**D.J.SANGHVI COLLEGE OF ENGINEERING**

INDUSTRIAL SUMMER TRAINING PROGRAMME

Project: Surface level detector

By: Vatsal Makani

**Executive Summary**

This project intends to build A Level Surface Device using an 2 access Accelerometer and display the x and the y co ordinates on the LCD screen and when the surface is perfectly horizontal display “done” on the computer through UART.

**Step 1: Project specification**

In the project two psoc boards are used. One is psoc3 and other is psoc4. In the accelerometer two access are used x access and y access. These x and y co ordinate are displayed on the lcd screen. When the surface is perfectly horizontal display “done” on the computer through UART.

A bray’s terminal software is used in this project. It is used for the serial communication between

the computer and the PSOC board. Through the Brays terminal we can send or receive values.

**Step 2: Component Selection**

1. 2 access accelerometer. Usually there are 3 access of the accelerometer x,y,z access.We make the use of the x and y. For this we require two analog pins accx and accy.

2.12 bit ADC. 12bit means it has a resolution of 12 bits.

3. Analog MUX (2:1)

4. UART.

5. LCD screen.

6. logic zero.

7. 3 digital output pins named as st, mode, en to enable the accelerometer.

**Step 3: Pin configuration**

1. Design pins should be equal to the physical pins. There should be 1 to 1 mapping.

2. Accelerometer consists of three input pins which is given to it by microcontroller. These pins are enable and select the mode of the accelerometer. These pins are

a. Enable: 1=enable, 0=disable

b. mode: 0=low power mode , 1= regular power mode.

c. st/mot (self test/motion): self testing mode=1, motion mode =0.

All the above pins are digital output pins.

3. There is two analog pin i.e accx and accy to the input of the AMux.

4. The output of the analog mux is given to the ADC.

5. aacx is connected to p3[5] and accy is connected to p3[6].

6. Pin configuration

St : p15[0]

En : p15[2]

Mode : p15[1]

7. connect the UART of PSOC4 to the PSOC3 TTL. Rx and Tx of UART of PSOC4 is connected to Tx and Rx of PSOC3.

That is with the help of external wire we connect p12[6] (Rx) of PSOC4 to p4[5] (Tx) of PSOC3 and connect p12[7] (Tx) of PSOC4 to p4[4] (Rx) of PSOC3.

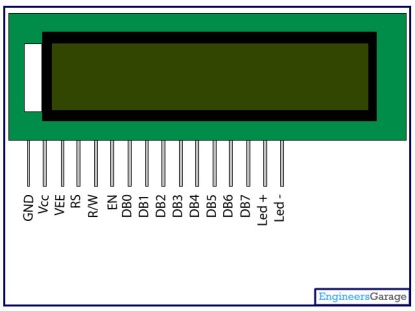
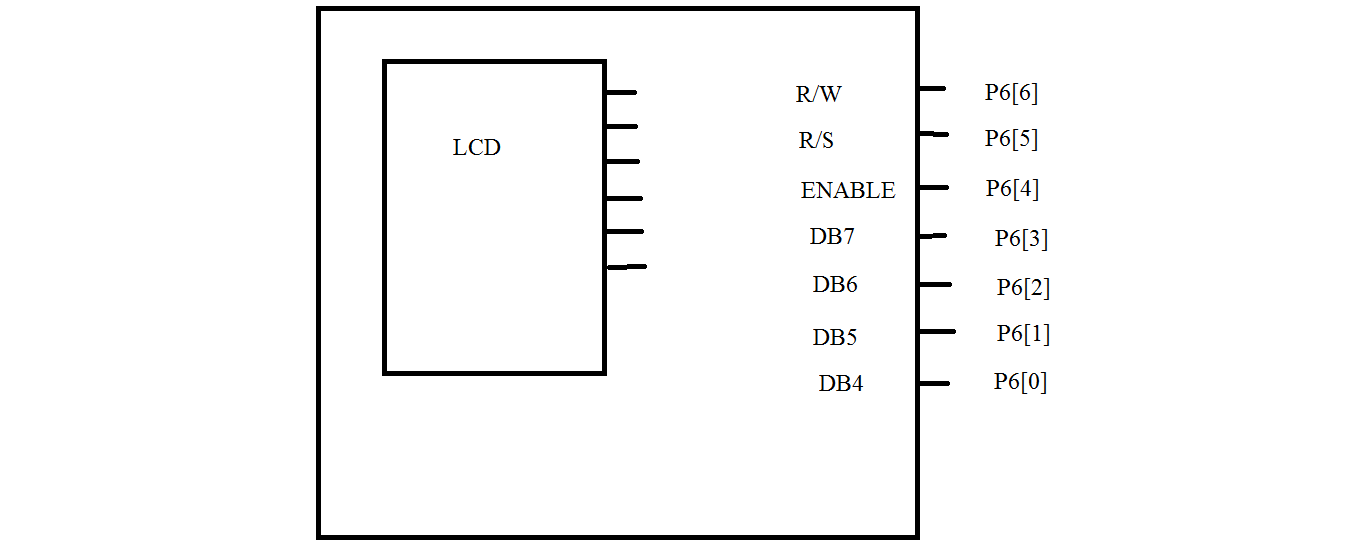
8. Pins of the lcd are connected to p6[6:0] ie to the port 6 of the psoc.

The pin configuration of the lcd is as follows:

Pin1: Ground

Pin2:Vcc

Pin3:contrast

Pin4: R/s register select.

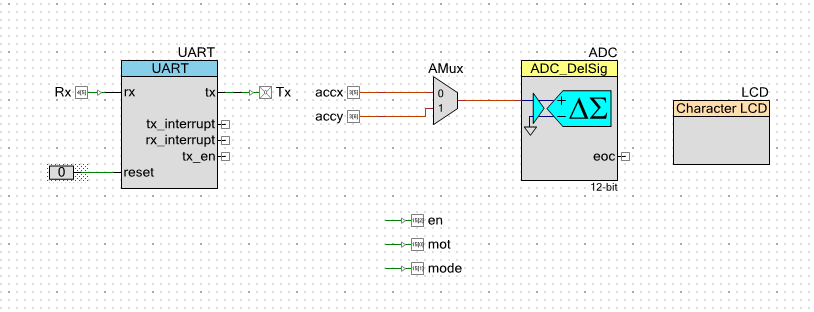
Pin5: R/W read/ Write.

Pin6: Enable.

Pin7-pin14: data bits DB0-DB7.

Pin15,pin16:back led.

**Step 4: Component configuration**



1. Uart configuration:

Mode : Full UART (Tx + Rx).

Baud rate: 9600

Data bits : 8

Stop bit : 1

Internal clock is been selected.

Rx : digital input (HW connection)

Tx : digital output (HW connection)

2. AMux configuration:

Channels : 2

Isolation : medium

3. ADC configuration

It is being used in multichannel. (1-multi sample)

Resolution is 12 bits.

Conversion rate is 10000.

Input range is 0 to 2vref.

Input mode is single ended.

No of configuration is 1.

4. A character lcd screen is been selected.

**Step 5: Algorithm**

It follows the following steps:

**Step 1**: Initialize the accelerometer, ADC, UART ,LCD.

**Step 2**: Select one channel of the AMux at a time. Get the value from ADC.

**Step 3**: Compare with 0g offset.

0g offset=1.65v

Vdd=3.3v

**Step 4**: Voltage range is from 0 to 3.3v but in order to save power, upper

bounded voltage is reduced from 3.3 to 2.048v.In psoc the reference line voltage

is 1.024v. The resolution is of 12 bits. Therefore 2.048 will correspond to 4096th

level and 0v will correspond to 0th level.

2.048v = 4096th level

1.65v = xth level

x= (1.65 \* 4096) / (2.048) = 3300

Therefore the 1.65v will correspond to 3300 level.

**Step 5:** The 12 bit resolution is lowered down to 5 bit resolution. This is done as

follows: The 12bit value is shifted 7 times to the right. Now the upper 5 bits of

the msb is made 0. So the 12 resolution is lowered down to 5 bit resolution.

**Step 6.** Hence we get the x co ordinate. Repeat the same procedure to get the y

co-ordinate by selecting another channel.

**Step 7:** Display the x and y co-ordinate on the lcd screen**.**

**Step 8:** Check whether the x and the y co-ordinates are zero. If yes display

“done” on the computer through the Bray’s terminal software.

Switch on the Bray’s terminal software, rescan it check the port to which the usb is connected

then connect it to the corresponding port.

**Code:**

#include <project.h>

#include <cypins.h>  **//include the necessary header files**

#include "stdio.h"

void initialize() **//generating a function that will initialize**

**all the components.**

{

en\_Write(1);

mode\_Write(1); **// initialize the accelerometer.**

mot\_Write(0);

CyGlobalIntEnable;

UART\_Start();

ADC\_Start(); **//enables the ADC**

ADC\_StartConvert()**; //starts the conversion**

}

int16 func(bit a)

{

int16 ADC\_b;

AMux\_Select(a); //**To select the corresponding channel of AMux.**

LCD\_Start();

ADC\_Stop(); **//To clear the garbage value of a previous channel.**

ADC\_Start();

ADC\_StartConvert();

ADC\_IsEndConversion(ADC\_WAIT\_FOR\_RESULT);

ADC\_b=ADC\_GetResult16();

ADC\_b = ADC\_b - 3000; // **Get the result from ADC**

ADC\_b = ADC\_b >> 8; //**compare with offset and change the resolution**.

return ADC\_b; //**return the value i.e the co-ordinate.**

}

int main()

{

char buff[15]={'\0'}; **//initialize the character buff.**

int16 ADC\_x, ADC\_y;

initialize(); //**call the function “initialize”**

LCD\_ClearDisplay()**; //clear the lcd display**

for(;;)

{

LCD\_ClearDisplay();

ADC\_x=func(0); **//call the function and get the co-ordinates of x**

**and y.**

ADC\_y=func(1);

LCD\_Position(0,1); **//set the lcd position where we have to display**

LCD\_PrintString("X: ");

LCD\_Position(0,2);

sprintf(buff,"%1d",(int16)ADC\_x);

LCD\_PrintString(buff);

**//Convert the numerical ADC\_x value of type uint16 into character and**

**then display it on LCD.**

LCD\_Position(1,1);

LCD\_PrintString("Y: ");

LCD\_Position(1,2);

sprintf(buff,"%1d",(int16)ADC\_y);

LCD\_PrintString(buff);

**//Convert the numerical ADC\_y value of type uint16 into character and then display it**

**On the LCD screen.**

if (ADC\_x==0 && ADC\_y==0)

{

UART\_PutString("DONE");

//**If the x and y co ordinates are zero ie it is perfectly horizontal it displays “done” on the computer through UART.**

}

CyDelay(100);

}

}

**Step 6. Test and debug:**

Do the necessary caliberation if required.

If the device is too sensitive we can further reduce the resolution.

The caliberation can be done by changing the value of offset i.e changing the value 3300.

In this project caliberation was required i.e we changed the value to 3000

To get the output when the board was perfectly horizontal.

The resolution was also decreased by 1 bit.

That is the resolution in this project was 4bit instead of 5bit.

The project designed was then tested and was successfully implemented.

**Conclusion:**

This project may be used by  [stonemasons](http://en.wikipedia.org/wiki/Stonemason), [bricklayers](http://en.wikipedia.org/wiki/Bricklayer), other building trades workers, [surveyors](http://en.wikipedia.org/wiki/Surveyor_(surveying)" \o "Surveyor (surveying)),[millwrights](http://en.wikipedia.org/wiki/Millwright) and other metalworkers, and in some [photographic](http://en.wikipedia.org/wiki/Photography) or [videographic](http://en.wikipedia.org/wiki/Videography" \o "Videography) work. Hence this project finds a wide application in day to day life.