquina contadora de monedas tiene opciones de manejo. En este sentido se optó por una pantalla LCD de cristal líquido la cual se aprecia en la Figura 3.12.

Este tipo de pantallas son de distintos tamaños, existen pantallas LCD 16x2 y 16x4, la principal diferencia es que la segunda tiene dos filas más que la primera. Por ende, se decidió utilizar la pantalla LCD 16X4 debido a que se necesita visualizar en el conteo el número total de monedas y la cantidad de dinero, además, la cantidad individual de

monedas para cada denominación, por lo tanto, se requería una pantalla que despliegue toda esa información y que sea de gran tamaño. No obstante, la pantalla de 16x2 no era lo suficientemente amplia, por esta razón se la descartó.

Adicionalmente, este tipo de pantalla se puede gestionar por medio de librerías ya existentes y facilita su programación, por estas razones se concluyó que esta pantalla cumple con los requerimientos que se necesita.



Figura 3.12 Pantalla LCD 16x4 [37]

La pantalla LCD 16x4 consta de 16 pines, los presentes cumplen una función específica, en la Tabla 3.9 se muestra con detalle la función de los mismos.

Tabla 3.9 Especificación de funciones de pines de la pantalla LCD 16x4 [38]

No. Pin	Símbolo	Función
1	V_{SS}	Fuente de alimentación pin negativo
2	V_{DD}	Fuente de alimentación pin positivo
3	V_O	Ajuste de contraste
4	RS	Señal de selección de datos/instrucciones
5	R/W	Señal de selección de lectura/escritura
6	E	Habilitador de señal
7 – 14	DB0 – DB7	Línea de bus de datos
15	A	Pin de alimentación positiva para el <i>led</i>
16	K	Pin de alimentación negativa para el <i>led</i>

Selección del regulador de voltaje

El prototipo contiene sensores y varios componentes electrónicos que necesitan una fuente de alimentación, en las especificaciones de los mismos se indica el voltaje de operación ideal siendo 5 (V_{DC}). La fuente de alimentación principal que se usa tiene una salida de voltaje de 12 (V_{DC}), por ende, se tiene la necesidad de obtener 5 (V_{DC}) de alimentación, para ello se necesitó la implementación de un regulador de voltaje, entre estos se consideraron dos, el regulador LM7805 y LM2596.

El regulador LM2596 soporta entradas de voltaje de 0.3 (V_{DC}) a 24 (V_{DC}) y corrientes de salida de hasta 2.5 (A_{DC}), posee un potenciómetro el cual regula la salida de voltaje. En cambio, el regulador LM7805 es del tipo lineal, soporta entradas de voltaje de hasta 35 (V_{DC}) y a la salida siempre entregará 5 (V_{DC}) [39] [40].

Acorde a los requerimientos se optó por el regulador de voltaje LM7805, principalmente porque siempre proporcionará una salida de voltaje de 5 (V_{DC}), no obstante, para el regulador LM2596 se debe ajustar por medio del potenciómetro la salida de 5 (V_{DC}). Además, según las especificaciones el regulador LM7805 funcionará correctamente ya que la entrada que recibirá de voltaje está acorde a los rangos de funcionamiento del mismo.

Adicionalmente, se diseñó un circuito de acondicionamiento para asegurar el correcto funcionamiento del regulador de voltaje, el cual se detalla más adelante. Este regulador es de tamaño compacto y posee tres pines de conexión como se muestra en la Figura 3.13, el primer pin es de entrada de voltaje, el segundo de conexión a tierra y el tercero es el pin de salida de voltaje.

El regulador LM7805 es económico en comparación al regulador LM2596, una gran desventaja del regulador LM2596 era que se necesitaba de un multímetro para conocer la salida de voltaje, por ende, si no se ajustaba bien la salida de voltaje se corría con el riesgo de dañar los componentes electrónicos o tener un mal funcionamiento de los mismos. Por estas razones se eligió el regulador LM7805.

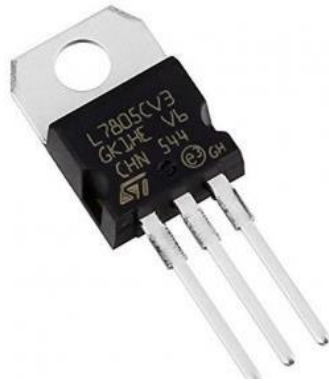


Figura 3.13 Regulador de voltaje LM7805 [41]

Selección del dispositivo de fecha y hora

Por otra parte, uno de los requerimientos para el envío de los mensajes de texto es que se envíe la fecha y hora en conjunto con la información del conteo de monedas. Esto se lo hace con el objetivo de mantener un registro.

Por ejemplo, se puede dar la situación de que el usuario se encuentre en un lugar en el cual no se tenga acceso a la señal de telefonía móvil, por lo que no recibirá el mensaje de texto a pesar de que este haya sido enviado. En muchos casos al momento que se tenga nuevamente señal celular y se reciba el mensaje de texto, los datos de fecha y hora en el registro del teléfono no serán exactos, por esta razón se envía la fecha y hora en el mensaje de texto para asegurar la exactitud del momento en el cual se envió el mensaje al usuario determinado.

Por estas razones se decidió el uso de un módulo RTC (*Real Time Clock*), como opciones se tomaron el módulo *Tiny* RTC DS1307 y DS3231. Los dos módulos poseen características similares incluso algunas iguales, por ejemplo, los dos funcionan por medio de comunicación I2C, funcionan con voltajes de 3.3 (V_{DC}) a 5 (V_{DC}), ambos son de bajo consumo, incorporan una pequeña memoria EEPROM, el precio en el mercado del módulo D1307 es menor al del D3231, pero la principal diferencia entre estos módulos es la batería [42].

La batería asegura que no se desfase la hora ni la fecha, para el módulo DS1307 puede durar años en cambio el módulo D3231 solo meses. Dada esta situación se optó por el módulo DS1307 [7] [9].

Selección del módulo GPRS/GSM

Para cumplir el requerimiento de enviar la información del conteo de monedas de forma inalámbrica y que se use la red de telefonía móvil, se escogió la implementación de un

módulo GPRS/GSM, entre los cuales se consideró dos tipos muy comunes y en base a los fundamentos se obtuvo la comparativa que se aprecia en la Tabla 3.10.

Tabla 3.10 Cuadro comparativo de módulos GPRS/GSM [10] [11] [19]

Módulo SIM800L	Módulo IOT-GA6
Posee 12 pines de conexión	Cuenta con 16 pines de conexión
Necesita una fuente de poder de entrada estable de entre 3.7 (V _{DC}) a 4.3 (V _{DC})	Tiene su propio regulador de voltaje interno, por lo que acepta 5 (V _{DC})
Tiene un solo <i>led</i> indicador	Posee dos <i>leds</i> indicadores
Soporta solo tecnología 2G	Trabaja con tecnología 2G, 3G, 4G
Corriente para modo <i>sleep</i> 0.7 (A _{DC})	Corriente de modo <i>sleep</i> 0.9 (A _{DC})

Algunas características similares entre estos dos módulos son, los dos funcionan con 2 (A_{DC}), ambos pueden hacer y recibir llamadas de voz, también envían y reciben mensajes de texto, los dos se manejan con comandos AT por lo que deben usar una UART, poseen conectores para conectar los dos tipos de antenas antes mencionadas, los dos son de un tamaño compacto y comparten un precio similar en el mercado. Ambos módulos presentaron un funcionamiento correcto, siempre y cuando se suministraban las condiciones de trabajo adecuadas (corriente principalmente).

Al hacer las pruebas con el módulo SIM800L se tuvieron problemas con la fuente de alimentación debido a que se tenía que suministrar exactamente 4.2 (V_{DC}) con 2 (A_{DC}) de lo contrario el módulo presentaba un funcionamiento incorrecto, incluso no lograba conectarse a la red de telefonía móvil, por ende, no se enviaba el mensaje de texto, además, en repetidas ocasiones se desconectaba de la red de telefonía móvil debido a que no se dispone de una fuente de poder externa estable regulable.

Sin embargo, se implementó un regulador de voltaje, el LM2596 antes mencionado para suministrar los 4.2 (V_{DC}), pero como desventaja este módulo reducía la corriente por ende no podía suministrar los 2 (A_{DC}) que necesita el módulo, entonces se descartó esta opción. No obstante, el módulo IOT-GA6 al tener su regulador de voltaje interno no presentó problemas en la parte de alimentación, se pudo configurar y programar correctamente, por ende, permitió enviar el mensaje de texto exitosamente.

En base a estos resultados obtenidos se optó por el módulo IOT-GA6, debido a que cumplió con el envío del mensaje de texto sin problemas, además de no presentar inconvenientes en la parte de alimentación, en consecuencia, tampoco en su funcionamiento. Adicionalmente se observó que el módulo se logra conectar a la red de

telefonía móvil en un lapso aproximado de 5 (segundos) desde el momento que recibe alimentación.

Selección de las fuentes de alimentación

El prototipo tiene diferentes elementos electrónicos los cuales necesitan un determinado voltaje y amperaje para funcionar correctamente, en la Tabla 3.11 se muestra el consumo de voltaje y corriente de funcionamiento de cada elemento.

Tabla 3.11 Consumo de corriente y voltaje de elementos electrónicos

Cantidad	Elemento	Consumo de voltaje	Consumo de corriente	TOTAL Consumo de corriente
1	Arduino Mega 2560	7 (V _{DC}) – 12 (V _{DC})	600 (mA _{DC})	600 (mA _{DC})
6	Módulo TCRT5000	5 (V _{DC})	60 (mA _{DC})	360 (mA _{DC})
1	Ventilador de PC	12 (V _{DC})	1.2 (A _{DC})	1.2 (A _{DC})
1	LM7805	Máximo 35 (V _{DC})	1 (A _{DC}) salida	1 (A _{DC})

Con el objetivo de economizar y reutilizar, se obtuvo una fuente de poder de una PC la cual no estaba en funcionamiento, por ende, se realizaron las pruebas de medición de voltaje y de corriente para verificar si la fuente de poder estaba en óptimas condiciones, como resultado se observó que funciona correctamente. Una gran ventaja es que posee 3 pines de salida tal como se muestra en la Figura 3.14 cada uno suministra un voltaje de 12 (V_{DC}) y corriente de 1.2 (A_{DC}), según las características que se observa en la placa.

De este modo, al tener 3 pines de salida se designó cada pin para ciertos elementos, a continuación, se detalla la distribución realizada.

- *Arduino Mega 2560*

En base a la Tabla 3.11, la placa Arduino Mega 2560 tiene un consumo de voltaje de 7 (V_{DC}) a 12 (V_{DC}) y un consumo de corriente de 600 (mA_{DC}). Por lo tanto, se le designó un pin de salida de la fuente de poder debido a que el pin de salida entrega 12 (V_{DC}) y el consumo de voltaje de la placa de Arduino Mega 2560 acepta 12 (V_{DC}), como consecuencia cumple con los niveles de voltaje que se necesita, por otra parte, el pin de salida entrega 1.2 (A_{DC}) y la placa de Arduino Mega 2560 consume 600 (mA_{DC}) por ende, también cumple con los requerimientos de amperaje.

- Ventilador de PC

El ventilador de PC requiere un voltaje de 12 (V_{DC}) y una corriente de 1.2 (A_{DC}), por ende, se le otorgó un pin de salida de la fuente de poder, ya que este cumple con los requerimientos de voltaje y corriente. Cabe mencionar que el ventilador también se lo reutilizó de la PC.

- Regulador de voltaje y módulo TCRT5000

El último pin de salida de la fuente de poder se lo asignó al regulador de voltaje LM7805 debido a que se encarga de proporcionar los 5 (V_{DC}) que necesitan los sensores detectores.

Por ende, el regulador LM7805 soporta la entrada de 12 (V_{DC}) de la fuente de poder y además funciona correctamente con la corriente de 1.2 (A_{DC}). Por lo tanto, la salida del regulador LM7805 dará un voltaje de 5 (V_{DC}) y una corriente de 1 (A_{DC}).

Según la Tabla 3.11 el módulo TCRT5000 funciona con 5 (V_{DC}) y tiene un consumo de corriente de 60 (mA_{DC}). En base a esta información, el voltaje entregado por el regulador LM7805 cumple con el consumo de voltaje para cada sensor y al tener un total de 6 sensores el consumo de corriente total es de 360 (mA_{DC}). El regulador LM7805 entrega una corriente de 1 (A_{DC}) como consecuencia cumple con la demanda de consumo de corriente.

En base a estos cálculos se concluyó que esta fuente de poder satisface la demanda de consumo de voltaje y corriente de los diferentes elementos.



Figura 3.14 Fuente de alimentación principal

Por otra parte, el módulo GPRS/GSM tiene un consumo de voltaje de 5 (V_{DC}) y corriente de 2 (A_{DC}), por lo que la fuente de alimentación principal no cumple con estos requerimientos, por ende, se necesitó otra fuente de poder que se consiguió de un cargador de teléfono móvil la cual se muestra en la Figura 3.15, según las

especificaciones y características se observó que posee un voltaje de 5 (V_{DC}) y corriente de 2 (A_{DC}) de salida, siendo así lo suficientemente capaz para satisfacer los requerimientos de voltaje y corriente del módulo GPRS/GSM.

Al hacer pruebas con el módulo GPRS/GSM se observó un funcionamiento correcto, por ende, se optó por esta fuente de poder.

La principal razón de la implementación de esta fuente de poder externa, es debido a que la fuente de poder principal no satisface el consumo de corriente de módulo GPRS/GSM.



Figura 3.15 Fuente de alimentación para módulo GPRS/GSM

Para la alimentación del módulo RTC y la pantalla LCD se utilizó las fuentes de poder que tiene la placa de Arduino Mega 2560, usando la fuente de 3.3 (V_{DC}) para el módulo RTC y la fuente de 5 (V_{DC}) para la pantalla LCD. Además, las fuentes de poder suministran la corriente adecuada de funcionamiento para cada elemento, la corriente máxima del módulo RTC es de 1.5 (mA_{DC}) y la fuente de poder de 3.3 (V_{DC}) da una corriente de 50 (mA_{DC}), por lo que es suficiente para asegurar un funcionamiento correcto del mismo.

3.2 Diseño de la máquina contadora

Selección del material para la barra clasificadora

En cuanto a la construcción del *hardware* se utilizó aluminio, específicamente un tubo de aluminio con un ángulo de 90 grados, el mismo que se puede ver en la Figura 3.16, se eligió este material debido a que las monedas circulan de manera efectiva sin inconvenientes, lo cual no sucedió con otros materiales que se consideraron como la madera y el plástico, ya que estos tuvieron problemas de circulación de monedas de 1 centavo y de 1 dólar específicamente, pero se dio solución al emplear tubo de aluminio.



Figura 3.16 Tubo de aluminio

Colocación y dimensionamiento de las medidas para cada moneda en la barra clasificadora

En primer lugar, se tomó en cuenta el diámetro de cada moneda, los cuales se encuentran desde la Tabla 3.1 hasta la Tabla 3.6, acorde al diámetro de cada moneda se realizó la medida necesaria.

Es importante mencionar que se necesitó un espacio de 5 (mm) en la parte inferior el mismo que se puede apreciar en la Figura 3.17, el cual da estabilidad a la moneda al momento de circular.

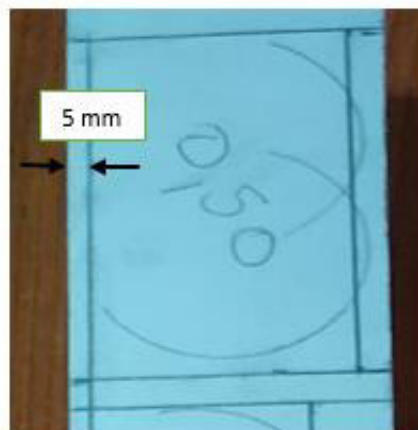


Figura 3.17 Medida del espacio inferior

A continuación, en la Figura 3.18 se muestra las medidas realizadas para la moneda de 10 centavos, las medidas de las monedas restantes se aprecian desde la Figura 3.19 hasta la Figura 3.23.

- Moneda de 10 centavos

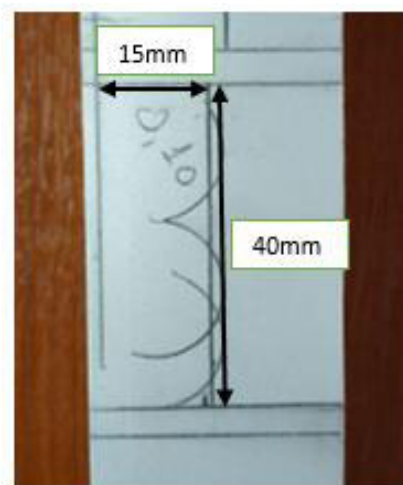


Figura 3.18 Medida de la moneda de 10 centavos

- Moneda de 1 centavo

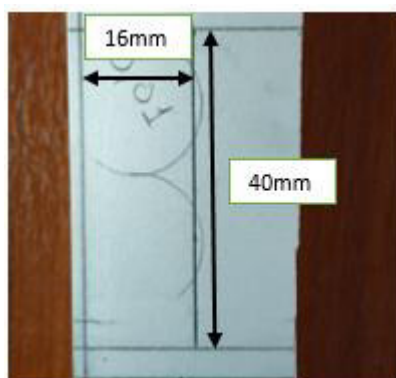


Figura 3.19 Medida de la moneda de 1 centavo

- Moneda de 5 centavos

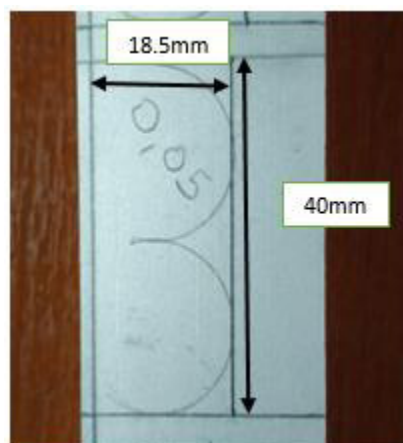


Figura 3.20 Medida de la moneda de 5 centavos

- Moneda de 25 centavos

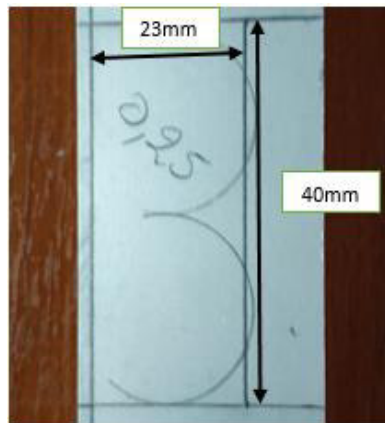


Figura 3.21 Medida de la moneda de 25 centavos

- Moneda de 1 dólar

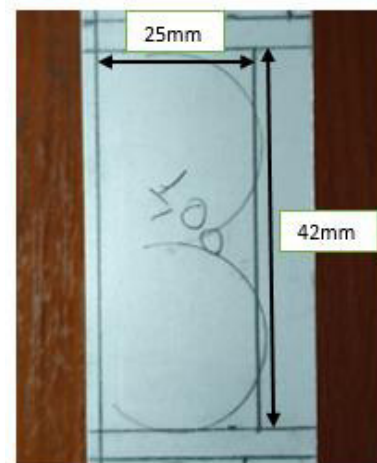


Figura 3.22 Medida de la moneda de 1 dólar

- Moneda de 50 centavos

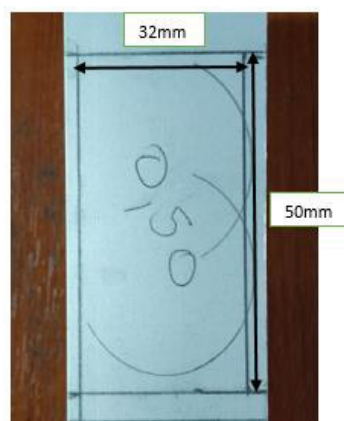


Figura 3.23 Medida de la moneda de 50 centavos

Diseño del separador de monedas

Por otra parte, se necesitó implementar un separador de monedas de forma triangular, 6 en total, los cuales cumplen con la función de llevar a la moneda por el camino correcto. Las medidas y diseño del separador se muestran en la Figura 3.24.

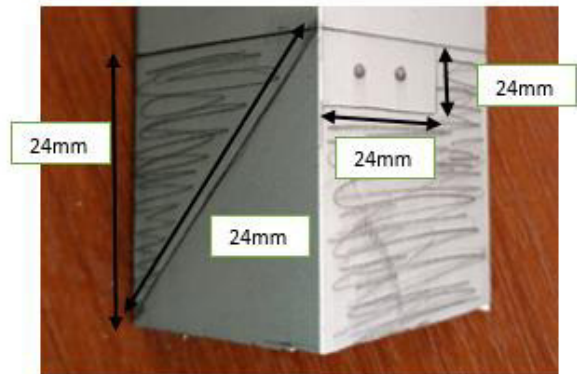


Figura 3.24 Diseño y medidas del separador de monedas.

Diseño del contenedor de monedas

Desde el punto de vista mecánico se necesitaba hacer un camino por el cual las monedas circulen y se clasifiquen hasta llegar a su respectivo depósito o contenedor. Por lo tanto, se diseñó una pieza de madera la cual se muestra en la Figura 3.25 la misma que cumple la función de soporte.

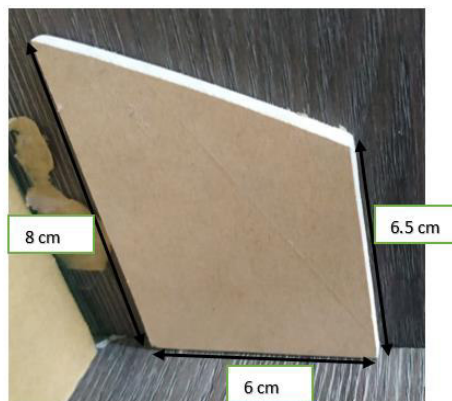


Figura 3.25 Soporte de camino

Diseño del distribuidor de monedas

En base a los requerimientos, se necesita un dispositivo el cual ayude a que las monedas puedan ingresar una por una a la máquina contadora para cumplir con el proceso de clasificación y conteo. Asimismo, que este dispositivo cuente con una estructura la cual guíe a las monedas dentro de la máquina contadora.

Por ende, se diseñó un envase en el cual se vierten las monedas y de manera automática este empieza a distribuir las mismas. El funcionamiento se basa en generar una vibración y de esta manera las monedas pueden caer. La estructura tendrá piezas que ayuden a las monedas a seguir el camino correcto.

Diseño del panel de control

Se diseñó un panel de control el cual se ubica en la parte frontal, el mismo se lo hizo de madera. Para el diseño se colocaron los pulsadores, *switch* y pantalla en la parte derecha. Tal como se muestra en la Figura 3.26 se colocó el orificio de la pantalla en la parte superior derecha en conjunto con el *switch* de encendido y apagado, abajo se colocaron los orificios para los pulsadores, *led* indicador y *switch* de selección de modo.

Básicamente se midieron los elementos y se colocaron las medidas respectivas para luego realizar los orificios para cada uno de ellos.

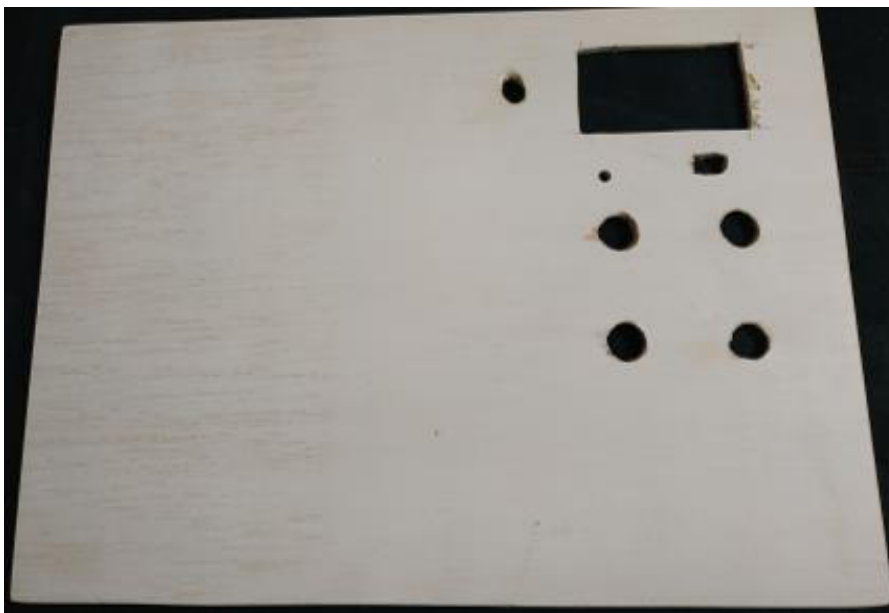


Figura 3.26 Diseño del panel de control

Diseño de los circuitos electrónicos

En primer lugar, se eligió el programa ISIS *Proteus* que sirve para crear y diseñar diferentes tipos de circuitos electrónicos, la gran ventaja que posee es su herramienta ARES la misma que permite crear circuitos impresos o PCB, se debe modificar las capas *Bottom Copper* y *Top Copper* para crear las pistas conductoras y la capa *Top Silk* para la señalización de los componentes electrónicos.

Fue necesario acorde a los requerimientos el diseño de 3 tipos de circuitos electrónicos los cuales son:

- Placa fuente reguladora de voltaje

Como se mencionó anteriormente los elementos electrónicos que se usan funcionan en su mayoría con 5 (V_{DC}), al tener una fuente de voltaje principal de 12 (V_{DC}), se implementó un regulador de voltaje, al cual se le diseñó un circuito de acondicionamiento para garantizar un funcionamiento correcto, el circuito en ISIS *Proteus* se lo aprecia en la Figura 3.27, su diseño en ARES en la Figura 3.28 y los componentes que lo conforman en la Tabla 3.12.

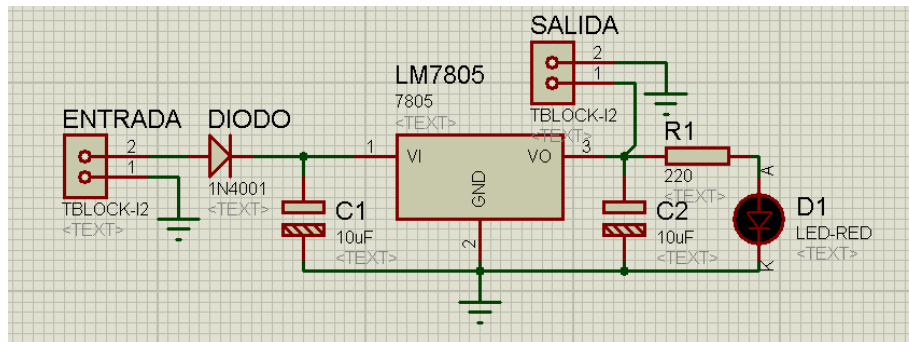


Figura 3.27 Diseño del circuito del regulador de voltaje

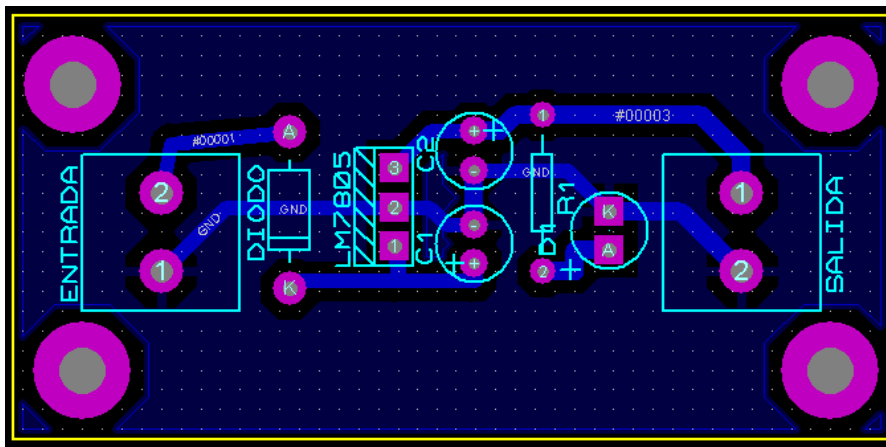


Figura 3.28 Diseño PCB del circuito del regulador de voltaje

Tabla 3.12 Lista de componentes del circuito del regulador de voltaje

Cantidad	Elemento electrónico
1	Regulador de voltaje LM7805
1	Diodo rectificador 1N4001
1	Resistencia de 220 (Ω)
1	Led
2	Capacitores de 10 (μ F)
2	Tblock de 2 posiciones

El circuito se basa en el regulador LM7805 al cual ingresan 12 (V_{DC}) por medio del *Tblock* denominado “ENTRADA”, tiene un diodo rectificador denominado “DIODO” el cual no dejará pasar la corriente en caso de estar conectada al revés la entrada de voltaje, se conecta dos capacitores electrolíticos denominados “C1 y C2” a la entrada y salida de voltaje del regulador para mantener el voltaje estable, se colocó un *led* denominado “D1” para conocer si la placa está en funcionamiento con su respectiva resistencia de protección llamada “R1”, finalmente se colocó un *Tblock* a la salida de voltaje del regulador denominado “SALIDA”, por medio de este *Tblock* se obtendrá los 5 (V_{DC}) que se necesita para alimentar a los elementos electrónicos.

- Placa de control de pulsadores y *led* indicador

El prototipo consta de pulsadores que activan funciones, un *switch* de dos posiciones que permite seleccionar el modo de funcionamiento y un *led* indicador. Por lo que en la Figura 3.29 se muestra el diseño en el programa ISIS Proteus, en la Figura 3.30 se aprecia su diseño PCB y en la Tabla 3.13 los componentes que conforman la placa.

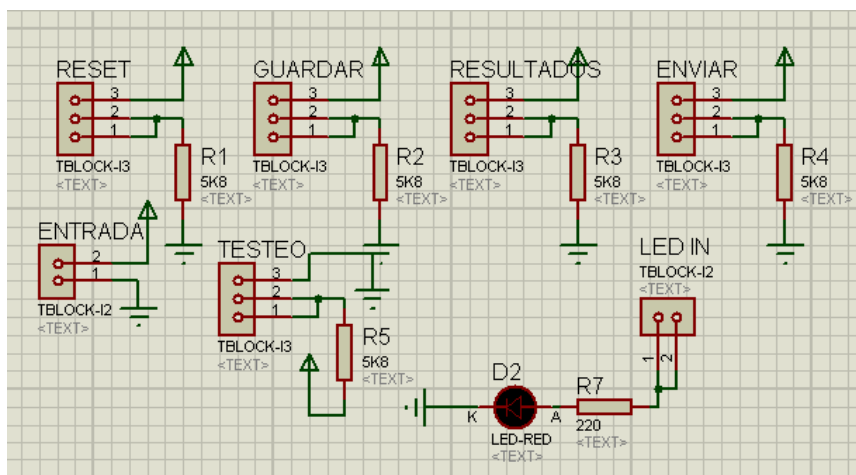


Figura 3.29 Diseño en *Proteus* de placa de control de pulsadores y *led* indicador

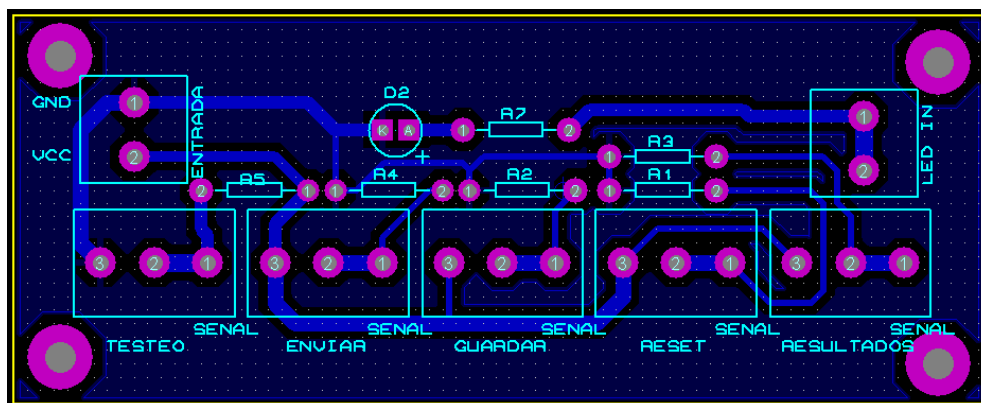


Figura 3.30 Circuito PCB de placa de control de pulsadores y *led* indicador

Tabla 3.13 Lista de elementos de placa de control de pulsadores y *led* indicador

Cantidad	Elemento electrónico
1	<i>Led</i>
1	Resistencia de 220 (Ω)
2	<i>Tblock</i> de 2 posiciones
5	Resistencia de 5.8 ($k\Omega$)
5	<i>Tblock</i> de 3 posiciones

Como se aprecia en la Figura 3.29 se utilizó la configuración de resistencias de *Pull-Down* para los pulsadores denominados “RESET”, “GUARDAR”, “RESULTADOS”, “ENVIAR”, y *Pull-Up* para el *switch* llamado “TESTEO” de este modo se establece un estado lógico alto o bajo que es detectado por el microcontrolador y posteriormente pone en marcha las respectivas funciones que se le han programado.

- Placa de control de pantalla LCD

Por otra parte, se tuvo que diseñar una placa de control para la pantalla LCD la cual se muestra en la Figura 3.31, básicamente que pueda abarcar todos los pines que tiene, además del potenciómetro que regula el nivel de luminosidad de la pantalla y por último la fuente de suministro de voltaje, en la Figura 3.32 se muestra el diseño PCB y finalmente en la Tabla 3.14 los componentes que conforman la placa.

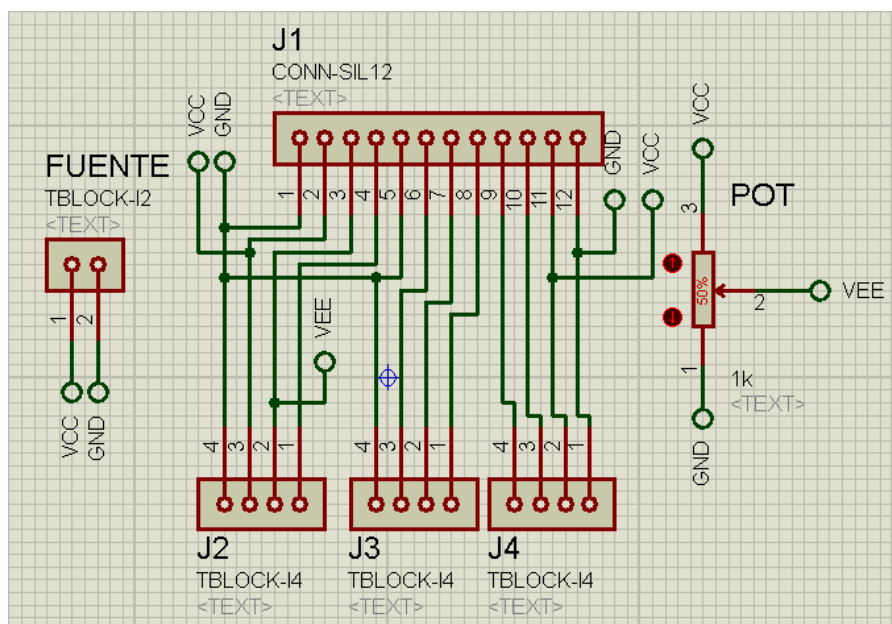


Figura 3.31 Diseño en *Proteus* de placa de control de pantalla LCD

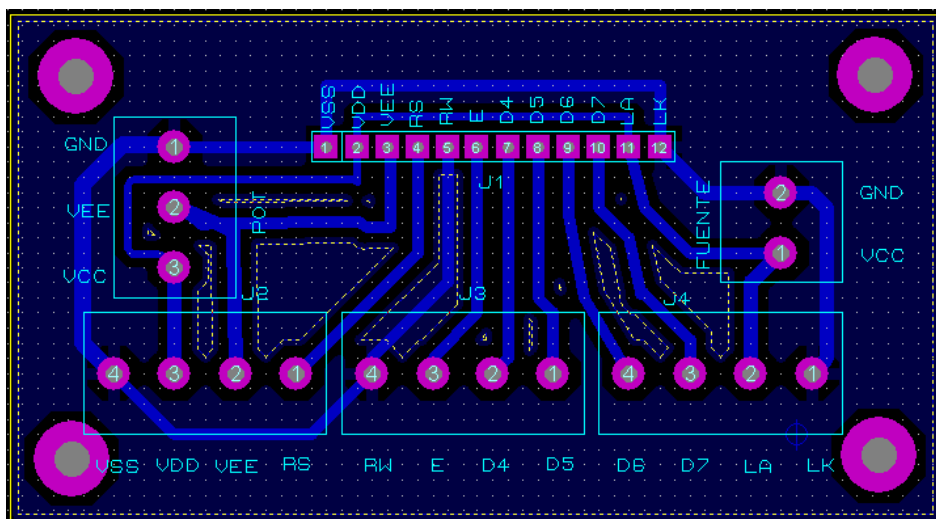


Figura 3.32 Circuito PCB de placa de control de pantalla LCD

Tabla 3.14 Lista de componentes de placa de control de pantalla LCD

Cantidad	Elemento electrónico
1	Regleta de conexión de 12 posiciones
1	<i>Tblock</i> de 2 posiciones
1	Potenciómetro de 1 (k Ω)
3	<i>Tblock</i> de 4 posiciones

Los *Tblock* de 4 posiciones denominados “J2”, “J3” y “J4”, alojan los cables de conexión que conectan la placa con el microcontrolador, la regleta llamada “J1” conecta los cables con los pines de la pantalla LCD, el potenciómetro denominado “POT” como se mencionó antes regula el nivel de brillo de la pantalla y el *Tblock* de dos posiciones llamado “FUENTE” recibe la alimentación para la pantalla.

Diseño de los circuitos de simulación

Más adelante se necesitará probar el código fuente de programación para las diferentes etapas del prototipo, por ende, se desarrolló y diseñó en ISIS *Proteus* una serie de circuitos, de los cuales se tiene:

- Pantalla LCD

Contiene una pantalla LCD de ISIS *Proteus* que ayudará a visualizar resultados, opciones y demás. Se conforma por una pantalla LCD y un potenciómetro tal como se muestra en la Figura 3.33.

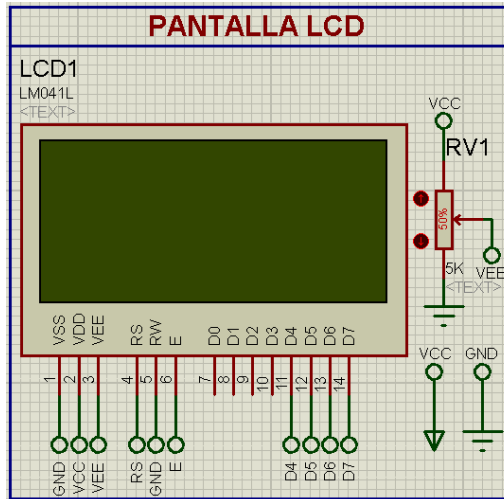


Figura 3.33 Diseño de simulación de pantalla LCD

- Selección de modo

Básicamente consta de un *switch* con una resistencia tal como se aprecia en la Figura 3.34, de este modo se podrá simular que se elige el modo de conteo de monedas o el de menú de selección.

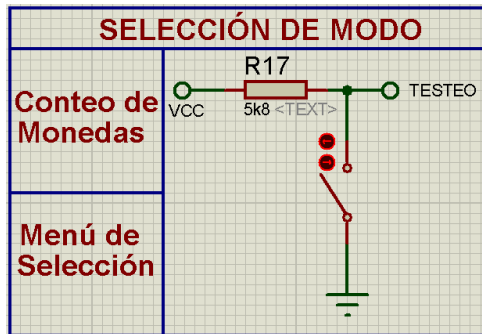


Figura 3.34 Diseño de circuito simulador de selección de modo

- Menú de opciones

Para activar las funciones se debe presionar pulsadores, se tiene 4 opciones, por ende, se seleccionó 4 pulsadores con resistencias tal como se muestra en la Figura 3.35, de este modo se puede simular la activación de los mismos.

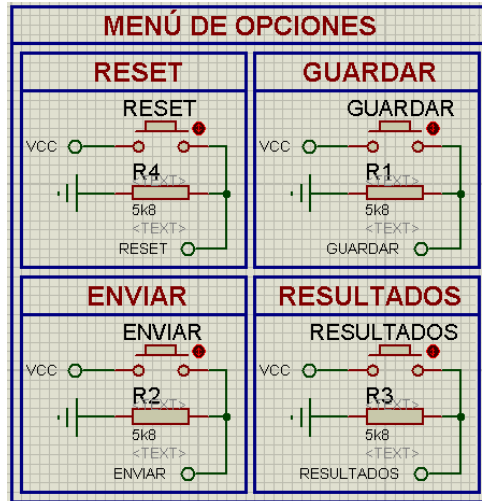


Figura 3.35 Diseño de circuito simulador de opciones

- Led indicador

Al momento de activar algunas funciones, se activará un *led* que actuará como indicador, por ende, se diseñó un circuito con un *led* y una resistencia de protección el cual se muestra en la Figura 3.36.

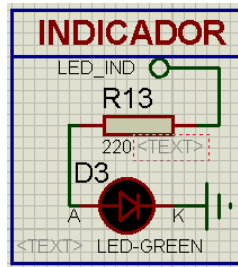


Figura 3.36 Diseño de circuito simulador de *led* indicador

- Simulación de sensores

Al no existir librerías de componentes del sensor detector de monedas que se eligió, se simuló por medio de pulsadores la función del sensor, es decir, que cuando se accione el pulsador se enviará una señal digital al microcontrolador emulando como si el sensor detectó una moneda. En la Figura 3.37 se aprecia el circuito.

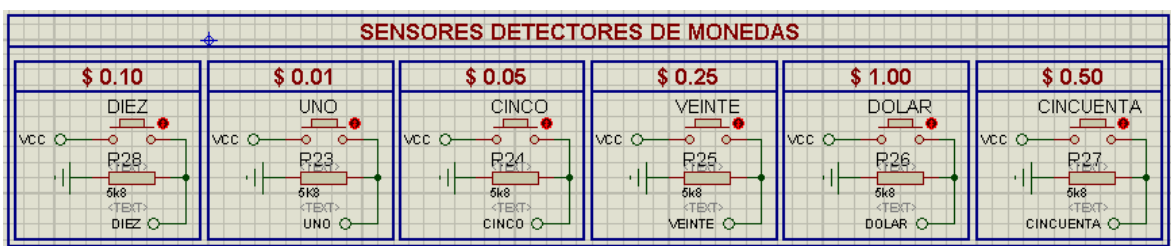


Figura 3.37 Circuito emulador de sensor detector de monedas

3.3 Desarrollo del software a emplearse en el microcontrolador

Para el desarrollo del *software* se utilizó lenguaje de alto nivel lo que permitió emplear algoritmos de una manera adecuada y adaptable a la capacidad cognitiva del ser humano.

Esquemas del código de programación

Para tener una visión clara de cómo realizar el código de programación se hizo diagramas de flujo, en la Figura 3.38 se aprecia el esquema general que tiene el código de programación. Subdivide dos modos de funcionamiento, el modo “OPCIONES” y el modo “PROCESO” los cuales se observan en la Figura 3.39 y Figura 3.40 respectivamente.

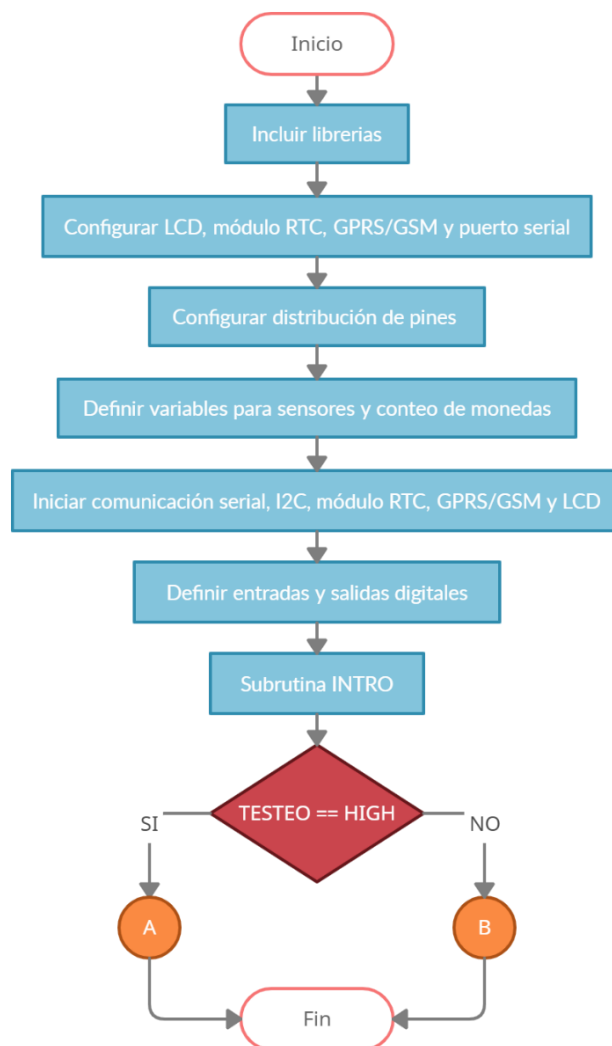


Figura 3.38 Diagrama de flujo general

En este diagrama de flujo se indica la forma en la que se programó el prototipo en general, se muestra la inclusión de librerías, la configuración e inicialización respectiva de cada elemento electrónico y módulo, también la configuración de pines, entradas y salidas digitales, además, de la animación introductoria.

Por último, muestra los dos modos de funcionamiento antes mencionados. Cabe resaltar que el prototipo siempre estará en un modo de funcionamiento ya sea el de "OPCIONES" o "PROCESO", no tiene un modo de funcionamiento neutro, debido a que así se diseñó el mismo.

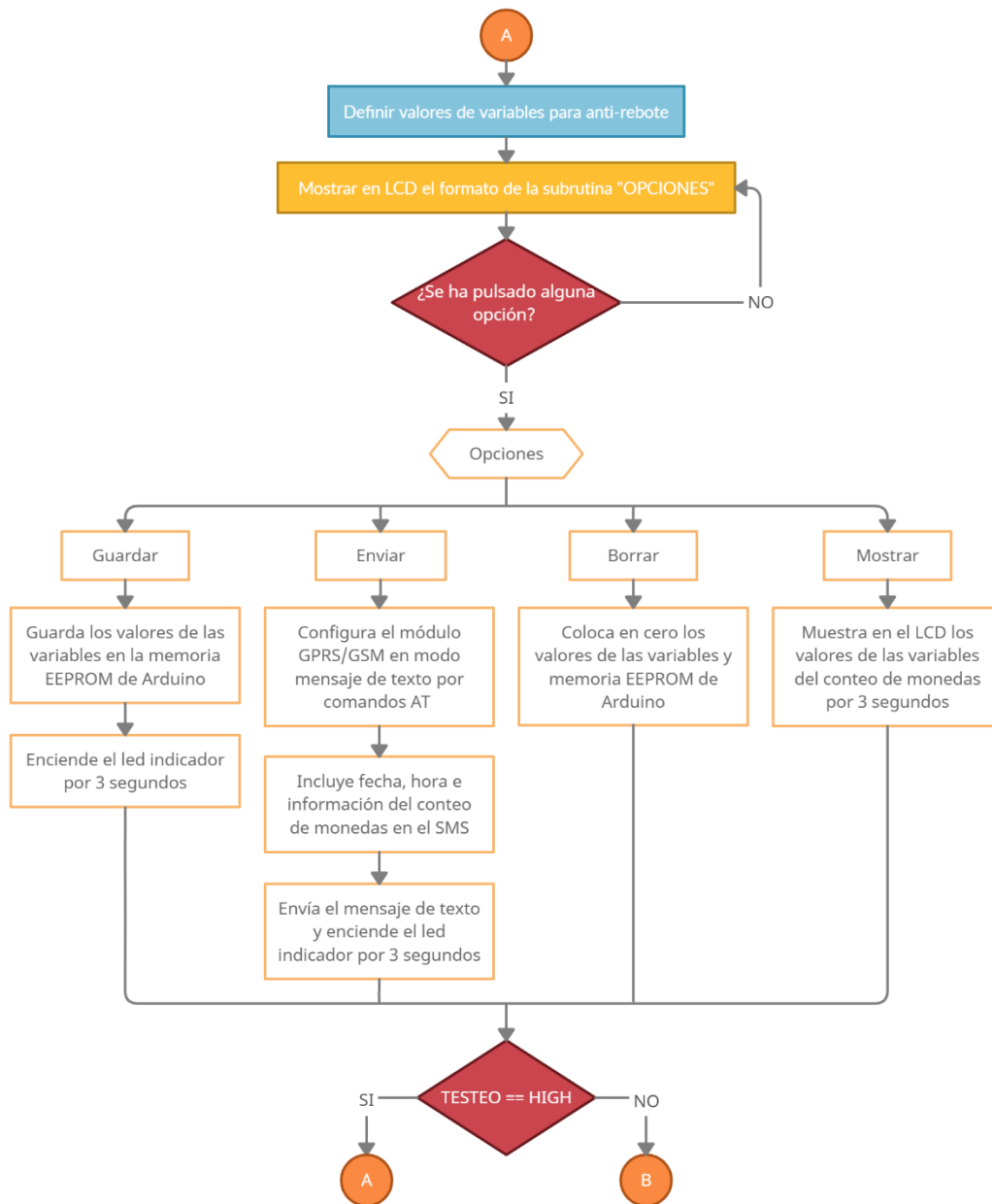


Figura 3.39 Diagrama de flujo de OPCIONES

Para este modo de funcionamiento se programó una subrutina, la cual comprende un formato en el cual se muestra un texto en conjunto con la fecha y hora. Este formato permanece en la pantalla siempre y cuando no se presione ninguna opción.

Tiene 4 opciones de las cuales son:

- Guardar

Esta opción hace uso de la memoria EEPROM de Arduino, en sí, escribe los valores de las variables del conteo de monedas en las direcciones de la memoria, por lo que si se tiene algún fallo de energía los datos del conteo no se perderán.

- Enviar

Permite enviar el mensaje de texto con toda la información pertinente al usuario. Hace uso de las funciones de comunicación que posee Arduino en este caso las de transmisión y recepción de datos. Por ende, se encarga de configurar el módulo GPRS/GSM por medio de comandos AT, luego incluye toda la información respectiva como la fecha, hora y los datos finales del conteo de monedas, por último, envía el mensaje de texto al número de teléfono móvil que se le programó.

- Borrar

Únicamente escribe el valor de cero en cada una de las direcciones de la memoria EEPROM que se usa y en las variables del conteo de monedas. Dejando así todos los valores en cero.

- Mostrar

Esta opción permite visualizar los datos del conteo de las monedas, es útil para verificar los datos que se enviarán en el mensaje de texto. Cabe resaltar que los datos que se muestra son los obtenidos del proceso de conteo y clasificación de monedas y no de la memoria EEPROM ya que el modo de funcionamiento "PROCESO" no permite guardar los datos en la memoria EEPROM, el modo "OPCIONES" permite guardar los datos.

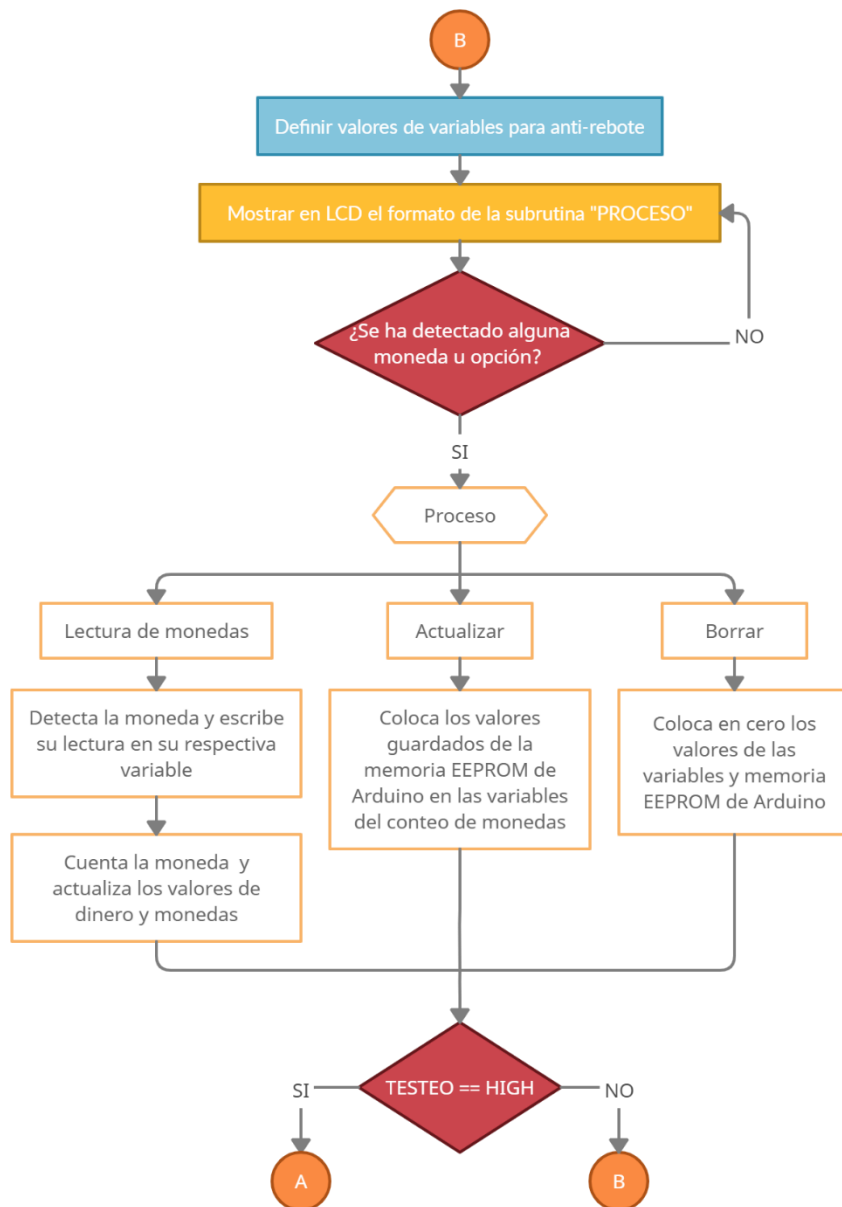


Figura 3.40 Diagrama de flujo de PROCESO

El modo de funcionamiento “PROCESO” se encarga del conteo de monedas, por ende, al seleccionar este modo se muestra en la pantalla LCD el formato del mismo. Este modo involucra el funcionamiento de los sensores detectores. Contiene 3 opciones siendo estas:

- Lectura de monedas

Este proceso se lo hace para cada una de las monedas, por ende, el sensor se encarga de detectar la moneda y enviar la señal a la placa de Arduino, una vez detectada se la

cuenta y enseguida se actualiza los valores del dinero y número de monedas en general, con esto se logra hacer que los valores coincidan y no se tenga errores de datos.

- Actualizar

Esta opción funciona siempre y cuando se hayan guardado los datos en la memoria EEPROM por medio de la opción Guardar. Por lo que sí existe algún fallo de energía, al volver a energizar el prototipo los valores de las variables de conteo serán cero, pero al activar esta opción se volverá a poner el valor que se guardó en la memoria EEPROM en las variables de conteo.

- Borrar

Reiniciar los valores de las variables de conteo y las direcciones de la memoria EEPROM, dejando todo en cero.

Programación del código en Arduino IDE

Al realizar el estudio de los diagramas de flujo se tuvo en cuenta varios aspectos importantes para la creación del código de programación, por ende, se tomó en cuenta la inclusión de librerías, la creación de variables, configuración de pines, configuración e inicialización tanto de elementos electrónicos como de módulos y por último la implementación de subrutinas que ayuden al funcionamiento correcto y óptimo del código de programación.

Inclusión de librerías en el código de programación

En primer lugar, en base a los elementos electrónicos seleccionados se usaron las librerías que se muestran en la Figura 3.41.

```
#include <LiquidCrystal.h>
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <Wire.h>
#include "RTClib.h"
```

Figura 3.41 Librerías

Por lo tanto, se incluyeron un total de 5 librerías, las cuales aseguran un manejo y funcionamiento adecuado de cada elemento electrónico, la función que comprenden se encuentra detallada a continuación:

- Librería *LiquidCrystal*

Permite gestionar la pantalla LCD de cristal líquido, contiene los recursos necesarios para implementar letras, números y signos que se muestran en la pantalla. Además, contiene todas las instrucciones u operaciones de bajo nivel, por lo que facilita la programación de la misma.

- Librería EEPROM

Esta librería hace uso de la memoria no volátil de Arduino, por ende, se encarga de escribir y leer los datos que se le configure por medio de su sintaxis de programación.

- Librería *SoftwareSerial*

Esta librería permite la comunicación serial de Arduino. Se usa para la comunicación de Arduino con el módulo GPRS/GSM.

- Librería *Wire*

Al tener un módulo RTC es necesaria la implementación de esta librería, la cual permite el uso del bus de comunicación I2C de Arduino, por ende, hace posible la comunicación entre el módulo RTC y la placa de Arduino.

- Librería *RTClib*

Adicionalmente para el módulo RTC, se incluyó esta librería para gestionar su funcionamiento. Esta librería permite usar un reloj por *software*.

Creación de variables para procesos y conteo de monedas

Para el funcionamiento adecuado del código de programación fue necesaria la implementación de variables de diferentes tipos, en sí, se utilizó 3 tipos de variables las cuales son:

- *Int*
- *Float*
- *Unsigned long*

Para el conteo de monedas se utilizó las variables *int* y *float*, las cuales se muestran en la Figura 3.42, al ser el valor de conteo de monedas individual y general un dato del tipo entero se utilizó *int*. Por otra parte, el valor del dinero total no es entero por lo que se empleó *float*.

```
int VALOR_UNO=0;
int VALOR_CINCO=0;
int VALOR_DIEZ=0;
int VALOR_VEINTE=0;
int VALOR_CINCUENTA=0;
int VALOR_DOLAR=0;
float DINERO_TOTAL=0;
int TOTAL_MONEDAS=0;
```

Figura 3.42 Variables para el conteo de monedas

Al tener pulsadores se implementó el anti-rebote con el fin de evitar falsas e incorrectas lecturas o detecciones, para ello fue necesaria la creación de variables las cuales se observan en la Figura 3.43.

```
int ANTERIOR_GUARDAR=0;
int ANTERIOR_ENVIAR=0;
int ANTERIOR_RESULTADOS=0;
int ANTERIOR_RESETEO=0;
```

Figura 3.43 Variables para anti-rebote

Con el fin de no utilizar la función “*delay*” debido a que dejaría el código de programación atrapado en una línea de código cierto tiempo, se optó por la función “*millis*” por lo que fue necesaria a creación de variables que permitan una correcta implementación de la misma, en la Figura 3.44 se muestra las variables.

Estas variables agregan un tiempo de 1 segundo en forma de “*delay*” para mostrar el formato de cada uno de los modos de funcionamiento, pero no deja atrapado el código en una sola línea, sino que permite ejecutar otras líneas de código simultáneamente permitiendo monitorizar los demás elementos electrónicos tales como sensores y pulsadores.

```
int PERIODO OPCIONES=1000;
int PERIODO PROCESO=1000;
unsigned long TIEMPO OPCIONES=0;
unsigned long TIEMPO PROCESO=0;
```

Figura 3.44 Variables para la función *millis*

Adicionalmente, se crearon variables que reemplacen la lectura digital de cada pin, con el objetivo de optimizar el código de programación y emprender una mejor comprensión del mismo. En la Figura 3.45 se muestran las variables.

```
int LECTURA_TESTEO=digitalRead(TESTEO);

int LECTURA_GUARDAR=digitalRead(GUARDAR);
int LECTURA_ENVIAR=digitalRead(ENVIAR);
int LECTURA_RESULTADOS=digitalRead(RESULTADOS);
int LECTURA_RESETEO=digitalRead(RESETEO);

int LECTURA_UNO=digitalRead(UNO);
int LECTURA_CINCO=digitalRead(CINCO);
int LECTURA_DIEZ=digitalRead(DIEZ);
int LECTURA_VEINTE=digitalRead(VEINTE);
int LECTURA_CINCUENTA=digitalRead(CINCUENTA);
int LECTURA_DOLAR=digitalRead(DOLAR);
```

Figura 3.45 Variables de lectura digital

Distribución de pines de conexión

Una parte fundamental es la asignación de pines para el funcionamiento de cada uno de los elementos electrónicos, también la configuración de cada pin ya sea como entrada o salida digital. Únicamente fue necesaria la configuración de los pines de los sensores detectores y pulsadores. Por otra parte, la configuración de pines de los demás elementos se encuentra en la librería empleada, por lo que no se configuró los mismos, esto se observa en la Figura 3.46.

```
int UNO=53;           pinMode(TESTEO, INPUT);
int DIEZ=49;         pinMode(GUARDAR, INPUT);
int CINCO=51;        pinMode(ENVIAR, INPUT);
int VEINTE=47;       pinMode(RESULTADOS, INPUT);
int CINCUENTA=45;    pinMode(RESETEO, INPUT);
int DOLAR=43;        pinMode(UNO, INPUT);
int GUARDAR=39;      pinMode(CINCO, INPUT);
int ENVIAR=37;       pinMode(DIEZ, INPUT);
int RESULTADOS=35;   pinMode(VEINTE, INPUT);
int RESETEO=33;      pinMode(CINCUENTA, INPUT);
int TESTEO=8;        pinMode(DOLAR, INPUT);
int LED_INDICADOR=14; pinMode(LED_INDICADOR, OUTPUT);
```

Figura 3.46 Distribución y configuración de pines de conexión

Para establecer el pin de conexión como entrada se utiliza la sintaxis “*INPUT*” y como salida “*OUTPUT*”.

Configuración de *Setup*

En esta parte se hace la inicialización de los diferentes elementos y módulos, por lo que, en la Figura 3.47 se aprecia la configuración.

```
Wire.begin();
RTC.begin();

//RTC.adjust(DateTime(__DATE__, __TIME__));

Serial.begin(9600);
lcd.begin(16, 4);
mySerial.begin(9600);
delay(1000);
```

Figura 3.47 Configuración de *Setup*

En primer lugar, se inicializa la comunicación I2C en conjunto con el módulo RTC, como siguiente se inicia el puerto serial para su respectiva comunicación, después la pantalla LCD y por último el módulo GPRS/GSM. Es importante mencionar que la línea de código que se encuentra comentada sirve para que el módulo RTC tome datos de la hora y fecha, en este caso toma los datos de la computadora, esta línea de código se debe cargar una sola vez y enseguida se debe volver a cargar el código sin esa línea. Debido a que si se carga esa línea de código la fecha y hora será la misma cada vez que se reinicie prototipo.

Subrutinas del código de programación

Como se mencionó antes la creación de subrutinas ayuda a que el código tenga un mejor funcionamiento y optimización. Por ende, fue necesaria la implementación de las siguientes subrutinas:

- Subrutina INTRO

Se hizo una animación introductoria la cual comprende un texto que muestra la bienvenida y la emulación de inicio de un sistema. Únicamente con el objetivo que al iniciar el prototipo se vea estético. En la Figura 3.48 se muestra la subrutina.

```

void INTRO() {
  lcd.clear();
  lcd.setCursor(4,0);
  lcd.print("BIENVENIDO");
  lcd.setCursor(0,1);
  lcd.print("CONTADOR MONEDAS");
  delay(777);
  lcd.setCursor(0,3);
  lcd.print("Iniciando.");
  delay(700);
  lcd.setCursor(0,3);
  lcd.print("Iniciando..");
  delay(700);
  lcd.setCursor(0,3);
  lcd.print("Iniciando...");
  delay(700);
}

```

Figura 3.48 Subrutina INTRO

- Subrutina OPCIONES

Esta subrutina comprende las opciones de los pulsadores, se muestra la inclusión de la función “*millis*” la cual ayuda a mostrar el formato de esta subrutina, simultáneamente permite detectar si se ha pulsado alguna de las opciones.

Para cada opción se utilizó un condicional “*if*” el cual detecta si se pulsó la opción y por medio del proceso de anti-rebote, confirma la pulsación y procede a ejecutar las líneas de código correspondientes a cada opción. En el Anexo: CÓDIGO DE PROGRAMACIÓN se muestra la subrutina.

- Subrutina PROCESO

Para el conteo de monedas fue necesaria la implementación de la subrutina “PROCESO” la cual tiene una estructura similar a la subrutina “OPCIONES” en la inclusión de la función “*millis*” ya que también muestra un formato, para esta subrutina se incluyó dos opciones, la de actualizar y borrar.

Para realizar el conteo de monedas se utilizó un condicional “*if*” el cual se ejecuta al momento que el sensor envía la señal digital, por ende, para cada lectura de moneda se implementó la estructura que se muestra en la Figura 3.49, para observar las líneas de código correspondientes a cada tipo de moneda se debe observar el Anexo: CÓDIGO DE PROGRAMACIÓN.

```

if (LECTURA_MONEDA!=ANTERIOR_MONEDA) {
  if (LECTURA_MONEDA==LOW) {
    VALOR_MONEDA=VALOR_MONEDA+1;
    lcd.setCursor(4,1);
    lcd.print(VALOR_MONEDA);
    TOTAL_MONEDAS=//Función matemática para calcular el total de monedas.
    DINERO_TOTAL=; //Función matemática para calcular el dinero total.
    lcd.setCursor(4,0);
    lcd.print(TOTAL_MONEDAS);
    lcd.setCursor(9,0);
    lcd.print(DINERO_TOTAL);
  }
}

```

Figura 3.49 Estructura del código de programación para la lectura y conteo de cada moneda

- Subrutina VERIFICAR

Esta subrutina se la creó para establecer la comunicación serial del Arduino con el módulo GPRS/GSM, en sí, permite enviar al módulo los comandos que se ingresen, además, recibe cualquier respuesta que se tenga por parte del módulo.

Comprende dos bucles “while” tal como se muestra en la Figura 3.50, en donde, el primer bucle establece que mientras se tenga datos almacenados en el búfer de recepción, se escriba en el módulo la lectura del puerto serial.

En cambio, en el segundo bucle implanta que mientras se tenga datos de respuesta por parte del módulo, se escriba en el puerto serial los datos recibidos del módulo.

```

void updateSerial() {
  delay(500);
  while (Serial.available()) {
    mySerial.write(Serial.read());
  }
  while (mySerial.available()) {
    Serial.write(mySerial.read());
  }
}

```

Figura 3.50 Subrutina VERIFICAR

Compilación del código de programación

Por otra parte, una vez que se terminó de programar todas las funciones necesarias, se procedió a verificar si el código de programación no tenía errores de sintaxis con la

misma herramienta de compilación del programa de Arduino. Para luego realizar las pruebas pertinentes con los circuitos de diseño de simulación.

3.4 Implementación del prototipo de la máquina contadora de monedas

Separador de monedas

Una vez colocadas las medidas, se procedió con el corte de sobrantes del separador de monedas, el cual se observa en la Figura 3.51.

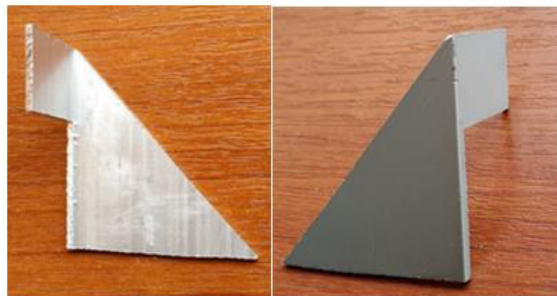


Figura 3.51 Separador de monedas

Barra clasificadora de monedas

En relación, se procedió a cortar los sobrantes de las medidas colocadas de cada moneda dando como resultado la barra clasificadora de monedas la misma que se muestra en la Figura 3.52.



Figura 3.52 Barra clasificadora de monedas

De este modo se implementaron los separadores en la barra clasificadora tal como se muestra en la Figura 3.53, con la ayuda de remaches los mismos que permiten sostener y mantener en el lugar correcto a los separadores.

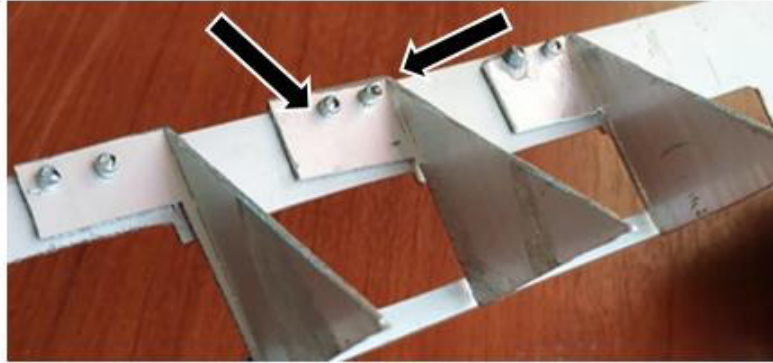


Figura 3.53 Implementación de separadores a la barra clasificadora

Es importante resaltar que los remaches deben ser colocados por encima del diámetro de la moneda de 50 centavos, con la finalidad de evitar que se afecte la trayectoria de la moneda, garantizando un funcionamiento correcto.

El resultado final de la implementación se muestra en la Figura 3.54.



Figura 3.54 Barra clasificadora de monedas finalizada

No obstante, se necesitó la implementación de una pieza base de madera la cual ayuda a que las monedas puedan iniciar el recorrido hacia los contenedores. Además, se colocó pequeños rectángulos de madera los cuales ayudan a que la moneda siga la trayectoria correcta tal como se aprecia en la Figura 3.55.



Figura 3.55 Barra clasificadora complementada con piezas de madera

Base del prototipo

Por otra parte, se implementó una base de madera para el prototipo la cual se muestra en la Figura 3.56, en la que se necesitó montar a la barra clasificadora de monedas.



Figura 3.56 Base del prototipo

Implementación de la barra clasificadora en la base del prototipo

Para implementar la barra clasificadora de monedas en la base del prototipo se colocó dos piezas de madera que sostienen a la misma, en la Figura 3.57 se aprecia las piezas y en la Figura 3.58 se muestra la forma en la que se implementa la barra con ayuda tornillos para fijar la barra a las piezas sostenedoras.

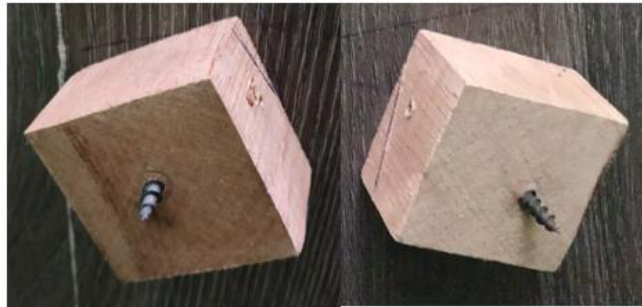


Figura 3.57 Sostenedores de la barra clasificadora

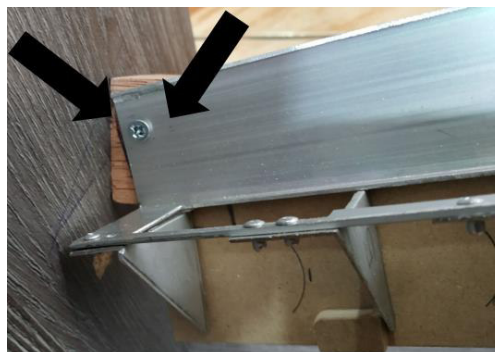


Figura 3.58 Implementación de la barra clasificadora a la base del prototipo

De este modo, es importante mencionar que se debe colocar la barra clasificadora con un ángulo de inclinación para que así las monedas puedan deslizarse sin problema alguno, tal como se lo muestra en la Figura 3.59.



Figura 3.59 Resultado final de la implementación de la barra clasificadora con la base del prototipo

Barra de plástico

Como siguiente se procedió a implementar una barra de plástico sobre la barra clasificadora tal como se muestra en la Figura 3.60, en sí, la función de esta es hacer caer a las monedas por el orificio respectivo y que no salgan desprendidas. Además,

ayuda a que las monedas puedan ir de la manera correcta para ser detectadas por el sensor.



Figura 3.60 Barra de plástico implementada

Contenedores de monedas

Con la ayuda de la pieza de soporte que se muestra en la Figura 3.25, se procedió a colocar 3 piezas tal como se muestra en la Figura 3.61, es importante mencionar que se cubrió la parte inferior de la barra clasificadora de moneda con cartón y no con madera, debido a que al estar la barra clasificadora inclinada se necesitaba una precisión milimétrica para encajar esa pieza en caso de ser de madera, por ende, se necesitó que fuese de cartón ya que se manipulo esa parte en la implementación de circuitos y al ser de cartón facilitó ese proceso.



Figura 3.61 Composición de piezas

Por otra parte, se tapó el espacio con madera dando así la forma al camino por el cual bajan las monedas, en la Figura 3.62 se muestra las piezas que se usó para cubrir ese espacio. Además, dando forma a los contenedores.



Figura 3.62 Pieza deslizador de monedas

Por último, se dividió en 6 partes debido a que se necesita 6 contenedores para cada moneda, en la Figura 3.63 se aprecia el resultado final, cabe mencionar que los separadores de la parte inferior son de madera y los separadores de la parte superior son de cartón, esto se debe a que en la parte inferior se necesita un material que sea resistente y se mantenga firme al momento de caer la moneda.

Al tener un ángulo de inclinación en la parte superior se usó cartón para marcar el camino hacia los contenedores. Como se mencionó antes el uso de cartón facilita la manipulación de esas piezas, lo que no sería fácil al usar madera.



Figura 3.63 Contenedores de monedas

Fabricación del diseño de circuitos electrónicos PCB

Para esto se implementó el método de impresión en papel fotográfico, en primer lugar, se exportó los circuitos PCB de ARES en un archivo pdf tal como se muestra en la Figura

3.64, el cual contenía las pistas conductoras y la señalización de los elementos electrónicos.

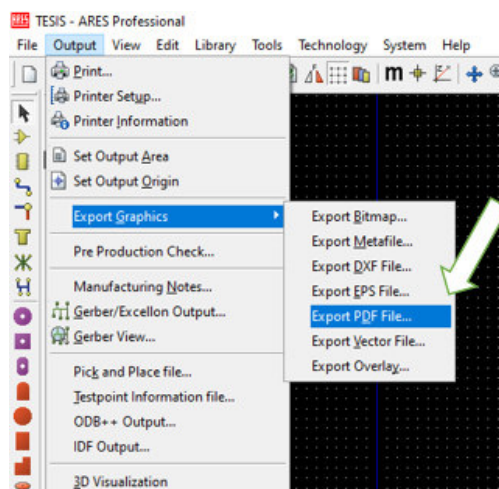


Figura 3.64 Exportación de circuitos PCB en pdf

Es importante mencionar que al momento de exportar los archivos pdf se debe tener en cuenta que para las pistas conductoras se debe seleccionar el modo normal para imprimir, en cambio para la señalización el modo espejo o “*mirror*”.

En la Figura 3.65 se aprecia la configuración de impresión para las pistas conductoras, por lo que en la opción “*Layers*” se debe seleccionar “*Bottom Cooper*” siendo esta la capa en la cual están las pistas. Y en la Figura 3.66 para la señalización de los elementos electrónicos, por ende, se debe seleccionar en “*Layers*” las opciones de “*Top Cooper*” y “*Top Silk*” debido a que esas capas conforman la señalización. Adicionalmente como se mencionó antes es muy importante seleccionar “*Mirror*” en “*Reflection*”, ya que de esta forma se podrá implementar en la placa de baquelita.

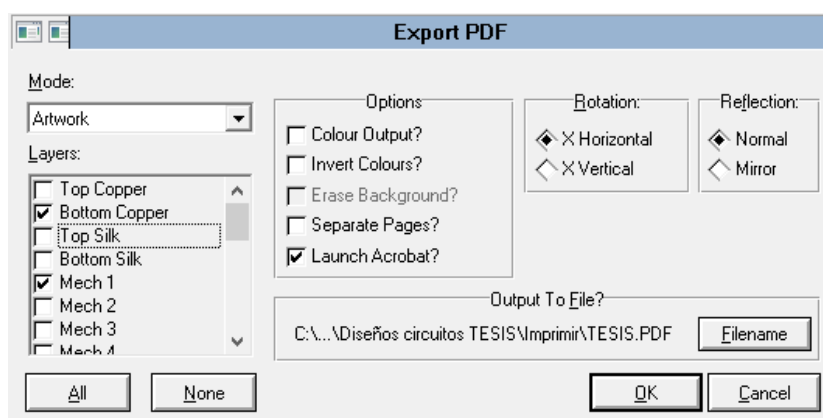


Figura 3.65 Configuración de impresión de pistas conductoras

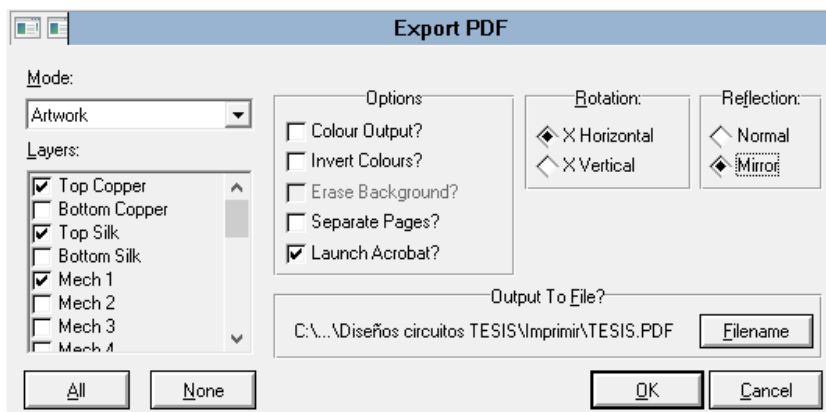


Figura 3.66 Configuración de impresión de señalización de elementos

Por otra parte, es importante mencionar que la impresión de los archivos se debe realizar en una impresora láser y no es una común, debido a que si se imprime en la común la tinta se desprenderá del papel al momento de traspasar las pistas a la baquelita, cosa que no ocurre con la impresión láser.

Finalmente, en la Figura 3.67 se aprecia la impresión de las pistas y en la Figura 3.68 señalización, es recomendable imprimir dos veces el mismo diseño ya que en ocasiones no sale de manera correcta el traspaso de pistas a la baquelita, entonces para evitar nuevamente el proceso de impresión se tiene el respaldo del diseño del circuito.

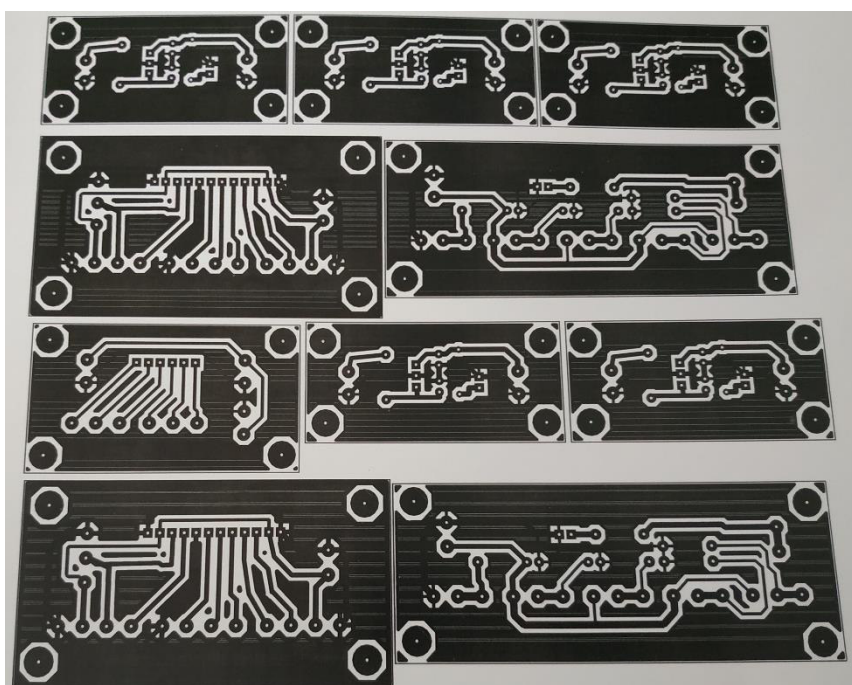


Figura 3.67 Impresión de pistas conductoras

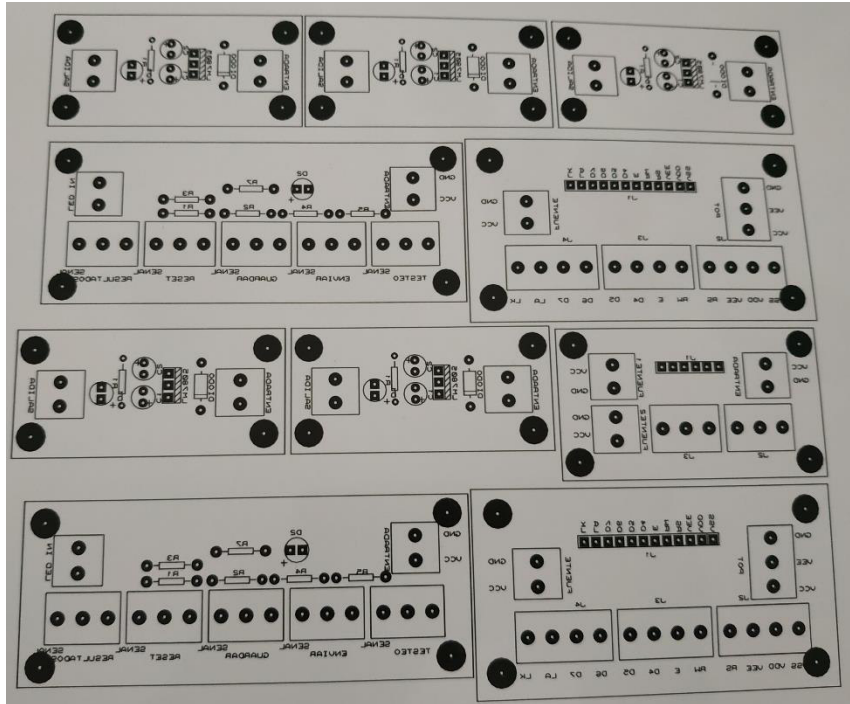


Figura 3.68 Impresión de señalización

Fabricación de los circuitos electrónicos en baquelita

Para esto se utilizó el método más práctico, en sí, el método del planchado. Por ende, se cortó la baquelita del mismo tamaño del circuito PCB y se colocó encima tal como se aprecia en la Figura 3.69.

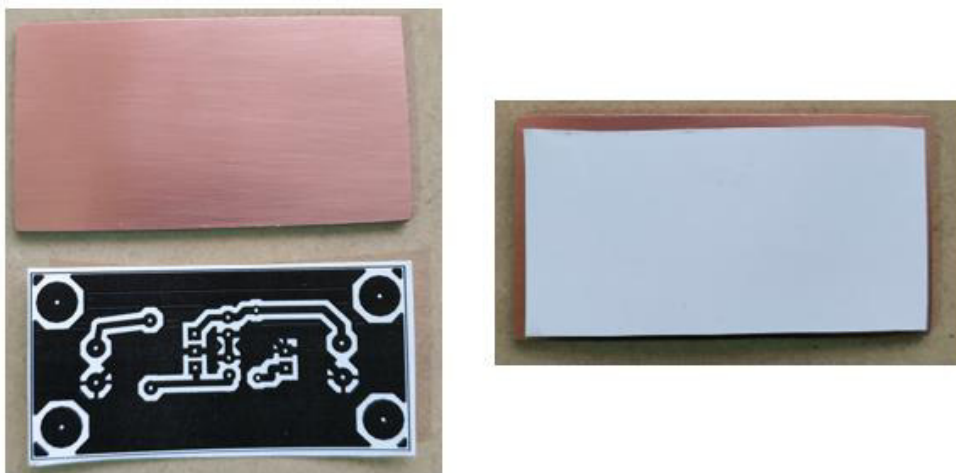


Figura 3.69 Implementación de la impresión PCB sobre baquelita

Después con una plancha de uso doméstico se procedió a planchar la baquelita en conjunto con la impresión y por medio de presión se aplicó calor tal como se muestra en la Figura 3.70 de esta forma se logró traspasar la tinta de la impresión al cobre de la baquelita. Es importante destacar que no se debe dejar por mucho tiempo la baquelita

al calor de la plancha debido a que el cobre puede deformarse dejando así la baquelita inservible.



Figura 3.70 Planchado de pistas sobre baquelita

Una vez completado el planchado, se debe poner la baquelita en un recipiente con agua fría, de este modo se logra enfriar la baquelita y se puede retirar el papel sobrante, tal como se muestra en la Figura 3.71.

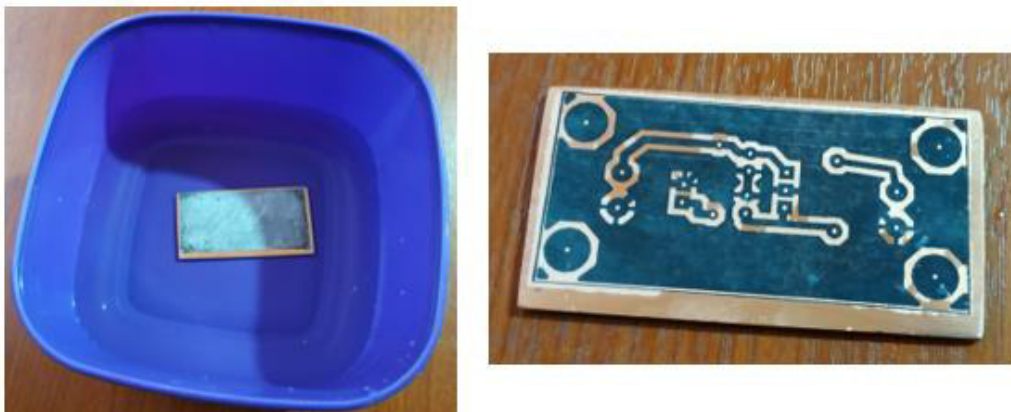


Figura 3.71 Enfriamiento de baquelita

En relación, se debe quitar todo el papel excedente con el objetivo de dejar la baquelita completamente limpia, siendo este un aspecto muy importante para el siguiente paso, denominado proceso de quemado, básicamente es colocar la baquelita en otro recipiente de agua al cual se le incorpora cloruro férrico tal como se aprecia en la Figura 3.72, como recomendación el recipiente debe ser desechable.

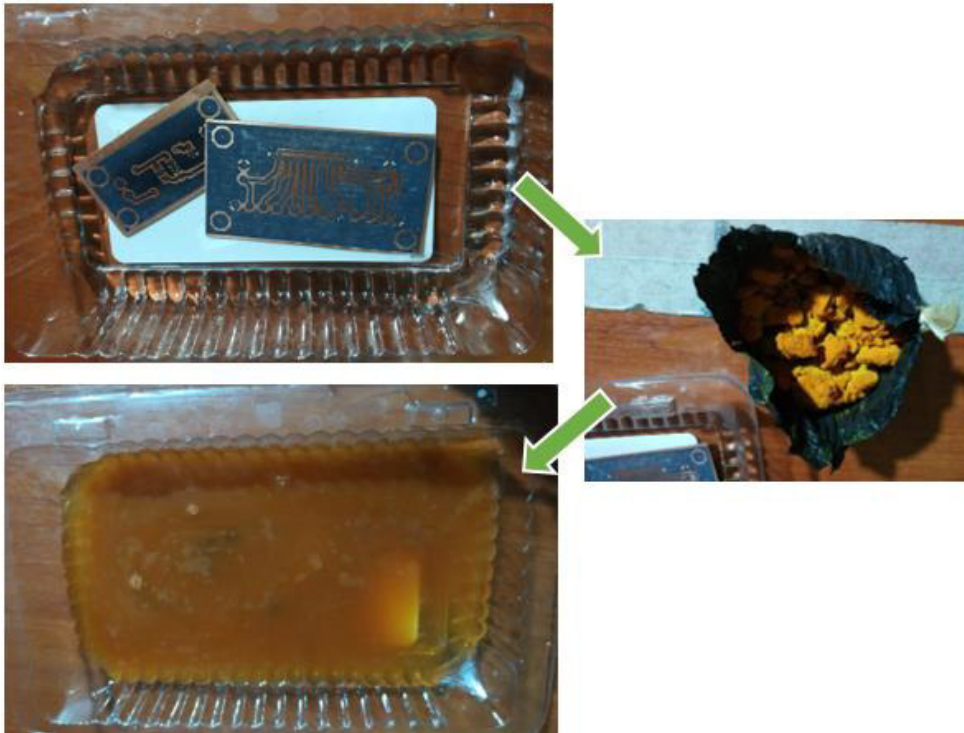


Figura 3.72 Proceso de quemado de baquelita

En este proceso el cloruro férrico se encarga de eliminar el cobre que no está cubierto por la tinta, por ende, después de aproximadamente 20 minutos se procede a retirar la baquelita del recipiente para luego con agua limpia quitar los residuos de cloruro férrico, luego se debe limpiar la baquelita, es decir, que se debe limpiar la tinta, esto se lo hace con diferentes materiales por ejemplo con una lija muy fina, un lustre de aluminio e incluso con un trapo mojado con thinner.

Posteriormente se procede a realizar los orificios para encajar los elementos electrónicos tal como se aprecia en la Figura 3.73.

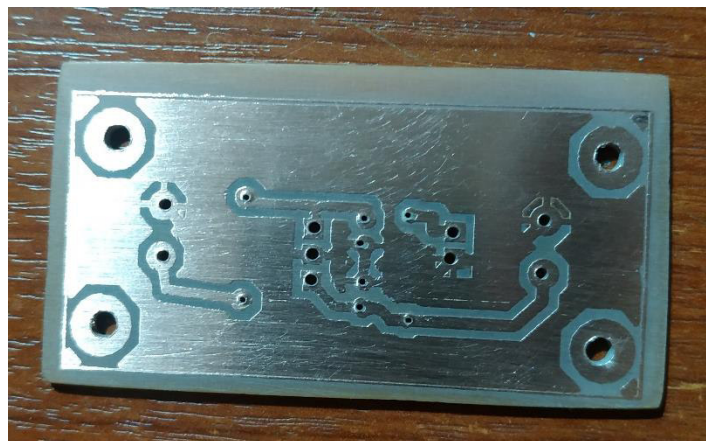


Figura 3.73 Pistas de la placa PCB

Como siguiente paso se realizó la señalización de los elementos, para esto se planchó la impresión respectiva en el lado contrario de las pistas conductoras y se retiró con agua el papel sobrante; el resultado se lo muestra en la Figura 3.74.

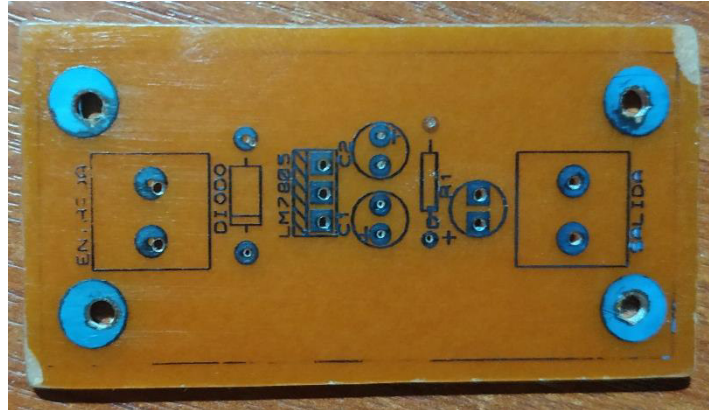


Figura 3.74 Señalización de elementos electrónicos

Como se mencionó anteriormente, algunos circuitos se los realizó en la placa de baquelita perforada, en la Figura 3.75 se aprecia el resultado final de esta implementación, básicamente se suelda los elementos y por medio de cable o estaño se une los mismos, siendo estos materiales (cable o estaño) las pistas conductoras de este tipo de circuitos.

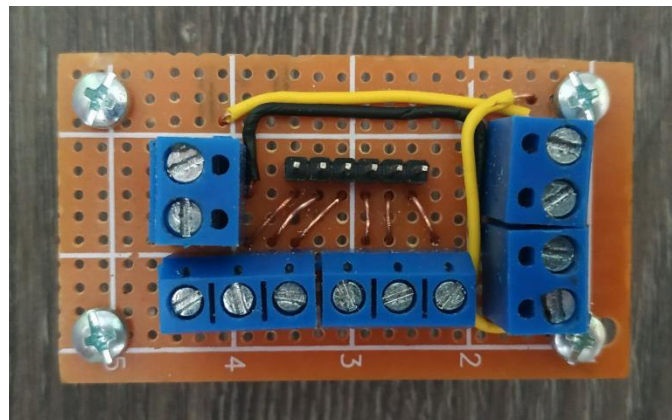


Figura 3.75 Circuito en baquelita perforada

Soldado de los elementos electrónicos

La señalización es parte fundamental en este proceso debido a que con ella se identifica el lugar donde se debe colocar cada elemento electrónico. Por consiguiente, se procede a ubicar los elementos en el lugar respectivo teniendo en cuenta la polarización de ciertos elementos, después con un cautín y estaño se procede a soldar los mismos, el resultado final se lo aprecia en la Figura 3.76.

Cabe mencionar que después de este proceso se debe limpiar las pistas para remover restos de pasta y estaño, se puede realizar con un cepillo de dientes viejo y con thinner.



Figura 3.76 Soldadura de elementos electrónicos

Una vez que se tuvo todos los circuitos electrónicos terminados, se procedió a instalarlos en una base de madera que se fabricó para cada uno, como se observa en la Figura 3.77. Esto se realizó para cada una de las placas, módulos y placas de baquelita perforada.



Figura 3.77 Placa PCB en base de madera

Instalación de los sensores detectores

Para esto se cortó los cables necesarios para cada uno de los sensores, en sí, se necesitó de 3 cables para cada sensor, 2 cables para alimentación y 1 cable de transmisión de señal digital. Después se colocó los sensores en la posición indicada en

la Figura 3.78, los cables de cada sensor se conectaron a la placa de control de sensores, la misma que se la fabricó con baquelita perforada.

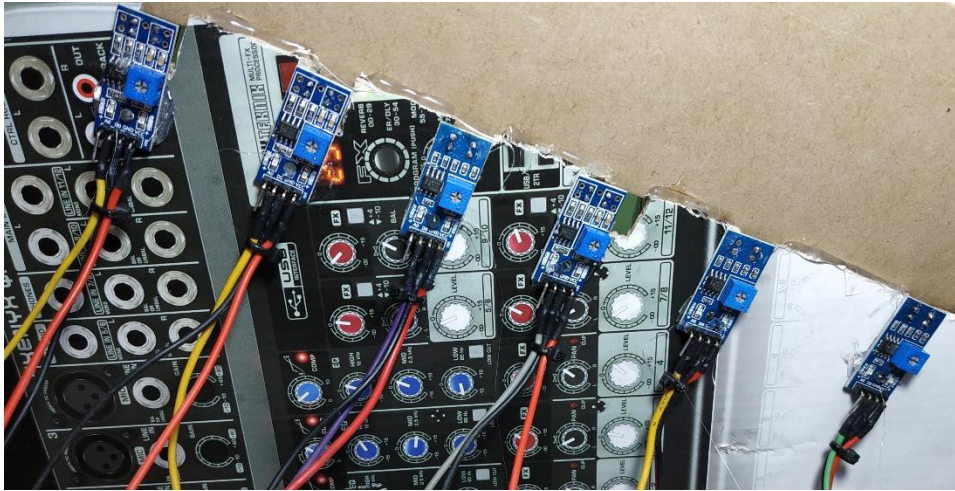


Figura 3.78 Instalación de sensores detectores de monedas

Implementación de los elementos electrónicos en panel de control

Para implementar los elementos electrónicos se procedió a soldar y colocar cada uno de los cables tanto en la pantalla LCD como en los pulsadores, *switch* y *led* indicador tal como se muestra en la Figura 3.79.

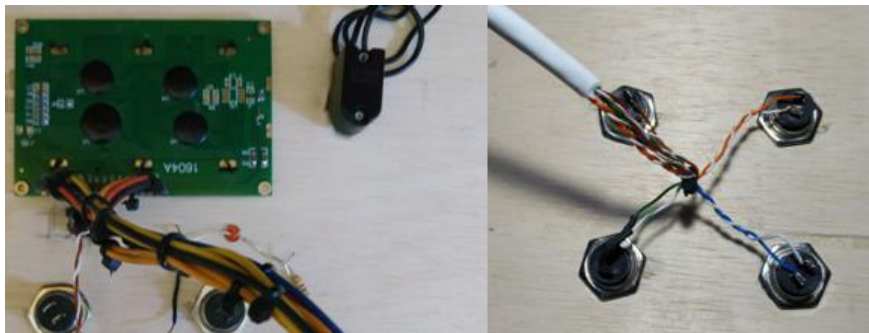


Figura 3.79 Conexión de cables a elementos electrónicos del panel de control

En los cables de la pantalla LCD se colocó amarras plásticas con el objetivo de ordenar los mismos. Para lograr diferenciar los pulsadores se utilizó cables de diferente color para cada uno. Cada uno de estos elementos se aseguró con silicona en caso de la pantalla LCD y con tuercas para los pulsadores.

Implementación de los circuitos electrónicos y módulos

Por otra parte, se procedió a montar cada una de las placas de control en la base del prototipo, se montó la placa controladora de pulsadores, *switch* y *led* indicador, la misma que se observa en la Figura 3.80, esta abarca los cables de conexión que llegan de los

elementos del panel de control. Desde esta placa se tiene cables de conexión a la placa de Arduino, de esta forma se establece la conexión entre los elementos del panel de control con la placa de Arduino.



Figura 3.80 Placa de control de pulsadores, *switch* y *led* indicador

Después se instaló la placa controladora de la pantalla LCD, esta se la puso de manera vertical y no en la base del prototipo, debido a que de esta forma se facilitó la conexión de cables provenientes del panel de control.

Cabe mencionar que no se utiliza todos los pines de conexión de los *Tblock* debido a que algunas conexiones están puestas en el mismo punto, además, no se soldó el potenciómetro en la placa directamente, esto se lo hizo por si el elemento electrónico sufría algún daño en el futuro, se lo pudiera cambiar sin necesidad de desmontar la placa, en la Figura 3.81 se muestra la placa.



Figura 3.81 Placa de control de pantalla LCD

Posteriormente se montó el módulo RTC, únicamente se necesitó la base de madera y los cables de alimentación y comunicación a la placa de Arduino. Al ser un dispositivo pequeño se lo colocó en una esquina de la base de prototipo tal como se muestra en la Figura 3.82.

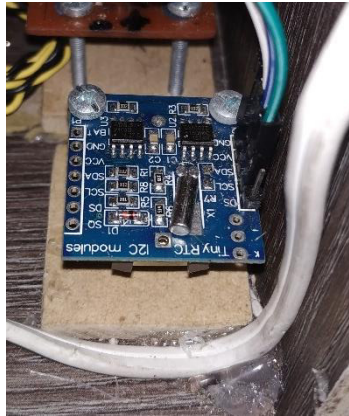


Figura 3.82 Placa del módulo RTC

No obstante, para la parte de alimentación del prototipo se realizó un circuito el cual involucra el *switch* de encendido y apagado. Se utilizó baquelita perforada para fabricar esta placa por la razón que es un circuito muy sencillo y se colocó de manera vertical como se observa en la Figura 3.83, de igual manera porque facilita la conexión de cables con el panel de control.

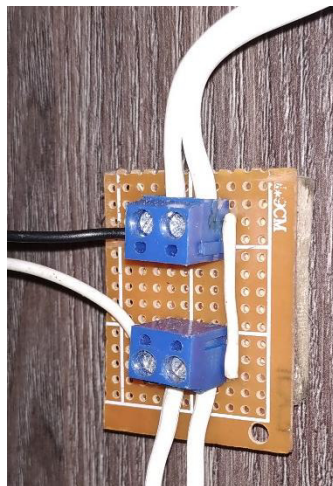


Figura 3.83 Placa de *switch* de encendido y apagado

En este sentido, la placa básicamente conecta un *switch* con la línea de energía eléctrica y permite el paso de la energía o corta el paso de la misma. De este modo se logra encender o apagar el prototipo con tan solo un *switch*.

Esta placa se conecta directamente con la fuente de alimentación principal, la cual se la colocó de forma vertical. La salida de voltaje de la fuente principal se conecta al circuito del regulador de voltaje LM7805, posteriormente la salida de voltaje de este circuito se conecta a la placa que distribuye el voltaje tanto para la placa de control de sensores como para la placa de control de pulsadores, *switch* y *led* indicador.

En la Figura 3.84 se muestra la placa, la misma que se fabricó con baquelita perforada, únicamente posee un *Tblock* que recibe los 5 (V_{DC}) de salida del circuito del regulador de voltaje y por medio de pines de conexión distribuye el voltaje. La regleta de pines del lado izquierdo representa la parte positiva y negativa los pines del lado derecho. Por ende, solo se debe conectar dos cables de conexión para alimentar las placas de control respectivas.

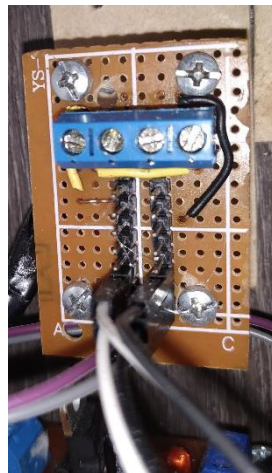


Figura 3.84 Placa distribuidora de voltaje

Por otra parte, se procedió a instalar el módulo GPRS/GSM, se conectó los cables de alimentación con la fuente de poder y se dejó los cables de transmisión y recepción de datos para conectarlos a la placa de Arduino.

Este módulo se colocó debajo de la placa de control de la pantalla LCD, se implementó una base de madera relativamente pequeña la cual ayuda a que el módulo se mantenga firme al momento de ingresar o retirar la tarjeta SIM, tal como se aprecia en la Figura 3.85.

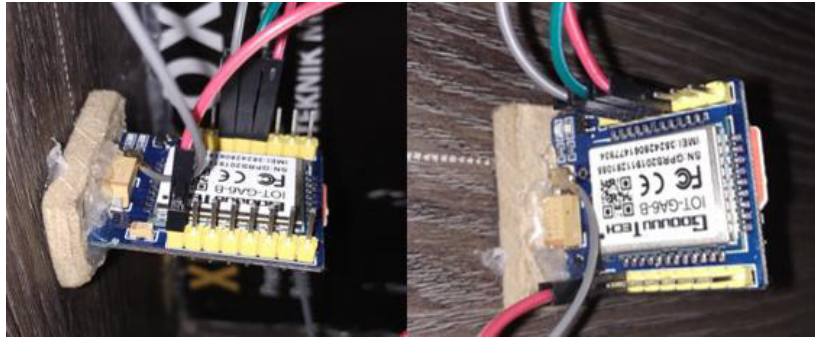


Figura 3.85 Placa del módulo GPRS/GSM

Finalmente, se montó la placa de Arduino en la base del prototipo, posteriormente se conectó cada uno de los cables provenientes de módulos y placas de control en los respectivos pines de conexión, tanto de alimentación como de comunicación. En la Figura 3.86 se muestra la placa ya implementada.



Figura 3.86 Placa de Arduino Mega 2560

Instalación del distribuidor de monedas

Este distribuidor de monedas consta de un envase plástico, en el cual se colocan las monedas, además, cuenta con un ventilador de PC en la parte inferior, que se encarga de generar la vibración, para lograr esto se colocó tornillos en una de las hélices tal como se muestra en la Figura 3.87, el objetivo era crear un desbalance en el ventilador generando peso en aquella hélice para dar como resultado un movimiento un poco brusco, generando así una vibración.

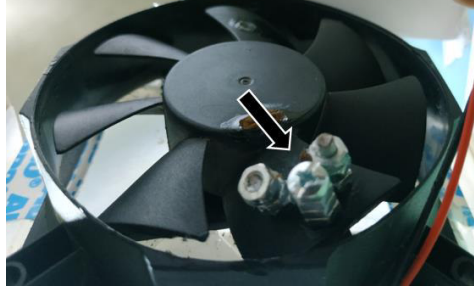


Figura 3.87 Ventilador de PC

Por consiguiente, se instaló la estructura que guía a las monedas hacia la máquina contadora la misma que se observa en la Figura 3.88, esta consta de dos piezas de aluminio, la pieza del lado izquierdo ayuda a que las monedas mantengan el curso de caída, la pieza del lado derecho encamina las monedas hacia la caída.



Figura 3.88 Estructura conductora de monedas

Por otra parte, se implementó dos piezas para el distribuidor de monedas, las mismas que sirven de soporte y se las aprecia en la Figura 3.89. La columna se fabricó de madera y la base que sostiene a la columna se la hizo de aluminio.

Estas piezas ayudan a mantener estable al distribuidor de monedas al momento en que se genera la vibración por parte del ventilador de PC, además, mantienen al distribuidor en un ángulo inclinado, de esta forma las monedas se deslizan y caen.

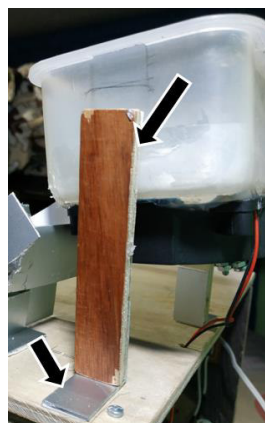


Figura 3.89 Soportes del distribuidor de monedas

Finalmente, se procedió a realizar el orificio por el cual caen las monedas tal como se muestra en la Figura 3.90, además, se colocó diferentes piezas dentro del envase con el objetivo de separar las monedas y que se dirijan al orificio.



Figura 3.90 Implementación de orificio y piezas de conducción

Implementación y finalización el prototipo

Como parte de la implementación se añadió color al prototipo, es decir, se lo pintó en este caso de color negro. Además, se añadió una barra de madera en los contenedores de monedas, la cual ayuda a que las monedas se queden en su respectivo contenedor y no salgan.

Adicionalmente, en la parte delantera del prototipo se montó en el panel de control una señalización, con el objetivo que sea más fácil manejar el prototipo.

Para el montaje de la barra de los contenedores se implementó bisagras de un tamaño adecuado y un cerrojo pasador. Las bisagras abren y cierran los contenedores mientras que el cerrojo mantiene cerrados los mismos con el objetivo que no salgan las monedas. En la Figura 3.91 se aprecia la implementación.

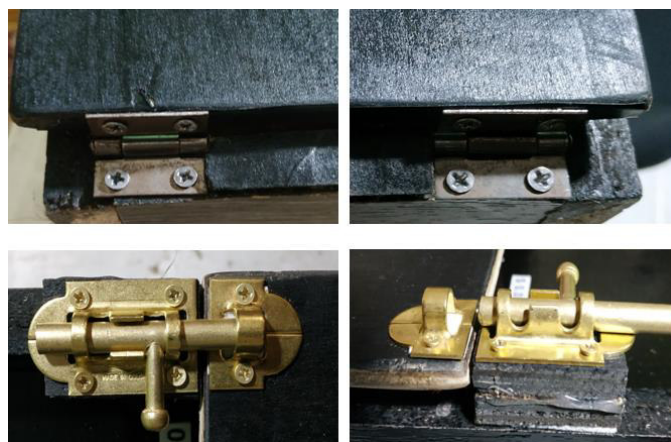


Figura 3.91 Implementación de la barra para contenedores

Por otra parte, en el panel de control se montó una señalización la cual se muestra en la Figura 3.92, esto permite seleccionar correctamente la opción a ejecutar.



Figura 3.92 Señalización del panel de control

Finalmente, en la Figura 3.93 se aprecia el prototipo ya finalizado y listo para realizar las respectivas pruebas de funcionamiento.



Figura 3.93 Prototipo finalizado

3.5 Realización de pruebas de funcionamiento de la máquina y del envío de notificaciones al usuario final

Pruebas de funcionamiento del *software*

En primer lugar, al encender el prototipo en la pantalla LCD se muestra un texto de introducción que se da por medio de la subrutina INTRO del código de programación, la cual se muestra en la Figura 3.94. Al visualizar el texto se aprecia que el texto está centrado por lo que no se debe realizar ninguna corrección. En la Figura 3.95 se aprecia en el prototipo.

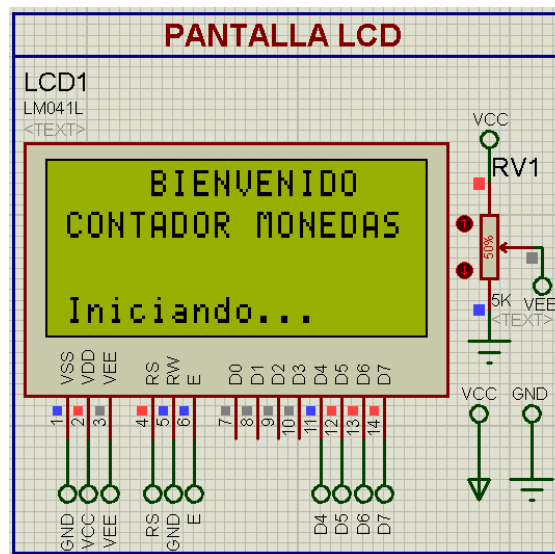


Figura 3.94 Animación introductoria en simulación



Figura 3.95 Animación introductoria en prototipo

Por otra parte, como se mencionó antes, el prototipo cuenta con dos modos de funcionamiento los cuales se pueden elegir por medio de un *switch*, el primero llamado OPCIONES y el otro PROCESO, cada uno de estos modos tiene un formato, en la Figura 3.96 se aprecia los mismos.

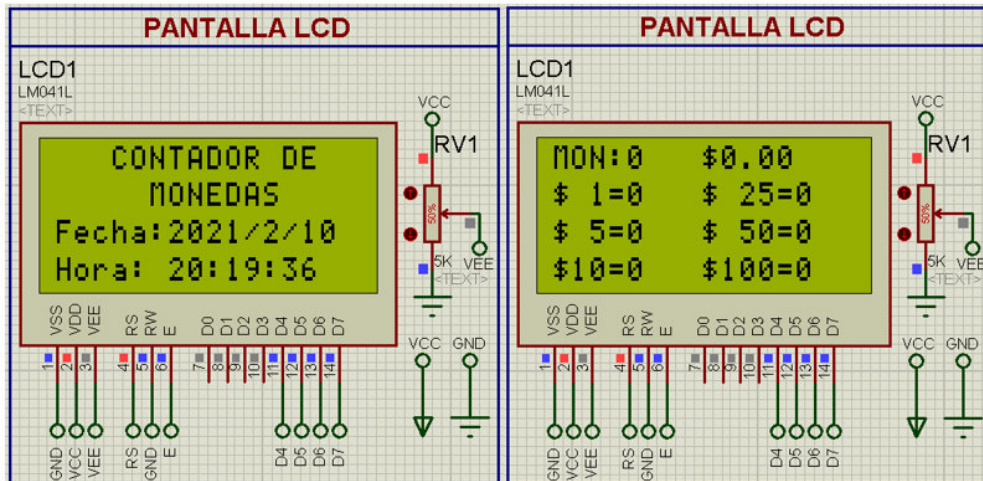


Figura 3.96 Formatos de modos de funcionamiento en simulación

Al simular en el *software* se aprecia que cada uno de los formatos de los modos de funcionamiento son correctos, por ende, no es necesario realizar ningún cambio. Y en la Figura 3.97 se muestran los formatos implementados en el prototipo.



Figura 3.97 Formatos de modos de funcionamiento en prototipo

Para el modo de funcionamiento PROCESO, no fue posible seleccionar como tal el sensor detector por medio de *software* pero se hizo un diseño del mismo, el cual se mostró en la Figura 3.37, por consiguiente al presionar un pulsador se envía la señal digital al microcontrolador Arduino, dando como resultado el conteo de monedas por parte de los sensores, en la Figura 3.98 se aprecia la simulación y en la Figura 3.99 se muestra en el prototipo.

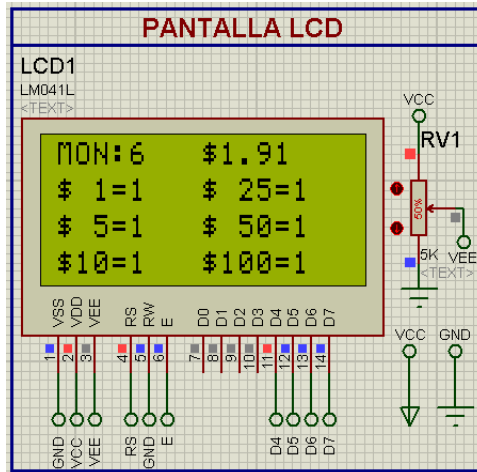


Figura 3.98 Simulación del conteo de monedas



Figura 3.99 Proceso de conteo de monedas en el prototipo

Para el modo de OPCIONES se tiene un total de 4 funciones, de las cuales cada una tiene su respectivo texto y función a ejecutar, se tiene las siguientes opciones:

- GUARDAR

Al presionar el pulsador, se observa que aparece el texto “GUARDANDO DATOS” y se enciende el led indicador por un periodo de 3 (segundos) tal como se muestra en la Figura 3.100 en simulación y en la Figura 3.101 en el prototipo.

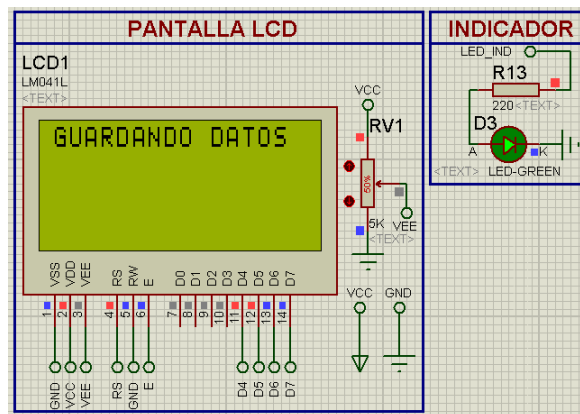


Figura 3.100 Simulación de la opción GUARDAR



Figura 3.101 Opción GUARDAR en prototipo

- ENVIAR

Al presionar esta opción, internamente el microcontrolador inicia el proceso de enviar el mensaje de texto con la información correspondiente, al iniciar este proceso se muestra en la pantalla el texto “ENVIANDO SMS”, mientras se muestra este texto se configura el módulo GPRS/GSM con comandos AT, una vez acaba la configuración se envía el mensaje de texto y en la pantalla aparecerá el texto “MENSAJE ENVIADO” y se encenderá el *led* indicador. Esto quiere decir que se ha enviado con éxito el mensaje, en la Figura 3.102 se aprecia este proceso en la simulación y en la Figura 3.103 en el prototipo.

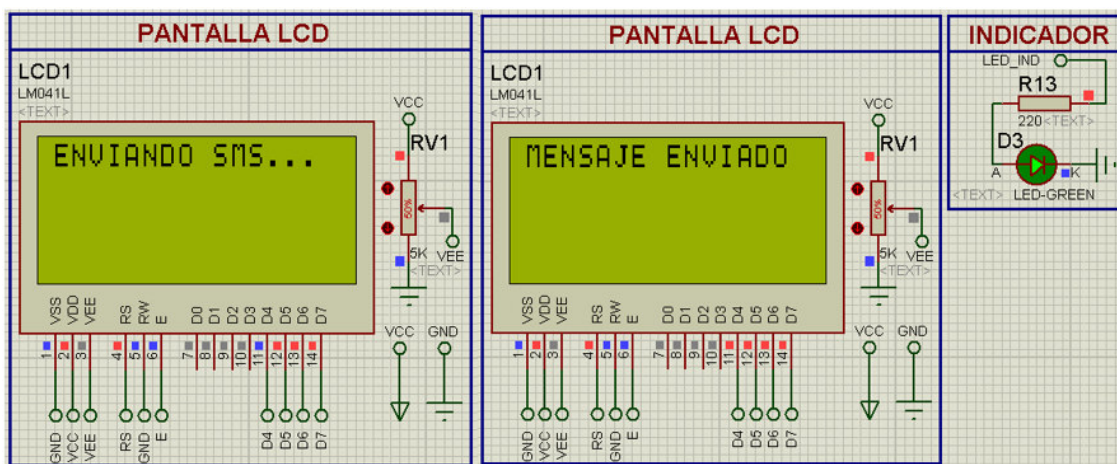


Figura 3.102 Simulación de la opción ENVIAR



Figura 3.103 Opción ENVIAR en el prototipo

Es importante mencionar que el proceso de envío de mensaje se cumple con normalidad siempre que se presione el pulsador de ENVIAR, pero la tarjeta SIM debe estar recargada con saldo, caso contrario no se enviará el mensaje de texto con éxito.

- RESULTADOS

Esta opción muestra los datos que se obtuvo del conteo de monedas por un tiempo de 3 (segundos), al realizar las pruebas se observó que los datos coinciden con el conteo de monedas, por ende, esta opción funciona correctamente, en la Figura 3.104 se muestra la opción simulada y en la Figura 3.105 en el prototipo

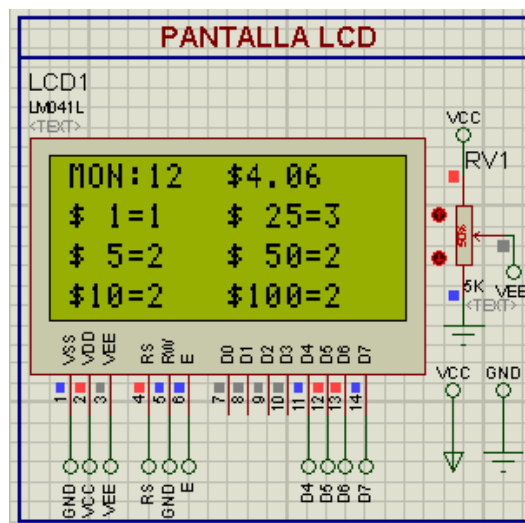


Figura 3.104 Simulación de la opción RESULTADOS

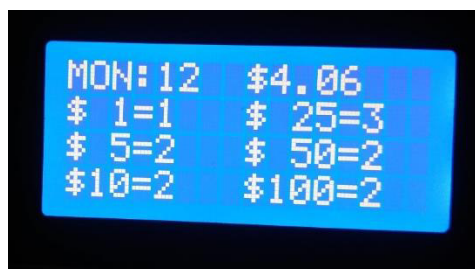


Figura 3.105 Opción RESULTADOS en prototipo

- RESETEO

Únicamente borra todos los datos del conteo que se tenga tanto en la memoria EEPROM como en las variables del conteo de monedas. Se muestra un texto indicando que se está eliminando la información, mientras el *led* indicador se apaga y se enciende, tal como se aprecia en la Figura 3.106 en simulación y en la Figura 3.107 en el prototipo.

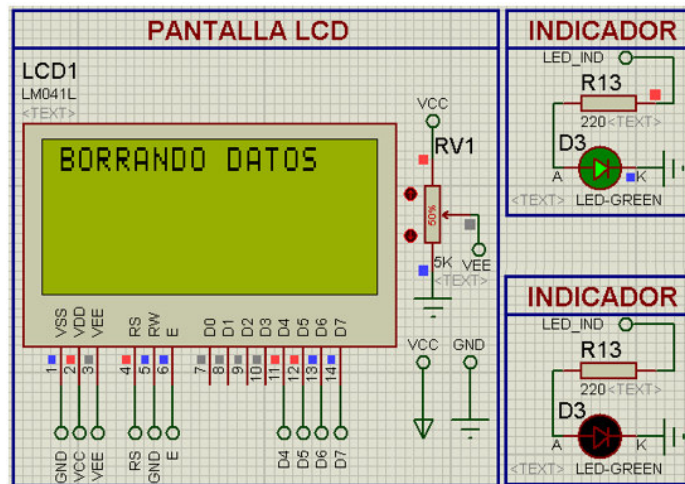


Figura 3.106 Simulación de la opción *RESET*



Figura 3.107 Opción *RESET* en prototipo

Pruebas del conteo de monedas

Como primer punto se colocó una moneda de cada tipo en el distribuidor de monedas tal como se muestra en la Figura 3.108, posteriormente se encendió el ventilador integrado al distribuidor el cual genera la vibración y de esta forma las monedas comenzaron el proceso de conteo y clasificación.



Figura 3.108 Monedas en el distribuidor

Una vez que las monedas salieron del distribuidor y cumplieron el proceso de conteo, como resultado se tuvo a cada moneda clasificada en su respectivo contenedor, en la Figura 3.109 se aprecia el resultado y en la Figura 3.110 se muestra el conteo en la pantalla LCD.



Figura 3.109 Monedas en los contenedores



Figura 3.110 Resultados del conteo de monedas

Comprobación de la llegada de la información final al usuario

Una vez hecho el conteo de monedas se procedió a realizar el envío de la notificación al usuario, por ende, se seleccionó el modo de OPCIONES y se presionó el pulsador

ENVIAR, en la Figura 3.111 se muestra la llegada del SMS, en donde se especifica la cantidad que se contó de dinero y se añade la fecha y hora en la que se envió el mensaje. La información es correcta por lo que no se necesitó realizar cambios en la configuración.



Figura 3.111 Notificación por SMS al usuario con fecha y hora

Como información importante cabe mencionar que el número telefónico al cual se debe realizar la recarga para que el prototipo envíe las notificaciones es: **0963111971** de la operadora Tuenti.

3.6 Costo del proyecto

En la Tabla 3.15 se aprecia el detalle del costo de implementación del prototipo.

Tabla 3.15 Costo del proyecto

Cantidad	Detalle	Precio unidad	Precio Total
1	Arduino Mega 2560	\$ 20.00	\$ 20.00
6	TCRT500	\$ 1.50	\$ 9.00
4	Pulsador	\$ 0.75	\$ 3.00
2	Switch	\$ 0.50	\$ 1.00
2	Led	\$ 0.15	\$ 0.30
1	Módulo RTC	\$ 2.99	\$ 2.99
1	Módulo GPRS/GSM	\$ 12.00	\$ 12.00
1	Antena PCB	\$ 2.00	\$ 2.00
1	Pantalla LCD	\$ 9.99	\$ 9.99
1	LM7805	\$ 0.80	\$ 0.80
1	Barra de aluminio	\$ 5.00	\$ 5.00
11	Remaches pequeños	\$ 0.20	\$ 2.20
1	Madera para piezas	\$ 10.00	\$ 10.00
1	Impresión láser de PCB	\$ 1.00	\$ 1.00
3	Baquelita de cobre 10x10 (cm)	\$ 2.50	\$ 7.50
1	Baquelita perforada 10x10 (cm)	\$ 1.50	\$ 1.50
1	Cloruro férrico	\$ 1.25	\$ 1.25
1	Aluminio para piezas	\$ 8.00	\$ 8.00
2	Bisagra	\$ 0.25	\$ 0.50
1	Cerrojo pasador	\$ 0.50	\$ 0.50
1	Tarjeta SIM	\$ 3.00	\$ 3.00
7	Resistencia	\$ 0.10	\$ 0.70
1	Potenciómetro	\$ 0.40	\$ 0.40
8	Tblock de 2 posiciones	\$ 0.30	\$ 2.40
8	Tblock de 3 posiciones	\$ 0.50	\$ 4.50
3	Tblock de 4 posiciones	\$ 0.80	\$ 2.40
1	Enchufe	\$ 1.50	\$ 1.50
1	Jack hembra	\$ 0.50	\$ 0.50
29	Tornillo y tuerca	\$ 0.05	\$ 1.45
1	Paquete de cables de conexión	\$ 8.00	\$ 8.00
1	Estructura del prototipo	\$ 20.00	\$ 20.00
1	Mano de obra	\$ 200.00	\$ 200.00
COSTO TOTAL DE IMPLEMENTACIÓN			\$ 343.38

4 CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- Al elaborar el prototipo se concluyó que es importante la creación e innovación de nuevos proyectos los cuales puedan cubrir ciertas actividades o procesos que pueden llegar a ser molestos o tediosos llegando a ocasionar malestar y que además influyen en la productividad de forma negativa.
- Se concluyó que un aspecto muy importante para identificar los requerimientos y necesidades fue realizar una investigación muy detallada sobre los elementos y herramientas que se requirió para la implementación tanto en *software* como son los diferentes programas a utilizar para el diseño y simulación de circuitos, además de los diferentes programas para crear el código de programación. Y a nivel de *hardware* como los sensores, módulos, *shields*, entre otros. De esta forma se logró elegir los elementos que se adaptaron de la mejor manera a los requerimientos establecidos y así garantizar un buen funcionamiento.
- Al realizar el estudio de consumo de corriente se concluyó que es fundamental elegir una fuente de poder capaz de satisfacer la demanda de corriente y voltaje, ya que, si no se suministra el voltaje y corriente adecuado, los elementos presentan un mal funcionamiento e incluso fallas.
- Al realizar el diseño del prototipo se concluyó que es importante la elección del tipo de material usado para cada estructura o elemento para que al momento de la implementación no se presenten inconvenientes. Por ejemplo, al realizar la barra clasificadora de monedas con madera se tuvieron problemas con las monedas de 1 dólar y 1 centavo, las cuales se atoraban y no permitían el desplazamiento correcto de las demás monedas. Por ende, se tuvieron que realizar pruebas con diferentes tipos de materiales y en base a los resultados obtenidos se eligió la mejor opción. Siendo así, el aluminio ya que este material permitió el correcto desplazamiento de todas las monedas.
- Al programar el microcontrolador se concluyó que el uso de diagramas de flujo es fundamental para tener una idea clara de cómo se debe estructurar el código de programación y con ello llevar un orden. Además, que al usar subrutinas se

puede crear subprogramas que contengan ciertas funciones a ejecutar solamente cuando se las requiera y de esta forma se agiliza y evita errores. Adicionalmente, con la creación de variables se facilita la modificación y comprensión del código fuente. Con el buen uso de estos recursos se programa más rápido.

- Para realizar los circuitos se concluyó que el mejor *software* para diseño y simulación es ISIS *Proteus*, el mismo abarca una gran cantidad de recursos y posibilidades, principalmente por su herramienta ARES que permite la creación y diseño de la placa PCB de una forma muy intuitiva y fácil. Además, que permitió la simulación de los diferentes circuitos que se implementó en el prototipo.
- Al realizar las pruebas de funcionamiento se concluyó que para lograr un óptimo funcionamiento se debe colocar el prototipo en una superficie plana y recta, ya que si hay alguna inclinación o deformación en la superficie esto afectará el proceso de conteo y clasificación, debido a que la barra clasificadora tiene un ángulo de inclinación adecuado para que las monedas se deslicen hacia los contendores. Al afectar este ángulo las monedas pueden deslizarse muy rápido o a su vez no lograr deslizarse.
- Se concluyó que se deben colocar los elementos que utilicen comunicación serial una misma velocidad de comunicación con Arduino, ya que al no estar sincronizados se tendrá un mal funcionamiento, en el caso del módulo GPRS/GSM se tuvo problemas para la configuración de comandos AT debido a que Arduino estaba en una tasa de transferencia de 9600 (baudios) y el módulo a una tasa de transferencia de 115200 (baudios). Entonces al enviar los comandos AT para la configuración del módulo se observaba unos caracteres desconocidos en el puerto serial. Por ende, no se tenía una sincronización entre estos elementos. Al configurar una misma tasa de transferencia se logró la configuración y funcionamiento correcto del módulo GPRS/GSM.

4.2 Recomendaciones

- Al realizar el diseño de los circuitos PCB es recomendable colocar un grosor de pista de conducción adecuado y poner toda la señalización necesaria con el objetivo de conocer la manera en la cual se debe colocar los elementos con

polaridad. Además, se recomienda hacer un diseño óptimo en donde se tenga una lógica de colocación de los elementos y así evitar un diseño no eficiente.

- Al ser un prototipo escalable se recomienda implementar mejoras tales como un teclado matricial el cual permita un mejor manejo, de igual forma se podría agregar un módulo *Wi-Fi* el cual permita subir y enviar información por medio de la red a una base de datos, entre otras opciones.
- Se recomienda conocer el funcionamiento de los elementos electrónicos ya que, si no se cuenta con los recursos para simulación, se puede crear algún circuito que emule el funcionamiento del mismo y con ello lograr la simulación de ciertos procesos. Tal es el caso del sensor detector TCRT5000 al cual se le diseñó un circuito emulador para realizar la simulación en el *software*.
- Para realizar la programación y simulación se recomienda tener un conocimiento pleno de los programas y herramientas a utilizar en este caso se utilizó *ISIS Proteus* y el entorno de programación de *Arduino*, con el objetivo de hacer un buen trabajo y no perder tiempo. Adicionalmente, es recomendable usar solamente los recursos necesarios y no incluir cosas que no sean relevantes para el desarrollo del mismo.
- Para la implementación de los elementos electrónicos es recomendable utilizar cables adecuados y soldar los mismos correctamente, ya que en varias ocasiones al manipular el prototipo se desoldaban los cables de los pulsadores ubicados en el panel de control, y al momento de realizar las pruebas no se tenía una respuesta adecuada de estos por este detalle.
- En ocasiones el prototipo presentaba un mal funcionamiento debido a la conexión de los cables, por lo que, se recomienda colocar los cables de conexión en los pines correctos y verificar la polarización de alimentación de los elementos y módulos.
- Al momento de realizar las pruebas con el módulo *GPRS/GSM* se observó que se tenía errores de conexión a la red debido a que no se le suministraba la corriente adecuada, por lo que el módulo no era capaz de mantener una conexión estable con la red, por lo tanto, se implementó una fuente de poder

externa que daba las garantías de voltaje y corriente al módulo. Además, al estar el módulo ubicado dentro del prototipo se tuvo una disminución del rango de señal por lo que se le implementó una antena de tipo PCB con el fin de aumentar el rango de recepción de señal y evitar fallos o desconexiones de la red.

- Es recomendable revisar el manual de mantenimiento en caso de tener fallas o un mal funcionamiento del prototipo, con el fin de encontrar la solución a los inconvenientes y evitar dañar el prototipo.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. E. Comercio, «El Comercio,» 12 octubre 2020. [En línea]. Available: <https://www.elcomercio.com/tendencias/ciencia/duracion-virus-covid19-pantallas-monedas.html>. [Último acceso: diciembre 2020].
- [2] J. Gonzáles, «Repositorio Universidad Politécnica de Cartagena,» diciembre 2013. [En línea]. Available: <https://repositorio.upct.es/bitstream/handle/10317/3797/pfc5630.pdf?sequence=1>.
- [3] Arduino, «ARDUINO.cl,» [En línea]. Available: <https://arduino.cl/arduino-mega-2560/#:~:text=El%20Arduino%20Mega%202560%20es,y%20un%20bot%C3%B3n%20de%20reseteo..> [Último acceso: noviembre 2020].
- [4] Y. Fernández, «xataka.com,» agosto 2018. [En línea]. Available: <https://www.xataka.com/basics/arduino-raspberry-pi-que-cuales-sus-diferencias>. [Último acceso: noviembre 2020].
- [5] A. García, «AZ adslzone,» marzo 2017. [En línea]. Available: <https://www.adslzone.net/noticias/seguridad/verkada-hackeo-150-000-camaras/>. [Último acceso: noviembre 2020].
- [6] Admin, «Geek Factory,» 20 marzo 2015. [En línea]. Available: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/ds1307-en-tinyrtc-con-arduino/>. [Último acceso: noviembre 2020].
- [7] A. ELECTRÓNICA, «arcaelectronica.com,» 2020. [En línea]. Available: <https://www.arcaelectronica.com/products/modulo-reloj-de-tiempo-real-rtc-ds1307>. [Último acceso: noviembre 2020].
- [8] Components, «COMPONENTS 101,» 2020. [En línea]. Available: <https://components101.com/ics/ds1307-i2c-real-time-clock-rtc>. [Último acceso: noviembre 2020].

- [9] I. Maxim Integrated Products, «maxim integrated TM,» 2015. [En línea]. Available: <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>. [Último acceso: noviembre 2020].
- [10] U. Electronics, «UNIT Electronics,» 2020. [En línea]. Available: <https://uelectronics.com/producto/iot-ga6-modulo-iot-g-a6-gsm-gprs-soporta-chip-4g/>. [Último acceso: diciembre 2020].
- [11] D. E. A2P, «YouTube,» 19 julio 2019. [En línea]. Available: <https://www.youtube.com/watch?v=XxJlyhz1JsU>. [Último acceso: diciembre 2020].
- [12] Educalingo, «Educalingo,» 2020. [En línea]. Available: <https://educalingo.com/es/dic-en/uart>. [Último acceso: diciembre 2020].
- [13] U. Electronics, «UNIT Electronics,» 2020. [En línea]. Available: http://uelectronics.com/wp-content/uploads/Modulo_IOT-GA6_PINOUT.png. [Último acceso: diciembre 2020].
- [14] K. S. Technology, «Indiamart,» 2020. [En línea]. Available: <https://www.indiamart.com/proddetail/gsm-pcb-antenna-20561213555.html>. [Último acceso: diciembre 2020].
- [15] Alibaba, «Turkish,» 2020. [En línea]. Available: <https://turkish.alibaba.com/product-detail/manufacture-supplier-copper-wire-spring-antenna-springs-60736685393.html?spm=a2700.7724857.0.0.556f205bBdjpHG>. [Último acceso: diciembre 2020].
- [16] T. Technology, «Makerfabs,» 28 febrero 2016. [En línea]. Available: https://www.makerfabs.com/desfile/files/A6_A7_A6C_datasheet-EN.pdf. [Último acceso: diciembre 2020].
- [17] NETTIGO, «NETTIGO EU,» 2020. [En línea]. Available: <https://nettigo.eu/products/sim800l-gsm-grps-module#:~:text=SIM800L%20is%20a%20miniature%20cellular,that%20require%20long%20range%20connectivity..> [Último acceso: noviembre 2020].

- [18] ELECTROPRO, «ELECTROPRO,» 23 diciembre 2017. [En línea]. Available: <http://electropro.pe/image/data/imgProductos/140.%20M%C3%B3dulo%20GSM%20SIM800/SIM800L.pdf>. [Último acceso: noviembre 2020].
- [19] I. m. ENGINEERS, «Last minute ENGINEERS,» 2020. [En línea]. Available: [https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/#:~:text=SIM800L%20GSM%20Module%20Pinout&text=NET%20is%20a%20pin%20where,3.4V%20to%204.4%20volts.&text=RxD%20\(Receiver\)%20pin%20is%20used,is%20used%20for%20serial%20communication..](https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/#:~:text=SIM800L%20GSM%20Module%20Pinout&text=NET%20is%20a%20pin%20where,3.4V%20to%204.4%20volts.&text=RxD%20(Receiver)%20pin%20is%20used,is%20used%20for%20serial%20communication..) [Último acceso: noviembre 2020].
- [20] S. Vishay, «AllDatasheet,» 2020. [En línea]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/26406/VISHAY/TCRT5000.html>. [Último acceso: noviembre 2020].
- [21] ELECTRO, «ELECTROCOMPONENTES,» 2020. [En línea]. Available: <https://www.electrocomponentes.es/movimiento/modulo-tcrt5000-seguidor-de-lineas-ir-obstaculos-637-.html>. [Último acceso: noviembre 2020].
- [22] Solectro, «SolectroShop,» 2020. [En línea]. Available: <https://solectroshop.com/es/sensores-de-distancia-y-proximidad-ir-y-ultrasonidos/93-modulo-tcrt5000-sensor-infrarrojo-ir-seguidor-linea-obstaculos.html#description>. [Último acceso: noviembre 2020].
- [23] VISHAY, «AllDatasheet,» 03 julio 2000. [En línea]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/26406/VISHAY/TCRT5000.html>. [Último acceso: noviembre 2020].
- [24] C. Plus Electronis, «Electronica.com.ve,» 2020. [En línea]. Available: https://electronica.com.ve/new/catalog/product_info.php?products_id=3793#:~:text=El%20TCRT5000%20es%20un%20sensor,objeto%20pasa%20enfrente%20del%20sensor.&text=La%20construcci%C3%B3n%20compacta%20detecta%20la,reflexiva%20del%20IR%20del%20objeto.. [Último acceso: noviembre 2020].
- [25] Iberobotics, «Iberobotics,» 2020. [En línea]. Available: <https://www.iberobotics.com/producto/modulo-sensor-siguelineas-tcrt5000-ir/>. [Último acceso: noviembre 2020].

- [26] GeekFactory, «GeekFactory,» 2020. [En línea]. Available: <https://www.geekfactory.mx/tienda/sensores/cny70-sensor-optico-de-reflexion/>. [Último acceso: noviembre 2020].
- [27] ElectronicosCALDAS, «Electronicos,» 2020. [En línea]. Available: <https://www.electronicoscaldas.com/es/infrarrojos/380-sensor-optico-reflectivo-cny70.html>. [Último acceso: noviembre 2020].
- [28] Juanpch, «Foros Ecuador,» mayo 2013. [En línea]. Available: <http://www.forosecuador.ec/forum/ecuador/educaci%C3%B3n-y-ciencia/198281-monedas-y-billetes-del-ecuador-historia-e-im%C3%A1genes>. [Último acceso: diciembre 2020].
- [29] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-1-dime-1965-2020/?tid=11650>. [Último acceso: diciembre 2020].
- [30] Juanpch, «Foros Ecuador,» 24 enero 2020. [En línea]. Available: <http://www.forosecuador.ec/forum/ecuador/educaci%C3%B3n-y-ciencia/198281-monedas-y-billetes-del-ecuador-historia-e-im%C3%A1genes>. [Último acceso: diciembre 2020].
- [31] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-1-cent-2010-2020/?tid=11528>. [Último acceso: diciembre 2020].
- [32] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-5-cents-2006-2020/?tid=11557>. [Último acceso: diciembre 2020].
- [33] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-1-4-dollar-1965-1998/?tid=11720>. [Último acceso: diciembre 2020].
- [34] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-1-dollar-2000-2008/?tid=11834>. [Último acceso: diciembre 2020].
- [35] UCOIN, «ES.UCOIN,» 2007. [En línea]. Available: <https://es.ucoin.net/coin/usa-1-2-dollar-1977-2020/?tid=11788>. [Último acceso: diciembre 2020].
- [36] R. Alonso, 03 abril 2020. [En línea]. Available: <https://hardzone.es/reportajes/comparativas/raspberry-pi-vs-arduino/>. [Último acceso: diciembre 2020].

- [37] I. CUENCA, «Mercado Libre,» 2020. [En línea]. Available: https://articulo.mercadolibre.com.ec/MEC-429845273-pantalla-lcd-16x4-1604-_JM#position=1&type=item&tracking_id=545d57e3-62e7-471a-8885-2346cbdc94f8. [Último acceso: diciembre 2020].
- [38] W. Display, «Winstar,» 2020. [En línea]. Available: <https://www.winstar.com.tw/es/products/character-lcd-display-module/lcd-16x4.html>. [Último acceso: diciembre 2020].
- [39] T. Instruments, «TI,» abril 2013. [En línea]. Available: https://www.ti.com/lit/ds/symlink/lm2696.pdf?ts=1615605783897&ref_url=https%253A%252F%252Fwww.google.com%252F. [Último acceso: diciembre 2020].
- [40] U. Electronics, «uelectronics,» 2020. [En línea]. Available: <https://uelectronics.com/producto/regulador-lm7805/>. [Último acceso: diciembre 2020].
- [41] E. CDMX, «CDMX electronica,» 2020. [En línea]. Available: <https://cdmxelectronica.com/producto/regulador-lm7805/>. [Último acceso: diciembre 2020].
- [42] L. Llamas, «Luis Llamas web,» 18 octubre 2016. [En línea]. Available: <https://www.luisllamas.es/reloj-y-calendario-en-arduino-con-los-rtc-ds1307-y-ds3231/#:~:text=El%20DS3231%20tiene%20una%20precisi%C3%B3n,errores%20en%20un%20desfase%20acumulado..> [Último acceso: diciembre 2020].

ANEXOS

Anexo: Código de programación.

Anexo: Manual de mantenimiento.

Anexo: Manual de uso.