



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

1. Título

PROYECTO FINAL INTERFACES: Comunicación serial microcontrolador-ordenador.

2. Resumen

El presente informe contempla todos los conocimientos adquiridos en el curso con el fin de aplicarlos en un proyecto con la finalidad de demostrar la aplicación de lo antes dicho, en el particular caso aquí presente sería el uso de los microcontroladores, la comunicación serial entre el ordenador y el dispositivo de control mediante un cable de comunicación USB-serial, y las interfaces visuales que en este caso se desarrollaron en Visual Basic, el diagrama en PROTEUS muestra la estructura de los elementos pertenecientes al circuito, el MICRCODE STUDIO se muestra las instrucciones que se alojan en el microcontrolador, finalmente tenemos la interfaz desarrollada en VISUAL BASIC la cual se encarga de hacer más fácil la manipulación del código.

3. Objetivos

Plantear una interfaz para comunicar la computadora con el microcontrolador y de esa manera implementar el paso de mensajes hacia el dispositivo.

Analizar los procesos que lleva el desarrollo para la comunicación desde el ordenador al dispositivo, y comentar sobre las dificultades y los requerimientos para el correcto funcionamiento de la plataforma.

4. Marco teórico

MANEJO DE DISPLAY LCD

Si bien muchas aplicaciones donde debemos mostrar información podemos resolverlas con display de 7 segmentos, estos presentan algunas limitaciones importantes, por ejemplo: no muestran caracteres alfanuméricos ni ASCII, y tienen un elevado consumo de corriente (recordemos que estamos encendiendo LEDs). Los módulos LCD (display de cristal líquido) solucionan estos inconvenientes y tienen algunas ventajas adicionales como la facilidad con que se pueden conectar a microprocesadores y microcontroladores, sumado a la óptima presentación de la información.

CARACTERISTICAS PRINCIPALES DE LOS MODULOS LCD

Los módulos LCD se encuentran en diferentes presentaciones, por ejemplo 2x16 (2 líneas de 16 caracteres), 2x20, 4x20, 4x40, etc. Es conveniente buscar en catálogos para encontrar aquel que cumpla con las necesidades de nuestro proyecto. La forma de utilizarlos y sus interfaces (como se conectan) son similares. En la hoja de datos del display viene la distribución de pines, alimentación y el controlador (CI interno del módulo), etc. Es fundamental también buscar la hoja de datos de CI interno o controlador de LCD, allí encontraremos la información para operar con el mismo. Describiremos a continuación como trabajar con el display 2x16 (FDCC1602E con controlador interno SPLC780) y se podrá considerar esta información genérica para displays similares. Se muestra en la figura un display de este tipo. Los pines de conexión de estos módulos incluyen un bus de datos de 8 bits, un pin de habilitación (E), un pin de selección que indica si lo que se está enviando por el bus es un dato o una instrucción (RS) y un pin que indica si se va a leer o escribir en el módulo (R/W).

Según la operación que se desee realizar en el módulo LCD, los pines de control E, RS#, RW# deben tener un estado determinado. Además debe tener en el bus de datos un código que indique un carácter para mostrar en la pantalla o una instrucción de control para el display. Los módulos LCD responden a un conjunto especial de instrucciones, estas deben ser enviadas por el microcontrolador o sistema de control al display, según la



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

operación que se requiera. Se muestran a continuación el conjunto de instrucciones del modulo LCD.

Nota: Cuando se va a cargar la dirección de la DDRAM donde se va a escribir el próximo caracter, nótese que el D7 siempre es 1. Por lo tanto cuando apunto a una dirección de memoria en el display hay que considerar este 1 adicional. Se verá con más detalle al describir el mapa de memoria del módulo LCD. La interfase entre el microcontrolador y el LCD se puede hacer con el bus de datos del pic trabajando a 4 u 8 bits. Las señales de control trabajan de la misma forma en cualquiera de los dos casos, la diferencia se establece en el momento de iniciar el display, ya que existe una instrucción que permite establecer dicha configuración. O sea tenemos que avisarle al LCD que vamos a operar en 8 o a 4 bits. Los caracteres que se envían al display se almacenan en la memoria RAM del módulo. Existen posiciones de memoria RAM, cuyos datos son visibles en la pantalla y otras que no son visibles, estas últimas se pueden utilizar para guardar caracteres que luego se desplazan a la zona visible.

CONTROL Y DATO	SEÑAL DE CONTROL		DATO / DIRECCIÓN								DESCRIPCIÓN	TIEMPO DE EJE
INSTRUCCIÓN	RS	RW	D7	D6	D5	D4	D3	D2	D1	D0	DESCRIPCION	
Borrar pantalla	0	0	0	0	0	0	0	0	0	1	Limpia todo el display y retorna el cursor a la posición de inicio, dirección 0	
Cursor a casa	0	0	0	0	0	0	0	0	1	-	Retorna el cursor a la posición inicio (dirección 0). También retorna el display, desplazando a la posición original. Los contenidos de la DDRAM permanecen sin cambios.	
Seleccionar modo	0	0	0	0	0	0	0	1	I/D	S	Configura la dirección de movimiento y si se desplaza o no el display. Esta operación es realizada durante operaciones de lectura escritura.	
Encender/apagar pantalla	0	0	0	0	0	0	1	D	C	B	Configura el estado ON/OFF de todo el display (D), el cursor (C) y el parpadeo del caracter en la posición del cursor.	
Desplazar Cursor / Pantalla	0	0	0	0	0	1	S/C	R/L	-	-	Mueve el cursor y desplaza el display sin cambiar los contenidos de la DDRAM.	
Activar función	0	0	0	0	1	D/L	N	F	-	-	Configura el tamaño de la interfase (DL), el número de líneas del display (N) y la fuente del carácter (F). N=0 es 1 línea. N=1 es 2 líneas.	
CG RAM	0	0	0	1	Dirección generador de RAM						Ajusta la dirección del generador de caracteres. El dato CG RAM es enviado y recibido después de este ajuste.	
DD RAM	0	0	1	Dirección de datos RAM							Ajusta la dirección de la DDRAM. La dirección es enviado y recibido después de este ajuste.	
Bandera de ocupado	0	0	BF	AC							Lectura de la bandera Busy Flag. Indicando que operaciones internas son realizadas y lectura de los contenidos del contador de direcciones.	
Escritura CG RAM/DD RAM	1	0	Escritura de dato								Escribe datos en la DDRAM o en la CGRAM	
Lectura CGRAM/DDRAM	1	1	Lectura de dato								Lectura de datos desde la DDRAM o la CGRAM	

Es importante anotar que solo se pueden mostrar caracteres ASCII de 7bits, por lo tanto algunos caracteres especiales no se pueden ver (es aconsejable tener a mano una tabla de caracteres ASCII para conocer los datos que son prohibidos). También se tiene la opción de mostrar caracteres especiales creados por el programador y almacenarlos en la memoria RAM que posee el módulo.

INTERFAZ CON MICROCONTROLADOR A 8 BITS

Esta forma de manejo es la mas sencilla de programar, pero tiene la desventaja de utilizar 8 pines del microcontrolador solo para el envío de datos y otros 2 pines para las señales de control. En principio en la mayor parte de las aplicaciones se va requerir escribir en el LCD y rara vez leer en el mismo, por lo tanto en este apunte nos dedicaremos exclusivamente a escribir en el LCD.



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

SIGNIFICADO DE LAS ABREVIATURAS	
I/D	= 1 incremental = 0 decremента
S	= 1 desplaza el mensaje en la pantalla = 0 mensaje fijo en la pantalla
D	= 1 encender (activar) la pantalla = 0 apagar la pantalla (desactivar)
C	= 1 activar cursor = 0 desactivar cursor
B	= 1 parpadea caracter señalado por el cursor = 0 no parpadea el caracter
S/c	= 1 desplaza pantalla = 0 mueve el cursor
RL	= 1 desplazamiento a la derecha = 0 desplazamiento a la izquierda
DL	= 1 datos de 8 bits = 0 datos de 4 bits
BF	= 1 durante operación interna del módulo = 0 finaliza la operación interna

Esto implica que el pin de selección de lectura/escritura (R/W) lo conectaremos siempre a tierra GND. Utilizaremos el puerto B como bus de datos (manejará los pines D0 a D7 del LCD) y el puerto A se encarga de manejar las señales de control (manejará los pines E y RS del LCD). Deberemos programar 2 subrutinas, una que llamaremos INSTRUC que será invocada cuando se quiera enviar una instrucción al módulo LCD, por ejemplo limpiar pantalla, indicar una posición de memoria, indicar si utilizaremos interfaz de 8 o 4 bits etc. Y otra subrutina que llamaremos ESCRIB que será invocada cuando se quiera escribir un dato para ser visualizado en el módulo LCD.

PUERTO SERIAL

Puerto es un concepto con múltiples usos. Su acepción más general refiere a una infraestructura que ofrece distintas clases de servicios. Dentro del contexto específico de la **informática**, puerto es la **interfaz** que permite **enviar y recibir datos digitales**. Puede tratarse de **puertos físicos** (que presentan una entrada en el hardware para que se conecte un periférico) o de **puertos virtuales** (interfaz lógica administrada mediante un programa informático).

Un **puerto serial** es aquel que **posibilita la transmisión de un bit a la vez**. Esta interfaz suele ser utilizada para la conexión del **teclado** o del **mouse (ratón)** a la **computadora (ordenador)**. En la actualidad, de todos modos, esta clase de puerto ha perdido popularidad, siendo reemplazada por el **puerto USB** u otras tecnologías más rápidas. Su principal ventaja sigue siendo el bajo costo y su sencillez.

Para comprender cómo funciona el puerto serial (que también suele denominarse **puerto serie**), puede compararse su lógica con el funcionamiento de un **puerto paralelo**. Mientras que en un puerto paralelo se transmiten muchos bits de manera simultánea y en dos direcciones diferentes, el puerto serial sólo envía o recibe de a un bit por vez.

Esto explica por qué se conoce como serial a este puerto: al considerar un **byte** de **información**, se encarga de la transmisión de esos **8 bits uno a uno**, creando el trayecto de la serie de bits.



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL



Además de todo lo expuesto, es necesario tener en cuenta otras consideraciones con respecto al puerto serial, entre las que destacan las siguientes:

- Los modelos más habituales son los llamados DB9, que se identifican por tener nueve pines.
- Hay que saber que para conectar y desconectar los dispositivos mediante ese puerto y que el ordenador en sí los reconozca es fundamental proceder a reiniciar el computador.
- Son conocidos por el nombre de RS-232.
- Se trata del primer tipo de interfaces que dieron la oportunidad de que los ordenadores se “relacionasen” con lo que era el mundo exterior.
- En lo que es la transmisión serial es imprescindible que existan dos bits de control: un bit de arranque y un bit de parada.
- Todos y cada uno de los puertos seriales, salvo excepciones muy concretas, están directamente integrados en lo que es la placa madre o placa base del ordenador en cuestión.
- Se considera que toda computadora, por regla general, tiene un mínimo de uno y un máximo de cuatro puertos seriales.
- La comunicación serial es la que se produce de manera totalmente asincrónica, es decir, sin necesidad de que tenga que existir una señal de sincronización.
- Como toda tecnología, los puertos serie o seriales también están avanzando de manera contundente. Por ese motivo, han comenzado a ir apareciendo en el mercado varios modelos que tienen la particularidad de ser de alta velocidad. Estos además son más baratos que sus antecesores y necesitan menos cableado que los anteriores, por lo que se facilita notablemente su uso.

EL PUERTO SERIE RS-232

El puerto serie RS-232 es el que se emplea en las computadoras, PC, módems, conmutadores e impresoras y tiene sus inicios en los años 60's por la EIA (Electronics Industries Association de los EE.UU), este fue creado para ofrecer una conexión entre aparatos que requieren comunicación de Datos. Durante los últimos 30 años que este estándar ha estado en uso, los equipos han evolucionado tremendamente, pero la norma inicial RS-232 ha cambiado muy poco y los pocos que se han producido han sido debidos a la interpretación propia de algunos fabricantes.

Los ordenadores se conectan con cualquier equipo periférico, a través de sus puertos paralelo o serie, o los más recientes como el USB (Universal Serial Bus, el cual deja desbancados a los otros con 12 Mb/s), pero en éste caso solo analizaremos el puerto serie RS-232 por ser un estándar impuesto en todos los equipos informáticos.

En un ordenador puede haber varios puertos serie, a los que normalmente se les denomina COM 1, COM 2, COM 3 (muchas veces los puertos serie a partir del COM 2 se denominan puertos virtuales o son debidos a ampliaciones de los puertos por tarjetas controladoras del tipo PCI), etcétera, por defecto el COM 1 suele pertenecer al ratón usando éste el IRQ 4, aunque también es posible encontrarle en el COM 2, así que lo normal es encontrarnos libre el puerto serie del COM 2 utilizando el IRQ 3.



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

El tener un puerto serie estándar internacionalmente, permite que distintos fabricantes produzcan aparatos que utilizando esa norma se interconectan entre sí, aumentando así las posibles aplicaciones y la posibilidad de conectividad entre equipos.

Para realizar una conexión en serie de datos ó información, se requiere como mínimo un cable de dos alambres, una conexión del tipo full-duplex como la de telefonía.

Si se quiere tener una comunicación bidireccional por un par de hilos y esta consiste en una serie de bits de información, se requieren otras terminales que indiquen al interface cuál de los aparatos interconectados transmite y cuál recibe, que tipo de información es, cuándo el aparato receptor esta listo para recibir, cuando el transmisor esta listo para transmitir, a que velocidad va ser la comunicación, etc., esto hace que el puerto serie tenga otras terminales que se usan para coordinar la comunicación entre los equipos.

Los equipos terminales de datos (conmutadores, PC, impresoras, etc), envían señales en 0's y 1's lógicos binarios, que el módem debe convertir a señales analógicas y enviarlas por la línea telefónica o canal de comunicación pero también es posible que se comuniquen siempre en digital.

Éste interface o puerto RS-232 trabaja entre +12 voltios y -12 voltios, de manera que un cero lógico es cuando la terminal esté entre +9 y +12 voltios, y un uno lógico cuando este entre -9 y -12 voltios de manera que un puerto serie que no esta transmitiendo, mantiene la terminal de transmisión en un 1 lógico es decir entre -9 y -12 volts.

LA ESTRUCTURA DE DATOS DEL PUERTO SERIE RS-232

La comunicación de datos en un puerto serie, se usa normalmente para efectuar comunicaciones asíncronas, osea sin tiempo preestablecido para iniciarse. Los datos llegan en paquetes de información, normalmente cada paquete es de 8 bits=1 byte, algunos equipos envían caracter por caracter, otros guardan muchos caracteres en la memoria y cuando les toca enviarlos los envian uno tras otro.

Uno de los parámetros más importantes en la comunicación serie, es la velocidad con la que los datos se transmiten, para el caso del RS-232, pueden transmitir de los 300 Baudios (1 Baudio=1 bit/seg) hasta 115,200 Baudios, la velocidad depende de los equipos conectados en el puerto serie y la calidad y longitud de los cables.

Otro de los parámetros importantes es el bit de inicio que le indica al puerto receptor que va a llegar un byte de información.

En este caso se envía un bit de inicio, una palabra de 7 bits (101 0110), que corresponde a la letra "V" en código ASCII y luego un bit de paro. La palabra puede también tener, longitud de 5, 6, 7 u 8 bits, el bit de inicio siempre es un paso de -12 volts a +12 volts, y el bit de paro queda en -12 volts.

Hay dos tipos de paridad adicional que se usan y estos son:

Marca (mark)

El bit de paridad que se intercala siempre es un uno.

Espacio (space)

El bit de paridad que se intercala siempre es un cero.

Es así como la comunicación serie RS-232 es la comunicación de Datos más empleada en el mundo, ya que utiliza pocos cables para lograrlo y mediante los módems, es la forma de intercomunicar computadoras, comunicarse a través de internet, control a distancia y muchas otras aplicaciones.

El estándar RS-232-EIA, es equivalente al V.24 del CCITT (Consultive Committee International Telegraph and Telephone), este comité es internacional y hace recomendaciones de carácter mundial.

Un circuito corto de corriente no debe exceder los 500mA. El driver debe utilizarse sin problema. (Tome importante nota de esto último)

Es interesante saber que el estandar RS232C especifica una tasa máxima de baudios de 20,000 BPS, la cual es demasiado pequeña para los estandares de hoy en día. De todas formas existe otra versión nueva más reciente, la RS-232D.



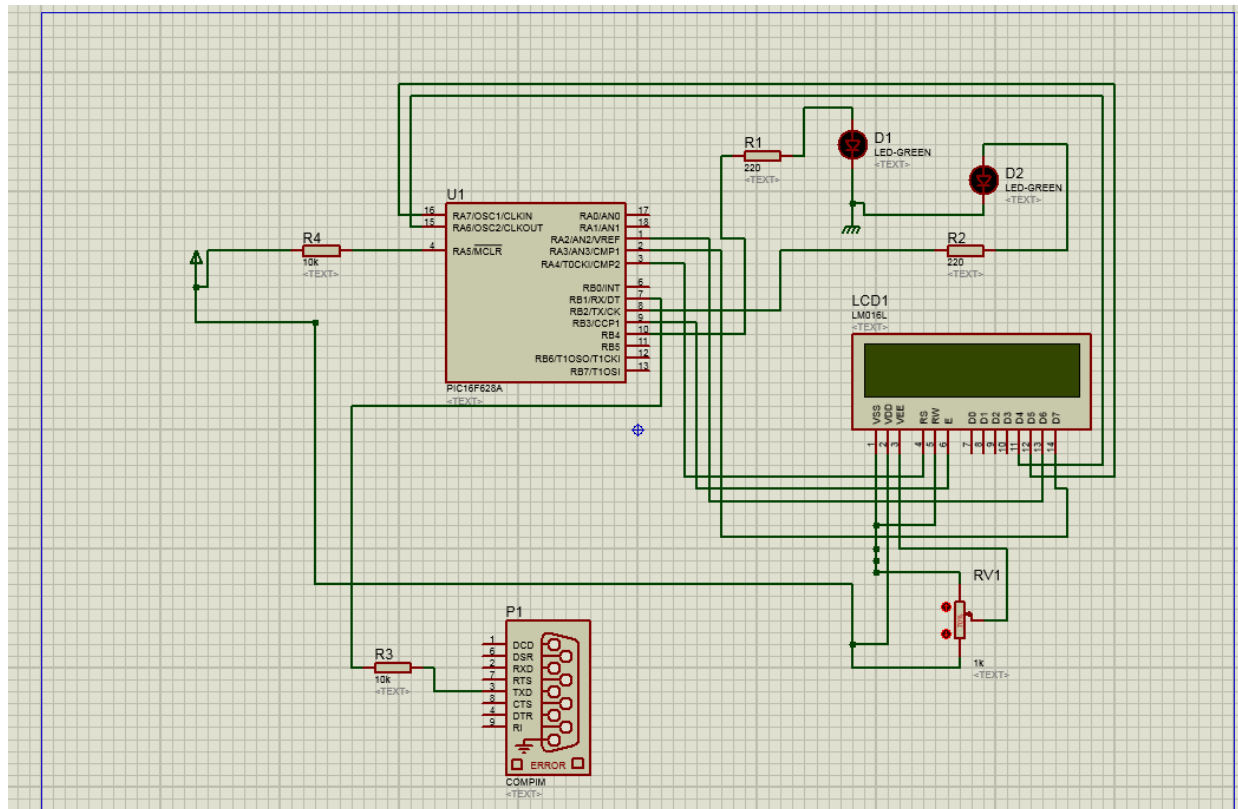
Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

5. Simulación



A continuación tenemos el diagrama del circuito que contempla el microcontrolador pic 16f628a el cual contiene las instrucciones las cuales permiten el trabajo con la pantalla lcd y con los leds que se muestran en la parte superior, estas instrucciones que están escritas en el lenguaje BASIC son las que permiten interactuar con el microcontrolador, estas instrucciones se transforman al hexadecimal y a ensamblador y se envían al dispositivo para que las interprete e interactúe con los demás dispositivos



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

```

MicroCode Studio - PICBASIC (Proyecto.pbp)
File Edit View Project Help
New Open Save Cut Copy Paste Undo Redo Print
Compile Compile Program 16F628 Read Verify Erase Information
Code Explorer
Includes
modedefs.bas
Defines
Constants
Variables
x
y
texto
dat
Alias and Modifiers
Symbols
Labels
Inicio
*****
* Name      : UNTITLED.BAS
* Author    : [select VIEW...EDITOR OPTIONS]
* Notice    : Copyright (c) 2017 [select VIEW...EDITOR OPTIONS]
*           : All Rights Reserved
* Date      : 18/1/2017
* Version   : 1.0
* Notes     :
*****
INCLUDE "modedefs.bas" ;incluyen los modos de comunicación

x VAR BYTE ;variable de almacenamiento
y VAR BYTE ;variable de almacenamiento
texto VAR BYTE ;variable de almacenamiento
dat VAR BYTE ;variable de almacenamiento
LCDOUT $fe,1, " PROYECTO FINAL " ;texto para verificar la conexión LCD
LCDOUT $FE, $C0, "INTERFACES - UTE" ;
PAUSE 4000 ;espera 4 seg.
LCDOUT $fe,1 ;limpia pantalla del Lcd

Inicio:

SERIN portb.1 ,N1200, dat ;esperar el dato y guardarlo en dat
LCDOUT, dat ;desplegar el dat en LCD

; RUTINA PRENDE Y APAGA LED AZUL

IF dat = "A" THEN

    LCDOUT $fe,1
    HIGH portb.2
    LCDOUT $fe,1, " ENCENDIDO "
    LCDOUT $FE, $C0, " LED AZUL "

endif

IF dat = "B" THEN

    LCDOUT $fe,1
    LOW portb.2

```

En la presente imagen esta la estructura del código en BASIC que se compila y se guarda en instrucciones en hexadecimal y en ensamblador, en este código se ofrecen diferentes actividades como son encender LEDs, la instrucción de polígrafo y el paso de palabras hacia el LCD, a continuación podremos encontrar el código que está en el microcontrolador.

INCLUDE "modedefs.bas" ;incluyen los modos de comunicación

x var byte ;variable de almacenamiento
y VAR BYTE ;variable de almacenamiento
texto VAR BYTE ;variable de almacenamiento
dat VAR BYTE ;variable de almacenamiento
LCDOUT \$fe,1, " PROYECTO FINAL " ;texto para verificar la conexión LCD
LCDOUT \$FE, \$C0, "INTERFACES - UTE" ;
PAUSE 4000 ;espera 4 seg.
LCDOUT \$fe,1 ;limpia pantalla del Lcd

Inicio:

SERIN portb.1 ,N1200, dat ;esperar el dato y guardarlo en dat
LCDOUT, dat ;desplegar el dat en LCD

; RUTINA PRENDE Y APAGA LED AZUL



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

If dat = "A" then

```
LCDOUT $fe,1
high portb.2
LCDOUT $fe,1, " ENCENDIDO "
LCDOUT $FE, $C0, " LED AZUL "
```

endif

If dat = "B" then

```
LCDOUT $fe,1
low portb.2
LCDOUT $fe,1, " APAGADO "
LCDOUT $FE, $C0, " LED AZUL "
```

endif

; RUTINA PRENDE Y APAGA LED ROJO

If dat = "C" then

```
LCDOUT $fe,1
high portb.4
LCDOUT $fe,1, " ENCENDIDO "
LCDOUT $FE, $C0, " LED ROJO "
```

endif

If dat = "D" then

```
LCDOUT $fe,1
LOW portb.4
LCDOUT $fe,1, " APAGADO "
LCDOUT $FE, $C0, " LED ROJO "
```

endif

; RUTINA PRENDE Y APAGA DOS LEDS

If dat = "E" then

```
LCDOUT $fe,1
```

```
FOR Y=1 TO 20
```

```
high portb.2
```

```
PAUSE 300
```

```
LOW portb.2
```

```
high portb.4
```

```
PAUSE 300
```

```
LOW portb.4
```



Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

NEXT

high portb.2
high portb.4

LCDOUT \$fe,1, " ENCENDIDO "
LCDOUT \$FE, \$C0, "LED ROJO Y AZUL"

endif

If dat = "F" then

LCDOUT \$fe,1
LOW portb.2
LOW portb.4
LCDOUT \$fe,1, " APAGADO "
LCDOUT \$FE, \$C0, "LED ROJO Y AZUL"

endif

; RUTINA ENVIA Y BORRA TEXTO DEL LCD

If texto = " " then

LCDOUT \$fe,1, texto;

endif

If dat = "H" then

LCDOUT \$fe,1
high portb.2
high portb.4
PAUSE 300
LOW portb.2
LOW portb.4

endif

GOTO inicio

END

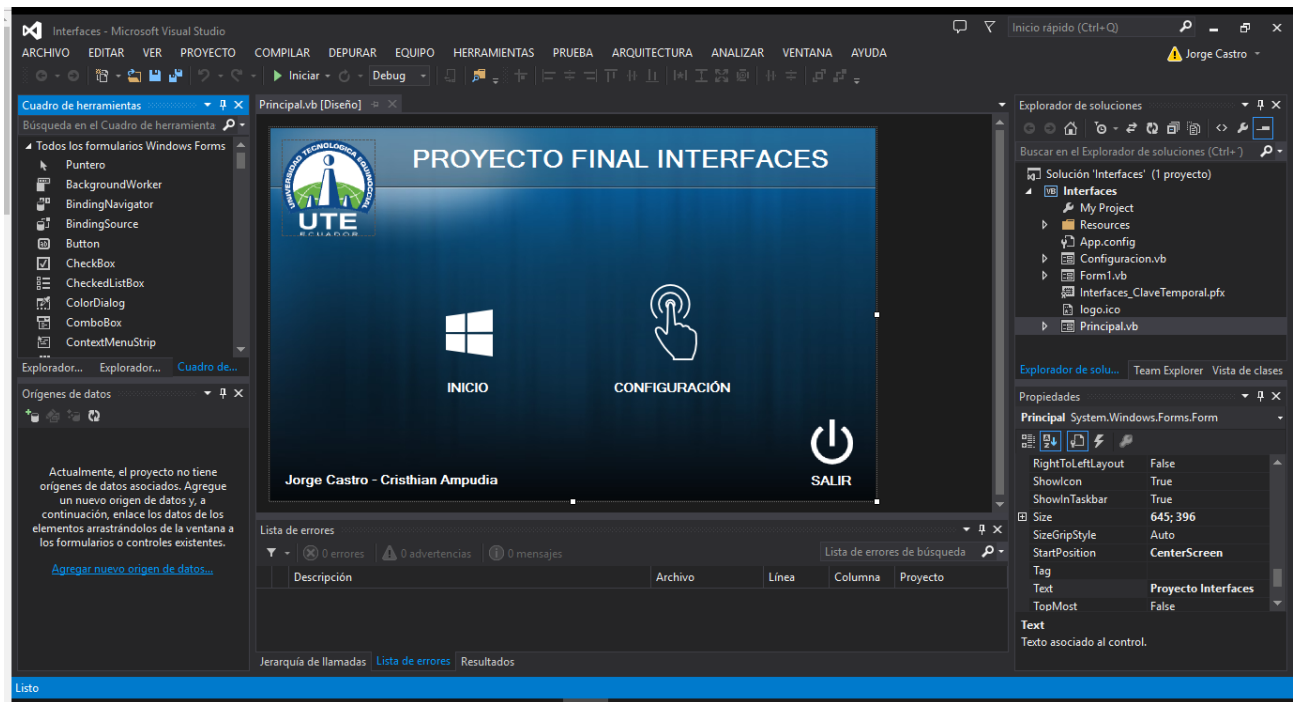


Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL



Como podemos observar tenemos la pantalla principal del sistema, aquí tenemos en el entorno de desarrollo como se ve la herramienta para el funcionamiento de la misma.

Configuracion

Buscar Puertos

COM1

Conectar

Para poder ejecutar el sistema, primero deberemos acceder a la configuración, donde buscaremos y habilitaremos los puertos para la comunicación, de no ser así no podremos usar el sistema como lo mostraremos a continuación:

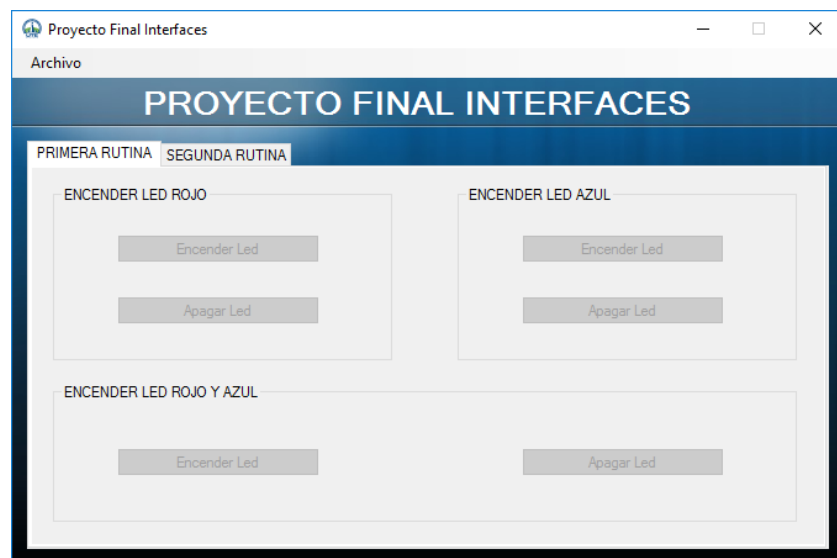


Cristhian Ampudia, Jorge Castro

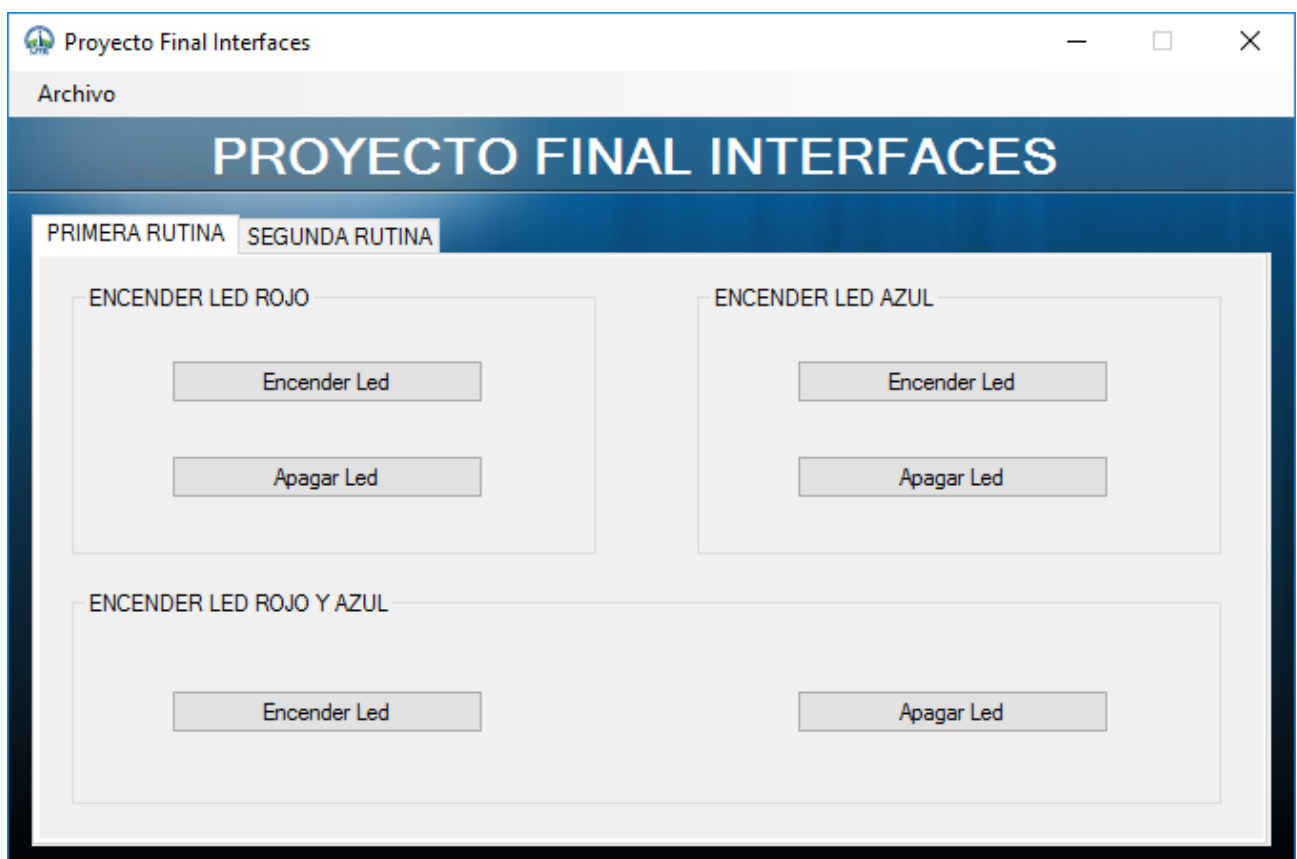
Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL



Aquí tenemos como aparece cuando no se configúralos puertos de acceso, las opciones se deshabilita y no permite mandar las instrucciones aunque permite ver que opciones se dispone para trabajar, las cuales funcionarían cuando estén habilitados los puertos.





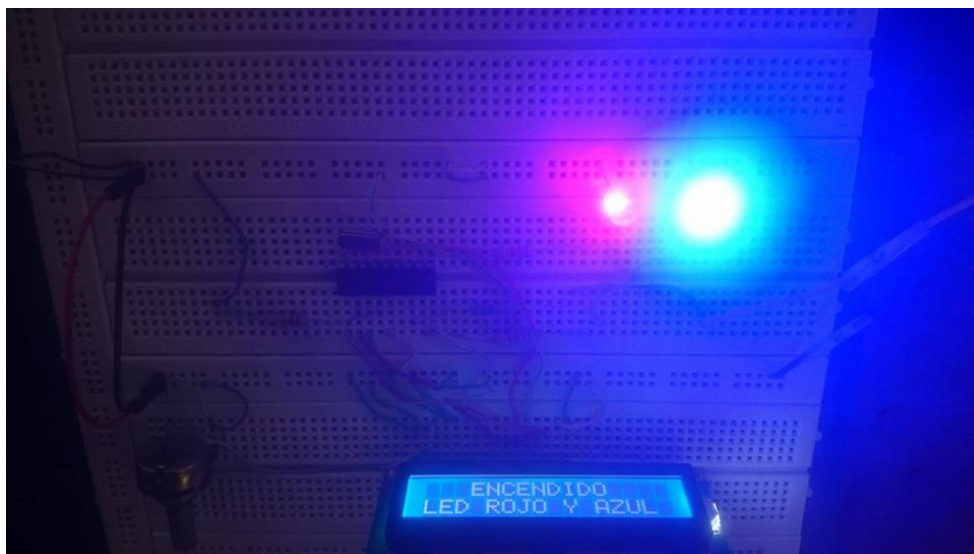
Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL

Aquí tenemos las opciones habilitadas gracias al acceder a la configuración y escogerlos puertos, como podemos apreciar tenemos diversas instrucciones, tenemos el encendido y apagado de los leds, y el activado de la rutina del policía.



Aquí tenemos la habilitación de los leds después de mandar la instrucción para encender ambos leds, incluso se proyecta un mensaje para mostrar que se han encendido ambos leds.



Aquí tenemos el funcionamiento del accionar en el botón para encender el led azul.

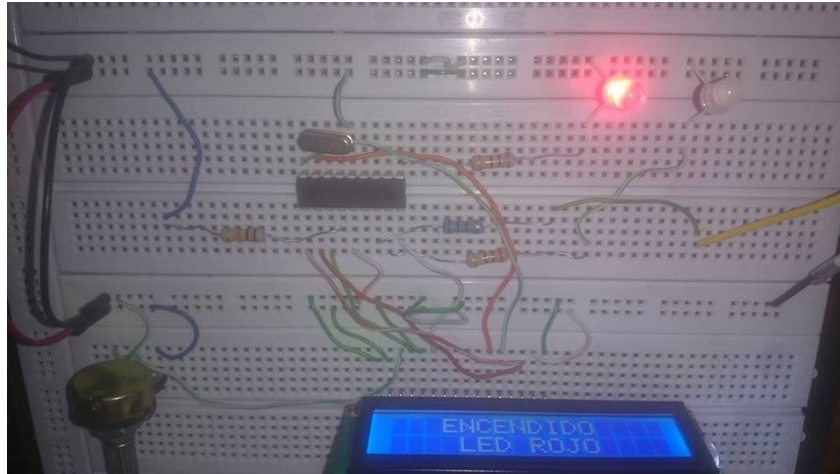


Cristhian Ampudia, Jorge Castro

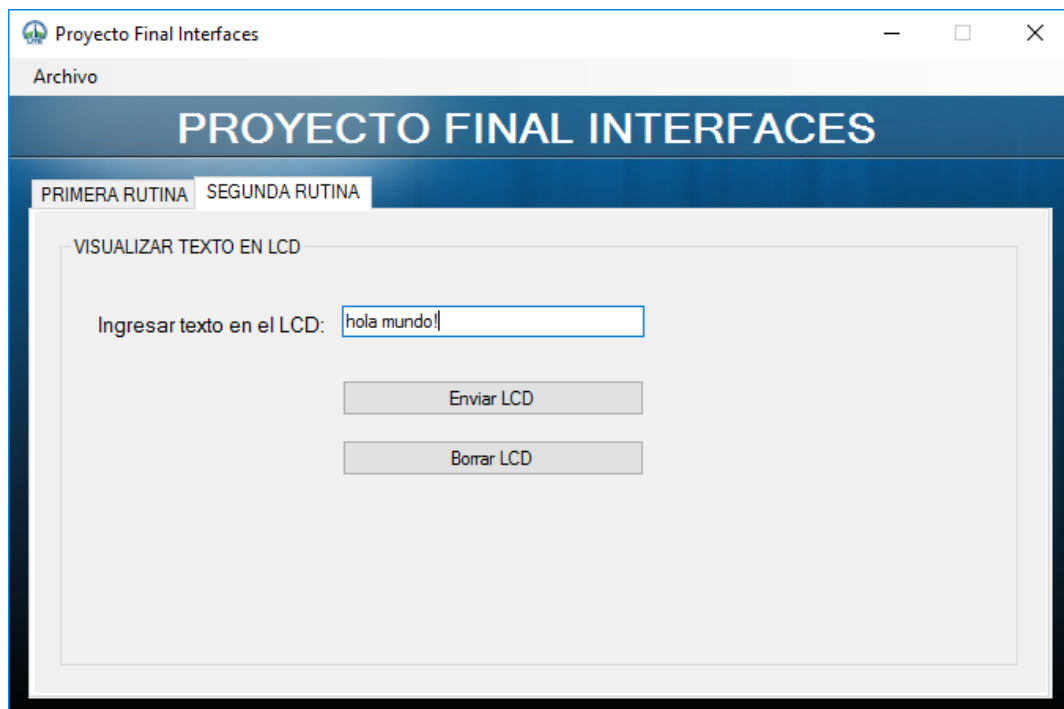
Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL



En cambio aquí tenemos el encendido del led rojo tras ejecutar la acción den encender el led dentro del panel.



Aquí podemos ver la segunda opción donde mandamos las frases al LCD, como podemos proyectar y borrar el mensaje en pantalla.

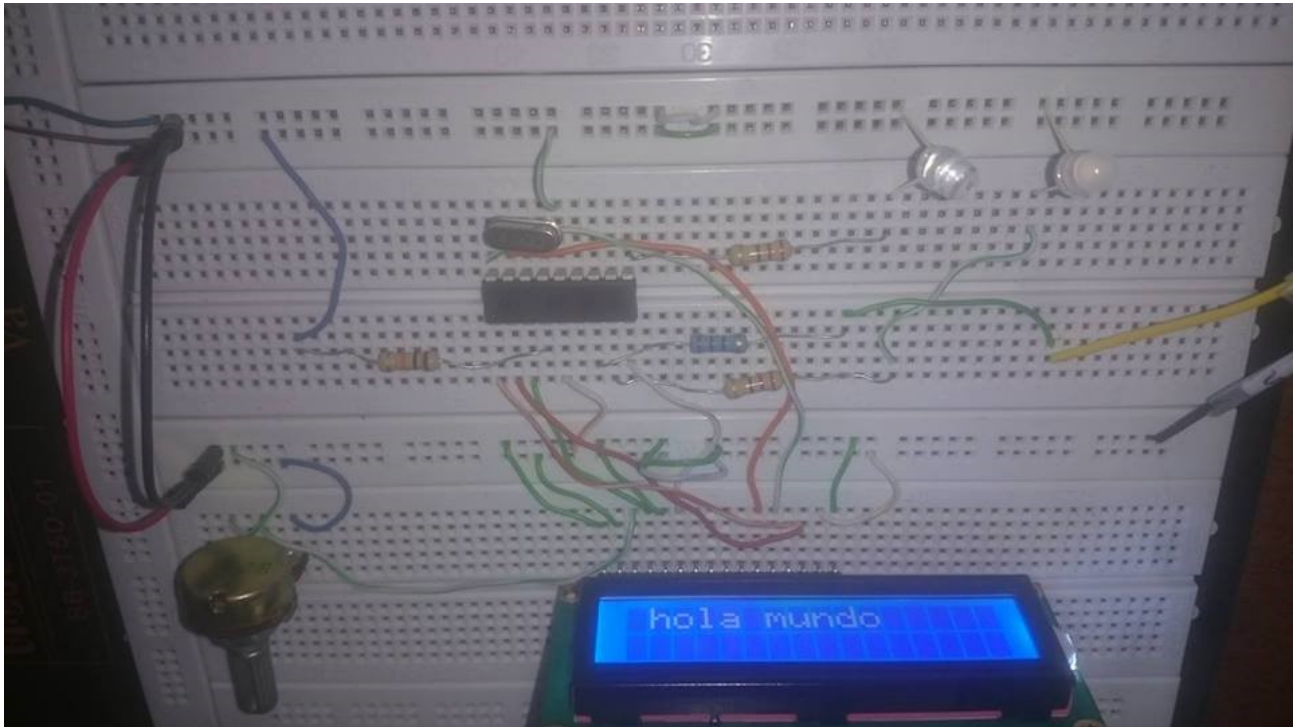


Cristhian Ampudia, Jorge Castro

Fecha de entrega:

25 de Enero del 2017

PROYECTO FINAL



Como se muestra aquí esta el mensaje que se mando desde el sistema hacia el circuito y posteriormente a la pantalla.

6. Cálculos Teóricos

7. Recolección de la información y Resultados (Observaciones)

Es importante recalcar que en un inicio resulto complejo el paso de información entre dispositivos, lo que se puede aclarar que es importante la instalación de los driver para que reconozca el cable comunicación de usb a serial, otra de las cosas importante de mencionar Es la facilidad de la herramienta que nos permite conectar directamente del sistema al circuito por que incluye ciertos módulos que resulta mas complejo en otras herramientas.

8. Análisis de Resultados

9. Conclusiones

En el proceso de desarrollo de la interfaz resulto muy satisfactorio los resultados conseguido ya que aunque presento cierta complicación en la transmisión de datos, sin embargo al momento de la investigación se soluciono el inconveniente y dio como resultado un trabajo gratificante porque el desarrollo de la plataforma y la comunicación con el circuito lo convirtió en un proyecto que abre puertas a otro tipo de tecnologías y que muestra que se puede hacer cosas aún más complejas.