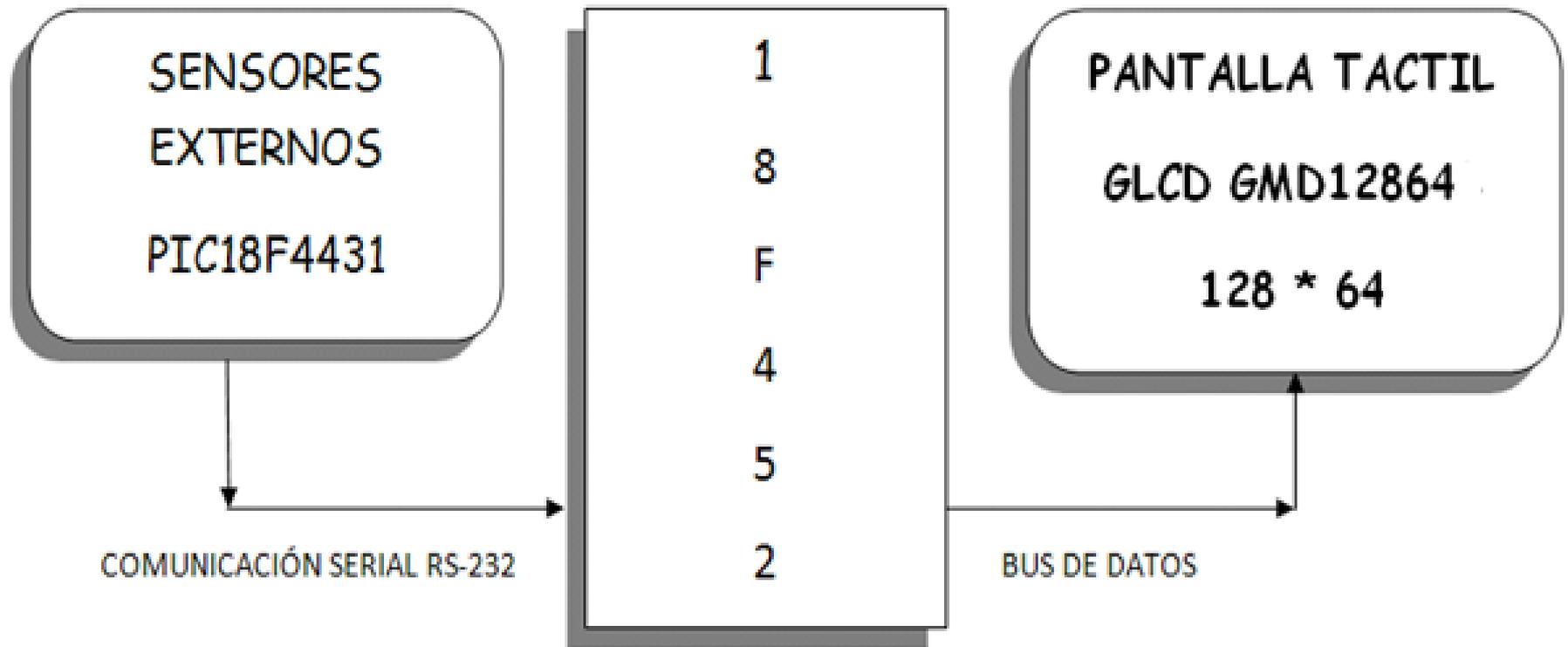


“Interfaz gráfica con pantalla LCD para analizar señales de sensores externos desarrollada con microcontroladores avanzados y capacidad de comunicación serial”

Giovanny Tapia
Tyrone Cazar

Diagrama de Bloques



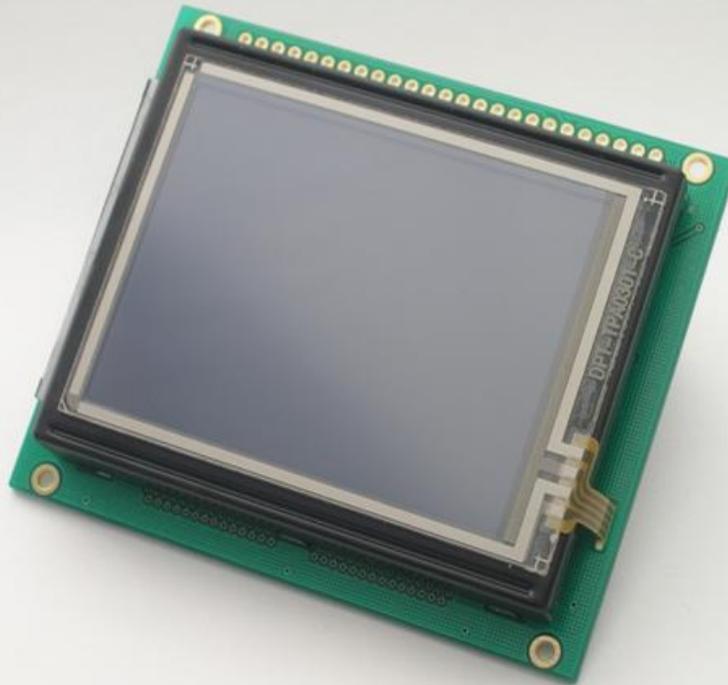
Objetivos del proyecto

Adquisición de datos por diferentes tipos de sensores externos, transmitidos mediante comunicación serial y presentados a través de la pantalla GLCD con panel táctil, a su vez diseñamos una variedad de menús que darán facilidad para manejar de una forma directa y sencilla la elección de cada proyecto a través del touch panel.

GLCD + Panel táctil

Pantalla táctil

Top > LCD Products > 128x64Q ALL-IN-ONE TOUCHSCREEN GLCD With RGB LED BACKLIGHT



**128x64Q ALL-
IN-ONE
TOUCHSCREEN
GLCD With
RGB LED
BACKLIGHT**

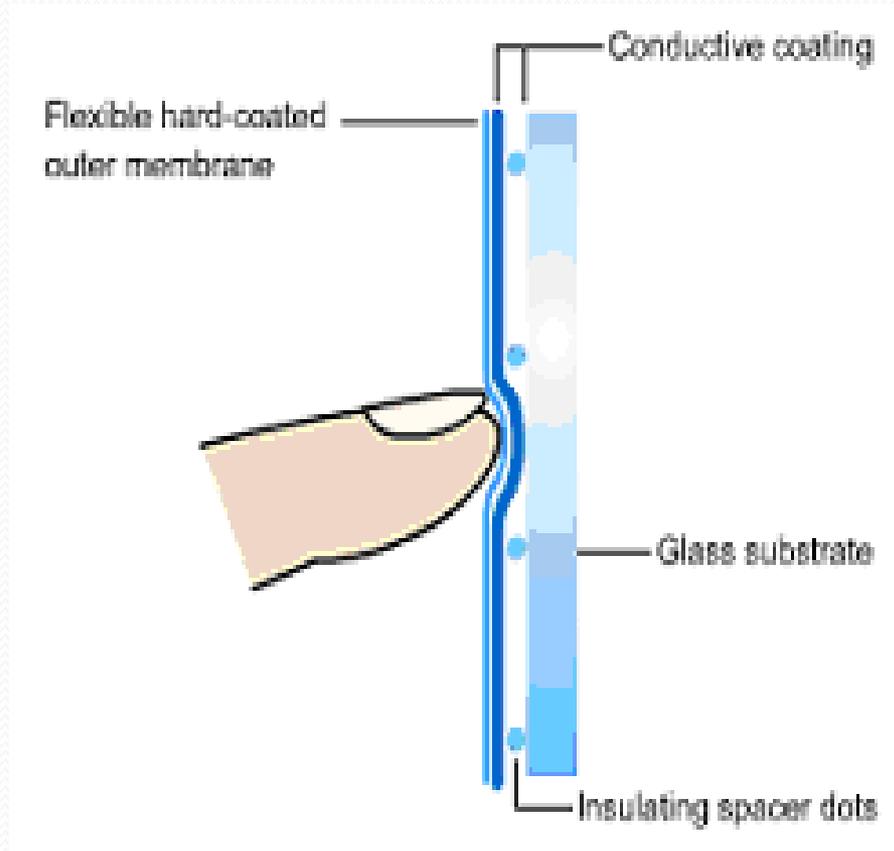
10+: \$34.50
25+: \$33.00
50+: \$31.00
100+: \$28.50
250+: CALL

SKU: 12864Q/TS/RGB
PRICE: **36.00**

128x64Q ALL-IN-ONE TOUCHSCREEN GLCD With RGB LED BACKLIGHT Description

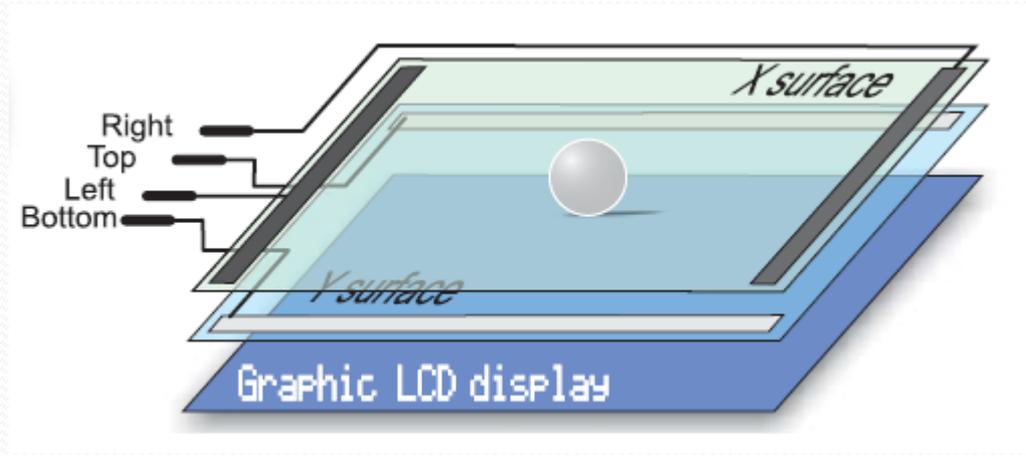
Panel táctil resistivo

Un panel táctil resistivo está compuesto por dos láminas rígidas transparentes.



Principio de funcionamiento

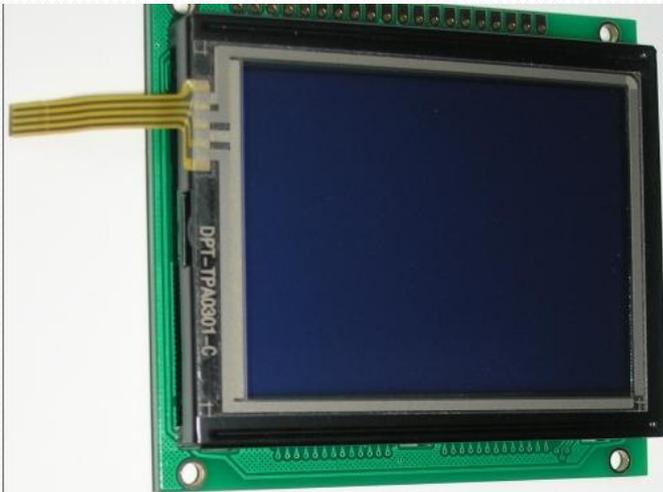
Primero determinar la coordenada en X después segundo la coordenada en Y.



Pantallas similares

CircuitED *Specializing in Electronic Development Products*

Home | Directory | My Account | TechComments | Contact | My Basket



128x64 BL/WH
TOUCHSCREEN
GLCD

SKU: 12864B/TS/TL
PRICE: 26.00

128x64 BL/WH TOUCHSCREEN GLCD Description

If you intend to use the mikroE builtin development board touchpanel controller you should first consider these 2 items:

- **128x64 GLCD with Soldered Connector**
- **TOUCHPANEL OVERLAY For 128x64 GLCD**

Top > LCD Products > **128x64 BK/YG Development Board GLCD w/Soldered Connector**

128x64
BK/YG
Development
Board GLCD
W/Soldered
Connector

SKU: 12864BNLY
PRICE: 20.00

128x64 BK/YG Development Board GLCD w/Soldered Connector Description

This 128x64 **KS0108** Graphical LCD is designed as a "Ready-To-Go" direct plug-in connection to mikroE development boards via a 20-pin pre-soldered Female SIP connector. Easy to connect and ready to use code

Inicialización de variables

```
dim GLCD_DataPort as byte at PORTD
```

```
dim GLCD_CS1 as sbit at LATB0_bit ' for writing to output pin always use latch (PIC18 family)  
GLCD_CS2 as sbit at LATB1_bit ' for writing to output pin always use latch (PIC18 family)  
GLCD_RS as sbit at LATB2_bit ' for writing to output pin always use latch (PIC18 family)  
GLCD_RW as sbit at LATB3_bit ' for writing to output pin always use latch (PIC18 family)  
GLCD_EN as sbit at LATB4_bit ' for writing to output pin always use latch (PIC18 family)  
GLCD_RST as sbit at LATB5_bit ' for writing to output pin always use latch (PIC18 family)
```

```
dim GLCD_CS1_Direction as sbit at TRISB0_bit  
GLCD_CS2_Direction as sbit at TRISB1_bit  
GLCD_RS_Direction as sbit at TRISB2_bit  
GLCD_RW_Direction as sbit at TRISB3_bit  
GLCD_EN_Direction as sbit at TRISB4_bit  
GLCD_RST_Direction as sbit at TRISB5_bit
```

```
dim DRIVE_A as bit ' sirve de salida digital para la pantalla tactil  
dim DRIVE_B as bit ' sirve de salida digital para la pantalla tactil  
dim i as byte ' contador
```

```
dim adc_rd1 as word ' variable donde se adquiere la coordenada X del panel  
dim adc_rd2 as word ' variable donde se adquiere la coordenada Y del panel  
dim muestras as byte ' variable usada para el conteo de las muestras a tomar
```

Inicialización de variables

```
dim trayectoria as byte      ' variables usadas para ingresar a los diferentes menus
dim opcion_menu as byte
dim conversion as byte
dim tipo_grafico as byte

dim x as word               ' variables donde se almacenan las coordenadas en X y Y para
dim y as word               ' verificar donde esta tocando el usuario en la pantalla tactil
dim x_coord128 as word
dim y_coord64 as word

dim y1 as byte             ' variables usadas para graficar en los diferentes ejes
dim y2 as byte
dim x1 as byte
dim x2 as byte

dim coordenadaX as byte
dim coordenadaY as byte
dim grados as byte

dim bandera as byte        ' banderas que me mantienen dentro de un menu especifico
dim bandera1 as byte      ' sirven para la presentacion
dim bandera2 as byte
dim bandera3 as byte
```

Subrutinas

```
'FUNCION QUE GENERA LA COORDENADA EN X DEL TOUCH PANEL
sub function GetX() as word

portb.6= 1           ' DRIVEA = 1 (LEFT drive on, RIGHT drive on, TOP drive off )
portb.7 = 0         ' DRIVEB = 0 (BOTTOM drive off )
Delay_ms(10)
adc_rd1 = adc_read(0) ' get ADC value from 2nd channel
x = adc_rd1

tmp4 = (x *127) / 1023

result = tmp4
end sub
'FUNCION QUE GENERA LA COORDENADA EN Y DEL TOUCH PANEL
sub function GetY() as word

portb.6 = 0           ' DRIVEA = 0 (LEFT drive off , RIGHT drive off , TOP drive on)
portb.7= 1           ' DRIVEB = 1 (BOTTOM drive on)
Delay_ms(10)
adc_rd2 = adc_read(1)
y = adc_rd2
|
tmp5 = 63 - ((y *63) / 1023)

result = tmp5
end sub
```

Subrutinas

```
' FUNCION ENCARGADA DE ELEGIR LOS PROYECTOS
]sub function ELEGIR_PROYECTO(dim x_coord as word, dim y_coord as word) as byte
]
]  if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=5) and (y_coord <=21))) then
]    result = 1    ' si el usuario desea elegir acelerometro
]  else
]    if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=24) and (y_coord <= 40))) then
]      result = 2    ' si el usuario desea elegir brujula
]    else
]      if (((x_coord >=28) and (x_coord <= 98))and ((y_coord >=43) and (y_coord <= 59))) then
]        result = 3    ' si el usuario desea elegir GPS
]      else
]        if (((x_coord >=5) and (x_coord <= 22))and ((y_coord >=44) and (y_coord <= 58))) then
]          result = 4    ' si el usuario desea salir
]        else
]          if (((x_coord >=103) and (x_coord <= 120))and ((y_coord >=44) and (y_coord <= 58))) then
]            result = 5    ' si el usuario desea elegir otro programa
]          else
]            result =6    ' si el usuario desea escoge cualquier otra opcion fuera de los rangos ;
]          end if
]        end if
]      end if
]    end if
]  end if
]end if
]end sub
```

Configuración de puertos de I/O

```
ADCON1 = ADCON1 or 0x05      ' Configure AN pins as digital
'CMCON = CMCON or 7         ' Turn off comparators
TRISA = 0xFF
TRISB = 0
TRISD = 0
Glcd_Init()                 ' Initialize Glcd
Glcd_Fill(0x00)             ' Clear Glcd
inicializar()
Glcd_Image (@tortuga_bmp)
|
while (banderaT = 0)
    x_coord128 = GetX()      ' obtiene la coordenada en X
    y_coord64 = GetY()      ' obtiene la coordenada en Y
    if ((x_coord128 >=0) and (x_coord128 <= 128)) and ((y_coord64 >=0) and (y_coord64 <=63)) then
        LIMPIAR()
        banderaT = 1
    else
        banderaT = 0
    end if
wend |

Glcd_Image (@PRESENTACION_bmp)

while (banderaP = 0)
```

```

while (bandera1 = 0)
  Glcd_Image(@proyectos_bmp)
  Move_Espera()
  x_coord128 = GetX()      ' obtiene la coordenada en X
  y_coord64 = GetY()      ' obtiene la coordenada en Y
  opcion_menu = ELEGIR_PROYECTO(x_coord128,y_coord64)  ' obtengo el proyecto
  ' OPCION_MENU = 1      ACELEROMETRO
  ' OPCION_MENU = 2      DISTANCIA
  ' OPCION_MENU = 3      BRUJULA
  ' OPCION_MENU = 4      SALIR
  ' OPCION_MENU = 5      OTRO MENU DE PROYECTOS
  ' OPCION_MENU = 6      MANTENGO EL LAZO
  if (opcion_menu = 1)  then  ' ACELEROMETRO
    while (bandera2 = 0)
      Glcd_Image(@menu_bmp)      ' trayectorias
      Move_Espera()
      x_coord128 = GetX()      ' obtiene la coordenada en X
      y_coord64 = GetY()      ' obtiene la coordenada en Y
      trayectoria = ELEGIR_TRAYECTORIA(x_coord128,y_coord64)
      ' si trayectoria es 1 voy al menu de las primeras muestras
      ' si trayectoria es 2 voy al menu de las segundas muestras
      ' si trayectoria es 3 regreso al menu anterior pa elegir otro proyecto
      ' si trayectoria es 4 me mantengo en el lazo hasta que presione correctamente
      if (trayectoria = 1)  then      'trayectoria 1
        while (bandera3 = 0)
          Glcd_Image(@graficos_bmp)  |
          Move_Espera()
          x_coord128 = GetX()      ' obtiene la coordenada en X
          y_coord64 = GetY()      ' obtiene la coordenada en Y

```

```

if (trayectoria = 1) then           'trayectoria 1
  while (bandera7 = 0)
    Glcd_Image(@graficos_bmp)
    x_coord128 = GetX()           'obtiene la coordenada en X
    y_coord64 = GetY()           'obtiene la coordenada en Y
    tipo_grafico = ELEGIR_GRAFICO(x_coord128,y_coord64)
    if (tipo_grafico = 1) then     'eligio grafico X vs Y
      Glcd_Image(@brujula_bmp)
      UART1_INIT(9600)
      Delay_ms(1000)
      muestras=0
      bit_inicio = 0x15
      while (muestras < 8)
        while(bit_confirmacion<>0x20)
          UART1_Write(bit_inicio)
          bit_confirmacion = uart1_read()
          Move_Delay()
        wend
        while (UART1_Data_Ready() = 0)
          .....
        wend
        coordenadaX = UART1_Read()
        while (UART1_Data_Ready() = 0)
          .....
        wend
        coordenadaY = UART1_Read()

      Glcd_Dot(coordenadaX, coordenadaY, 1)

```

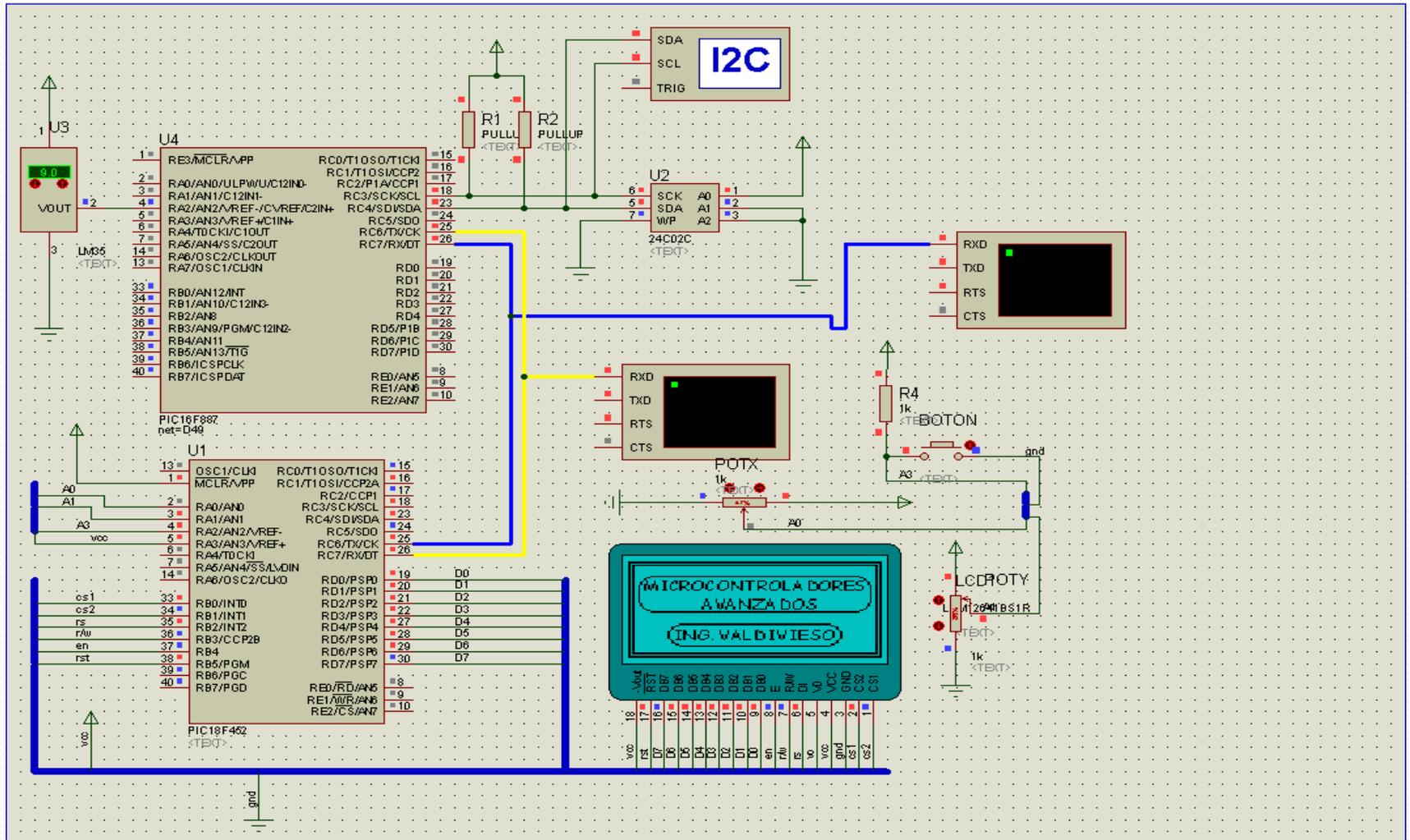
```

if (tipo_grafico = 2) then ' eligio grafico ANGULO vs TIEMPO
LIMPIAR()
Glcd_Image(@angulo_bmp)
UART1_INIT(9600)
Delay_ms(1000)
muestras=0
x1 = 16
y1 = 54
bit_inicio = 0x40
while (muestras < 8)
    while (bit_confirmacion<>0x30)
        UART1_Write(bit_inicio)
        bit_confirmacion = uart1_read()
        Move_Delay()
    wend

    while (UART1_Data_Ready() = 0)
    wend
    angulo = UART1_Read()
    grados = (angulo * 48)/255
    x2 = x1 + 12
    y2 = 54 - grados
    Glcd_Line(x1 ,y1 ,x2, y2, 1)
    x1 = x2
    y1 = y2
    muestras= muestras +1
wend

```

Simulación



Simulación

The image displays a simulation environment with two Virtual Terminal windows on the left and a large diagram on the right.

The top Virtual Terminal window shows the text: `00 01 02`

The bottom Virtual Terminal window shows the text: `14 10 E2 0A`

The main diagram on the right is a large cyan oval with a black border. It is divided into four quadrants by a vertical line and a horizontal line. The quadrants are labeled with letters: 'N' at the top, 'S' at the bottom, 'E' on the left, and 'C' on the right. Above the oval, the text 'A0' is visible. The diagram is set against a light blue background with a grid pattern.

Conclusiones

- Mediante nuestro proyecto hemos logrado implementar de una manera sencilla una interfaz gráfica que tiene una amplia gama de aplicaciones, permitiendo visualizar los datos obtenidos en cualquier lugar que queramos tomar muestras de campo.

Conclusiones

- Para la elaboración de nuestra interfaz gráfica usamos el PIC18F452 ,debido a que este posee más cantidad de memoria que es indispensable para visualizar las imágenes y menús que son parte de la presentación de la materia de graduación de la ESPOL, además que nuestro código fuente se encarga de graficar datos de 5 proyectos los cuales demandan mucha memoria (2000 líneas)

Conclusiones

- Para comunicarnos con los demás proyectos usamos la comunicación serial RS-232, al ser un tipo de comunicación asíncrona tuvimos que enviar parámetros para iniciar la transmisión, es decir que somos los encargados de definir la sincronización para el envío y recepción de datos a la interfaz gráfica.

Conclusiones

- Podemos concluir mediante pruebas experimentales que la configuración de los pines X y Y del panel táctil deben conectarse de una forma específica, ya que sino su funcionamiento seria erróneo.

Recomendaciones

- Revisar y entender el manual de especificaciones de la pantalla táctil para su buen funcionamiento y de esta manera no cometer errores en la conexión de sus pines sobre todo en los de las coordenadas X y Y, también en los pines de control de contraste y colores.
- En la utilización de los transistores para el convertidor A/D, verificar que todos estos sean del mismo fabricante porque sino la ganancia varía y el convertidor no funcionaria y a su vez el panel táctil tampoco.

Recomendaciones

- Una recomendación que cabe recalcar es de leer detenidamente el datasheet del PIC18F452, para revisar la configuración de sus pines, y así evitar cometer errores en las conexiones, logrando la prevención de quemar algún componente del proyecto.