

Using Touch Screen LCDs In EMBEDDED Applications

Professor Dr Dogan Ibrahim of the Near East University, Cyprus, describes the basic principles and types of touch screen LCDs and gives an example to show how they can be used in microcontroller-based systems

IF YOU ARE PLANNING of replacing your mobile phone the chances are that you will get one with a touch screen. More and more consumer electronic products are now available with touch screen inputs. For example, electronic games, MP3 players, GPS receivers, mobile phones, PDAs, ATM machines, industrial control systems, remote control devices, point-of-sale (POS) terminals, advertisement show screens, information displays and many more similar products offer special screens where items are selected from a menu by simply touching the relevant part of the screen.

Perhaps the biggest advantage of a touch screen display is that it eliminates the need for a keyboard input, resulting in a cheaper and a lighter overall design. The user input facilities in such devices are usually provided in the form of soft keypads where the layout of a keypad is displayed on a touch screen panel, and required characters and numbers can be entered by simply touching the required key positions on the touch

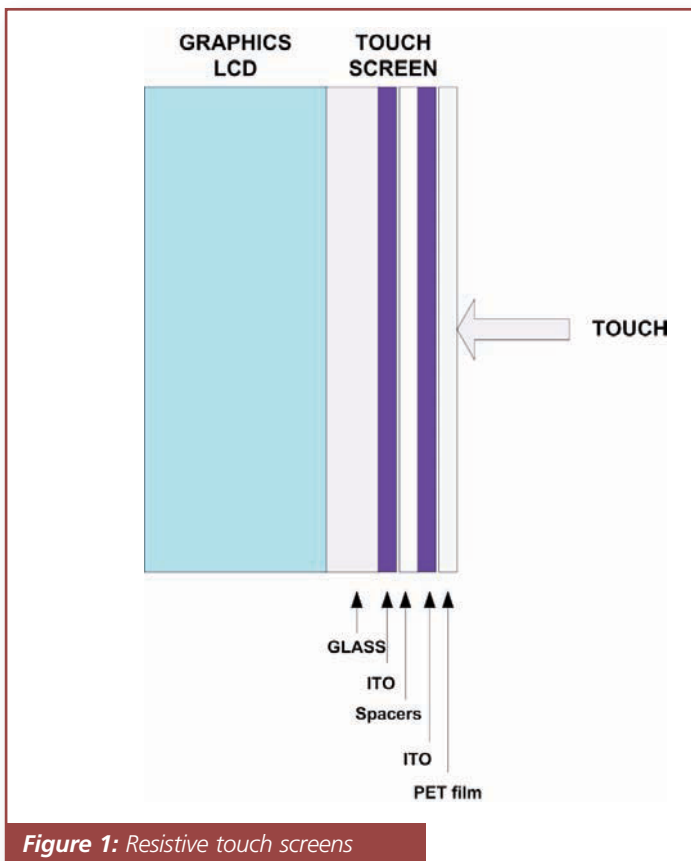


Figure 1: Resistive touch screens

screen. Soft keypad also makes it easier to enter and edit data quickly.

Another advantage of a touch screen display is that it is usually much quicker to navigate around the screen than using a keyboard or a mouse type inputs. Also, in some applications, such as GPS mapping and navigation, a desired geographical point can easily be selected by simply touching the desired point on the screen. It may take more time and effort to accurately select a point on a map using a keyboard and a mouse.

Touch screen displays are also used in most POS systems, for example in restaurants and in supermarket check-outs to select a purchased item from a menu quickly, easily and reliably. Perhaps the biggest advantage in such applications is the speed of making a correct selection.

One of the biggest disadvantages of touch screens is that the screen may get dirty and oily, and fingerprints can be accumulated on the screens after long usage by the finger and as a result, it may become less sensitive to a touch. Also, the screen can easily become scratched, especially if a hard object is used to touch and navigate through the screen. Touch screens can also cause stress on human fingers when used for more than a few minutes at a time, since pressure is required to make a selection. A touch device (e.g. a stylus) or fingernails can be used to prevent issues of direct touch. Another disadvantage is that a touch screen LCD display is usually more expensive than a standard LCD display. The choice of whether or not to use a touch screen display depends entirely on the nature of the application, the cost and the level of user experience.

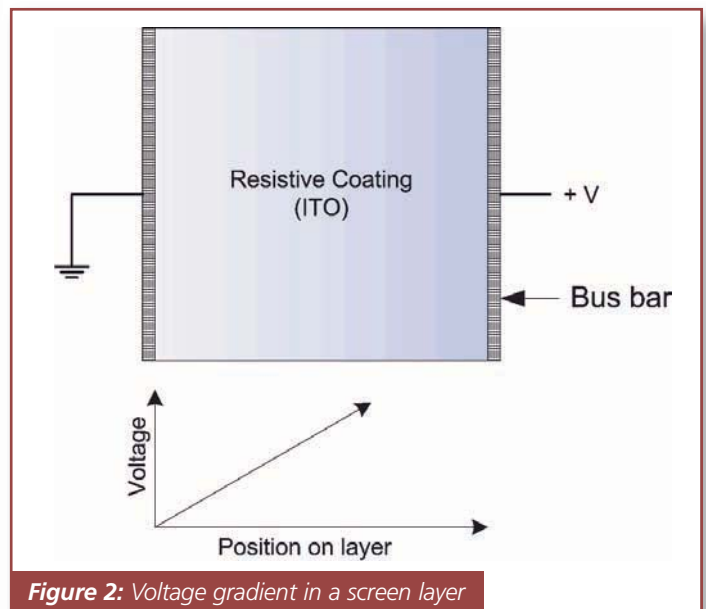


Figure 2: Voltage gradient in a screen layer

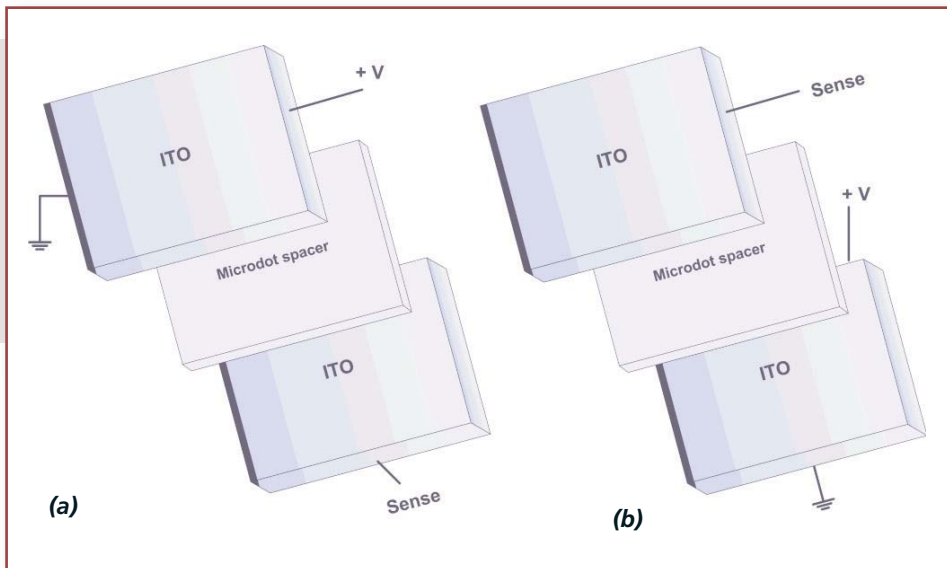


Figure 3: Determining the X and Y co-ordinates (4-wire)
 (a) Determining the X co-ordinate; (b) Determining the Y co-ordinate

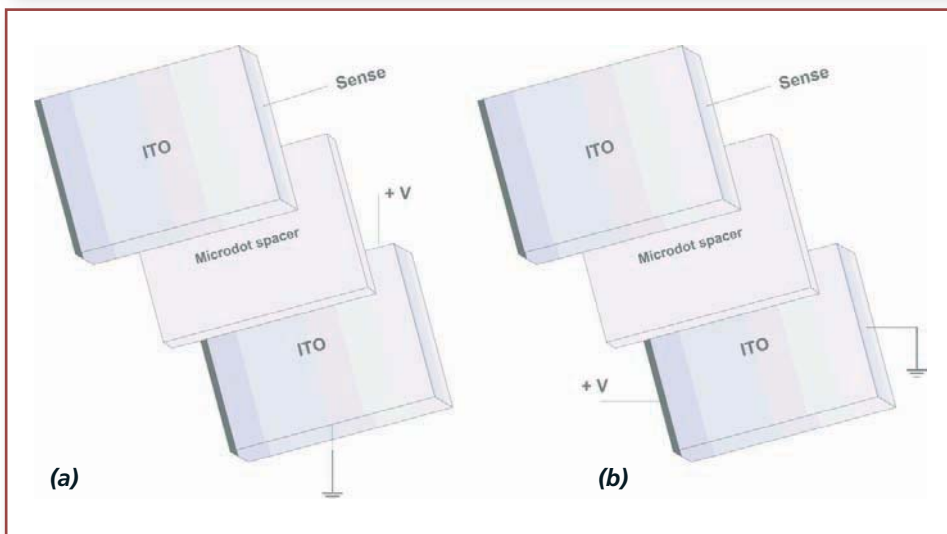


Figure 4: Determining the X and Y co-ordinates (5-wire)
 (a) Determining the X co-ordinate; (b) Determining the Y co-ordinate

Types of Touch Screen Displays

Touch screen displays are in the form of either large screen monitors, such as those used in PC systems, or small LCDs, used in microcontroller-based systems. Although the principle of operation is the same in either case, in this article the small LCD type touch screen displays are considered. Such displays usually have resolutions of 128x64 pixels and are used in battery-operated intelligent devices.

A touch screen LCD is basically a combination of a graphics LCD (GLCD) and a touch sensitive panel mounted on top of the GLCD. The two parts are independent of each other: The panel senses the co-ordinates where the user touched and the GLCD displays graphical information on the LCD display based upon user's selection.

There are several types of touch screen LCDs, such as resistive, capacitive, surface acoustic wave, optical imaging, strain gauge and so on. The most commonly used types are the resistive and capacitive ones and some information about each type is given below.

A resistive touch screen consists of several layers, where two electrically conductive resistive layers are separated by a very small gap and a flexible layer is used at the top. One of the layers is connected to a voltage source.

When a point is pressed on the screen, the touched points of both conductive layers make a contact and if the voltage is read at the other layer this voltage will be proportional to the position of the point touched because of the voltage dividing effect. Further details about resistive touch screens are given later.

A capacitive touch screen panel is coated with a material that stores electrical charges. When the panel is touched, a small amount of charge is drawn to the point of contact and the charge is measured at each corner of the panel and is then processed to determine the point touched.

Resistive touch screens have the advantages that the screen responds when touched with any kind of object, e.g. finger, stylus, nail, etc. On the other hand, the capacitive screens respond only when touched by a naked finger (but they will not respond when touched with an object or if wearing gloves for example). On the other hand, capacitive touch screens are lower power devices, have higher granularities and also provide higher clarity.

In this article the resistive touch screen is used in a practical microcontroller-based application and further information about resistive touch screens is given in the next section.

Resistive Touch Screens

Resistive touch screens are used in most low cost, medium resolution systems. A resistive touch screen consists of at least three layers. As shown in **Figure 1**, the touch screen is mounted on a graphics LCD (GLCD). The bottom layer is a glass (or acryl), coated with a resistive Indium Tin Oxide (ITO) solution. On top of this, a resistive ITO coated poly Ethylene Terephthalate (PET) flexible film is used. The two conductive ITO layers are separated from each other with microdot spacers so that there is no contact between them when the screen is not touched.

When a pressure is applied to the top of the screen, e.g. by touching the screen, the two ITO layers will make contact at the point of the touch. Electrical circuits are then used to determine the point of the contact. Usually a 4-wire, 5-wire, or an 8-wire circuit is used to determine the co-ordinates of the point touched by the user. These circuits are described below in greater detail.

4-Wire Resistive Touch Screen

These are the least expensive and most commonly used types of resistive touch screens. Conductive bus bars with silver ink are implanted at the

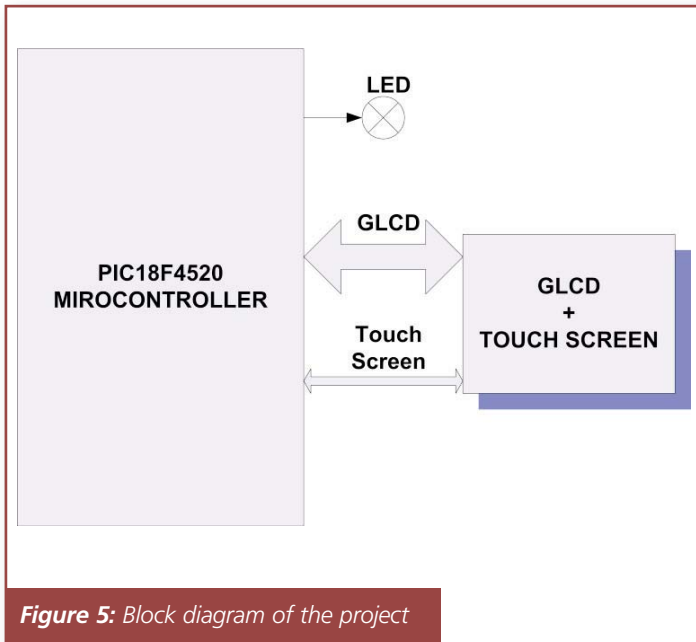


Figure 5: Block diagram of the project

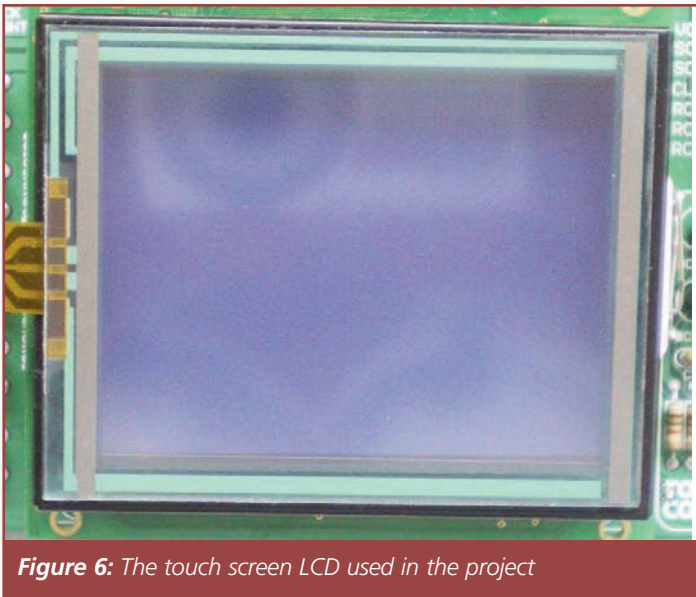


Figure 6: The touch screen LCD used in the project



Figure 7: Data shown on the GLCD

opposite edges of a screen layer. The principle of operation is such that, as shown in **Figure 2**, if one side of a layer is connected to +V and the other side to ground, a potential gradient results on the screen layer, and the voltage at any point on this layer becomes directly proportional to the distance from the +V side.

In a 4-wire touch screen two measurements are made one after the other one to determine the X and Y co-ordinates of the point touched by the user. **Figure 3a** shows how the X co-ordinate can be determined. Here, the right and left hand sides of the top layer can be connected to +V and ground respectively. The bottom layer can then be used to sense and measure the voltage at the point touched by the user. An A/D converter is used to convert this analogue voltage to digital and then determine the X co-ordinate.

Similarly, **Figure 3b** shows how the Y co-ordinate can be determined. Here, the upper and lower sides of the bottom layer can be connected to +V and ground respectively. The top layer can then be used to sense and measure the voltage at the point touched by the user. Again, an A/D is used to convert the voltage to digital and then to determine the Y co-ordinate.

5-Wire and 8-Wire Resistive Touch Screen

This is a modification of the basic 4-wire system where one layer (usually the top layer) is used for sensing and measuring the voltage, while the other layer is where the voltage gradient is created in X and Y directions. As shown in **Figure 4a**, to determine the X co-ordinate, the upper and lower sides of the bottom layer can be connected to +V and ground respectively. The top layer is then used to sense and measure the voltage.

To determine the Y co-ordinate we simply have to reverse polarity and sides of the bottom layer (see **Figure 4b**). The Y co-ordinate is then read from the top layer.

8-wire touch screen is used when more accurate measurements of the screen co-ordinates are required. In 4 and 5 wire implementations the resistance of the bus bars and the connection circuitry usually introduce offset errors in voltage measurements. These offset errors can drift with temperature, humidity and time. 8-wire touch screens compensate for drift by adding 4 additional reference lines, thus enabling the voltage to be measured directly at the touch screen bus bars. 8-wire touch screens are generally more expensive than others and are not covered further in this article.

An Application Example

An example is given here to demonstrate how a 4-wire resistive touch screen LCD can be used in a real application. In this example a touch screen with a GLCD is connected to a PIC18F type microcontroller. In addition, an LED is connected to one of the microcontroller port pins. The objective of this example is to flash the LED at a rate specified by the user.

Four options (1 second, 5 seconds, 10 seconds and 20 seconds) are displayed on the touch screen and the user is expected to make a choice by touching the required option and then start the flashing.

The block diagram of the project is shown in **Figure 5**. The project is built around a PIC18F4520 type high end microcontroller (www.microchip.com), operating at 8MHz. A 4-wire resistive touch screen (model no: TTW4028001) with graphics LCD display (KS0108 controller compatible) is used in the project (see **Figure 6**).

The touch screen graphics display is expected to show the images as in **Figure 7**.

Rectangles and boxes are drawn on the screen with text inside them. The screen is 128 pixels horizontal and 64 pixels vertical with the origin at the top left corner, the X-axis to the right and the Y-axis downwards.

The co-ordinates of the rectangles are also shown in Figure 7. The GLCD screen format can be designed using either a graph paper with a pen and pencil, or a bitmap design program can be used to draw the screen images and then convert it to hex data that can be downloaded to the GLCD controller (e.g. bmp2lcd, fastlcd, GLCDFontCreator etc).

The operation of the circuit is as follows: After power-up, the user will select the LED flashing rate by touching the required option box on the screen. The selected rate will be shown at the bottom of the GLCD screen. Flashing will then start as soon as the START box is touched. The display will show an appropriate message (e.g. "flashing...") at the bottom to indicate that flashing has started.

The Circuit Diagram

The circuit diagram of the project is shown in **Figure 8**. Ports B and D of the microcontroller are used to drive the GLCD. The connections between the microcontroller and the GLCD are as follows:

Microcontroller

RD0 – RD7
RB0
RB1
RB2
RB3
RB4
RB5

GLCD

D0 – D7
CS1
CS2
RS
R/W
E
RST

The background light of the GLCD is turned ON permanently by connecting the LED input to +5V via a resistor, and the GLCD contrast is adjusted using a 10K potentiometer. A touch screen controller circuit is used to interface the microcontroller to the 4 touch screen pins. In a microcontroller-touch screen interface, a controller circuit is usually required to provide the correct logic levels to the touch screen pins.

Normally, logic 0, logic 1 and OFF state are required. The OFF state can be provided using an open-drain microcontroller pin in input mode. Alternatively, touch screen controller chips such as AD785 or AD7846 can be used to provide the necessary interface voltage levels. In Figure 8,

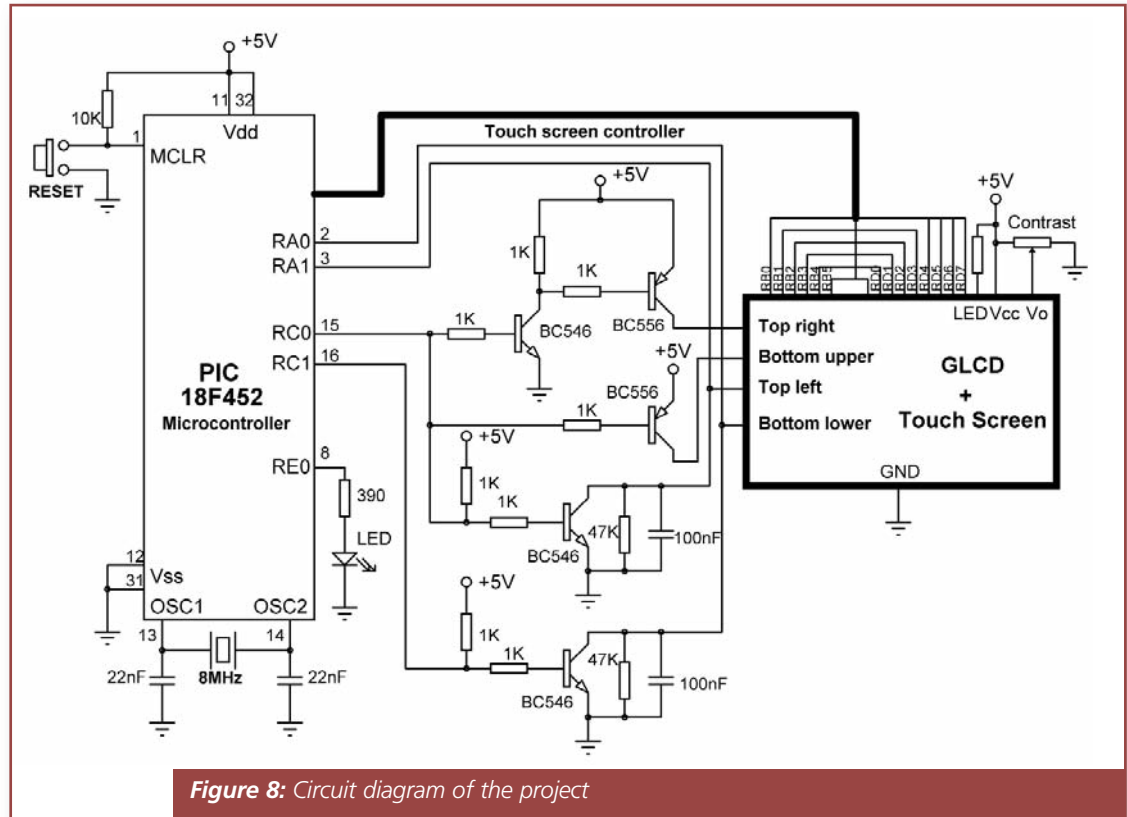


Figure 8: Circuit diagram of the project

switching transistors are used as the touch screen controller. For example, when RC0 is set to logic 1, Top Right pin becomes 1, Top Left pin becomes 0 and Bottom Upper pin becomes OFF.

Measuring the X Co-ordinate

In reference to **Figure 9** and assuming the top layer has contacts Top Right and Top Left and the bottom layer has contacts Bottom Upper and Bottom Lower, the following setup is required to determine the X co-ordinate:

Top Left:	GND
Top Right:	+5V
Bottom Lower:	To A/D converter (X co-ordinate)
Bottom Upper:	OFF

Similarly, to determine the Y co-ordinate, the following setup should be made:

Top Left:	To A/D converter (Y co-ordinate)
Top Right:	OFF
Bottom Lower:	GND
Bottom Upper:	+5V

In Figure 9 for example, the X co-ordinate can be read into analogue port RA0 when:

RC0 = 1	(Top Left = 0, Top Right = 1, Bottom Upper = OFF)
RC1 = 0	(Bottom Lower = OFF)
Read RA0	(Read Bottom Lower)

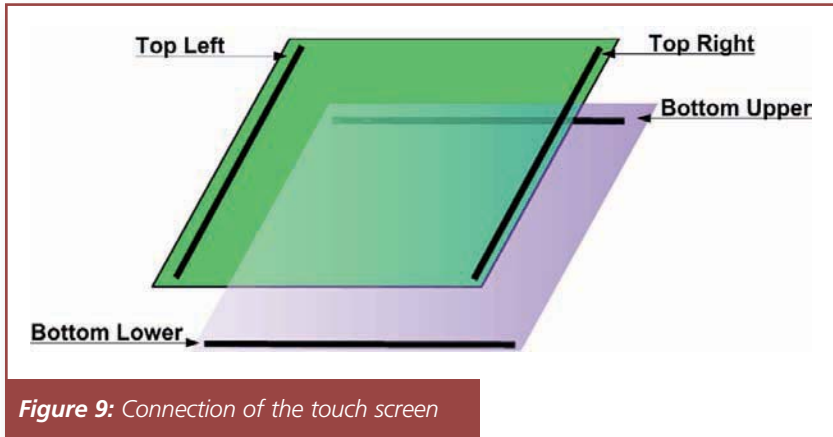


Figure 9: Connection of the touch screen

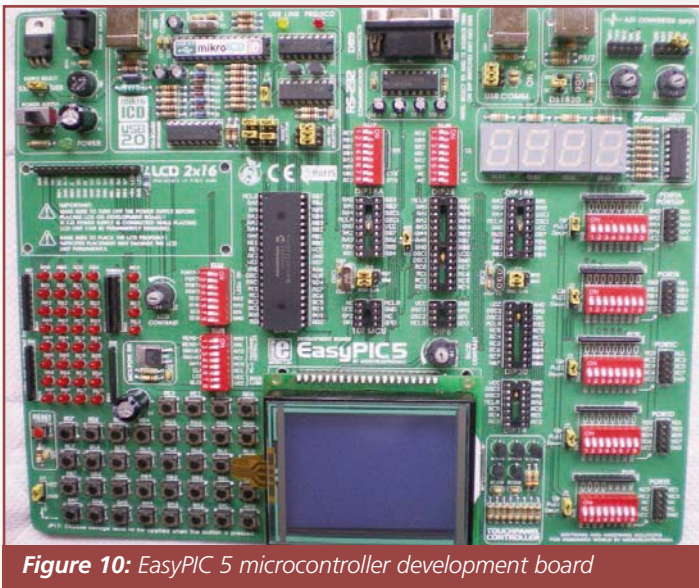


Figure 10: EasyPIC 5 microcontroller development board

Similarly, to read the Y co-ordinate:

RC0 = 0 (Top Left = OFF, Top Right = OFF, Bottom Upper = 1)
 RC1 = 1 (Bottom lower = GND) (Read Top Left)
 Read RA1

The Construction

The project was constructed easily using an EasyPIC 5 microcontroller development board (see Figure 10). Using this board one can develop and test touch screen software in a relatively short time. This board (www.mikroe.com) offers the following features:

- Socket for 12/16/18/20 and 40 pin PIC microcontrollers
- On board crystal, reset circuit and power supply
- LCD and touch screen GLCD interface with controller
- 7 segment 4-digit display
- 4 LEDs and 40 push-button switches
- RS232 and USB interfaces
- In-circuit debugger
- Programmer
- Access to microcontroller port pins

The board includes a touch screen controller circuit built using transistors as in Figure 8. A programmer is available on the board so that compiled programs can very easily be downloaded to the program memory of the microcontroller.

```

BEGIN
Configure the GLCD
Configure I/O ports
Display selection boxes on GLCD

DO FOREVER
Get X and Y co-ordinates of touched point
IF box "1s" touched Rate = 1
ELSE IF box "5s" touched Rate = 5
ELSE IF box "10s" touched Rate = 10
ELSE IF box "20s" touched Rate = 20
ENDIF
Display "Rate = <Rate>"

IF box "START" touched
DO FOREVER
Turn LED ON
Wait Rate seconds
Turn LED OFF
Wait Rate seconds
ENDDO
ENDIF

ENDDO
END
    
```

Figure 11: PDL of the project

The Software

The software was developed using the mikroC Pro language. This is a high-level C programming language for PIC microcontrollers, developed by mikroElektronika (www.mikroe.com) and it offers a large number of built-in libraries to interface various devices to PIC microcontrollers. Some of the libraries are for SD cards, Compact Flash cards, RS232, RS485, CAN bus, USB, LCD, GLCD, I2C bus and many more.

The compiler also includes a software simulator in the form of an in-circuit debugger that can be very useful during program development.

Figure 11 shows operation of the software as a PDL. The complete program listing is shown in Figure 12.

At the beginning of the program the GLCD-microcontroller interface is defined and Glcd_Init command is used to initialise the GLCD. Then the microcontroller I/O ports are configured. The images shown in Figure 7 are then created and displayed on the GLCD. Command Glcd_Rectangle draws a rectangle with the specified top-left and bottom-right co-ordinates. Similarly, command Glcd_Box draws a box with the specified co-ordinates. Texts are then written inside the boxes using the Glcd_Write_Text commands by specifying the starting co-ordinates of the texts. Inside the main loop of the program the X and Y co-ordinates of the option boxes are read by functions ReadX and ReadY respectively, to see if the user touched any of the boxes and, if so, the appropriate flashing rate is selected. mikroC built-in functions ADC_Read(0) and ADC_Read(1) are used to read the analogue data from inputs RA0 and RA1 respectively. The physical X co-ordinate is calculated by multiplying the read X value by 128 and then dividing by the 1024 (maximum value for 10-bit A/D). Similarly, the physical Y co-ordinate is calculated by multiplying the read Y value first by 64 and dividing by 1024 and then taking away from 64 (since the +5V is applied to Bottom Upper layer during Y measurements). Finally, touching the START box starts flashing the LED connected to port RE0 at the selected rate.

Figure 13 shows a typical run of the program where the flashing rate

```

/*****
TOUCH SCREEN EXAMPLE
*****

In this example, an LED is flashed at a rate determined by the user by selecting
the rate from a touch screen GLCD. The project is based on a PIC18F452 type
microcontroller, operating at 8MHz. The software is based on the mikroC Pro.

*****/
char GLCD_DataPort at PORTD;          Glcd_Init();
sbit GLCD_CS1 at RB0_bit;             Glcd_Set_Font(Font_Glcd_5x7,5,7,32);
sbit GLCD_CS2 at RB1_bit;             Glcd_Fill(0);
sbit GLCD_RS at RB2_bit;
sbit GLCD_RW at RB3_bit;
sbit GLCD_EN at RB4_bit;
sbit GLCD_RST at RB5_bit;

char GLCD_CS1_Direction at TRISB0_bit;
char GLCD_CS2_Direction at TRISB1_bit;
char GLCD_RS_Direction at TRISB2_bit;
char GLCD_RW_Direction at TRISB3_bit;
char GLCD_EN_Direction at TRISB4_bit;
char GLCD_RST_Direction at TRISB5_bit;

char msg_Start[] = "START";
char msg_1s[] = "1s";
char msg_5s[] = "5s";
char msg_10s[] = "10s";
char msg_20s[] = "20s";
char msg_Rate[] = "Rate=";
char msg_Flashing[] = "flashing...";

long x_real,y_real;
char rate;

long ReadX(void)
{
    long x;
    PORTC.F0 = 1;
    PORTC.F1 = 0;
    Delay_Ms(5);
    x = ADC_Read(0);
    x = x*128/1024;
    return(x);
}

long ReadY(void)
{
    long y;
    PORTC.F0 = 0;
    PORTC.F1 = 1;
    Delay_Ms(5);
    y = ADC_Read(1);
    y = 64 - ((y*64)/1024);
    return (y);
}

void Delay_Seconds(char d)
{
    char i;

    for(i=0; i<d; i++)Delay_Ms(1000);
}

void main()
{
    char i;
    TRISA = 0x03;
    TRISC = 0;
    TRISE0_bit = 0;
    ADCON1 = 0x0F;
    PORTA = 0;
    PORTC = 0;

    Glcd_Rectangle(5,12,40,42,1); // START rectangle
    Glcd_Box(7,14,38,40,1); // START box
    Glcd_Rectangle(60,10,85,30,1); // 1s rectangle
    Glcd_Box(62,12,83,28,1); // 1s box
    Glcd_Rectangle(95,10,120,30,1); // 5s rectangle
    Glcd_Box(97,12,118,28,1); // 5s box
    Glcd_Rectangle(60,35,85,55,1); // 10s rectangle
    Glcd_Box(62,37,83,53,1); // 10s box
    Glcd_Rectangle(95,35,120,55,1); // 20s rectangle
    Glcd_Box(97,37,118,53,1); // 20s box

    Glcd_Write_Text(msg_Start,8,3,0); // START text
    Glcd_Write_Text(msg_1s,67,2,0); // 1s text
    Glcd_Write_Text(msg_5s,102,2,0); // 5s text
    Glcd_Write_Text(msg_10s,65,5,0); // 10s text
    Glcd_Write_Text(msg_20s,100,5,0); // 20s text
    Glcd_Write_Text(msg_Rate,10,7,1); // RATE= text

    for(;;)
    {
        x_real = ReadX();
        y_real = ReadY();

        if((x_real >= 62 && x_real <= 83) && (y_real >= 12 && y_real <= 28))
        {
            rate = 1; msg_Rate[5]='1'; msg_Rate[6]='s'; msg_Rate[7]=' ';
        }
        else if((x_real >= 97 && x_real <= 118) && (y_real >= 12 && y_real <= 28))
        {
            rate = 5; msg_Rate[5] = '5'; msg_Rate[6] = 's'; msg_Rate[7]=' ';
        }
        else if((x_real >= 62 && x_real <= 83) && (y_real >= 37 && y_real <= 53))
        {
            rate = 10; msg_Rate[5]='1'; msg_Rate[6]='0'; msg_Rate[7]='s';
        }
        else if ((x_real >= 97 && x_real <= 118) && (y_real >= 37 && y_real <= 53))
        {
            rate = 20; msg_Rate[5]='2'; msg_Rate[6]='0'; msg_Rate[7]='s';
        }
        Glcd_Write_Text(msg_Rate,1,7,1);

        if ((x_real >= 7 && x_real <= 38) && (y_real >= 14 && y_real <= 40))
        {
            for(i=0; i<11; i++)msg_Rate[i+9]=msg_Flashing[i];
            Glcd_Write_Text(msg_Rate,1,7,1);
            for(;;)
            {
                PORTE.F0 = 1;
                Delay_Seconds(rate);
                PORTE.F0 = 0;
                Delay_Seconds(rate);
            }
        }
    }
}

```

Figure 12: Program listing



Figure 13: A typical run of the program (selecting a Rate)

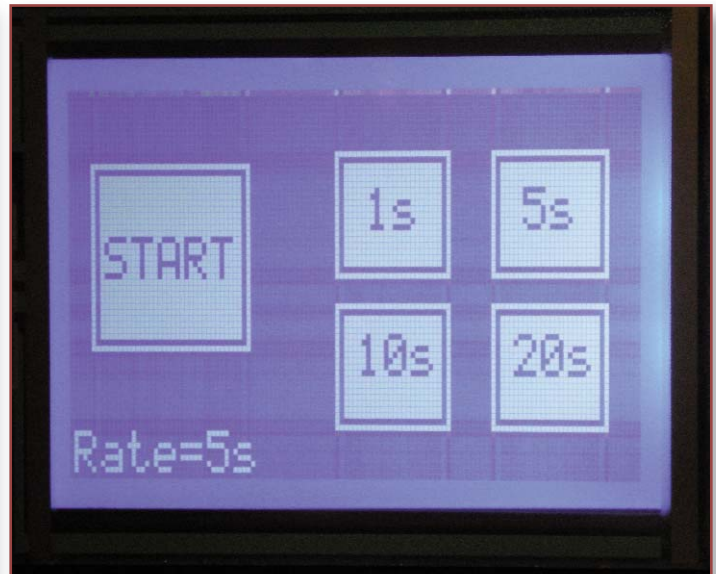


Figure 14: A typical run of the program (starting the flashing)

was selected as five seconds. Touching the START box starts to flash the LED and the GLCD display changes to indicate that flashing has started, as in Figure 14.

The touch screen GLCD example given here is very simple, but using these principles very complex touch screen GLCD based projects can easily be designed. ■

RELIABLE LOW POWER RADIO MODEMS FOR PERFORMANCE CRITICAL APPLICATIONS

ASCII in, ASCII out, 9600 baud wireless link, minimum effort

- Takes care of all over-air protocols
- European license-free 433 MHz ISM band & Custom frequencies
- Line-of-sight range over 500m
- Transmit power: +10dBm (10mW)
- Receiver sensitivity: -107dBm (for 1% BER)
- Addressable point-to-multipoint
- Conforms to EN 300 220-3 and EN 301 489-3
- No additional software required

Ideally suited for fast prototyping / short design cycle time

TXL2
& RXL2



Producing VHF and UHF, ISM band modules for over 25 years.

T: +44 (0) 20 8909 9595 sales@radiometrix.com
www.radiometrix.com

RADIOMETRIX
WIRELESS DATA TRANSMISSION