ECE 2036: Lab #3 – mBED Hardware Starter Lab

Category: Getting Started with MBED ~ 1 week to complete

ECE2036a - Due Date: Monday September 28 @ 11:59 PM

ECE2036b –Due Date: Tuesday September 29 @ 11:59 PM

Score Part 1:____ Score Part 2: ____ Score Part 3: ____ Score Part 4: ____ Score Part 5: ______

Directions: To complete this lab, you will need to demo your working mbed projects by posting videos to Youtube of your components working, and putting the link to the videos on t-square dropbox. Please make a folder called Lab 3 on t-square's dropbox. You may call each main.cpp file for each section: mainPart1.cpp, mainPart2.cpp, mainPart3.cpp, mainPart4.cpp, and mainPart5.cpp. Please upload the main.cpp files that you create to your Lab 3 folder on t-square. You will also need to take pictures and videos according to the following specifications. Make sure that you start each video with a shot of your face and state your name clearly.

1. We would like a selfie with you and your completed board (posted to t-square).
2. Take a 10 second video of you demonstrated Part 1 (post YouTube)
3. Take a video of you demonstrating Part 2 as long as you need (post YouTube)
4. Take a 20 second video of you demonstrating Part 3 (post YouTube)
   (hold sensor to heat it up on the video!!)
5. Take a 25 second video of you demonstrating a sound with Part 4 (post YouTube)
6. Upload the file that you created on the microSD card to t-square dropbox for Part 5.

IMPORTANT: AFTER COMPLETING THIS LAB DO NOT TAKE APART YOUR HARDWARE BECAUSE YOU WILL USE IT FOR SUBSEQUENT MBED LABS IN THIS CLASS!!

Embedded devices account for 98% of the world's microprocessors. For every desktop computer, there are over 100 embedded devices. A high-end car can contain up to 100 microprocessors. Embedded devices contain a computer with software that is typically not changed by the user (called "firmware"). Most users are not aware that their cell phones, cameras, audio players, and TVs contain a computer with firmware. C/C++ is currently the most widely used language for embedded devices. ARM processors, which are similar to the one found in the *mbed* module, are used in about 80% of embedded devices including most cell phones.

This assignment will start with getting a variety of external components working with your *mbed* board. In addition to your *mbed* module, you will need a protoboard, wire kit, and LCD board which should come with your mBED kit that you purchased.

Instructions for Part 1 (20%): "What's up mBED!"

1. Watch this YouTube video FIRST. This tells you how NOT to break the pins on your *mbed* board! VERY IMPORTANT - COULD WIND UP COSTING YOU $50!

Video link: http://mbed.org/blog/entry/104/

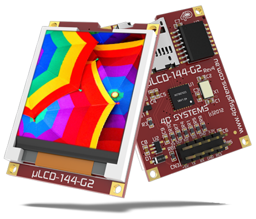
2. Create your *mbed* account using the instructions in your kit. After placing the *mbed* board on the protoboard, you can connect this to your USB port. It should show up on your computer as an external drive. You will put files in this drive that will be automatically downloaded to the flash drives on your *mbed* board.

3. Import the skeleton code into your *mbed* account that is found at http://mbed.org/users/4180_1/code/mythermostat/

You should import skeleton code to your *mbed* cloud account so that you can use the libraries provided for this and other labs.

4. On your protoboard, please put the LCD on the side of the MBED chip where you have the pins 27-29. See the wiring table below for the pin connections needed. You might also want to read the mbed wiki page is (Please note some of the pin changes in the table in Figure 1!)

https://mbed.org/users/4180_1/notebook/ulcd-144-g2-128-by-128-color-lcd/



CAUTION

- If you switch the Rx/Tx pins, you will get the LCD splash screen message.
- MAKE SURE NOT TO SWITCH POWER AND GROUND PINS. If you do, you will have to BUY a new device!!!!

LCD Wiring for LCD test program

Mbed	uLCD pin
5V=VU	5V (1)
Gnd	Gnd (7)
TX=P28	RX (5)
RX=P27	TX (3)
P29	Reset (9)
Note: Bend over LCD pin 10, or use LCD's cable set per wiki instructions	

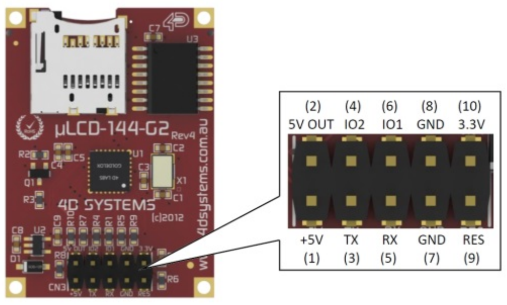


Figure 1: These are the LCD pin connections needed for this assignment. Please note that there are some pins moved between this figure and the mbed cookbook wiki demo code. Note that if you choose to use the cable, the RX and TX pins are swapped on the cable connector label (per wiki).

5. Create a new program in your compiler environment and type in the following code. You will need to copy uLCD_4DGL.h from the skeleton code project in third step into this new project. Cut and paste can be used in the compiler's left column that shows project source files. You can also use the import library link found on the LCD wiki page.

```
#include "mbed.h"
#include "uLCD_4DGL.h"

uLCD_4DGL uLCD(p28, p27, p29); // create a global lcd object

int main() {
  uLCD.printf("\nWhat's up mBED!\n");
}
```

6. Compile this program. On some machines, this will automatically download into your *mbed* if it is connected to your computer; however, on other machines the file might be in your download folder. In this case, all you need to do is drag it to the *mbed* drive, and it will be automatically downloaded to the board.

7. Pressing the reset button on the chip will automatically run the program with the most recent timestamp that is downloaded to the board. Do this and you should see your first message on the LCD display!

Instructions for Part 2 (20%): "Bouncing Ball"

I would like for you to modify the following program for a bouncing ball on the LCD. In the modification you will have a larger and faster bouncing ball! I would like for you to first show the regular bouncing ball on the screen; pause for two seconds; then start your code for the larger bouncing ball. Also I would like you to make the following changes to the code. You will have to use the *mbed* resources to help you make the appropriate changes. You may find some problems might arise, AND you can fix them if you would like!

1. Change the border to another color besides white.

2. Your large ball should be larger than the original ball with a radius of at least 10 pixels.

3. Noticeably speed up the movement of the ball in both the x and y direction.

Here is some sample code for a ball moving across the screen

```
// uLCD-144-G2 demo program for uLCD-4GL LCD driver library
#include "mbed.h"
#include "uLCD_4DGL.h"

uLCD_4DGL uLCD(p28, p27, p29); // serial tx, serial rx, reset pin;

int main()
{
  uLCD.display_control(PORTRAIT);
  uLCD.cls();
  uLCD.printf("Bouncing Ball");
  uLCD.baudrate(BAUD_3000000); //jack up baud rate to max for fast display
  wait(1.0);

  //Set up initial conditions
  float fx=50.0,fy=21.0,vx=0.4,vy=0.3;
  int x=50,y=21,radius=4;
  uLCD.background_color(BLACK);
  uLCD.cls();

  //draw borders
  uLCD.line(0, 0, 127, 0, WHITE);
  uLCD.line(127, 0, 127, 127, WHITE);
  uLCD.line(127, 127, 0, 127, WHITE);
  uLCD.line(0, 127, 0, 0, WHITE);

  for (int i=0; i<1500; i++)
  {
    //draw ball
    uLCD.circle(x, y, radius, RED);

    //bounce off edge walls and slow down a bit
   if ((x<=radius+1) || (x>=126-radius)) vx = -.95*vx;
   if ((y<=radius+1) || (y>=126-radius)) vy = -.95*vy;

   //erase old ball location
   uLCD.filled_circle(x, y, radius, BLACK);

  //calculate new ball position
  fx=fx+vx;
  fy=fy+vy;
  x=(int)fx;
  y=(int)fy;
  } //end for loop
}//end main
```

Instructions for Part 3 (20%) : "Digital Thermometer"

An TMP36 wiki page is provided that shows how to connect the TMP36 sensor and read the temperature using C/C++ on the mbed module. BE CAREFUL because it looks just like the 2N3904 transistor in the kit, so check the tiny marks on the case's flat area for the part number. This assignment will verify your hardware connections and sensor operation before trying your next mbed lab.

1. In previous sections you got your LCD working, please leave this hooked up!
   Now hook up the temperature sensor using the information on the wiki page at:

http://mbed.org/users/4180_1/notebook/lm61-analog-temperature-sensor/

2. Create a new program and type in the following code. Once this is working, please modify the program so that you make the display of the temperature look visually appealing.

```
#include "mbed.h"
#include "uLCD_4DGL.h"
```

```
uLCD_4DGL uLCD(p28, p27, p29); // serial tx, serial rx, reset pin;

//Setup a new class for TMP36 sensor
class TMP36
{
public:
   TMP36(PinName pin);
   TMP36();
   float read();
private:
//class sets up the AnalogIn pin
   AnalogIn _pin;
};

TMP36::TMP36(PinName pin) : _pin(pin) {} //This is an initializer list … more to come in class
// _pin(pin) means pass pin to the AnalogIn constructor

float TMP36::read()
{
//convert sensor reading to temperature in degrees C
   return ((_pin.read()*3.3)-0.500)*100.0;
}

//instantiate new class to set p15 to analog input
//to read and convert TMP36 sensor's voltage output

TMP36 myTMP36(p20);

int main()
{
   float tempC, tempF;

   while(1) {
      tempC = myTMP36.read();
      //convert to degrees F
      tempF = (9.0*tempC)/5.0 + 32.0;
      //print current temp
     uLCD.cls();
     uLCD.printf("%5.2f C %5.2f F \n\r", tempC, tempF);
      wait(.5);
   }
}
```

Instructions for Part 4 (20%): "Jazzy Tunes"

In the fourth part of this lab you will build a basic 3-key musical keyboard. Do not take off the other components! The point of this exercise is to build up and verify your components to help with your future mBED lab. I would advise that you try to get the keys working first with some of the LEDs and then add functionality with the speaker. I will not spell out everything explicitly here, but instead I will require you to get information from the mBED website.

Pushbuttons

Read the pushbutton wiki page at

http://mbed.org/users/4180_1/notebook/pushbuttons/

and watch the videos for additional help using pushbuttons with mbed. Small pushbuttons are available for use on your breadboard. I would like for you to look at the constructor calls in the sample code to determine the pin connections to the mbed board.

Speaker

WARNING! WARNING! WARNING WILL ROBINSON!!!

THE SPEAKERS ARE VERY DELICATE. DO NOT PULL UP ONCE IT IS ON THE BREADBOARD. YOU MUST EASE IT OUT BY PRYING IT OFF FROM UNDERNEATH THE SPEAKER. IT IS TRICKY.. . BE CAREFUL.

Use the driver transistor and speaker to make tones when each button is pressed. See the speaker wiki page for additional hardware and software help using speakers with mbed which is found at:

https://mbed.org/users/4180_1/notebook/using-a-speaker-for-audio-output/

I would like for you to look at the constructors in the sample code to determine the pin connections to the mbed board.

1. Please leave all previous component hooked up on your board (i.e. LCD and temperature sensor)

2. Now hook up three push buttons using the information on the wiki page at:

http://mbed.org/users/4180_1/notebook/pushbuttons/

I would suggest testing the code on the following page without the PlayNote functions to see if the push buttons work with the LEDs built into the mbed board.

3. Now hook up the speaker and transistor driver using the information on the wiki page found at:

https://mbed.org/users/4180_1/notebook/using-a-speaker-for-audio-output/

4. Create a new program and type in the following code. You will need to copy the Speaker.h file over from the skeleton code that you downloaded to your mbed account in part 1 of this lab.

```
#include "mbed.h"
#include "Speaker.h"
#include "PinDetect.h"

DigitalOut myled1(LED1);
DigitalOut myled2(LED2);
DigitalOut myled3(LED3);
DigitalOut myled4(LED4);

PinDetect pb1(p28); //you can use different pins
PinDetect pb2(p29); //you can use different pins
PinDetect pb3(p30); // you can use different pins
```

```
// setup instance of new Speaker class, mySpeaker using pin 21
// the pin must be a PWM output pin
Speaker mySpeaker(p21);

 //---------------------------------------------------------------------------------------------
// Callback routine is interrupt activated by a debounced pb1 hit
// That is ... this code runs with interrupt is generated by first button press

void pb1_hit_callback (void)
{ myled1 = !myled1;
mySpeaker.PlayNote(200.0,0.25,0.1);
}

//---------------------------------------------------------------------------------------------
// Callback routine is interrupt activated by a debounced pb2 hit
// That is ... this code runs with interrupt is generated by second button press

void pb2_hit_callback (void)
{  myled2 = !myled2;
   mySpeaker.PlayNote(400.0,0.25,0.1);
}

//---------------------------------------------------------------------------------------------
// Callback routine is interrupt activated by a debounced pb3 hit
// That is ... this code runs with interrupt is generated by third button press

void pb3_hit_callback (void)
{  myled3 = !myled3;
mySpeaker.PlayNote(800.0,0.25,0.1);
}

//---------------------------------------------------------------------------------------------
int main()
{
//setup push buttons
   pb1.mode(PullUp);
   pb2.mode(PullUp);
   pb3.mode(PullUp);
   // Delay for initial pullup to take effect
   wait(.01);
   // Setup Interrupt callback functions for a pb hit
   pb1.attach_deasserted(&pb1_hit_callback);
   pb2.attach_deasserted(&pb2_hit_callback);
   pb3.attach_deasserted(&pb3_hit_callback);
   // Start sampling pb inputs using interrupts
   pb1.setSampleFrequency();
   pb2.setSampleFrequency();
   pb3.setSampleFrequency();
// pushbuttons now setup and running

   while(1)
   {
    myled4 = !myled4;
    wait(0.5);
   }
} //end main
```

Instructions for Part 5 (20%): “Hello Micro SD Card”

In this part you will need to add the micro SD card to your setup.

The micro SD card (http://mbed.org/cookbook/SD-Card-File-System) or a USB flash drive (<2G FAT 16) can be used for storage. For this lab you will use the SD card. Please use the constructor call in the below sample code that WRITES a simple message to file on your SD card to determine the pin connections that you need. The microSD card can be put into the larger SD card adapter to read the data files on a PC if you would like. An example file that you might use to verify that your SD card is working is given below. To read the files you will need to look up documentation online using fscanf. (There are others, but here is one link: http://developer.mbed.org/users/ColonelPewter/notebook/Read_SD_card/)

```
#include "mbed.h"
#include "SDFileSystem.h"
#include "uLCD_4DGL.h"

SDFileSystem sd(p5, p6, p7, p8, "sd");

uLCD_4DGL uLCD(p28, p27, p29); // serial tx, serial rx, reset pin;

int main() {

  uLCD.printf("Hello Micro SD Card!\n");

  mkdir("/sd/mydir", 0777);

  FILE *fp = fopen("/sd/mydir/sdtest.txt", "w");
  if(fp == NULL) {
    uLCD.printf("Error Open \n");
  }
  fprintf(fp, "I love ECE2036");
  fclose(fp);
}
```

PLEASE NOTE: Writing files without an SD card plugged in can result in the mBED “blue LEDs of death”.