



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS DE VIDEOJUEGO

Pedro Pablo Samuel Muñoz Taniel

Asesorado por el Ing. Carlos Eduardo Guzmán Salazar

Guatemala, marzo de 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE
COMPUTADORES Y CONSOLAS DE VIDEOJUEGO**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

PEDRO PABLO SAMUEL MUÑOZ TANIEL

ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, MARZO DE 2015

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Narda Lucía Pacay Barrientos
VOCAL V	Br. Walter Rafael Véliz Muñoz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADORA	Inga. Ingrid Salomé Rodríguez de Loukota
EXAMINADOR	Ing. Julio Rolando Barrios Archila
EXAMINADOR	Ing. José Antonio de León Escobar
SECRETARIA	Inga. Marcia Ivonne Véliz Vargas

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS DE VIDEOJUEGO

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, el 6 de marzo de 2014.

Pedro Pablo Samuel Muñoz Taniel

Guatemala, 10 de octubre de 2014

Señor
Coordinador del Área de Electrónica
Escuela de ingeniería Mecánica eléctrica
Facultad de ingeniería
Universidad de San Carlos de Guatemala

Estimado coordinador:

Por este medio informo a usted que he dado mi aprobación al trabajo de graduación del señor Pedro Pablo Samuel Muñoz Taniel, titulado **DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMATICO DE COMPUTADORES Y CONSOLAS DE VIDEOJUEGO**. Pues ha cumplido con los objetivos que se propusieron en el protocolo respectivo.

Por lo que, tanto el señor Muñoz Taniel y el suscrito en su calidad de ASESOR del trabajo de graduación somos, responsables del contenido del mismo.

Sin otro particular, lo saludo atentamente.



Carlos Guzmán Salazar
Ingeniero Electricista
Colegiado No. 2762

CARLOS GUZMAN SALAZAR
Ingeniero Electricista
Col. No. 2762



FACULTAD DE INGENIERIA

Ref. EIME 55 2014
Guatemala, 21 de Octubre 2014.

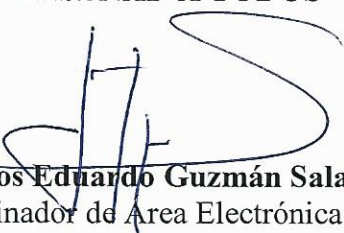
Señor Director
Ing. Guillermo Antonio Puente Romero
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

**Me permito dar aprobación al trabajo de Graduación titulado:
DISEÑO DE CONTROLADOR Y TEMPORIZADOR
AUTOMÁTICO DE COMPUTADORES Y CONSOLAS DE
VIDEOJUEGO del estudiante, Pedro Pablo Samuel Muñoz Taniel,
que cumple con los requisitos establecidos para tal fin.**

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS


Ing. Carlos Eduardo Guzmán Salazar
Coordinador de Área Electrónica.





REF. EIME 55. 2014.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; PEDRO PABLO SAMUEL MUÑOZ TANIEL titulado: DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS DE VIDEOJUEGO, procede a la autorización del mismo.


Ing. Guillermo Antonio Puente Romero



GUATEMALA 20 DE NOVIEMBRE 2,014



DTG. 109 .2015

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS DE VIDEOJUEGO**, presentado por el estudiante universitario: **Pedro Pablo Samuel Muñoz Taniel**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Murphy Olimpo Paiz Recinos
Decano



Guatemala, 13 de marzo de 2015

/gdech

ACTO QUE DEDICO A:

Jesucristo

Por ser mi Dios, mi Salvador, la principal razón de mi existencia. Porque me da la vida, la sabiduría, las fuerzas para seguir adelante día con día, brindándome todo su amor y bendición de forma incondicional. Pues todo lo que tengo es por Él y su misericordia

Mi madre

Candelaria Taniel, por haberme ayudado durante toda mi vida, toda mi carrera y por ser una gran bendición; por haberme apoyado sin importar cuál sea mi problema o mi situación, quien ha conocido, vivido junto a mí los sacrificios por los que he pasado para lograr esta meta.

Miguel Ángel Ramírez

Quien como pastor de mi iglesia, ha sido un importante apoyo moral y espiritual, brindándome consejos en muchos aspectos de mi vida.

Mis hermanos

David, Jonás, Raquel y Rebeca Muñoz Taniel, quienes en muchas ocasiones me han apoyado y compartido su conocimiento y consejos para mi vida personal y profesional.

Mis mejores amigos

Jorge Roberto García Velásquez, Julio Roberto Luna Reynoso, Evelyn Oliva y Karla Ríos; con quienes he compartido los esfuerzos y problemas de la carrera.

AGRADECIMIENTOS A:

**Universidad San Carlos
de Guatemala**

Por ser mi casa de estudios, mi alma máter;
por haberme formado como profesional.

Facultad de Ingeniería

Por ser el lugar donde pude formarme como
ingeniero y haber conocido muchos aspectos
de la vida y la ciencia que desconocía.

**Ing. Carlos Eduardo
Salazar Guzmán**

Por haberme brindado su apoyo e instruido
en varios cursos durante la carrera.

Jehu Reyes

Supervisor de mis prácticas finales; quien me
enseñó e instruyó en varios temas de
telecomunicaciones y convivencia con mis
compañeros; así también, por haberme
brindado su amistad.

Mis amigos de SENCOM

Por apoyarme en la realización de mis
prácticas y por brindarme su valiosa amistad.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	VII
LISTA DE SÍMBOLOS	IX
GLOSARIO	XI
RESUMEN.....	XV
OBJETIVOS.....	XVII
INTRODUCCIÓN	XIX
1. CIBERCAFÉ.....	1
1.1. Computador.....	1
1.1.1. Software.....	1
1.1.1.1. Sistema operativo	2
1.1.1.2. Aplicación de software	2
1.1.2. Hardware	3
1.1.3. Computador personal.....	3
1.2. Cibercafé	5
1.2.1. Historia.....	5
1.2.2. El cibercafé actualmente	6
1.2.3. Formas de administración.....	7
1.2.3.1. Forma manual.....	7
1.2.3.2. Por software.....	8
1.2.3.3. Por mecanismos de hardware	8
1.2.4. Clasificación de los cibercafés	9
1.2.4.1. Enfocado a videojuegos.....	9
1.2.4.2. Enfocado a internet.....	9

2.	SALÓN DE VIDEOJUEGOS	11
2.1.	Videoconsola.....	11
2.1.1.	Historia y generaciones de videoconsolas.....	11
2.1.1.1.	Primera generación	12
2.1.1.2.	Segunda generación	12
2.1.1.3.	Tercera generación	13
2.1.1.4.	Cuarta generación	13
2.1.1.5.	Quinta generación	14
2.1.1.6.	Sexta generación.....	14
2.1.1.7.	Séptima generación.....	15
2.1.1.8.	Octava generación	15
2.1.2.	Videojuego.....	15
3.	DESCRIPCIÓN DE LOS COMPONENTES FÍSICOS PARA EL DISEÑO PROPUESTO.....	17
3.1.	Componentes pasivos.....	17
3.1.1.	Dispositivo receptor de monedas.....	17
3.1.2.	Resistor eléctrico	18
3.1.3.	Relé	19
3.1.4.	Oscilador electrónico de cristal	20
3.1.5.	Capacitor	20
3.1.5.1.	Capacitor cerámico.....	21
3.1.5.2.	Capacitor electrolítico	21
3.1.6.	Microswitch	22
3.1.7.	Transformador	22
3.1.8.	Puente de rectificador.....	23
3.1.9.	Conector RCA.....	23
3.2.	Conector USB estándar	24
3.3.	Componentes activos.....	25

3.3.1.	Transistor de unión bipolar (BJT)	25
3.3.2.	Circuito integrado FT232RL	28
3.3.3.	El microcontrolador	29
3.3.3.1.	Partes principales de un microcontrolador	29
3.3.3.1.1.	CPU	29
3.3.3.1.2.	Memoria.....	31
3.3.3.1.3.	Unidades de entrada y salida	33
3.3.3.1.4.	Reloj principal	33
3.3.3.2.	Recursos especiales del microcontrolador	34
3.3.3.2.1.	Temporizadores o <i>timers</i>	34
3.3.3.2.2.	Perro guardián o <i>watchdog</i>	34
3.3.3.2.3.	Brownoutreset.....	34
3.3.3.2.4.	Estado de bajo consumo	35
3.3.3.2.5.	Conversores A/D y D/A.....	35
3.3.3.2.6.	PWM.....	35
3.3.3.2.7.	Puertos de comunicación.....	36
3.3.3.3.	Herramientas de desarrollo.....	37
3.3.3.4.	AVR	38
3.3.3.4.1.	ATmega328	40
3.3.3.4.2.	Arduino Duemilanove	41
3.3.4.	Módulo RF	44
3.3.5.	Módulo LCD 20x4	45
3.3.6.	Servomotor	47

4.	DESARROLLO FÍSICO DEL PROTOTIPO.....	49
4.1.	Receptor de monedas	49
4.1.1.	Componentes mecánicos del receptor de monedas.....	49
4.1.2.	Circuito electrónico	53
4.2.	Interfaz de conexión con un computador	54
4.3.	Circuito de conexión con una videoconsola	55
4.4.	Módulo LCD	56
4.5.	Módulo RF o llave electrónica	57
4.6.	Candado electrónico	57
4.6.1.	Parte mecánica.....	58
4.6.2.	Circuito servomotor.....	58
4.6.3.	Fuente de alimentación propuesta.....	59
4.7.	Circuito general	60
4.7.1.	Diseño de PCB	61
4.8.	Diagrama de conexión del prototipo.....	63
4.8.1.	Conexión para control de computador	63
4.8.2.	Conexión para control de videoconsola.....	64
4.9.	Contenedor del prototipo	64
5.	DESARROLLO DEL SOFTWARE DEL PROTOTIPO	67
5.1.	Programación del microcontrolador	67
5.2.	Programación del software para la PC.....	76
5.2.1.	Parte gráfica	76
5.2.2.	Código del programa	77
6.	FUNCIONAMIENTO DEL PROTOTIPO	83
6.1.	Funcionamiento como controlador de videoconsolas	83
6.2.	Funcionamiento como controlador de computadores	84
6.2.1.	Funcionamiento de software dentro del computador	85

6.2.1.1.	Funcionamiento como administrador	85
6.3.	Funcionamiento de control (llave electrónica)	87
CONCLUSIONES		89
RECOMENDACIONES.....		91
BIBLIOGRAFÍA.....		93

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Interior de un <i>microswitch</i>	22
2.	Conectores USB estándar A y B	24
3.	Polarización de un transistor BJT	27
4.	Diagrama de conexión circuito integrado FT232	28
5.	Patillaje del Arduino Duemilanove	44
6.	Patillaje de modulo LCD 20x4	47
7.	Diagrama interno de un servomotor	48
8.	Vista lateral 1 de receptor de monedas mecánico.....	50
9.	Vista lateral 2 receptor de monedas mecánico.....	51
10.	Vista interna de receptor de monedas mecánico	52
11.	Vista frontal de receptor de monedas mecánico	53
12.	Conexión de <i>microswitch</i>	54
13.	Conexión USB entre computador y Arduino.....	54
14.	Diagrama de circuito de interfaz con videoconsola	55
15.	Diagrama de circuito de LCD	56
16.	Diagrama de circuito de conexión módulo RF.....	57
17.	Representación de la mecánica de la chapa electrónica	58
18.	Diagrama de conexión de servomotor.....	59
19.	Propuesta para elaboración de fuente de 5 voltios	60
20.	Diagrama general de todo el circuito.....	60
21.	Diseño de placa PCB circuito de adaptación.....	61
22.	Simulación de circuito impreso sin Arduino	62
23.	Simulación de circuito impreso con Arduino.....	62

24.	Diagrama de conexión para control de computador	63
25.	Diagrama de conexión para control de videoconsola	64
26.	Vista frontal y posterior del dispositivo abierto	65
27.	Ubicación de los componentes dentro del contenedor	65
28.	Diseño del programa de control en computador	76
29.	Diseño de la ventana de bloqueo.....	77
30.	Diagrama de flujo como controlador de videoconsolas.....	83
31.	Diagrama de flujo como controlador de computadores.....	84
32.	Icono de la interfaz para el computador	85
33.	Interfaz control de tiempo.....	85
34.	Inicio de funcionamiento del software	86
35.	Modo administrador de software de computadores	86
36.	Menú de opciones con control RF.	87

TABLAS

I.	Código de colores para resistores eléctricos	19
II.	Código de colores conectores y cables RCA	23
III.	Descripciones de pines conectores y cables USB	25
IV.	Resumen de características de ATmega 328	41
V.	Resumen características Arduino Duemilanove	43
VI.	Función y nombre de pines para módulo LCD 20X4	46

LISTA DE SÍMBOLOS

Símbolo	Significado
A	Amperio, medida de corriente eléctrica
cm	Centímetros, unidad de distancia
F	Faradio, medida de capacitancia
Hz	Hertz, medida de frecuencia igual a 1/s
K	Kilo, índice de notación exponencial $1 \cdot 10^3$
μ	Micro, índice de notación exponencial $1 \cdot 10^{-6}$
mm	Milímetros, unidad de distancia
n	Nano, Índice de notación exponencial $1 \cdot 10^{-9}$
Ω	Ohm, medida de resistencia eléctrica
Q	Quetzal, moneda oficial de la república de Guatemala
S	Segundos
W	Vatio, medida de potencia eléctrica
V	Voltio, medida de potencial eléctrico

GLOSARIO

Árcade	Término que se refiere a una máquina recreativa de videojuegos que se encuentra en lugares públicos.
Baudios	Unidad de velocidad de bits por segundo.
Bits	Unidad básica de datos 0 o 1.
Bytes	8 bits. Tipo de variable en programación, que admite enteros entre 0 y 255.
C#	Lenguaje de programación orientado a objetos desarrollado por la empresa de software Microsoft.
Circuito	Conjunto de componentes electrónicos, conectados entre sí, que tienen al menos una trayectoria cerrada.
<i>Datasheet</i>	Hoja de datos. Documento realizado por el fabricante de un producto electrónico, el cual menciona todas sus características y modos de funcionamiento.

Diagrama	Diseño o representación de un circuito electrónico.
<i>Firmware</i>	Bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria.
GND	Del término inglés <i>groud</i> , es la referencia o voltaje cero de circuito.
IC	<i>Integrated circuit</i> . Circuito Integrado.
Lenguaje de programación	Lenguaje que sirve para expresar procesos o funciones que pueden ser llevados a cabo por un computador.
Livewire	Programa especializado para la realización y simulación de componentes electrónicos.
Placa de circuito	Placa física de baquelita o fibra de vidrio que se utiliza para un circuito impreso.
PCB	<i>Printed circuit board</i> . Tarjeta de circuito impreso.
PCBWizard	Programa de diseño de tarjeta de circuito impreso.

PinOut

Patillaje. Término en inglés que se refiere a una imagen o representación que muestra la numeración y nombre de cada pin de un componente electrónico.

Pull-down

Resistor eléctrico que se conecta entre la referencia y la entrada de un circuito de alta impedancia, para evitar errores por ruido electromagnético.

Visual studio

Entorno de desarrollo integrado creado por Microsoft.

Vcc

Voltaje de alimentación de un circuito.

RESUMEN

El presente trabajo de graduación abarca los conocimientos teóricos necesarios para el diseño completo de un controlador y temporizador de videoconsolas o de computadores.

En el primer capítulo se da información detallada de lo que es un cibercafé, sus funciones y la forma en que este se administra. Todo con el propósito de dar una justificación teórica del porqué de la construcción del dispositivo controlador.

En el segundo capítulo se muestra una visión general de los videojuegos, su historia y las salas de videojuegos. Esto con el propósito de entender las diferencias entre cada generación de video consola.

En el tercer capítulo se incluye una descripción de los componentes físicos mecánicos y electrónicos utilizados para la construcción de un prototipo propuesto. También se presentan de forma detallada las características que poseen los distintos componentes utilizados.

En el cuarto y quinto capítulo se describe el desarrollo físico y digital del prototipo, es decir se proveen los diagramas de componentes y cómo construirlos. En el cuarto capítulo se analiza la conexión de los componentes utilizados hacia el microcontrolador Atmega328 sobre la placa Arduino y la elaboración del circuito de adaptación a la misma. En el quinto capítulo se presenta la programación del microcontrolador y de software desarrollado para la administración de un computador que utilice el sistema operativo Linux.

En el sexto capítulo se describe un breve manual que explica el funcionamiento del dispositivo, tanto desde el punto de vista de usuario, como del administrador.

OBJETIVOS

General

Presentar el diseño de un dispositivo que permita el control y la temporización de computadores y consolas de videojuego.

Específicos

1. Presentar los antecedentes y componentes de un cibercafé.
2. Presentar los antecedentes de las consolas de videojuego y las características principales de cada generación de videoconsola.
3. Describir los componentes físicos utilizados para el diseño propuesto.
4. Explicar el funcionamiento de la programación desarrollada para el dispositivo controlador.
5. Presentar un resumen que explique al usuario y administrador acerca del funcionamiento del dispositivo.

INTRODUCCIÓN

Un cibercafé o café internet es un local de acceso al público en general, donde se ofrece a los clientes acceso a internet o entretenimiento a través de distintos dispositivos electrónicos. Para ello, el local dispone de computadoras y usualmente cobra una tarifa fija por un período determinado para el uso de dichos equipos.

Por otro lado, una videoconsola o consola de videojuegos es un sistema electrónico de entretenimiento que ejecuta juegos electrónicos y que pueden formar parte de un café internet.

Actualmente no existe un controlador o temporizador que integre los mecánicos de software y hardware para cibercafés y videoconsolas. Por lo que se diseñó una propuesta de un controlador y temporizador de computadores que integrara tanto sistemas de software como de hardware, para que tenga un fácil uso y permita al administrador ahorrar tiempo.

La modernización y los servicios de computadores han llegado a formar parte de la vida cotidiana del ser humano. Asimismo muchos negocios prestan estos servicios a diversos usuarios siendo no solo el uso de internet sino también otros servicios como la utilización de ofimática e incluso servicios de entretenimiento como los videojuegos, por lo que la creación de herramientas para su administración debe ser segura y de fácil manejo.

1. CIBERCAFÉ

Para comprender el dispositivo que se desea realizar y entender a totalidad el presente trabajo de graduación es necesario tener una base teórica para que apoye las ideas planteadas. A continuación se procederá a explicar los conceptos básicos necesarios ligados a un cibercafé.

1.1. Computador

Es un conjunto de componentes electrónicos que recibe y procesa datos para convertirlos en información útil. Un computador es un conglomerado de circuitos integrados, condensadores, resistencias, bobinas y otros componentes interconectados; además puede ejecutar con gran exactitud y velocidad algún servicio requerido por un usuario o varios de los mismos. Todo computador posee dos componentes principales los cuales software y hardware.

1.1.1. Software

Comprende todos los elementos lógicos del computador, es decir la parte no tangible, los cuales permiten el soporte informático al hardware (partes tangibles del computador). Este ordenamiento lógico está constituido desde simples códigos binarios hasta extensos y complejos algoritmos de programación que dan lugar a la ejecución de distintas aplicaciones o programas que permiten al usuario interactuar y manipular la información de una manera útil para sus propósitos.

El software es el encargado de la administración de los componentes físicos; sin este, el computador no sería más que solo un grupo de componentes electrónicos sin ningún sentido útil; entre los elementos más importantes del software de un computador se puede encontrar:

1.1.1.1. Sistema operativo

Es un programa o grupo de programas que permite administrar los recursos de hardware del computador, es decir la ejecución lógica y secuencial de cada uno de los componentes físicos. El sistema operativo es el software base de todas las demás aplicaciones que se utilicen en el computador.

1.1.1.2. Aplicación de software

En informática, una aplicación es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos. Esto lo diferencia principalmente de otros tipos de programas como los sistemas operativos (que hacen funcionar al ordenador), las utilidades (que realizan tareas de mantenimiento o de uso general), y los lenguajes de programación (con el cual se crean los programas informáticos).

Suele resultar una solución informática para la automatización de ciertas tareas complicadas como la contabilidad, redacción de documentos, o la gestión de un almacén. Algunos ejemplos de programas de aplicación son los procesadores de textos, hojas de cálculo, y base de datos.

Ciertas aplicaciones desarrolladas a medida suelen ofrecer una gran potencia, ya que están exclusivamente diseñadas para resolver un problema específico.

Otras aplicaciones llamadas paquetes integrados de software, ofrecen menos potencia pero a cambio incluyen varias aplicaciones, como un programa procesador de textos, de hoja de cálculo y de base de datos.

1.1.2. Hardware

Se refiere a todos los componentes físicos (piezas electrónicas, soportes, discos, etc.) que posee un sistema informático; su principal función es proveer el soporte necesario para que el software se ejecute de manera virtual.

Dependiendo de la funcionalidad, el fabricante y el dispositivo que se utilice contarán con distintas características de hardware que limitarán las capacidades máximas de software del sistema.

1.1.3. Computador personal

El ordenador personal es un equipo con suficiente potencia y rendimiento para satisfacer las necesidades de un usuario medio. El ordenador personal es también conocido por la abreviatura PC (*personal computer*). El término computadora personal comienza a extenderse en los años 80 para distinguir el nuevo microprocesador de gama baja (computadora personal) de los de la gama media.

Los ordenadores personales son equipos menos potentes, pero también más baratos que las minicomputadoras. La arquitectura de un ordenador personal está diseñada para su uso en modo autónomo (usuario único). Sin embargo, los componentes de hardware de diseño modular permiten convertir un PC en un terminal de conexión a un equipo remoto.

Los ordenadores personales se dividen en tres categorías distintas:

- *Desktop*: el ordenador personal de escritorio es un modelo de escritorio con monitor, situado encima de las cosas y el teclado alfanumérico externo. Es el primer tipo de PC que surgió en el mercado. El recipiente de la placa base es de fácil acceso. Los componentes modulares y la facilidad de acceso a la placa base facilitan las operaciones de montaje y de mejora de las especificaciones de la computadora. Para agregar una nueva tarjeta, se tiene que insertarla en la ranura correspondiente
- *Tower*: la torre de ordenador personal tiene una configuración de hardware similar al escritorio, pero el caso está dispuesto en una posición vertical. Esto le permite colocar el contenedor de los componentes de hardware del ordenador, incluso debajo de la mesa o en el suelo; la desvinculación de la proximidad a la pantalla y el teclado. Los componentes de hardware y especificaciones son las mismas que una computadora de escritorio.
- *Portátil (notebook)*: los ordenadores personales son claramente diferentes de escritorio y la torre. La computadora portátil integra una plana de cristal líquido de la pantalla LCD, el caso y el teclado en un solo cuerpo. A menudo también integran un área de la pantalla táctil del teclado, toque sensible, en lugar del ratón. Cuando se termina de usar la pantalla se puede cerrar como un "*notebook*". Por esta razón se conocen también como ordenadores portátiles *notebook*. Los ordenadores portátiles han reducido tamaño y peso. Se caracterizan por una potencia de cálculo inferior a la torre. En compensación se debe integrar una batería que permita el uso de la computadora por un par de horas, incluso en ausencia de energía eléctrica.

1.2. Cibercafé

En términos generales un cibercafé es un negocio que se dedica al alquiler de computadores por determinado tiempo para proveer servicios de internet, juegos, impresiones, escaneo de documentos, entre otros, que se puedan realizar con computadores. Aunque normalmente el nombre hace pensar que dicho negocio vende bebidas con cafeína, esto no ocurre necesariamente, aunque originalmente estos proveían dicho servicio, actualmente no es algo tan común.

1.2.1. Historia

Los primeros cibercafés, fueron abiertos en Londres, Inglaterra. El primer cibercafé fue el Café Cyberia, que abrió sus puertas en septiembre del noventa y cuatro. Su fundadora, Eva Pascoe, dice que la idea se le vino a la cabeza a principios de los años 90, cuando pasaba mucho tiempo lejos de su familia trabajando en su tesis doctoral.

En esos tiempos era de las pocas personas que tenía acceso a una cuenta de correo electrónico, servicio puramente académico por aquellos días; pero al no tener nadie más en su familia una dirección de correo electrónico, debía gastar cantidades considerables de dinero en cuentas telefónicas.

Un día, sentada en un café cerca de su universidad, pensó que podría ser divertido poder ir a ese establecimiento, con su ordenador portátil y enviar correos mientras se tomaba un descanso en su rutina habitual. Echó un vistazo, alrededor y pudo reconocer a algunos amigos, de los que sabía que tenían conexión a internet desde sus casas.

Después, hablando con ellos, pensaron en cómo sería tener conexión permanente a internet desde un café y pagar una pequeña tarifa para poder intercambiar mensajes con sus amigos y familiares, enviar correo y tener mensajería instantánea. Tres meses después, en septiembre, abrieron el primer café Internet en Londres. Desde ese momento hasta ahora, los cibercafés se han multiplicado por todo el mundo.¹

1.2.2. El cibercafé actualmente

Actualmente el cibercafé se ha convertido en un negocio bastante común en la mayor parte del mundo, inclusive en varias partes de Guatemala; esto debido al crecimiento exponencial de internet, las redes sociales y otros servicios en los que usuarios normalmente requieran el uso de un computador.

Los cibercafés se han convertido en un negocio bastante difundido y viable ya que permiten a usuarios que no disponen de un computador o servicios de internet en su hogar puedan utilizar el servicio de una forma fácil y rápida, como si se tratase de cualquier servicio público.

Cabe también destacar el hecho que aunque este negocio es bastante común en gran parte del mundo, actualmente en Guatemala se carece de regularizaciones que prohíban servicios no aptos para menores o ilegales (pornografía, juegos violentos, etc.), todo esto debido a que el negocio es relativamente nuevo para el país. Dada la gran rentabilidad de los cibercafés, se ha vuelto común que algunas empresas o comerciantes particulares pongan a la venta cantidades determinadas o “paquetes” que permitan que cualquier persona con el capital suficiente pueda iniciar un negocio desde cero.

¹ www.cibercafe.us/internet/negocios/cafe/historia_del_cybercafe/. Consulta: 05 de febrero 2014.

1.2.3. Formas de administración

Habitualmente todo cibercafé provee servicios por determinada cantidad de tiempo equivalente al dinero que el cliente ha pagado; así que es necesario algún método o mecanismo que permita que el usuario tenga acceso a los servicios prestados por el computador.

Para el presente trabajo de graduación se desea proponer el diseño de un dispositivo que utilice una mezcla de mecanismos de hardware y un software dentro del computador para administrarlo de forma más eficiente, confiable y automatizada. A continuación se exponen las formas más comunes para administrar un cibercafé:

1.2.3.1. Forma manual

Es la forma más barata de administración y facilita a cualquier tipo de usuario administrar una red de computadoras en un cibercafé. Aunque es una forma simple y barata, es la que más problemas presenta debido a que el administrador del cibercafé tiene a su cargo toda la responsabilidad de las computadoras.

En este tipo de administración, el usuario suele utilizar un cronómetro para proveer el servicio a los usuarios del cibercafé, con lo cual se introducen errores al sistema, ya que el tiempo no es exacto. Asimismo, es un sistema viable solo para cibercafés pequeños (menores a 3 o 4 computadoras), ya que si se posee demasiados, resulta ser una tarea demasiado difícil para ser realizada por una sola persona, más aún si el cibercafé presta más de un solo servicio.

1.2.3.2. Por software

Suele ser la forma más común para la administración de un cibercafé, encontrando software de licencia libre o de pago. Su forma de funcionamiento suele ser bastante simple, en algunos casos hasta bastante compleja.

Existe una amplia gama de controladores para cibercafé, los cuales permiten determinar cuánto tiempo se encuentra habilitado el computador, envío de impresiones, procesos en uso, entre otros.

Aunque suele ser una forma bastante eficiente, en algunas ocasiones, se presenta el inconveniente de que un usuario experimentado puede fácilmente violar la seguridad del programa y permitirse el uso del computador más tiempo. Asimismo, se requiere de un administrador para la red y de un cierto conocimiento de software para utilizar la mayoría de controladores, es decir aunque suele ser una forma bastante eficiente en la mayoría de ocasiones, no es un sistema totalmente automatizado.

1.2.3.3. Por mecanismos de hardware

Esta es la forma de administración más cara debido a que requiere comprar algún dispositivo externo que administre el tiempo del computador; suele ser una forma bastante eficiente y casi totalmente automatizada en su mayoría. Estos mecanismos permiten administrar los computadores de forma eficiente, deshabilitando muchos servicios del computador; pero también presenta inconvenientes, ya que en ocasiones es incómodo para el usuario no contar en la pantalla del computador con una referencia de cuánto tiempo o cuánto saldo posee al momento de utilizarlo.

1.2.4. Clasificación de los cibercafés

Como antes se explicó, un cibercafé es un negocio bastante viable y extendido a lo largo del mundo. Se ha visto el fenómeno de que existen distintos negocios, los cuales se clasifican dependiendo del tipo de clientes que atienden y los servicios que prestan hacia los mismos.

1.2.4.1. Enfocado a videojuegos

Estos cibercafés se caracterizan por tener clientela joven y proveer el servicio de videojuegos populares. Para este tipo de cibercafé se requiere de computadores que exigen muchos recursos de hardware (memoria RAM, procesador, tarjeta de vídeo, etc.) para poder ejecutar de una manera fluida los videojuegos.

1.2.4.2. Enfocado a internet

En este tipo de cibercafés se puede encontrar clientes tanto jóvenes como adultos, quienes acuden al servicio de internet para realizar trabajos o revisar sus cuentas de correo electrónico, redes sociales y otras páginas web que les resultan útiles para sus trabajos estudios u ocio.

No se requiere del uso de computadores con hardware avanzado, pero sí con gran cantidad de servicios como: impresiones, fax, digitalización de imágenes, conexiones fiables y de un ancho de banda considerablemente grande para poder proveer un servicio óptimo a sus usuarios.

2. SALÓN DE VIDEOJUEGOS

Una sala de videojuegos es un local público que provee a los usuarios entretenimiento a través del alquiler de equipos en los que se puedan utilizar juegos de video. Aunque estas salas tienen menos funciones que los cibercafés, es un negocio con una historia más antigua, el cual inicia con los salones de videojuego arcade. En este trabajo de graduación se estudiarán solo los términos generales y conceptos de las videoconsolas para las cuales se desea proveer un servicio que permita a cualquier persona administrarlo de forma fácil, sin transformar la videoconsola en una consolar de arcade en su totalidad.

2.1. Videoconsola

Una videoconsola es un aparato electrónico que genera señales de video para un monitor integrado o conectado externamente de un juego electrónico que permite a sus usuarios la interacción a través de controles conectados a la misma.

2.1.1. Historia y generaciones de videoconsolas

Los videojuegos han evolucionado, desde presentarse integrados en el sistema, a estar almacenados en otros dispositivos que son introducidos en dicha consola. Estos dispositivos empezaron en forma de cartucho y pasaron por las cintas, disquetes, tarjetas de memoria, CDs, DVDs, etc.; ahora mismo, muy pocas son las videoconsolas que siguen utilizando el cartucho como medio de almacenamiento, siendo usado generalmente el formato de disco.

La industria de los videojuegos ha clasificado las videoconsolas en distintas generaciones, clasificación que determina cuándo y cómo (fecha de lanzamiento y tecnología existente), así como el ancho de bus del procesador en bits, que ha ido creciendo de generación en generación desde la segunda.

2.1.1.1. Primera generación

Aunque los videojuegos empezaron en la década de los 50, estos usaban pantallas vectoriales, (empezaron usando los osciloscopios), y no de vídeo que pone el prefijo en las consolas. Fue avanzando en 1972 cuando apareció la Magnavox Odyssey, creada por el padre de los videojuegos Ralph Baer, para la compañía electrónica Magnavox. Tuvo un éxito moderado, al menos hasta que Atari lanzó su juego arcade Pong, desbancando en popularidad a Magnavox en 1975.

2.1.1.2. Segunda generación

Esta generación se inicia con la salida de la Atari 2600 en 1976, originalmente llamada Atari, VCS logra un gran éxito durante la primera mitad de los 80. Este dominio intentó ser derribado por la consola Colecovision, que tenía el doble de colores, y por la Intellivision de Mattel, que llevaba el primer procesador de 16 bits en una consola.

A pesar del realismo en gráficos y sonido que esto permitía, solo llegó a vender un tercio de lo que vendió la Colecovision (2 millones frente a los 6 de la de Coleco). Luego, intentando no quedarse rezagada, saldría la Atari 5200, pero con la crisis de los videojuegos y la llegada de los microordenadores se hundieron las ventas en casi todos los sistemas, curiosamente, salvo la Atari 2600, que siguió tirando hasta los 90.

Estas consolas eclipsaron la salida al mercado de un par de compañías que se harían más famosas más adelante, Nintendo con su TV-game 6, y Sega, con la SG-1000.

2.1.1.3. Tercera generación

Con el *crash* en el que se sumió América del Norte dando resultado a la crisis del videojuego del 83, las industrias americanas dejaron la producción de las consolas, Japón aprovechó este momento para lanzar un golpe maestro en el 84 por parte de Nintendo: la Famicon, posteriormente conocida como la NES (Nintendo Entertainment System) tras su rediseño del 85.

Aunque en la generación anterior ya se utilizaban los 8 bits e incluso los 16, fue a esta a la que se denominó la era de los 8 bits.

Nintendo con su NES, SEGA con sus tres consolas, SG-1000 Mark II, Mark III y Sega Master System, Atari con la Atari 7800 y la poco conocida Epoch con su consola de casetes Epoch Cassette Vision, son las consolas que englobaron esta generación.

2.1.1.4. Cuarta generación

La era de los 16 bits; comenzó el 30 de octubre de 1987, con la salida del PC Engine (TurboGrafx-16 en América del Norte) por parte de la Nippon Electric Company, siendo la primera de esta generación. No le sirvió de mucho ante el gran dominio que supusieron la Súper Nintendo (Súper Famicon o comúnmente denominada SNES) y la Sega Mega Drive, que se hicieron aún más populares, gracias al gran impacto sobre el público de sus personajes más carismáticos, como Mario y Sonic.

El resto de consolas no tuvieron éxito, excepto la Neo-Geo de SNK, las que lo tuvieron un intenso periodo de guerra comercial que fue denominado como la época dorada de los videojuegos.

2.1.1.5. Quinta generación

Más conocida como la era de los 32 bits. Esta era comienza en el 94, con la comercialización de la Sega Saturn y la PlayStation. Fue también conocida porque pasó de los entornos 2D a los 3D. Aquí empezó el declive de otras empresas, la Atari Jaguar, la 3DO Interactive Multiplayer (de la unión de Panasonic, Sanyo y Goldstar) debido a su pobre *marketing*, fallando a la hora de crear impacto. También fue una generación importante en cuanto a que empezó a preocuparse verdaderamente por la ergonomía de los mandos.

2.1.1.6. Sexta generación

La era de los 128 bits empieza con la Sega Dreamcast, a finales de noviembre del 98. Esta generación empieza a tomar parecidos arquitectónicos con los ordenadores personales, empezando a encaminarse hacia la multimedia, claro ejemplo de ello fue la PlayStation 2, que podía reproducir películas en DVD, y empezando a añadir las conexiones a internet, de las que la Dreamcast fue pionera con un modem de 33.6 Kb, aunque a pesar de tales novedades, esta fue la última consola de Sega, debido a un gran fallo en su seguridad anti-copia de los juegos.

De esta generación también forma parte la Game Cube, de Nintendo, un intento de recuperar el mercado perdido frente a la PlayStation con la N64, y la Xbox, innovando con la inclusión de un disco duro para guardar la información dentro de la propia consola.

2.1.1.7. Séptima generación

La era actual está marcada por los procesadores multinúcleo y el juego en red, y en otra medida la competición Blue-ray vs. Hd-dvd que mantuvieron PlayStation 3 y XBox360, respectivamente. Desmarcándose de la competición de potencia de esas casas, la Wii (Nintendo) siguió otro camino, la innovación, con un mando que hacía el juego más realista al apuntar y notar las aceleraciones.

2.1.1.8. Octava generación

La próxima generación de videoconsolas destacable por el uso de internet como eje central de la funcionalidad de las consolas. Aunque las generaciones anteriores de videoconsolas normalmente se han sucedido en ciclos de cinco años, la transición de la séptima a la octava generación ha durado más de seis años. La transición es igual de inusual en que la consola de la anterior generación que tuvo más ventas.²

2.1.2. Videojuego

Un videojuego consiste en una simulación digital sonora y de imágenes que se ejecutan en un dispositivo electrónico como una computadora o consola, con el propósito proveer entretenimiento. Dependiendo de la plataforma, un videojuego tendrá diferentes tipos de controles como el ejemplo de un computador hace uso del teclado o el *mouse*. También provee audio y vídeo; muchos videojuegos proveen otros métodos interactivos para que el usuario perciba una cierta realidad virtual.

² Autora: Lidia contreras. (<http://histinf.blogs.upv.es/2011/02/15/historia-de-las-videoconsolas/> consulta: 15 de febrero 2014).

3. DESCRIPCIÓN DE LOS COMPONENTES FÍSICOS PARA EL DISEÑO PROPUESTO

3.1. Componentes pasivos

Los componentes pasivos de la parte electrónica son aquellos que no necesitan una fuente de energía para su correcto funcionamiento. Los componentes pasivos que integran el diseño propuesto se describen a continuación.

3.1.1. Dispositivo receptor de monedas

Un receptor o selector de monedas es un dispositivo que puede seleccionar monedas de cierta denominación y rechazar las que pertenecen a otro tipo. Asimismo, el receptor de monedas tiene la característica de devolver las monedas que no son aceptadas. Entre los distintos transceptores de monedas están:

- **Mecánico:** consiste en un dispositivo enteramente mecánico que selecciona la moneda con base en su peso, anchura y diámetro. Todo lo que este dispositivo mueva o accione, no depende de ningún componente eléctrico.
- **Electrónico:** suele ser más complejo que el mecánico debido a que selecciona la moneda con base en el material del cual está hecho, midiendo su resistencia eléctrica. También algunos miden el peso, diámetro y anchura de la moneda, utilizando sensores ópticos.

- Electromecánicos: mezclan ambas tecnologías en distintas formas para tener una mayor exactitud para distinguir entre las monedas correctas e incorrectas.

3.1.2. Resistor eléctrico

Es un componente electrónico que aprovechando el efecto Joule produce calor a partir de la oposición a una corriente eléctrica. Se le suele llamar con distintos nombres como: resistor o resistencia.

Sus aplicaciones son bastante variadas, pero generalmente se usan para disminuir la corriente que circula a través de un circuito, disipando la corriente excesiva en forma de calor. Cabe destacar que el calor producido por un resistor también puede averiarlo, así que es importante calcular cuánta potencia se desea que soporte.

Los valores más comunes utilizados para circuitos de electrónica digital son de 0.25 W, 05 W y 1 W. También existen resistencias que soportan valores de potencia que superan 1 W hasta los KW (generalmente utilizados para dispositivos de audio o electrónica de potencia).

Los resistores son considerados componentes pasivos debido a que no necesitan una fuente de energía para que funcione adecuadamente. Su aplicación va desde limitar corrientes hasta construcción de calefactores.

Un resistor normalmente posee unas bandas de colores que indican sus valores de resistencia, tolerancia y otras características. A continuación se muestra la tabla correspondiente al código de colores de un resistor.

Tabla I. **Código de colores para resistores eléctricos**

Color	Primera cifra	Segunda cifra	Multiplicador	% de tolerancia
Negro	0	0	1	No existe
Marrón	1	1	10	1
Rojo	2	2	100	2
Naranja	3	3	1 000	No existe
Amarillo	4	4	10 000	4
Verde	5	5	100 000	0,5
Azul	6	6	1 000 000	0,2
Morado	7	7	10 000 000	0,1
Gris	8	8	100 000 000	0,05
Blanco	9	9	1 000 000 000	No existe
Dorado	No existe	No existe	0,1	5
Plateado	No existe	No existe	0,01	10
Ninguno	No existe	No existe	No existe	20

Fuente: elaboración propia.

3.1.3. Relé

El relé o relevador es básicamente un pulsador o interruptor que se activa de forma electrónica al recibir una corriente a través de él. Es una forma fácil y sencilla de aislar 2 circuitos totalmente independientes.

Un relé está formado por una bobina a la cual se le induce una corriente eléctrica activa de forma mecánica para formar un circuito independiente.

3.1.4. Oscilador electrónico de cristal

Es un oscilador electrónico que utiliza la resonancia mecánica de un cristal mineral (generalmente cristal de cuarzo) para crear una señal electrónica a una frecuencia específica, la cual se utiliza para realizar un conteo de tiempo. Es usado normalmente en relojes digitales, emisores y receptores de radio, para obtener estabilidad de la señal de reloj y el conteo de tiempos.

La adición de capacitores a través de un cristal, hará que la resonancia se desplace hacia abajo. La adición de inductancia a través de un cristal hará que la resonancia se desplace hacia arriba.

Los cristales pueden ser fabricados para la oscilación en un amplio rango de frecuencias, desde unos pocos kilohercios hasta varios cientos de megahercios.

Muchas aplicaciones requieren un cristal de frecuencia del oscilador, convenientemente relacionada con alguna otra frecuencia deseada, por lo que cientos de frecuencias de cristal estándar se hacen en grandes cantidades y se almacenan por los distribuidores de electrónica.

3.1.5. Capacitor

Un capacitor o condensador es un componente pasivo que puede almacenar carga en forma de campo eléctrico por medio de un par de placas conductoras, las cuales están separadas entre sí por un dieléctrico.

La carga presente en un capacitor estará dada por la diferencia de voltaje aplicado en sus extremos y por la capacitancia del mismo. La capacitancia será determinada por la distancia entre sus placas y el material del cual está construido.

Cuando a un capacitor se le aplica una corriente continua, se comporta de manera ideal como un circuito abierto, evitando el paso de corriente después de cierto tiempo. Su comportamiento en corriente alterna suele ser totalmente distinto debido a que este presentará una reactancia; asimismo este adelanta la corriente respecto a la corriente aplicada.

3.1.5.1. Capacitor cerámico

Los capacitores de cerámico suelen ser los más simples ya que se tratan solamente de dos placas circulares paralelas entre sí, a las cuales se les adiciona cables para poder conectarlo al circuito. Este tipo de capacitores proveen capacitancias bajas y soportan voltajes al igual muy bajos, esto debido a su simplicidad. Su tolerancia ronda el 5 %.

3.1.5.2. Capacitor electrolítico

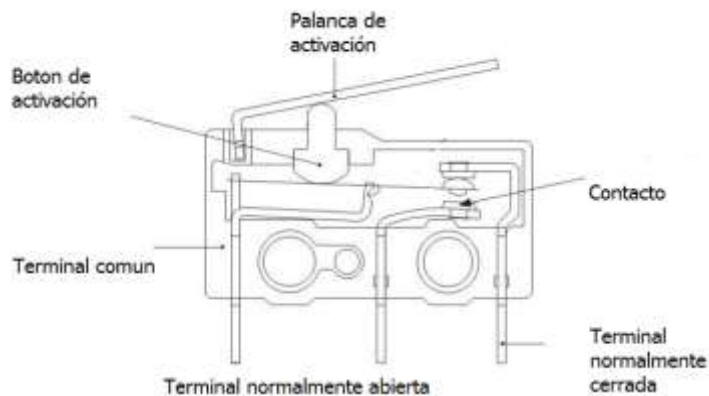
Capacitor que usa un líquido conductor para una de sus placas. Este tipo de capacitores logran tener voltajes y capacitancias mucho más altas que los cerámicos. Cabe destacar que ellos se poseen una polaridad, lo cual es de suma importancia debido a que el terminal negativo es más corto que el positivo y que a voltajes mayores a 1.5 voltios podría fácilmente romperlo e iniciar un corto circuito.

3.1.6. Microswitch

Es un interruptor simple de tamaño pequeño, que posee una especie de palanca, la cual le permite activar el interruptor con una fuerza bastante pequeña. Variando el tamaño de la palanca se podrá graduar la fuerza necesaria para activar el interruptor interno.

A continuación se muestra el diagrama interno típico de un *microswitch*:

Figura 1. Interior de un *microswitch*



Fuente: http://i00.i.aliimg.com/img/pb/131/067/404/404067131_970.jpg.

Consulta: 01 de abril de 2014.

3.1.7. Transformador

Es un dispositivo electrónico pasivo que permite aumentar o disminuir el voltaje, variando la cantidad de corriente y manteniendo la potencia en ambos lados. Está construido por dos bobinas de material conductor, devanadas por un núcleo cerrado de material ferromagnético. La corriente que circula por una de ellas induce en la otra bobina una corriente y voltaje diferente.

3.1.8. Puente de rectificador

Es un componente electrónico utilizado para rectificar voltaje alterno y convertirlo en voltaje directo. Está formado por 4 diodos que aprovecha la parte positiva y negativa de la corriente, haciendo que fluya en un solo sentido.

3.1.9. Conector RCA

Es un tipo de conector electrónico que se utiliza para audio y vídeo, es junto con el conector de audio jack uno de los más utilizados. El conector posee en su centro el polo positivo y alrededor un anillo metálico que es la referencia, separados por un aislante de plástico o caucho.

Tabla II. **Código de colores conectores y cables RCA**

Audio analógico	Izquierda/mono	Blanco
	Derecho	Rojo
	Central	Verde
	Envolvente izquierdo	Azul
	Envolvente derecho	Gris
	Envolvente trasero izquierdo	Marrón
	Envolvente trasero derecho	Marrón claro
	Subwoofer	Violeta
Audio digital	S/PDIF	Naranja
Vídeo analógico	Vídeo compuesto	Amarillo
Vídeo por componentes	Y	Verde
	Pb	Azul
	Pr	Rojo

Continuación de la tabla II.

Vídeo por componentes	R	Rojo
	G	Verde
	B	Azul

Fuente: <http://electronica-teoriaypractica.com/significado-de-los-colores-en-el-conector-rca/>.

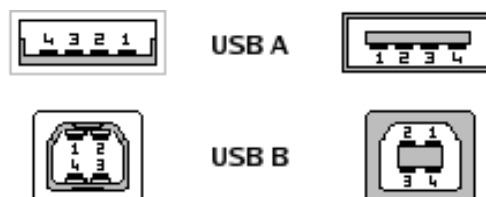
Consulta: 01 de abril de 2014.

3.2. Conector USB estándar

Por sus siglas en inglés Universal Serial Bus; el USB es una interfaz de entrada y salida serial. Existen dos tipos de conectores USB estándar:

- Tipo A, que tienen forma rectangular y se utilizan generalmente para dispositivos que consumen poco ancho de banda.
- Tipo B, que tienen forma cuadrada y se utilizan generalmente para dispositivos con altos requisitos de ancho de banda.

Figura 2. Conectores USB estándar A y B



Fuente: <http://www.ucontrol.com.ar/forosmf/explicaciones-y-consultas-tecnicas/pinout-conector-usb-tipo-b-para-circuito-impreso/>. Consulta: 06 de mayo 2014

Tabla III. **Descripciones de pines conectores y cables USB**

Pin	Señal	Descripción	Color
1	Vcc	5 voltios	Rojo
2	D-	Data-	Blanco
3	D+	Data+	Verde
4	GND	Tierra	Negro

Fuente: elaboración propia.

3.3. Componentes activos

Los componentes activos son aquellos que son capaces de controlar los circuitos o de realizar ganancias. Fundamentalmente son los generadores eléctricos y ciertos componentes semiconductores. Estos últimos, en general, tienen un comportamiento no lineal, esto es, la relación entre la tensión aplicada y la corriente demandada no es lineal.

Los componentes activos utilizados para el diseño del prototipo del dispositivo controlador y temporizador se muestran a continuación:

3.3.1. Transistor de unión bipolar (BJT)

Es un transistor que está formado por dos uniones PN, similar a la conexión de dos diodos, es el más común de los transistores y pueden estar formados de silicio o germanio dopado. Existen dos tipos de transistores BJT: el NPN y PNP y el flujo de corriente del mismo está indicado por el gráfico correspondiente a cada transistor; se usa en electrónica para amplificar corrientes, voltajes o para la construcción de compuertas analógicas.

El transistor de unión bipolar posee tres regiones físicas denominadas: emisor, base y colector. El emisor, que se diferencia de las otras dos por estar fuertemente dopada, comportándose como un metal y su nombre se debe a que esta terminal funciona como emisor de portadores de carga. La base separa al emisor del colector y es muy estrecha comparada al colector. El colector es la parte más robusta del transistor para soportar una mayor corriente. Un transistor BJT tiene tres regiones de operación que se explicarán continuación:

- Región de corte: donde ambas uniones están polarizadas de forma inversa. La corriente (denominada corriente de fuga) de base es muy pequeña, esta región puede ser utilizada como un interruptor abierto.
- Región lineal o activa: el transistor actúa como un amplificador lineal. La unión base-emisor está conectada en directo y la unión colector-base está en reversa. Típicamente se utiliza para amplificar señales analógicas o continuas.
- Región de saturación: ambas uniones están conectadas en directo. Cuando un transistor trabaja en esta región, este funciona como interruptor cerrado.

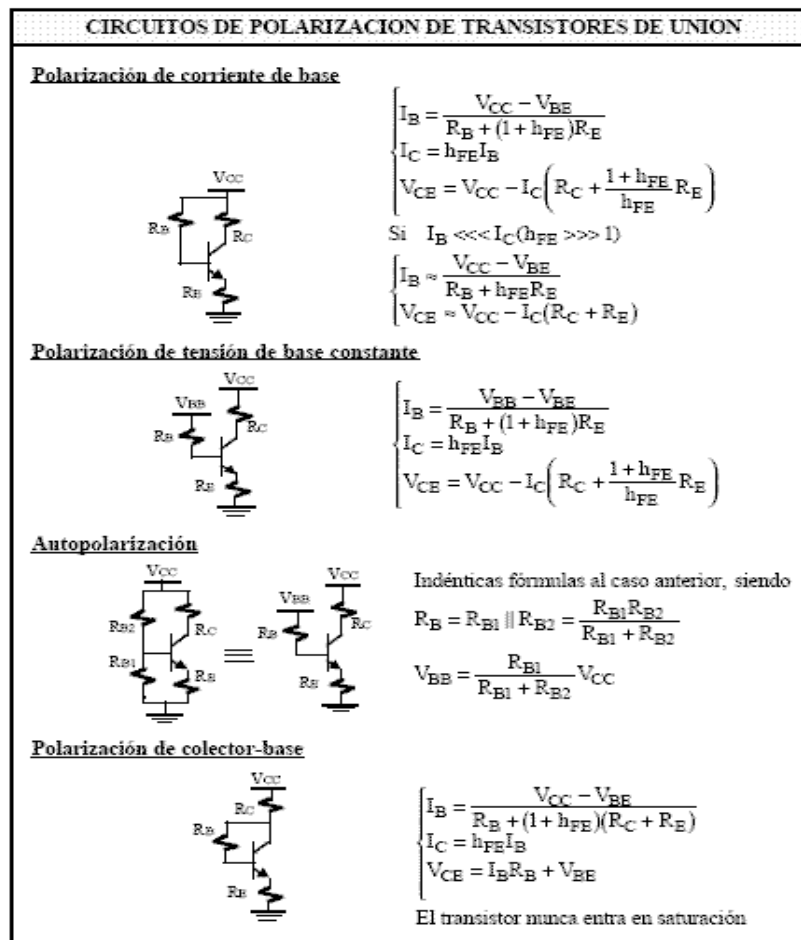
También debe tomarse en cuenta que existen dos regiones que deben ser evitadas a toda costa, para que el transistor sea útil y sufra daños en su funcionamiento:

- Región de ruptura: que determina el límite físico de operación del transistor. En esta región el transistor se avería, debido a la corriente de excesiva a la que se expone.

- Región Inversa: no tiene implicaciones útiles para amplificación o conmutación. La unión base-emisor está conectada en inverso y la unión colector-base está en directa.

A continuación se muestran esquemas de polarización y algunas fórmulas para poder implementar estos circuitos.

Figura 3. Polarización de un transistor BJT



Fuente: http://imagenes.unicrom.com.s3.amazonaws.com/circuitos_polarizacion_transistores.gif.

Consulta: 02 de abril de 2014.

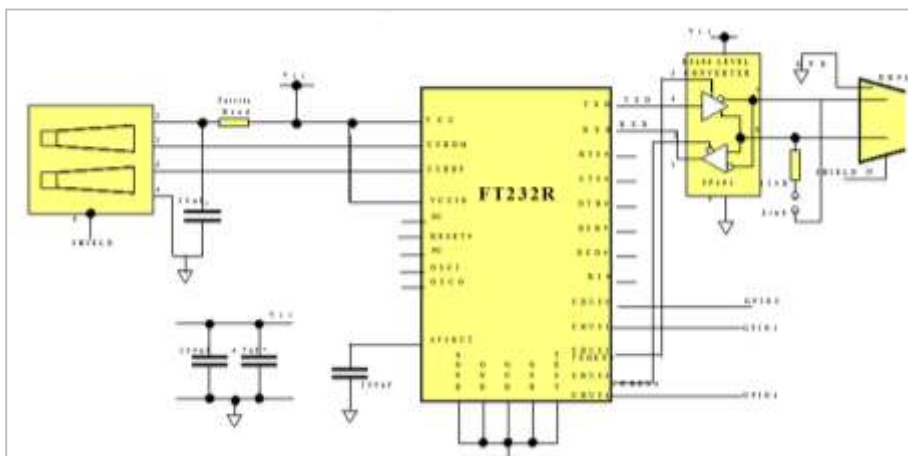
3.3.2. Circuito integrado FT232RL

Es uno de los circuitos integrados de uso más común, utilizados para convertir USB señales UART señales. Este proceso es muy útil ya que permite comunicar y subir código a un Arduino u otro microcontrolador, sin necesidad de un programador externo.

Tiene como función de interfaz entre USB a UART de forma bidireccional, además de comportarse digitalmente como un simulador serial. Es un circuito integrado bastante complejo que además de tener las funciones de simulador serial, también convierte los niveles de voltaje equivalentes entre integrados TTL y el puerto USB del computador.

A continuación se presenta el diagrama de conexión que normalmente posee este dispositivo para la conversión USB-Serial.

Figura 4. Diagrama de conexión circuito integrado FT232



Fuente: <http://html.alldatasheet.com/html-pdf/144591/FTDI/FT232RL/1830/18/FT232RL.html>.

Consulta: 02 de abril de 2014.

3.3.3. El microcontrolador

Es un circuito integrado al cual se le puede cambiar su programación para poder realizar tareas varias. Contenido dentro del mismo se encuentra una gran cantidad de otros circuitos integrados embebidos, los cuales lo convierten en una microcomputadora con periféricos capaces de realizar distintas funciones.

Los microcontroladores varían sus capacidades dependiendo de las necesidades o características que el usuario necesite para poder aplicarlo. Actualmente los microcontroladores pueden encontrarse en gran cantidad de aparatos eléctricos, debido a que están diseñados para minimizar costos y su fácil utilización y programación. Un microcontrolador difiere en gran medida de un microprocesador debido a que en un mismo circuito integrado posee la mayoría de partes que un microprocesador necesita ser utilizado.

3.3.3.1. Partes principales de un microcontrolador

A continuación se explican los componentes más comunes a todos los modelos actuales de microcontroladores:

3.3.3.1.1. CPU

La unidad central de proceso (CPU por sus siglas en inglés: *central processing unit*) es el componente central del microcontrolador y cualquier computador, en este es donde se produce la mayoría de cálculos matemáticos y lógicos. En términos de potencia es el elemento más importante de un sistema informático, ya que en él se realizan todas las operaciones.

Desde el punto de vista lógico y funcional, un microprocesador está compuesto por varios registros, una unidad de control y una unidad aritmética y lógica; el microprocesador ejecuta instrucciones almacenadas de forma binaria dentro de la memoria RAM. El tiempo que tardan ejecutarse estas instrucciones vendrá dado principalmente por la frecuencia de reloj y la complejidad de la misma.

El CPU del microcontrolador consta de partes más pequeñas especializadas para funciones distintas:

- Decodificador de instrucciones: es la parte que traduce las instrucciones del programa y las hace entendibles al CPU, para poder ejecutar otras partes del circuito. Gracias a este las instrucciones pueden ser escritas en un lenguaje comprensible para el ser humano y ser ejecutadas por el microcontrolador.
- Unidad lógica aritmética (ALU de sus siglas en inglés *arithmetic logical unit*): dentro de este componente se realizan todas las operaciones matemáticas y lógicas.
- Acumulador: es el registro de trabajo y está íntimamente ligado a la unidad lógica aritmética. Se utiliza para almacenar datos sobre los que se va realizar alguna operación, para un procesamiento futuro.
- Bus de datos y de direcciones: el bus de datos se utiliza para conectar todos los circuitos dentro del microcontrolador y enviar los datos de un circuito a otro. El bus de direcciones se utiliza para transmitir la dirección del CPU a la memoria. Ambos buses comunican el microcontrolador con el exterior.

3.3.3.1.2. Memoria

Se entiende como memoria a todo tipo de forma de almacenamiento que un dispositivo electrónico utilice para almacenar información de forma binaria. Típicamente un microcontrolador contará con los siguientes tipos de memoria:

- Memoria de solo lectura (ROM por sus siglas en inglés: *read only memory*): es una memoria no volátil de solo lectura, cuyo contenido se graba durante la fabricación del chip. Si el microprocesador deja de ser alimentado, todo lo contenido dentro de la memoria permanecerá sin borrarse, pero no puede cambiarse durante la ejecución del programa. Algunas veces es utilizado para almacenar datos de tipo constante y números de serie e instrucciones de arranque.
- OTP (*one time programmable*): el microcontrolador contiene una memoria no volátil de solo lectura, programable una sola vez por el usuario. Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde un PC. La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.
- Memoria de acceso aleatorio (RAM de sus siglas en inglés *random access memory*): este tipo de memoria pierde su contenido una vez se retira la fuente de alimentación del microcontrolador. El contenido de la misma puede ejecutarse durante la ejecución del programa de la RAM y se utiliza para almacenar datos de resultados inmediatos creados por el microcontrolador.

- Memoria EPROM (de sus siglas en inglés *erasable programmable read only memory*): pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde un PC. Si posteriormente se desea borrar el contenido, disponen de una ventana de cristal en su superficie, por la que se somete a la EPROM a rayos ultravioleta durante varios minutos.
- Memoria EEPROM (*electrically erasable programmable read only memory*): el contenido de esta memoria puede modificarse durante la ejecución de un programa, al igual que la RAM; pero si el microcontrolador pierde su fuente de energía, el contenido dentro de esta memoria permanecerá intacto. Se usa principalmente para almacenar datos y resultados de operaciones variados.
- Memoria FLASH: se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM, pero consume menos y es más pequeña. A diferencia de la ROM, la memoria FLASH es programable en el circuito, es más rápida y de mayor densidad que la EEPROM. La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado. Son idóneas para la enseñanza y la ingeniería de diseño.

Asimismo, existen dos tipos de arquitecturas típicos para almacenar y utilizar los datos de la memoria, los más conocidos son: la arquitectura Harvard y la arquitectura Von Neumann.

En la arquitectura Harvard, los datos y el programa están almacenados en dos memorias físicamente separadas y cada una con un bus de memoria propio. En el caso de la arquitectura Von Neumann, la memoria de programa y la de datos se encuentran físicamente colocadas en el mismo dispositivo, teniendo así un mismo bus de memoria.

3.3.3.1.3. Unidades de entrada y salida

Son los sistemas y vías que utiliza el microcontrolador para comunicarse con el exterior, asimismo estos permiten introducir información en el microcontrolador. Todos los microprocesadores destinan algunos pines para soportar líneas de entrada y salida de tipo digital.

3.3.3.1.4. Reloj principal

Todo microcontrolador y microprocesador debe poseer un circuito oscilador que genere una onda de alta frecuencia usada para sincronizar todas las operaciones, a esta señal se le denomina reloj. Entre mayor sea la velocidad del reloj mayor será la velocidad que tarde el microcontrolador en hacer una tarea.

Generalmente la mayoría de microcontroladores disponen de los siguientes tipos de reloj para distintos propósitos:

- RC: oscilador de resistencia y condensador
- XT: oscilador con cristal
- HS: oscilador de cristal de alta velocidad
- LP: cristal de baja frecuencia y bajo consumo de energía
- Os: oscilador interno (que no necesita componentes externos)

3.3.3.2. Recursos especiales del microcontrolador

Cada fabricante de microprocesadores posee arquitecturas distintas y ofrece distintos recursos, dependiendo los propósitos y el modelo de microprocesador. Los recursos especiales más en un microcontrolador se describen a continuación.

3.3.3.2.1. Temporizadores o *timers*

Son utilizados para controlar periodos de tiempo y contar acontecimientos que suceden cada cierto tiempo en el exterior.

Para el conteo de tiempos se carga un valor desde algún registro y se incrementa o reduce su valor a un pulso constante, hasta esperar el desborde o llegada a cero, produciendo un aviso.

3.3.3.2.2. Perro guardián o *watchdog*

Consiste en un temporizador que cuando se desborda o llega a cero provoca un reinicio automático del microcontrolador. El *watchdog* permite que el programa no se quede encerrado en un bucle infinito; si el programa funciona correctamente, el *watchdog* se refrescará cada cierto tiempo; de lo contrario provocará un reinicio de todo el sistema.

3.3.3.2.3. Brownoutreset

El *Brownoutreset* o protección ante fallo de alimentación se encarga de reiniciar el microcontrolador.

Todo microcontrolador posee cierto rango permisible de voltaje en su alimentación, si el voltaje es demasiado bajo al nominal, automáticamente reiniciará el funcionamiento del microcontrolador, evitando así errores en su funcionamiento.

3.3.3.2.4. Estado de bajo consumo

Uno de los recursos especiales con los que cuentan los microprocesadores y que es de vital importancia para dispositivos portátiles es el estado de bajo consumo. Este consiste en una especie de reposo o hibernación en la cual el microprocesador consumirá solo la energía necesaria para poder estar encendido y al activarse, debido a una interrupción, volverá a su funcionamiento normal.

3.3.3.2.5. Conversores A/D y D/A

Los conversores A/D consisten en pines de salida especiales que transforman valores analógicos a señales digitales dentro del interior del microcontrolador. Todo microcontrolador presentará cierto ruido de cuantización al transformar los valores analógicos a digitales. En el caso de los conversores D/A realizan el proceso contrario a los conversores A/D; convierten valores digitales a analógicos en las salidas del microcontrolador.

3.3.3.2.6. PWM

Es un circuito especializado que varía la anchura de los pulsos a la salida de un pin del microcontrolador. Generalmente se utiliza para variar la intensidad de la luz, la velocidad de algún motor o simular algún voltaje analógico menor al máximo que puede proveer un pin de salida.

3.3.3.2.7. Puertos de comunicación

Los puertos de comunicación del microcontrolador permiten que este se comunique con otros dispositivos electrónicos. Asimismo, los microcontroladores cuentan con distintas formas de comunicación hacia el exterior, algunos de los más importantes son:

- UART, adaptador de comunicación, serie asíncrona.
- USART, adaptador de comunicación, serie síncrona y asíncrona.
- Puerto paralelo esclavo para conectarse con los buses de otros microprocesadores.
- USB (*universal serial bus*), el conocido bus serie para los PC.
- Bus I2C, que es un interfaz serie de dos hilos desarrollado por Philips.
- Interface SPI, un puerto serie síncrono.
- CAN (*controller area network*), para permitir la adaptación con redes de conexión multiplexado, desarrollado conjuntamente por Bosch e Intel para el cableado de dispositivos en automóviles. En EE.UU. se usa el J1850.
- TCP/IP, ya existen microcontroladores con un adaptador de comunicación para este protocolo.

3.3.3.3. Herramientas de desarrollo

Para programar las distintas tareas de un microcontrolador, se cuenta con herramientas que permiten modificar sus acciones a voluntad. Las principales herramientas de desarrollo son:

- **Ensamblador:** la programación en lenguaje ensamblador puede resultar un tanto ardua para el principiante, pero permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema.
- **Compilador:** la programación en un lenguaje de alto nivel (como C o Basic) permite disminuir el tiempo de desarrollo de un producto y si además está familiarizado con C o Basic, es una buena opción. No obstante, cuando el compilador convierta el código del programa a un lenguaje ensamblado, cada línea de código del programa en lenguaje de alto nivel habrá generado bastantes más líneas de código en lenguaje ensamblador, normalmente en una relación de uno a tres. Esto significa que para utilizar un lenguaje de alto nivel se necesita un microcontrolador con una capacidad de memoria relativamente grande.
- **Simulador:** se trata de software capaz de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso.

- Placas de evaluación: se trata de pequeños sistemas con un microcontrolador ya montado y que suelen conectarse a un PC desde el que se cargan los programas que se ejecutan en el microcontrolador. Las placas suelen incluir visualizadores LCD, teclados, LEDs, fácil acceso a los pines de E/S, etc. Pueden incluir un programa de control o sistema operativo que recibe el nombre de programa monitor.
- Programador: es un dispositivo que conectado a un PC permite grabar en el microcontrolador el programa desarrollado. Algunos puede fabricarlos uno mismo y resultan muy económicos. De esta manera se puede programar al microcontrolador una vez esté montado en la placa del circuito utilizando una conexión de dos, tres o cuatro terminales.

3.3.3.4. AVR

Son una familia de microcontroladores de arquitectura Harvard desarrollados por la compañía Atmel. Se pueden dividir en los siguientes grupos:

- ATxmega: procesadores muy potentes con 16 a 384 KB de memoria *flash* programable, encapsulados de 44, 64 y 100 pines, capacidad de DMA, eventos, criptografía y amplio conjunto de periféricos con DACs.
- ATmega: microcontroladores AVR grandes, con 4 a 256 KB de memoria *flash* programable, encapsulados, de 28 a 100 pines, conjunto de instrucciones extendido y amplio conjunto de periféricos.

- ATtiny: pequeños microcontroladores AVR con 0.5 a 8 KB de memoria *flash* programable, encapsulados, de 6 a 20 pines, y un limitado set de periféricos.,
- AT90USB: ATmega integrado con controlador USB.
- AT90CAN: ATmega con controlador de bus CAN.
- Tipos especiales: algunos modelos especiales, por ejemplo, para el control de los cargadores de baterías, pantallas LCD y los controles de los motores o la iluminación.

El AVR está diseñado para funcionar con código C. Como este lenguaje utiliza profusamente punteros para el manejo de variables en memoria, los tres últimos pares de registros internos del procesador son usados como punteros de 16 bits al espacio de memoria externa, bajo los nombres X, Y, Z.

Los microcontroladores AVR tienen una cañería con dos etapas (cargar y ejecutar), que les permite ejecutar la mayoría de las instrucciones en un ciclo de reloj, lo que los hace relativamente rápidos entre los microcontroladores de 8 bits.

El set de instrucciones de los AVR es más regular que la de la mayoría de microcontroladores de 8 bits (por ejemplo, los PIC). Como los PIC, tiene una comunidad de seguidores (ejemplificadas por el foro de internet *AVR freaks*), principalmente debido a la existencia de herramientas de desarrollo gratuitas o de bajo coste. Estos microcontroladores están soportados por tarjetas de desarrollo de costo razonable, capaces de descargar el código al microcontrolador, y por una versión de las herramientas GNU.

3.3.3.4.1. ATmega328

El ATmega 328 es un chip microcontrolador RISC de alto rendimiento creado por Atmel y pertenece a la serie megaAVR. Este microcontrolador integra:

- 32 KB ISP flash.
- Una memoria con la capacidad de leer mientras escribe.
- 1 KB de memoria EEPROM.
- 2 KB de SRAM.
- 23 líneas de E/S de propósito general.
- 32 registros de proceso general.
- Tres temporizadores flexibles/contadores con modo de comparación.
- Interrupciones internas y externas.
- Programador de modo USART.
- Interface serial orientada a byte de 2 cables.
- SPI puerto serial, 6 canales de 10 bits, conversor A/D (8 canales en TQFP y QFN/MLF).
- *Watchdog timer* programable con oscilador interno.
- Cinco modos de ahorro de energía seleccionables por software.

El dispositivo opera entre 1.8 y 5.5 voltios. Por medio de la ejecución de poderosas instrucciones en un solo ciclo de reloj, el dispositivo alcanza una respuesta de un millón de instrucciones por segundo, balanceando consumo de energía y velocidad de proceso.

Tabla IV. **Resumen de características de ATmega 328**

Parámetros	Valores
Flash	32 Kilobytes
RAM	2 Kilobytes
Cantidad de pines	28
Frecuencia máxima de operación	20 MHz
CPU	8 bits AVR
Número de canales variables	16
Pines máximos de E/S	26
Interrupciones externas	24

Fuente: <http://www.arduino.cc/es/>, consulta: 20 de junio de 2014.

3.3.3.4.2. Arduino Duemilanove

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el ATmega 168, ATmega328, ATmega1280 y ATmega8, por su sencillez y bajo costo permiten el desarrollo de múltiples diseños. Por otro lado, el software consiste en un entorno de desarrollo que implementa el lenguaje de programación *Processing/wiring* y el cargador de arranque que es ejecutado en la placa.

El Arduino Duemilanove ("2009") es una placa con microcontrolador basada en el ATmega168 o el ATmega328. Tiene 14 pines con entradas/salidas digitales (6 de las cuales pueden ser usadas como salidas PWM), 6 entradas analógicas, un cristal oscilador a 16 Mhz, conexión USB, entrada de alimentación, una cabecera ISCP y un botón de reinicio.

El Arduino Duemilanove puede ser alimentado vía conexión USB o con una fuente de alimentación externa. El origen de la alimentación se selecciona automáticamente.

Las fuentes de alimentación externas (no USB) pueden ser tanto un transformador o una batería. El transformador se puede conectar usando un conector macho de 2.1 mm con centro positivo en el conector hembra de la placa. Los cables de la batería pueden conectarse a los pines Gnd y Vin en los conectores de alimentación (*power*).

La placa puede trabajar con una alimentación externa de entre 6 a 20 voltios. Si el voltaje suministrado es inferior a 7 voltios el pin de 5 voltios puede proporcionar menos de 5 voltios y la placa puede volverse inestable; si se usan más de 12 voltios, los reguladores de voltaje se pueden sobrecalentar y dañar la placa. El rango recomendado es de 7 a 12 voltios. Los pines de alimentación son los siguientes:

- 5V: la fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5 voltios.

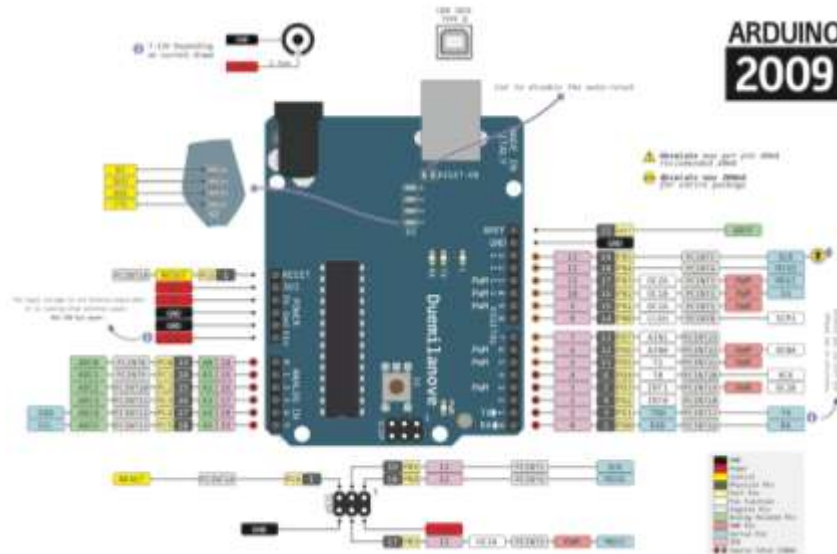
- 3V3: una fuente de voltaje a 3.3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada 50 mA.
- GND: se refiere los pines de toma de tierra.

Tabla V. **Resumen características Arduino Duemilanove**

Microcontrolador	ATmega 368 (ATmega 168 en versiones anteriores)
Voltaje de funcionamiento	5 voltios
Voltaje de entrada (recomendado)	7-12 voltios
Voltaje de entrada (limite)	6-20 voltios
Pines E/S digitales	14 (6 proporcionan salida PWM)
Pines de entrada analógica	6
Intensidad por pin	40 mA
Intensidad en pin 3,3V	50 mA
Memoria <i>flash</i>	16 KB (ATmega168) o 32 KB (ATmega328) de las cuales 2 KB las usa el gestor de arranque(<i>bootloader</i>)
SRAM	1 KB (ATmega168) o 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328)
Velocidad de reloj	16 MHz

Fuente: <http://www.arduino.cc/es/>. Consulta: 20 de junio de 2014.

Figura 5. **Patillaje del Arduino Duemilanove**



Fuente: <http://www.arduino.cc/es/>. Consulta: 20 de junio de 2014.

3.3.4. **Módulo RF**

Un módulo RF o de radio frecuencia es un dispositivo electrónico utilizado para recibir y/o transmitir señales de radio entre dos dispositivos. Es un sistema embebido que permite la fácil comunicación de dos dispositivos electrónicos, ya sea por forma serial o directamente, proporcionando pulsos digitales.

Los módulos de RF son los más utilizados en los productos de media y baja de volumen para aplicaciones de consumo, tales como puertas de garaje, sistemas de alarma inalámbricos, controles remotos industriales, etc. Típicamente maneja frecuencias de 433,92 MHz, 315 MHz, 868 MHz, 915 MHz y 2 400 MHz. Estas frecuencias se utilizan debido a las regulaciones nacionales e internacionales que controlan la segunda mano de la radio para la comunicación.

También se puede aplicar a los módulos a través de una gran variación de funcionalidad y capacidad. Los módulos de RF típicamente incorporan una placa de circuito impreso , transmisión o recepción de circuito, la antena y la interfaz de serie, para la comunicación con el procesador principal. Los tipos de módulos RF que se pueden encontrar son:

- El módulo transmisor: solamente transmite señales
- El módulo receptor: solamente recibe señales
- Módulo de transceptor: recibe y transmite señales de radio frecuencia

3.3.5. Módulo LCD 20x4

Es un módulo de visualización electrónica que cuenta con una amplia gama de aplicaciones, es básico y muy utilizado en distintos dispositivos y circuitos. Este módulo integra pines de entrada y salida para su control y funcionamiento.

Este tipo de LCD es capaz de mostrar 20 caracteres por línea, contando con 4 lineales para tal función; asimismo, cada carácter está formado de 5x7 pixeles. El LCD posee dos registros: comando y datos.

El registro de comandos almacena las instrucciones de comando dadas a la pantalla LCD. Un comando es cualquier instrucción que se le dé a la LCD, entre estas están: posición del cursor, control de pantalla, borrar todo en pantalla, etc. Los datos son dados por valores ASCII, correspondientes al carácter que se desea mostrar en dicha pantalla.

El patillaje o *pinout* de los módulos LCD variará según el fabricante. Para el diseño del prototipo se propone utilizar una LCD 20x4 de 8 bits de datos y 2 pines para RS y *enable*.

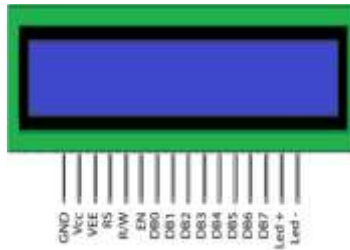
A continuación se muestra la función y el nombre de cada uno de los pines del módulo LCD:

Tabla VI. **Función y nombre de pines para módulo LCD 20X4**

Número de pin	Función	Nombre del pin
1	Tierra (0 V)	GND
2	La tensión de alimentación; 5 voltios (de 4.7 a 5.3 voltios)	VDD
3	Ajuste de contraste; a través de una resistencia variable	VEE
4	Selecciona comando de registro.	RS
5	Selecciona el modo de escritura o lectura.	RW
6	Envía datos a los pines de datos cuando una señal de activación.	E
7	Pines de datos de 8 bits	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Conexión de alimentación para la luz de fondo de alimentación (5 voltios)	Led +
16	Conexión de tierra para la luz de fondo de alimentación (0 voltios)	Led-

Fuente: elaboración propia.

Figura 6. **Patillaje de módulo LCD 20x4**



Fuente: elaboración propia.

3.3.6. Servomotor

Un servomotor o simplemente servo es un dispositivo controlado con un eje fijo, el cual posee un circuito interno que permite llevarlo a posiciones angulares específicas, dependiendo de la señal de entrada. El servo convierte por medio de un engranaje la fuerza de un motor DC, aumentando su torque en gran medida y dejando el eje en una posición específica. También provee potencia proporcional para cargas mecánicas. Un servo, por consiguiente, no consume mucha energía.

El motor del servo tiene circuitos de control (amplificador de error o comparador) y un potenciómetro conectado al eje central del servomotor. Este potenciómetro permite a la circuitería de control supervisar el ángulo actual del servomotor. Si el eje está en el ángulo correcto, entonces el motor está apagado.

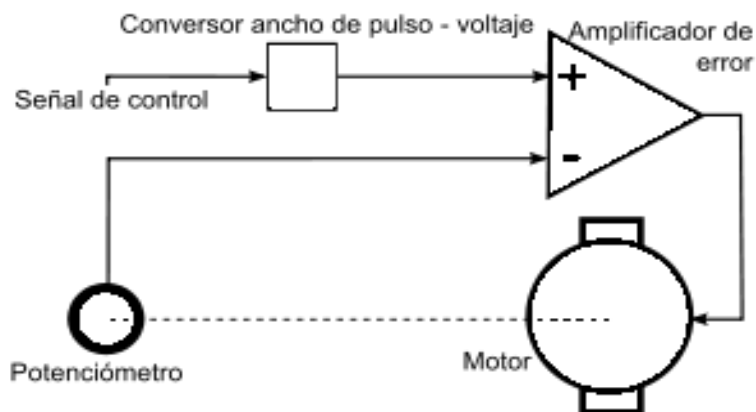
Si el circuito chequea que el ángulo no es el correcto, el motor girará en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de girar alrededor de los 180 grados. Normalmente, en algunos llega a los 210 grados, pero varía según el fabricante.

Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los cincuenta hercios, así las señales PWM tendrán un periodo de 20 milisegundos.

La cantidad de voltaje aplicado al motor es proporcional a la distancia que este necesita viajar. Así, si el eje necesita regresar una distancia grande, el motor regresará también a toda velocidad. Si este necesita regresar solo una pequeña cantidad, el motor correrá a una velocidad más lenta. A esto se le llama control proporcional.

En la siguiente figura se podrá notar el diagrama típico de un servomotor:

Figura 7. **Diagrama interno de un servomotor**



Fuente: <http://upload.wikimedia.org/wikipedia/commons/thumb/8/8e/ControlServo.svg/300px-ControlServo.svg.png>. Consulta: 22 de junio de 2014.

4. DESARROLLO FÍSICO DEL PROTOTIPO

4.1. Receptor de monedas

El receptor de monedas está compuesto por la parte mecánica y electrónica. La parte mecánica determina si el tamaño y peso de la moneda son correctos, mientras que la parte electrónica envía pulsos eléctricos que son activados cuando la moneda es la correcta.

4.1.1. Componentes mecánicos del receptor de monedas

Para realizar la recepción de las monedas de Q 1.00 se procedió a diseñar un dispositivo sencillo, el cual permite recibir las monedas que tengan un peso, ancho y diámetro equivalente al de una moneda de un quetzal. Sabiendo de antemano que una moneda de un quetzal tiene un diámetro de 2,9 cm y un peso aproximado de once gramos, dependiendo de la aleación, se procedió a diseñar un dispositivo que rechace alguna moneda distinta. Las medidas del dispositivo seleccionador de monedas se muestran a continuación:

En la vista lateral 1 se puede notar que el *microswitch* es colocado en una pestaña diseñada para soportarlo y que el largo de su palanca activadora dependerá del peso que se desee que se active. Para el caso de las monedas de un quetzal se pretende que el dispositivo se active con un peso aproximado de 10 gramos. También se puede notar una abertura trapezoidal que tiene medidas de aproximadas entre 2.8 cm a 5 cm; esto con el propósito de rechazar monedas menores al diámetro deseado.

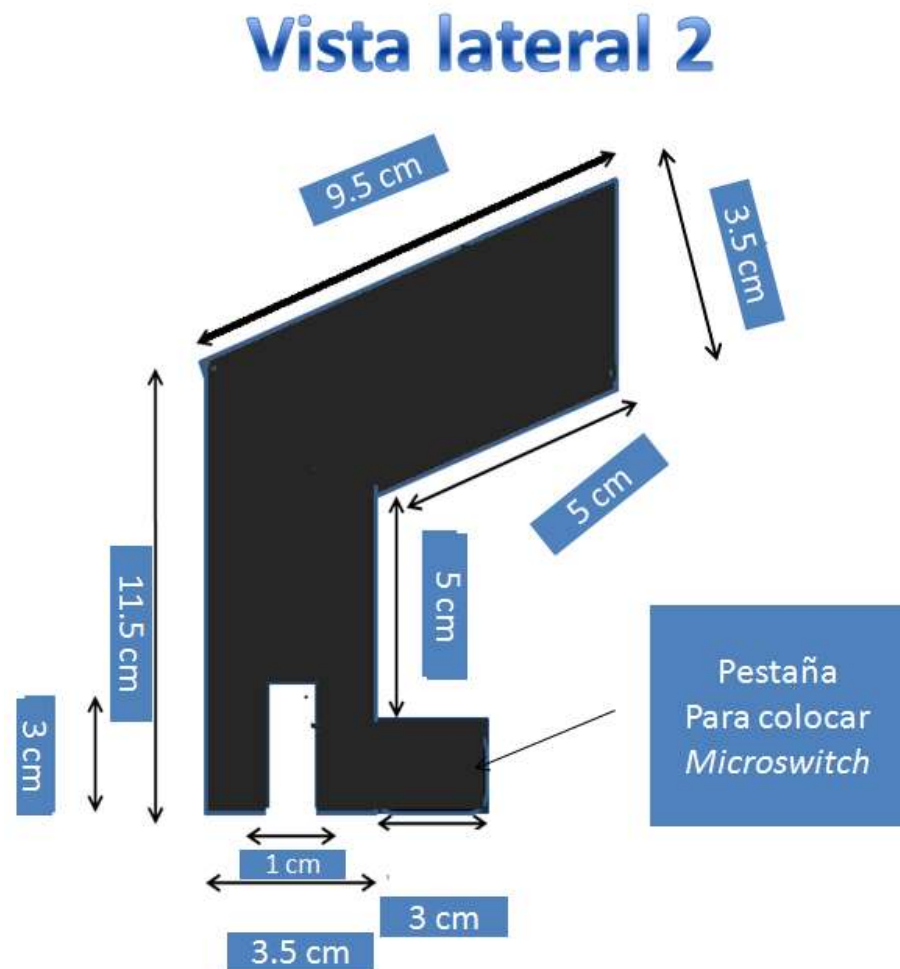
A continuación se muestra una de las vistas laterales del dispositivo receptor de monedas:

Vista lateral 1



En la vista lateral 2 se puede notar que no se posee la abertura de rechazo de monedas, solamente se puede notar la abertura para colocar la palanca del *microswitch* y la pestaña de apoyo para el cuerpo del mismo.

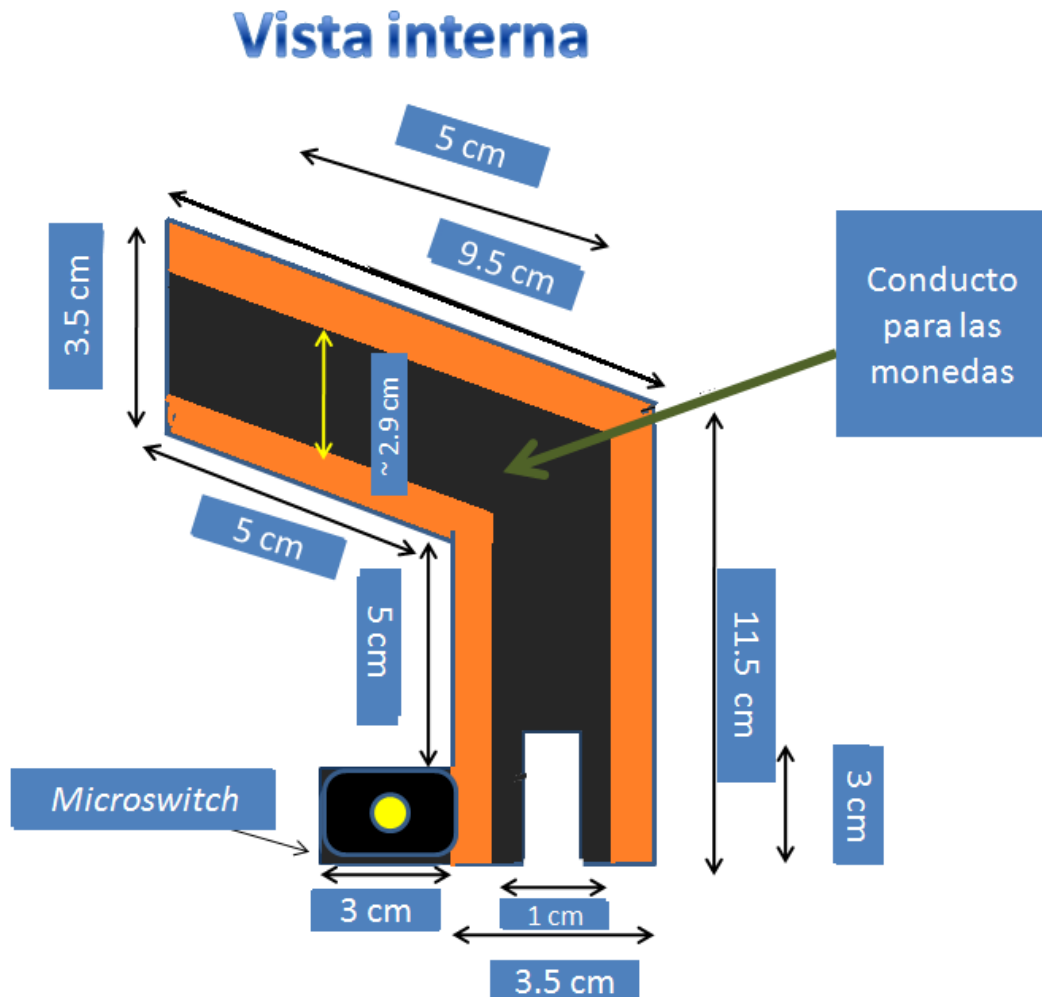
Figura 9. Vista lateral 2 receptor de monedas mecánico



Fuente: elaboración propia, con programa Microsoft Power Point 2007.

Entre la tapa lateral 1 y 2 irá un conducto que corresponde al diámetro y ancho de la moneda de Q1.00; este conducto tiene un diámetro aproximado de 2,9 cm.

Figura 10. Vista interna de receptor de monedas mecánico



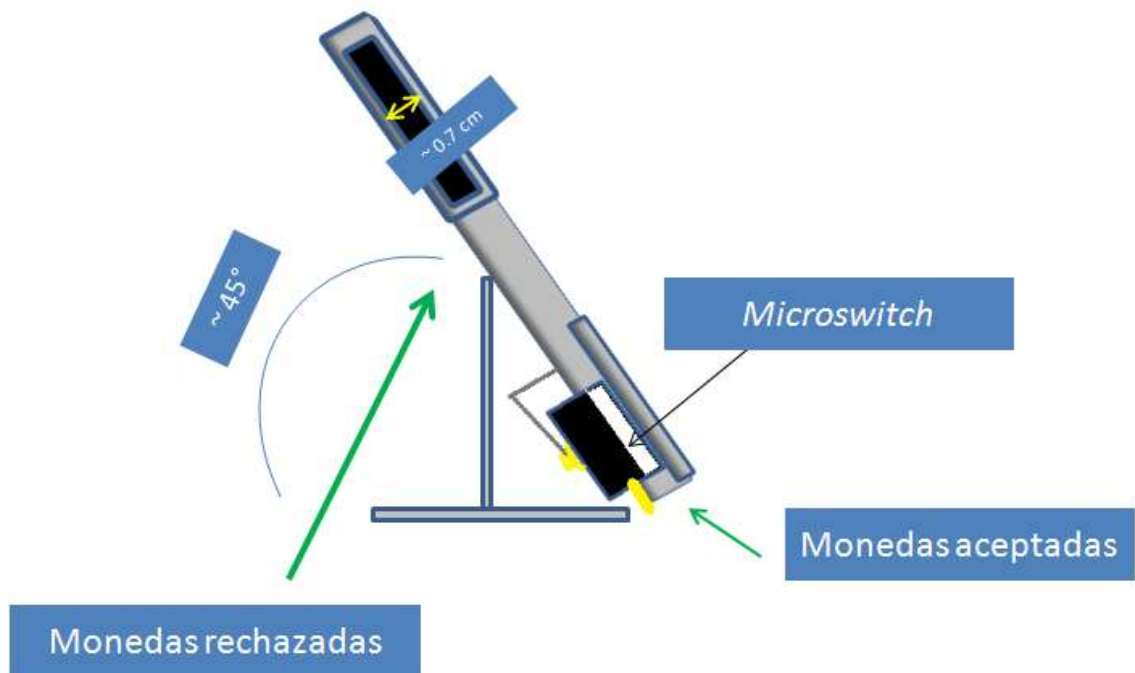
Fuente elaboración propia, con programa Microsoft Power Point 2007.

El dispositivo seleccionador de monedas debe colocarse a una inclinación aproximada de 45 grados, para permitir que las monedas con un diámetro menor a 2,9 cm caigan por el extremo de rechazo.

La parte interior del receptor debe tener la menor fricción posible para evitar que las monedas queden atoradas.

Figura 11. Vista frontal de receptor de monedas mecánico

Vista frontal sin tapa



Fuente elaboración propia, con programa Microsoft Power Point 2007.

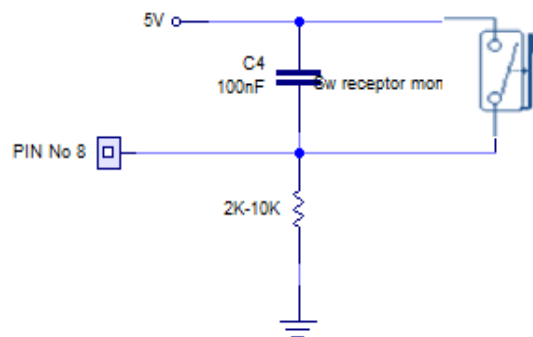
4.1.2. Circuito electrónico

El *microswitch* estará conectado en paralelo a un capacitor cerámico de 100 nanofaradios para reducir el rebote mecánico y proveer un pulso más uniforme, asimismo se coloca una resistencia de *pull-down* para evitar el ruido electromagnético.

Se colocará el circuito al pin número 8 del Arduino, que recibirá una señal de 5 voltios al momento de activarse el *microswitch*.

A continuación se muestra el diagrama de conexión entre los pines del Arduino y el *microswitch*.

Figura 12. **Conexión de *microswitch***

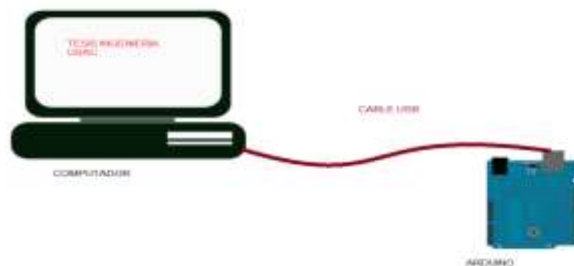


Fuente: elaboración propia, con programa Live Wire 1.30.

4.2. Interfaz de conexión con un computador

Para conectar el dispositivo controlador al computador se procederá a utilizar la conexión USB del mismo, a través de un cable A/B. El computador detectará al dispositivo Arduino como un puerto serial, debido al simulador y conversor serial-USB.

Figura 13. **Conexión USB entre computador y Arduino**



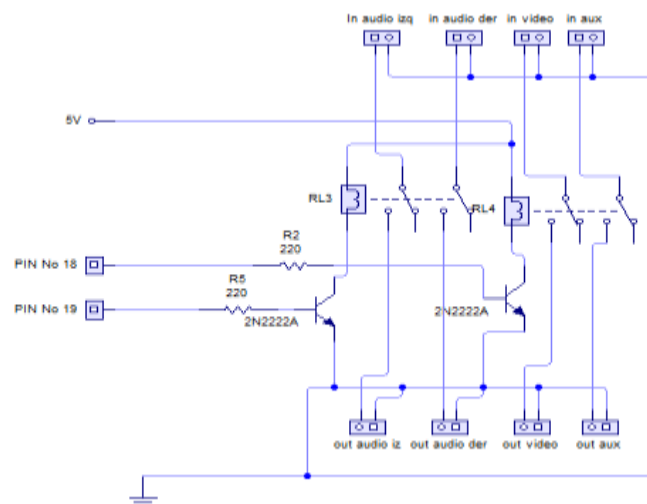
Fuente: elaboración propia.

4.3. Circuito de conexión con una videoconsola

En el caso de controlar una videoconsola se posee dos relés dobles que son activados y desactivados por los pines 18 y 19 del Arduino. A la salida de estos pines se agregan unos transistores NPN 2N2222 y resistencias de 220 ohmios. El propósito de los transistores y las resistencias es disminuir la corriente exigida al microcontrolador y amplificar la corriente a los niveles requeridos por cada relé.

Para cada entrada/salida de audio y video se colocará un conector RCA y por último se pretende dejar una conexión auxiliar en el caso que el usuario necesite activar un dispositivo diferente. Cabe destacar que cada conexión soporta un máximo de 1 amperio, se debe tener un especial cuidado cuando se utilice la conexión auxiliar para no exceder los niveles especificados.

Figura 14. Diagrama de circuito de interfaz con videoconsola



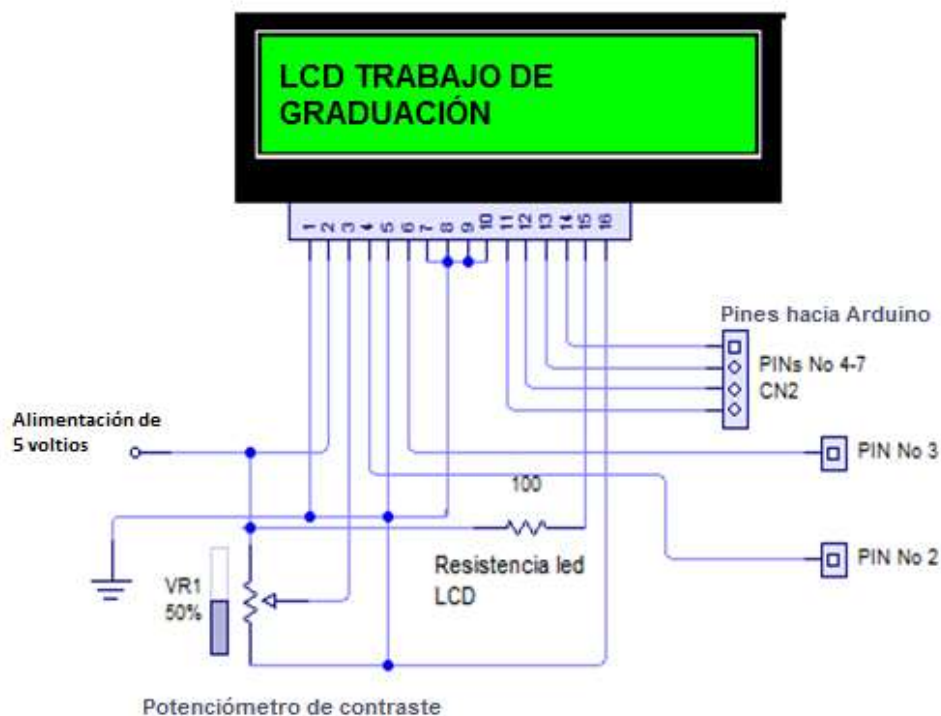
Fuente: elaboración propia, con programa Live Wire 1.30.

4.4. Módulo LCD

Se muestra a continuación la conexión del módulo LCD, notando que los pines RS y *Enable* están ubicados en los pines 2 y 3 del Arduino. Los pines de datos D4 a D7 de la LCD a los pines 4 a 7 del Arduino. También se colocó un potenciómetro alimentado por 5V al pin VEE de la LCD para variar el contraste.

Los pines que no son utilizados para alimentación (Vcc, GND) o para la iluminación (A y K) son aterrizados para evitar errores en los datos mostrados, debido a fuentes de ruido electromagnético.

Figura 15. Diagrama de circuito de LCD

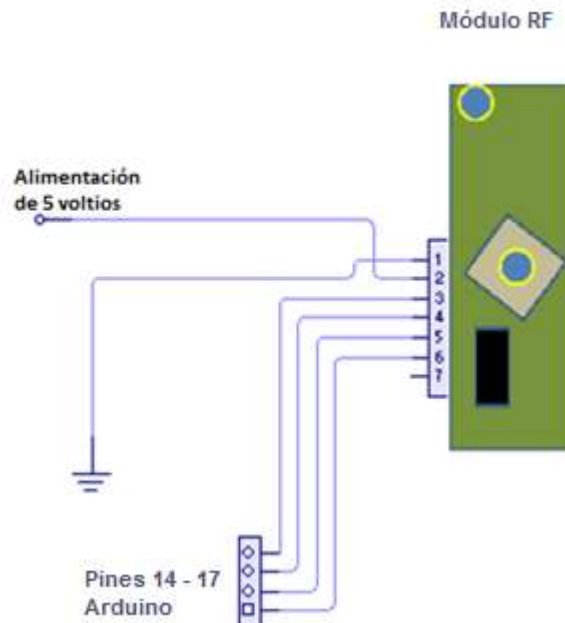


Fuente: elaboración propia, , con programa Live Wire 1.30.

4.5. Módulo RF o llave electrónica

Con el propósito de proveer una llave al usuario se utilizó un módulo RF PT2262/PT2272 4-CH, el cual es alimentado con voltaje en los pines 1 y 2. Sus 4 salidas son conectadas a los pines 14,15, 16 y 17 del Arduino.

Figura 16. Diagrama de circuito de conexión módulo RF



Fuente: elaboración propia, con programa Live Wire 1.30.

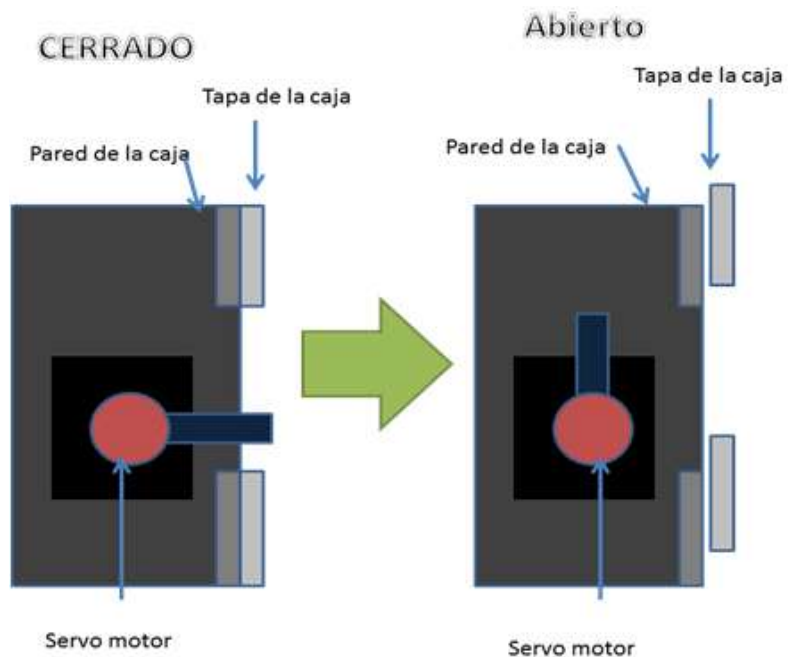
4.6. Candado electrónico

Para el candado electrónico se optó por utilizar un servomotor de 5 voltios, ya que con este se podrá mover la chapa a la posición deseada, teniendo un gran torque y utilizando un amperaje relativamente pequeño.

4.6.1. Parte mecánica

El servomotor estará por defecto en la posición de cerrado, pero cuando el circuito de control dé la orden, este moverá su eje, permitiendo la liberación de la puerta de la caja donde se encuentra el dispositivo.

Figura 17. **Representación de la mecánica de la chapa electrónica**

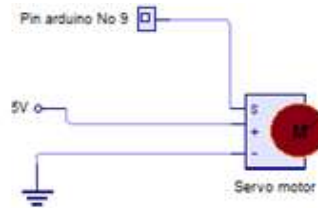


Fuente: elaboración propia, con programa Microsoft Power Point 2007.

4.6.2. Circuito servomotor

El servomotor será alimentado por 5 voltios y se conectará al pin 9 del Arduino que simulará una salida analógica con ayuda de un PWM, incorporado al microcontrolador.

Figura 18. **Diagrama de conexión de servomotor**



Fuente: elaboración propia, con programa Liwe Wire 1.30.

4.6.3. Fuente de alimentación propuesta

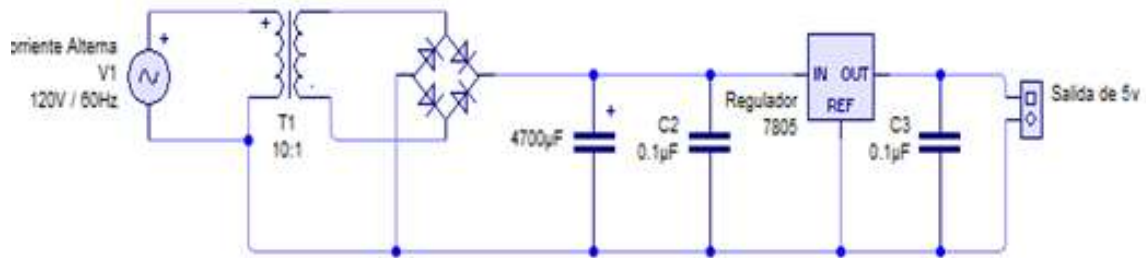
Para alimentar el circuito se podrá utilizar la entrada de alimentación externa del Arduino, la cual soporta de 9 a 12 voltios o utilizar la entrada USB, proporcionándole 5 voltios. Todo esto previamente rectificado y regulado.

Asimismo, en caso de que el usuario no posea una fuente de alimentación que provea 12 o 5 voltios, a continuación se presenta el diseño de una fuente de alimentación.

Esta se hará con la ayuda de un transformador 10:1 para reducir el voltaje alterno de 120 a 12 voltios, seguido de un puente rectificador de diodos para convertir el voltaje AC en DC. Seguido de esto se procederá a colocar un capacitor electrolítico de 4700 μF para estabilizar el voltaje con lo cual se obtendrán aproximadamente 12 voltios de voltaje DC directo. Por último se procederá a utilizar el regulador LM7805 junto con 2 capacitores cerámicos de 0.1 μF , para regular el voltaje a 5 voltios.

Esta fuente puede también ser reemplazada por otro tipo de fuente o cargador que provea 5 voltios con una corriente de 2 amperios.

Figura 19. Propuesta para elaboración de fuente de 5 voltios

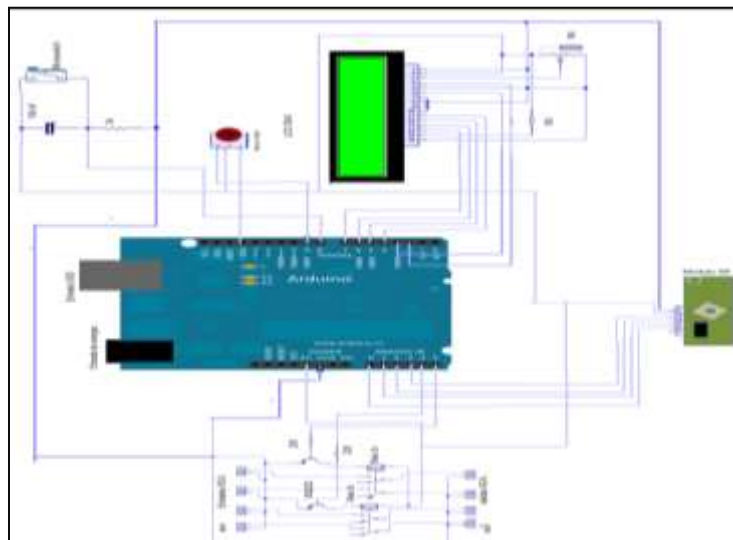


Fuente: elaboración propia, con programa Live Wire 1.30.

4.7. Circuito general

La conexión de todos los componentes electrónicos hacia el Arduino se muestra a continuación; con ello se plantea elaborar una placa de circuito impreso que sea fácil de conectar.

Figura 20. **Diagrama general de todo el circuito**



Fuente: elaboración propia, con programa Live Wire 1.30.

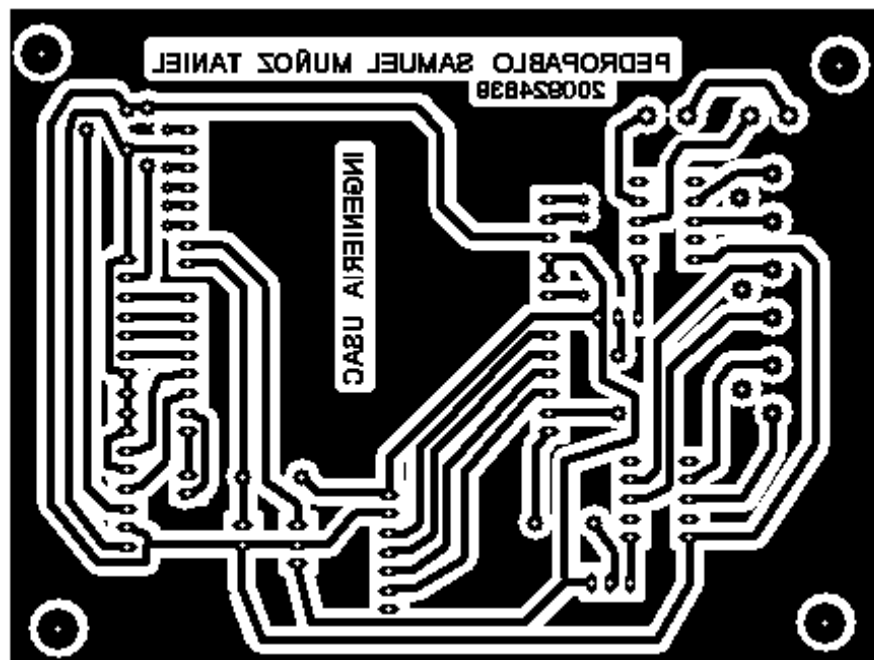
4.7.1. Diseño de PCB

La placa está diseñada para que el Arduino se coloque con los pines, encajando con cada uno de los que sean necesarios para el funcionamiento.

Se ha minimizado el circuito para ser capaz de permitir la conexión de todos los componentes en una placa de un tamaño moderado, sobre la cual se debe colocar el Arduino Duemilanove.

En la figura 21 se puede notar el diseño de PCB invertido para la construcción en placa de cobre virgen:

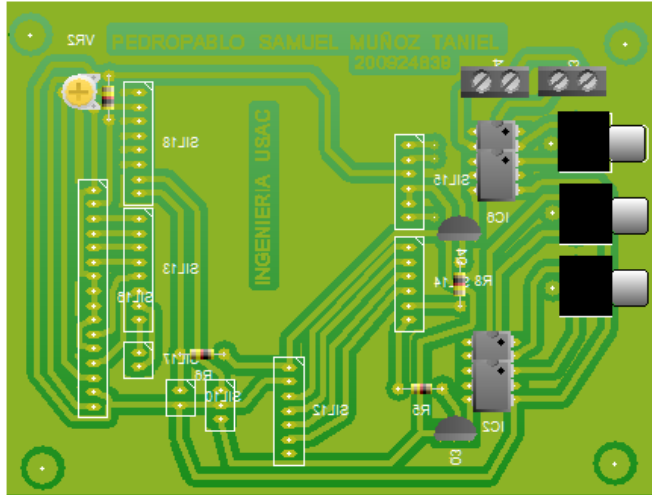
Figura 21. **Diseño de placa PCB circuito de adaptación**



Fuente: elaboración propia, con programa PCB Wizard 1.30.

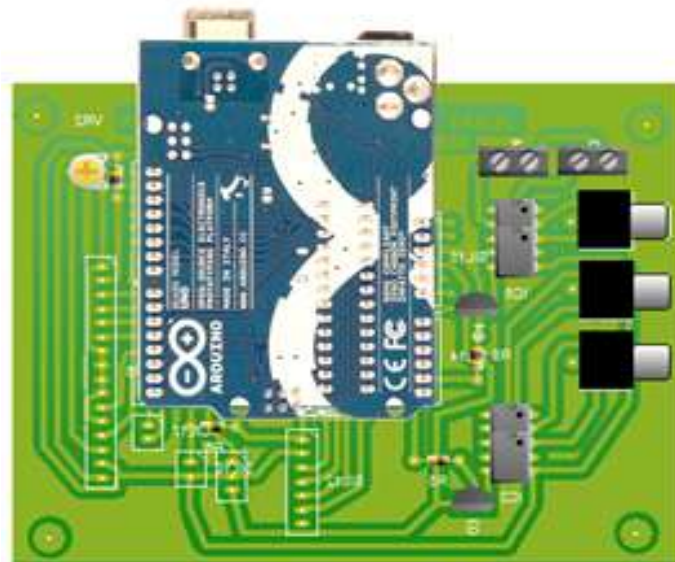
En las figuras 22 y 23 se muestra la simulación del prototipo construido.

Figura 22. **Simulación de circuito impreso sin Arduino**



Fuente: elaboración propia, con programa PCB Wizard 1.30.

Figura 23. **Simulación de circuito impreso con Arduino**



Fuente: elaboración propia, con programa PCB Wizard 1.30.

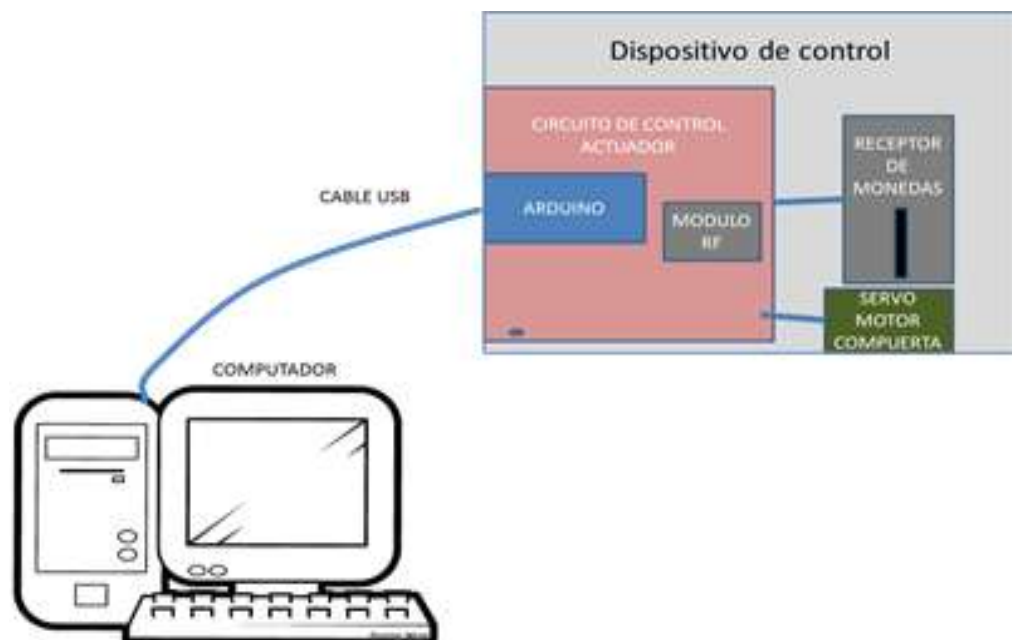
4.8. Diagrama de conexión del prototipo

Hay que destacar que el diseño está pensado para controlar una videoconsola o un computador. La forma de conectar cambiará según sea la necesidad. A continuación se explica la forma correcta de conexión del controlador.

4.8.1. Conexión para control de computador

El dispositivo presenta la opción de controlar videoconsolas o computadores; en el caso de utilizar un computador, solamente debe tomarse un cable USB y conectarlo hacia el controlador.

Figura 24. Diagrama de conexión para control de computador

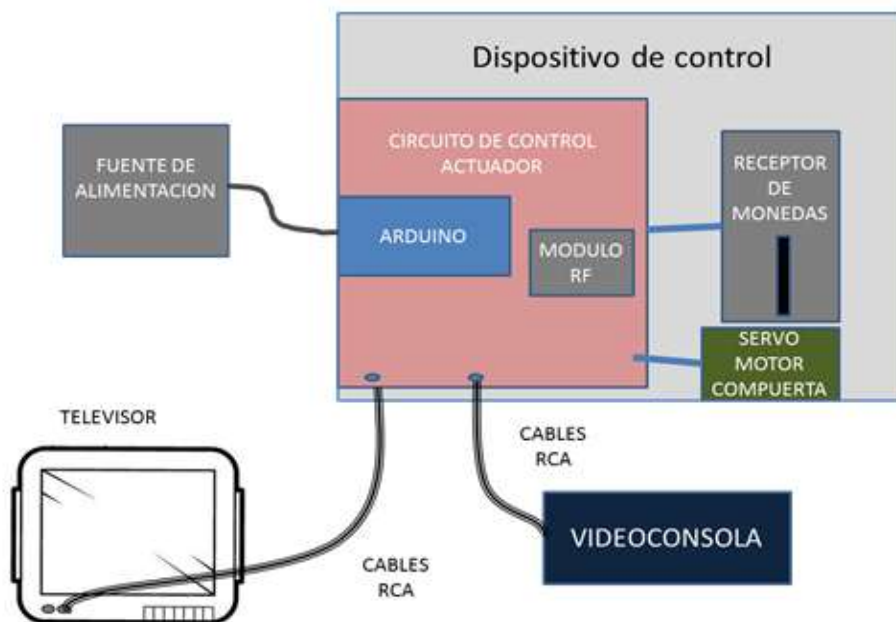


Fuente: elaboración propia, con programa Microsoft Power Point 2007.

4.8.2. Conexión para control de videoconsola

En el caso de la conexión con una videoconsola, se deberá alimentar el circuito externamente con 5 voltios por la entrada USB del Arduino o con un voltaje de entre 9 a 12 voltios a través del conector correspondiente del circuito. Asimismo se deben colocar los 3 cables RCA entre la consola, el controlador y 3 cables más, a la salida del controlador y la videoconsola.

Figura 25. Diagrama de conexión para control de videoconsola



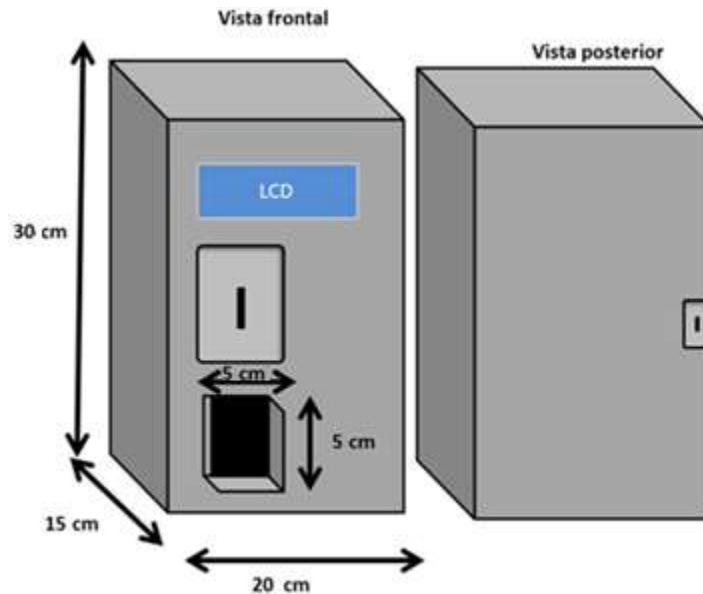
Fuente: elaboración propia, con programa Microsoft Power Point 2007.

4.9. Contenedor del prototipo

Para el prototipo se plantea la creación de una caja de material resistente (madera, metal, acrílico, etc.) para contener el dispositivo.

El tamaño mostrado en las figuras 26 y 27, puede ser variable y se podría modificar dependiendo de las necesidades.

Figura 26. **Vista frontal y posterior del dispositivo abierto**



Fuente: elaboración propia, con programa Microsoft Power Point 2007.

Figura 27. **Ubicación de los componentes dentro del contenedor**



Fuente: elaboración propia, con programa Microsoft Power Point 2007.

5. DESARROLLO DEL SOFTWARE DEL PROTOTIPO

La programación del controlador y temporizador fue realizado en dos diferentes lenguajes: Arduino (para la placa de desarrollo del mismo nombre) y Visual Studio (para la interfaz gráfica en el computador). A continuación se muestra la programación asociada a cada una de las partes.

5.1. Programación del microcontrolador

El siguiente código es el *firmware* encargado de recibir y procesar las instrucciones que llegan en forma de señales eléctricas hacia la placa de desarrollo Arduino. Este código controla cada uno de los pines de salida que activan la chapa electrónica a través de PWM. También es el encargado de procesar el pulso provocado por el transductor tragamonedas y las señales digitales del control o llave electrónica.

A continuación se muestra la programación contenida dentro de la memoria flash del Arduino.

```
/******  
**UNIVERSIDAD SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERÍA**  
**TRABAJO DE GRADUACIÓN**  
**CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS  
DE VIDEOJUEGO**  
**PEDROPABLO SAMUEL MUÑOZ TANIEL CARNET 200924839 INGENIERIA  
ELECTRONICA**  
**ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR**  
*****/  
    ///**LIBRERIAS**///  
    #include<LiquidCrystal.h> //libreria LCD  
    #include<Servo.h> //libreria de servo  
    #include<MsTimer2.h> //libreria de tiempo  
    #include<EEPROM.h> //libreria para memoria eeprom
```

```

/**PINES Y VARIABLES**/
//variable
LiquidCrystalcd(2, 3, 4, 5, 6, 7); //se los pines de la LCD: RS 2, Enable 3, datos 4-7
//variables y constantes
String tiempo;
String secs, mins, hours, tiempo2, secs2, mins2, hours2, enviar;
int bandera2=0;
int bandera=0;
int monedas=0;
int monedas2=0;
int contador=0;
int segundos=0;
int minutos=0;
int horas=0;
int segundos2=0;
int minutos2=0;
int horas2=0;
int cuota=EEPROM.read(5);
//Entradas digitales
constintcontrolB = 14;
constintcontrolD = 15;
constintcontrolA= 16;
constintactivador=8; //moneda
constintcontrolC = 17;

//RELES (salidas digitales)
constint rele1 = 18;
constint rele2 = 19;

//servo
Servo candado;
///*****LOOP DE CONFIGURACION***** //
///*****//
void setup() {

//Leer los datos de la eeprom para el tiempo//
monedas=EEPROM.read(1);
segundos=EEPROM.read(2);
minutos=EEPROM.read(3);
horas=EEPROM.read(4);
////////////////////

Serial.begin(9600); //velocidad del puerto serie 9600 BAUDIOS
lcd.begin(20, 4); // Filas y columnas de la lcd

//////////mensaje de bienvenida//////////
lcd.setCursor(0,0);
lcd.print("Guatemala USAC 2014");
lcd.setCursor(0,1);
lcd.print("Facultad ingenieria");
lcd.setCursor(0,2);
lcd.print("Escuela EIME") ;

```

```

lcd.setCursor(0,3);
lcd.print("Ing. Electronica") ;
delay(2000);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Estudiante:");
lcd.setCursor(0,1);
lcd.print("PedroPablo Samuel");
lcd.setCursor(0,2);
lcd.print("Munoz Taniel ") ;
lcd.setCursor(0,3);
lcd.print("Carnet 200924839") ;
delay(2000);
////////////////////
////////mensaje por defecto////////
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Dispositivo apagado");
lcd.setCursor(0,1);
lcd.print("Listo para funcionar");
lcd.setCursor(0,2);
lcd.print("Q1.00="+String(cuota)+"min") ;
////////////////////
MsTimer2::set(100, temporizador); // timmer a 100mS
candado.attach(9); //Variable servo asignada al pin 9

//////// Modo E/S pines digitales////////
pinMode(rele1, OUTPUT);
pinMode(rele2, OUTPUT);
pinMode(controlB, INPUT);
pinMode(controlD, INPUT);
pinMode(controlA, INPUT);
pinMode(controlC, INPUT);
pinMode(activador, INPUT);
}

//*****//
//*****PROGRAMA PRINCIPAL***** //
//*****//

voidloop() {

    ////////////si hay monedas funcionar//////////inicio//////////
    if (monedas>0){

        digitalWrite(rele1, HIGH);
        digitalWrite(rele2, HIGH);
        if (bandera==0){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Dispositivo Encendid");
            bandera=1;
            MsTimer2::start();

```

```

    }
}
//////////si hay monedas funcionar//////////FIN //////////
//////////si no hay monedas apagar//////////inicio//////////
if (monedas==0){
  minutos=0;
  segundos=0;
  horas=0;
  EEPROM.write(1, 0);
  EEPROM.write(2, 0);
  EEPROM.write(3, 0);
  EEPROM.write(4, 0);
  if (bandera==1){
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Dispositivo Apagado ");
    lcd.setCursor(0,1);
    lcd.print("Credito: Q"+String(monedas)+".00");
    lcd.setCursor(0,2);
    lcd.print(tiempo);
    lcd.setCursor(0,3);
    lcd.print(tiempo);
    Serial.print("F");
    digitalWrite(rele1, LOW);
    digitalWrite(rele2, LOW);
    MsTimer2::stop();
    bandera=0;
    delay(1000);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Dispositivo apagado");
    lcd.setCursor(0,1);
    lcd.print("Listo para funcionar");
    lcd.setCursor(0,2);
    lcd.print("Q1.00="+String(cuota)+"min");
  }
}
//////////si no hay monedas apagar//////////final//////////

//////////Si recibe una moneda//////////inicio//////////
if ((digitalRead(controlA) == HIGH)||digitalRead(activador) == HIGH) {
  while((digitalRead(controlA) == HIGH)||digitalRead(activador) == HIGH){
    delay(100); //antirebo
  }
  monedas=monedas+1;
  lcd.setCursor(0,1);
  lcd.print("Credito: Q"+String(monedas)+".00 ");
  Serial.print("G");
}
//////////desde la computadora//////////
serialcompu();

```

```

////////////////////////////////////Si recibe una moneda////////////////////////////////////final////////////////////////////////////

if (digitalRead(controlB) == HIGH) {
while (digitalRead(controlB)==HIGH){
}
Serial.print(27);
}
////////////////////////////////////Llave electronicaprecionarboton C////////////////////////////////////inicio
if (digitalRead(controlC) == HIGH) {
while (digitalRead(controlC)==HIGH){
}
bandera2=1;
Serial.print("K");
MsTimer2::stop();
lcd.clear();
lcd.setCursor(0,0);
lcd.print("La llave electronica");
lcd.setCursor(0,1);
lcd.print("ha sido detectada");
lcd.setCursor(0,2);
lcd.print("Menuactivandose") ;
salgo:
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Menu:A.Cuota");
lcd.setCursor(0,1);
lcd.print(" B.Abrir candado");
lcd.setCursor(0,2);
lcd.print(" C.Reset general");
lcd.setCursor(0,3);
lcd.print(" D.salir");
while (bandera2==1){
if (digitalRead(controlA) == HIGH) {
while (digitalRead(controlA)==HIGH){
}
}
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Cuota: Q1.00="+String(cuota)+"min");
lcd.setCursor(0,1);
lcd.print("B.+ D.- C.Salir") ;
retorno:
if (digitalRead(controlB) == HIGH) {
cuota=cuota+1;
if (cuota >60){cuota=1;}
EEPROM.write(5, cuota);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Cuota: Q1.00="+String(cuota)+"min");
lcd.setCursor(0,1);
lcd.print("B.+ D.- C.Salir") ;
delay(200);
}
}
}

```

```

if (digitalRead(controlD) == HIGH) {
  cuota=cuota-1;
  if (cuota <1){cuota=60;}
  EEPROM.write(5, cuota);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Cuota: Q1.00="+String(cuota)+"min");
  lcd.setCursor(0,1);
  lcd.print("B.+ D.- C.Salir") ;
  delay(200);
}
if (digitalRead(controlC) == HIGH) {
  goto salgo;
}
goto retorno;
lcd.clear();
lcd.setCursor(0,0);
lcd.print("Menu:A.Cuota");
lcd.setCursor(0,1);
lcd.print("  B.Abrir candado");
lcd.setCursor(0,2);
lcd.print("  C.Reset general");
lcd.setCursor(0,3);
lcd.print("  D.salir");
}
if (digitalRead(controlB) == HIGH) {
  while (digitalRead(controlB)==HIGH){
  }
  candado.write(150);
  delay(5000);
  candado.write(90);
}
if (digitalRead(controlD) == HIGH) {
  while (digitalRead(controlD)==HIGH){
  }
  bandera2=0;
  bandera=0;
  MsTimer2::stop();
  // MsTimer2::start();
  setup();
  break;
}
if (digitalRead(controlC) == HIGH) {
  while (digitalRead(controlC)==HIGH){
  }
  monedas=0;
  segundos=0;
  minutos=0;
  contador=0;
  horas=0;

  EEPROM.write(1, 0);

```

```

EEPROM.write(2, 0);
EEPROM.write(3, 0);
EEPROM.write(4, 0);
    // EEPROM.write(5, 10);
}

}
}
}

//*****//
//*****//

//SUBROUTINA DE INTERRUPCION POR TIMMER //
//*****//
void temporizador()
{
    contador=contador+1;
    Serial.print(enviar);
        //////////////////////////////////////////////////ccontador de tiempo////////////////////////////////
    if (monedas < 10){

        if (cuota< 10){
            enviar="M "+String(monedas)+mins2+"."+secs2+" "+String(cuota);
        }else {
            enviar="M "+String(monedas)+mins2+"."+secs2+String(cuota);    }

        } else{
            if (cuota< 10){
                enviar="M"+String(monedas)+mins2+"."+secs2+" "+String(cuota);
            }else
            {
                enviar="M"+String(monedas)+mins2+"."+secs2+String(cuota); }
            }

            //////////////////////////////////cuenta ascendente////////////////////////////////
        if (contador>9) {
            contador=0;
            segundos=segundos+1;
            lcd.setCursor(0,1);
            lcd.print("Credito: Q"+String(monedas)+".00 ");
            lcd.setCursor(0,2);
            lcd.print(tiempo);
            lcd.setCursor(0,3);
            lcd.print(tiempo2);
            //guardar en la eeprom cada segundo//
            EEPROM.write(1, monedas);
            EEPROM.write(2, segundos);
            EEPROM.write(3, minutos);
            EEPROM.write(4, horas);
            //////////////////////////////////
            //String enviar="valor"+String(monedas)+tiempo+tiempo2+String(cuota);
        }
        //////////////////////////////////

```

```

if (segundos>59) {
    segundos=0;
    minutos=minutos+1;
}
if (minutos>59) {
    minutos=0;
    horas=horas+1;
}
if (segundos < 10){
    secs="0"+String(segundos);
} else{
    secs= String(segundos);
}
if (minutos< 10){
    mins="0"+String(minutos);
} else {
    mins= String(minutos);
}
if (horas< 10){
    hours="0"+String(horas);
} else {
    hours= String(horas);
}
tiempo=hours+":"+mins+":"+secs;

if ((minutos== cuota)&&(segundos==0)&&(contador==0)){
    if (monedas >0){
        monedas=monedas-1;
        lcd.setCursor(0,1);
        lcd.print("Credito: Q"+String(monedas));
        contador=0;
        segundos=0;
        minutos=0;
    }
}
////////////////////////////////////
//////////cuenta regresiva//////////
if (segundos==0){
    minutos2=cuota-minutos;
    segundos2=0;
}else{
    minutos2=cuota-minutos-1;
    segundos2=60-segundos;
}
if (segundos2 < 10){
    secs2="0"+String(segundos2);
} else{
    secs2= String(segundos2);
}
if (minutos2 < 10){
    mins2="0"+String(minutos2);
} else {

```

```

        mins2= String(minutos2);
    }
    if (horas2 < 10){
        hours2="0"+String(horas2);
    } else {
        hours2= String(horas2);
    }
    tiempo2=hours2+"."+String(mins2)+"."+String(secs2);
}
//*****//
//*****//
void serialcompu(){
    goto saltar;
    mensaje:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.setCursor(0,0);
    lcd.print("Dispositivo apagado");
    lcd.setCursor(0,1);
    lcd.print("Listo para funcionar");
    lcd.setCursor(0,2);
    lcd.print("Q1.00="+String(cuota)+"min" ) ;
    saltar:
    if (Serial.available() > 0) {
        // lee el byte entrante:
        intuso=Serial.read();
        if (uso==76){
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Cerradura abierta");
            lcd.setCursor(0,1);
            lcd.print("desde computador");
            candado.write(150);
            delay(5000);
            candado.write(90);
            goto mensaje;

        }
        if (uso==68){
            EEPROM.write(1, 0);
            EEPROM.write(2, 0);
            EEPROM.write(3, 0);
            EEPROM.write(4, 0);
            monedas=0;
            minutos=0;
            horas=0;
            segundos=0;
        }
    }
}

```

5.2. Programación del software para la PC

El programa de comunicación y bloqueo para videoconsolas fue desarrollado en el lenguaje de programación C#, en la herramienta de desarrollo Visual Studio; capacitado para funcionar con computadores que cuenten con Windows 98 en adelante. A continuación se expone la parte gráfica y los comandos del programa.

5.2.1. Parte gráfica

Para la creación del programa se hicieron 2 formularios: se desarrolló el formulario principal (form1) de 251x157 pixeles al cual se agregaron varias etiquetas (señalización y de hora, inicio, tiempo y crédito disponible), así como un cuadro combinado para selección del puerto serie (se conecta físicamente por un puerto USB, pero es detectado por el computador como puerto serie) y un cuadro de texto para el ingreso de contraseñas. También se añadieron 6 botones para distintas funciones que se explicarán más adelante.

Figura 28. Diseño del programa de control en computador



Fuente: elaboración propia, con programa Visual Studio 2010.

Para el segundo formulario se procedió a crear un formulario en pantalla completa de gran tamaño

Figura 29. Diseño de la ventana de bloqueo



Fuente: elaboración propia, con programa Visual Studio 2010.

En ambos formularios se configuró la opción de no presentar botones de maximizar, minimizar y cerrar, así como no tener la propiedad de aparecer en la barra de tareas. Esto para que el usuario no pueda cerrar el programa con métodos convencionales.

5.2.2. Código del programa

El diseño de la parte gráfica del programa fue realizado con ayuda de Visual Studio. Dentro de la programación del formulario 1 se podrá encontrar el siguiente código:

```
/*
*****
*****
**UNIVERSIDAD SAN CARLOS DE GUATEMALA_____**
**FACULTAD DE INGENIERÍA_____**
**TRABAJO DE GRADUACIÓN_____**
**CONTROLADOR Y TEMPORIZADOR AUTOMÁTICO DE COMPUTADORES Y CONSOLAS
DE VIDEOJUEGO**
**PEDROPABLO SAMUEL MUÑOZ TANIEL CARNET 200924839 INGENIERIA
ELECTRONICA**
**ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR**
*****
*****/
////////////////////librerias////////////////////
using System;
using System.Collections.Generic;
```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Media;
using System.IO.Ports;
using System.Threading;
namespace CC
{
    public partial class Form1 : Form
    {
        int c = 0;
        int monedas = 0;
        int com = 0;
        string contra = "contra";
        Form2 frm = new Form2();
        public Form1()
        {
            InitializeComponent();

            Control.CheckForIllegalCrossThreadCalls = false;
            foreach (string s in SerialPort.GetPortNames())
            {
                comboBox1.Items.Add(s);
            }
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (button1.Text == "Empezar") //Comprueba que el botón sea para empezar
            {
                timer1.Start(); //Empieza el timer
                button1.Text = "Pausa"; //Pone que el botón sea para pausar
            }
            else //Si es para pausar
            {
                timer1.Stop(); //Parar el timer
                button1.Text = "Empezar"; //Pone el botón para empezar
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            serialPort1.Write("D");
            //Parar del todo y poner a 0
            timer1.Stop(); //Parar el timer
            button1.Text = "Empezar"; //Pone el botón para empezar por si se había cambiado previamente
            label2.Text = "00:00:00"; //Pone la cuenta a 0
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}

```

```

privatevoid timer1_Tick(object sender, EventArgs e)
{
    ////////////////////////////////////SI MONEDAS no hay monedas ////////////////////////////////////
    if (monedas==0)
    {
        SystemSounds.Exclamation.Play();
        timer1.Stop();
        frm.Show();
        label2.Text = "00:00:00"; //Pone la cuenta a 0
    }
}

privatevoid button3_Click(object sender, EventArgs e)
{
    Application.Exit();
}

privatevoid label6_Click(object sender, EventArgs e)
{
}

privatevoid timer2_Tick(object sender, EventArgs e)
{
    if (c == 0)
    {
        button5.Visible = false;
        button6.Visible = false;
        c = 1;
    }
    label4.Text = DateTime.Now.ToLongTimeString();
    if (com == 1)
    {
        timer1.Start();
        com = 0;
    }

    if (monedas>0)
    {
        timer1.Start();
    }
}

privatevoid serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    ////////////////////////////////////recibir atraves del puerto serial////////////////////////////////////
    stringdato_reciv = serialPort1.ReadExisting();
    if (dato_reciv == "F")
    {
        monedas = 0; //desactivar
    }
    if (dato_reciv == "K") //llave electronica detectada
    {
        button4.Visible = true;
        textBox1.Visible = true;
        frm.Hide();
    }
}

```

```

        c = 0;
    }
    try
    {
        string nuevo = dato_reciv.Substring(0, 1);
        string nuevo2 = dato_reciv.Substring(3, 5);
        if (nuevo == "M")//recibe los datos y los muestra en pantalla
        {

            label3.Text = "Q1.00= "+dato_reciv.Substring(8, 2)+"min";
            label8.Text = "Q."+ dato_reciv.Substring(1, 2)+".00";
            label2.Text = "00:"+nuevo2;
            monedas = 1;
            frm.Hide();
        }
    }
    catch
    {
    }
    ///////////////////////////////////activar PC////////////////////////////////////
    if (dato_reciv == "G")
    {
        com = 1;
        frm.Hide();
        if (monedas==0)
        {
        }
        Thread.Sleep(2);
        monedas = 1;
    }
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    serialPort1.PortName = comboBox1.Text;
    try
    {
        serialPort1.Open();
    }
    catch
    {
        MessageBox.Show("Puerto no válido");
        return;
    }
    comboBox1.Enabled = false;
    comboBox1.Visible = false;
}
private void button4_Click(object sender, EventArgs e)
{
    if (textBox1.Text==contra)
    {
        button5.Visible = true;
        button2.Visible=true;
    }
}

```

```

button3.Visible = true;
MessageBox.Show("contraseña ingresada con exito");
button6.Visible = true;
timer1.Stop();
}
else
{
frm.Show();
MessageBox.Show("contraseña invalida Utilice la clave para activar PC");
}
    textBox1.Text = "";
}
private void button5_Click(object sender, EventArgs e)
{
serialPort1.Write("L");
serialPort1.Write("J");
}
private void button6_Click(object sender, EventArgs e)
{
    comboBox1.Enabled = false;
    comboBox1.Visible = false;
    button1.Visible = false;
    button2.Visible = false;
    button3.Visible = false;
    button4.Visible = false;
    textBox1.Visible = false;
    button5.Visible = false;
    button6.Visible = false;
frm.Show();
}
private void button6_Click_1(object sender, EventArgs e)
{
    comboBox1.Enabled = false;
    comboBox1.Visible = false;
    button1.Visible = false;
    button2.Visible = false;
    button3.Visible = false;
    button4.Visible = false;
    textBox1.Visible = false;
    button5.Visible = false;
    button6.Visible = false;
frm.Show();
}
}
}
}

```

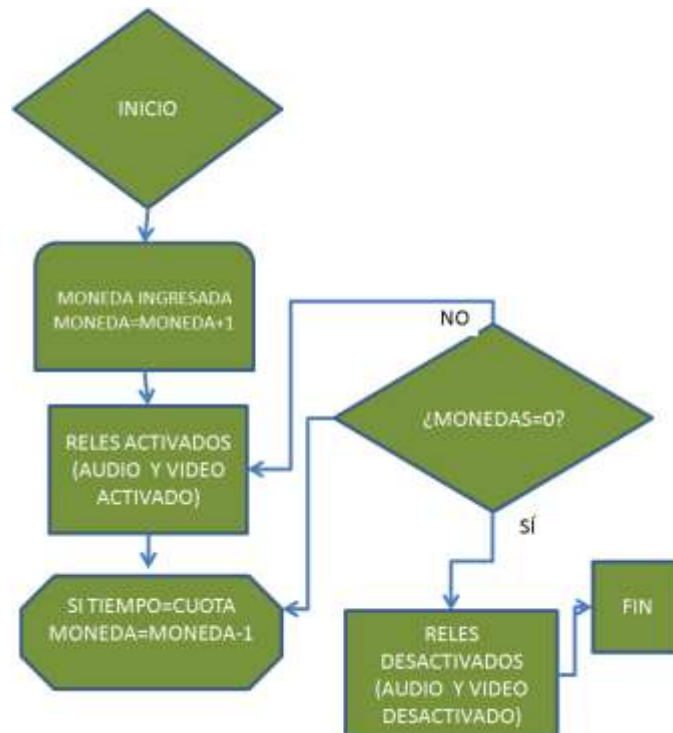

6. FUNCIONAMIENTO DEL PROTOTIPO

Una vez conectado el circuito correspondiente al funcionamiento deseado podrán controlarse los dispositivos de la forma que a continuación se describe.

6.1. Funcionamiento como controlador de videoconsolas

En el siguiente diagrama de flujo puede observarse el funcionamiento del dispositivo conectado a una videoconsola.

Figura 30. Diagrama de flujo como controlador de videoconsolas

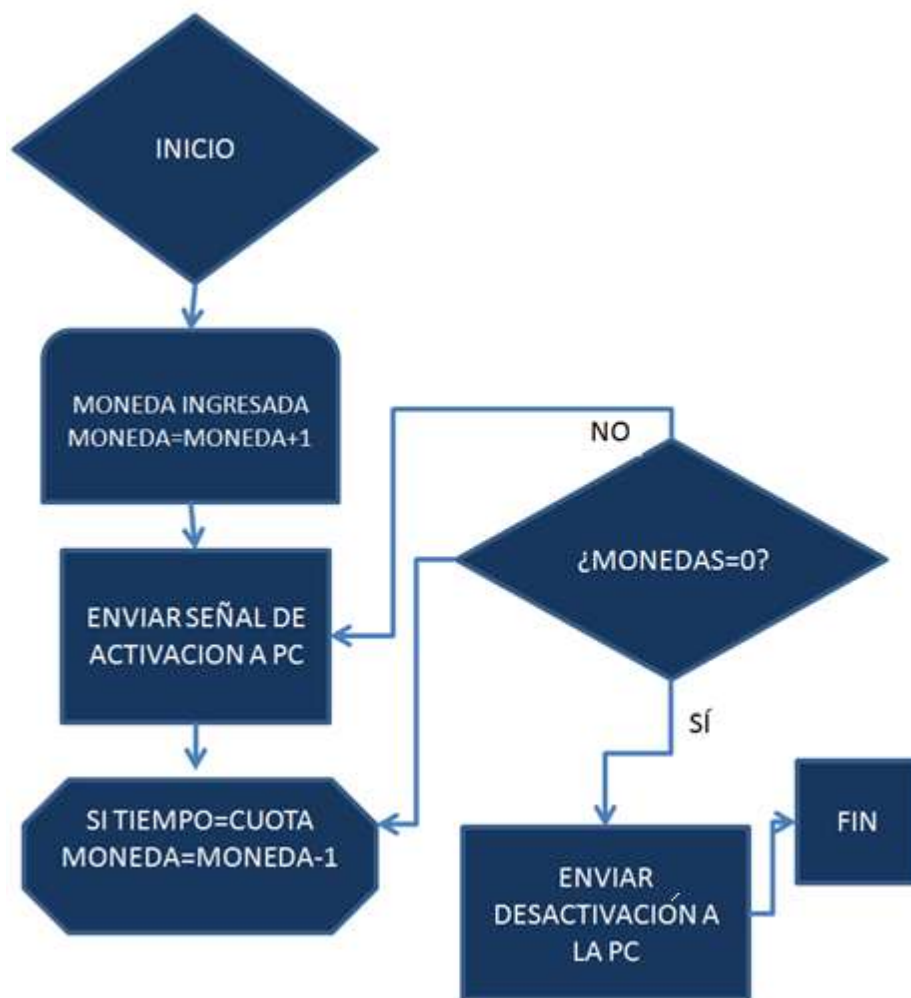


Fuente: elaboración propia, con programa Microsoft Power Point 2007.

6.2. Funcionamiento como controlador de computadores

El funcionamiento del dispositivo como controlador de computadores puede observarse a continuación.

Figura 31. Diagrama de flujo como controlador de computadores



Fuente: elaboración propia, con programa Microsoft Power Point 2007.

6.2.1. Funcionamiento de software dentro del computador

Para abrir el programa bastará con dar clic en el icono mostrado en la figura 32, para luego ejecutar el programa, el cual posee el interfaz mostrado en la figura 33.

Figura 32. Icono de la interfaz para el computador



Fuente: elaboración propia, con programa Visual Studio 2010.

Figura 33. Interfaz control de tiempo



Fuente: elaboración propia, con programa Visual Studio 2010.

6.2.1.1. Funcionamiento como administrador

Una vez abierto el programa de control en modo inicio, se debe seleccionar el puerto serie correspondiente.

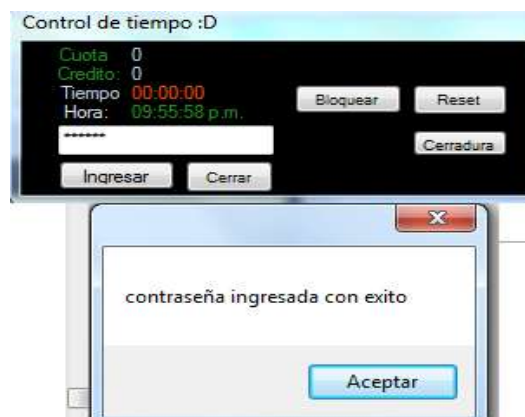
Figura 34. **Inicio de funcionamiento del software**



Fuente: elaboración propia, con programa Visual Studio 2010.

Una vez seleccionado el puerto correspondiente, se debe ingresar la contraseña. Si la contraseña es correcta mostrará un mensaje y se abrirá el modo administrador. Si la contraseña se ingresó incorrectamente, el computador se bloqueará y no se podrá ingresar dicha contraseña, a menos que se use el control (llave electrónica).

Figura 35. **Modo administrador de software de computadores**



Fuente: elaboración propia, con programa Visual Studio 2010.

Una vez esté en modo administrador, se podrá volver a bloquear el computador, reiniciar el controlador, reanudar la administración de tiempo (si se tiene monedas), y abrir y cerrar la chapa. Para poder ingresar al modo administrador, si la computadora está bloqueada, se debe usar la llave electrónica presionando el botón B.

6.3. Funcionamiento de control (llave electrónica)

El módulo RF que se utilizó trae por defecto un control que posee 4 botones (A, B, C y D). Para activar el menú de opciones debe presionarse el botón C (esto también activará el modo administrador en la PC si es que está conectado a la misma).

Figura 36. **Menú de opciones con control RF**



Fuente: elaboración propia, con programa Microsoft Power Point 2007.

Dependiendo de la selección que se haga, podrán administrarse distintas funciones del controlador:

- A. Cuota: cambiar la cantidad de minutos que se dan por Q1,00.
- B. Abrir candado: abre el candado durante 5 segundos y activa el modo administrador en la PC.

- C. Borra todo lo ingresado y coloca valores por defecto.
- D. Salir de modo administrador.

CONCLUSIONES

1. Se determinó que los dispositivos de control y temporización de computadores y consolas de videojuego actuales no son eficientes porque los sistemas no proporcionan una seguridad efectiva.
2. Se estableció que los antecedentes y formas de administrar un cibercafé son por métodos manuales, de software, y de hardware.
3. Pudo describirse el historial de los videojuegos y las consolas de videojuego, las cuales se han caracterizado por su evolución en generaciones; la primera data de los años cincuenta y la última se denomina de la era actual.
4. Los componentes físicos más relevantes encontrados dentro de la propuesta del diseño son: el microcontrolador Arduino, los conectores USB y RCA, el *microswitch*, el servomotor, entre otros.
5. La programación del software que permite el funcionamiento del prototipo del dispositivo controlador cuenta con una plataforma de administración para sistemas operativos basados en Windows y Linux en conjunto con el software interno del microcontrolador.
6. El desarrollo del presente diseño determinó que debe presentarse al lector una forma entendible respecto del uso del dispositivo, específicamente en el caso de ser utilizado como controlador de computadores y videoconsolas.

RECOMENDACIONES

1. Tomar en cuenta que el diseño de un dispositivo de control de videoconsolas y computadores optimiza en gran medida la administración de estos dispositivos, reduciendo costos y permitiendo al encargado dedicar un menor tiempo al negocio.
2. Para optimizar los dispositivos de control y temporización de computadores y consolas actuales, es importante considerar los antecedentes y las formas de administrar un cibercafé, y de ellas tomar la más eficiente.
3. Al establecer los precedentes de las consolas de videojuegos y sus principales características, se deben tomar en cuenta las diferencias entre cada marca y generación de videoconsola para poder realizar un dispositivo que pueda controlar a la mayoría de clases existentes.
4. Es importante conocer el funcionamiento y las características de todos los componentes utilizados para la solución propuesta, ya que actualmente no existe un controlador que administre y temporeice videoconsolas y computadores.
5. Es importante para el funcionamiento de la programación desarrollada que el dispositivo sincronice el sistema operativo del computador y el software dentro del microcontrolador. También que permita controlar y temporizar las distintos tipos de videoconsola, independientemente de su software.

6. Para lograr el funcionamiento del dispositivo es importante establecer un resumen que explique al usuario y al administrador cómo conectar el dispositivo, cómo hacer uso del mismo y cómo utilizar la llave electrónica de seguridad.

BIBLIOGRAFÍA

1. ARDUINO. *The Arduino Playground*. [en línea].
<<http://playground.arduino.cc/>>. [Consulta: 20 de junio de 2014].
2. CIBERCAFE.US. *Historia del cibercafé*. [en línea].
<www.cibercafe.us/internet/negocios/cafe/historia_del_cybercafe/>
[Consulta: 20 de mayo de 2014].
3. Club Planeta. *Un cibercafé y sus servicios*. [en línea].
<http://www.trabajo.com.mx/un_cibercafe_y_sus_servicios.htm>.
[Consulta: 30 de abril de 2014].
4. ENTEGRATE. *¿Qué necesitas para un café internet?* [en línea].
<<http://e-ntegrate.com/soporte/quenecesitasparaunci.html>>.
[Consulta: 20 de marzo de 2014].
5. ITAMI, Hiroyuki; ROEHL, Thomas. *Mobilizing invisible assets*.
Cambridge: Harvard University Press, 1991. 200 p.
6. KENT, Steve. *The ultimate history of videogames* Estados Unidos:
Three Rivers Press, 2001. 608 p.
7. LANDA, Cosio; ARRIOJA, Nicolás. *Guía total del programador*.
Argentina: Fox Andina, 2010. 404 p.

8. Nómadas Electrónicos. *Puerto serial en C sharp*. [en línea].
<<http://nomadaselectronicos.wordpress.com/2013/01/11/puerto-serial-en-c-para-no-programadores/>>. [Consulta: 20 de mayo de 2014].
9. TEKNOFLOP. *No te pierdas con los conectores USB*. [en línea].
<<http://www.teknoplof.com/2010/08/03/no-te-pierdas-con-los-conectores-usb/>>. [Consulta: 4 de mayo de 2014].
10. WIKIPEDIA. *Videoconsola*. [en línea].
<<http://es.wikipedia.org/wiki/Videoconsola>>. [Consulta: 28 de abril de 2014].