

The goal of this exercise is to prepare the infrastructure necessary to build an embedded system using the mbed LPC1768 development board (<http://developer.mbed.org/>). This will be used in a project that involves interfacing to the mbed, programming, testing and deploying it in an embedded application.

The mbed is an ARM-based microcontroller manufactured by NXP Semiconductors. ARM is a RISC instruction set architecture originally developed by Acorn Computers in the early 1980s. It has become the most widely used 32-bit architecture in terms of number manufactured. There is at least one ARM processor in 99% of all smartphones and tablets and, as of 2013, over 50 billion products containing an ARM chip have been shipped ([Vance 2014](#)). The mbed's ARM Cortex-M3 core runs at 96MHz with 512KB flash memory and 64KB RAM, and several communication interfaces, including Ethernet, USB Device, CAN, SPI, I2C, and others.

Task 1: Setup: USB connection and Account Activation

Follow the instructions in the Setup Guide found on the sheet included in the mbed box. Connect your mbed to your laptop/PC via the USB cable. Click the mbed.html link on the flash drive that appears. Click signup and note the last few digits of the serial number of your board on the web page (lower right) as you might need it later (e.g., for serial I/O driver installation). Set up your account, which will allow you to use the cloud compiler to develop, compile, and download code for the mbed to execute.

Task 2: Learn proper insertion/removal of mbed

The mbed microcontroller is fragile and inserting it into a breadboard will require some force. Ensure you line up the holes, then press evenly until it goes into the board fully. If you are not careful, the pins will break.

Never try to remove the mbed by pulling on the USB lead or USB connector. Do not push down on the USB connector when pulling out the USB cable. These actions may pull off the USB connector itself.

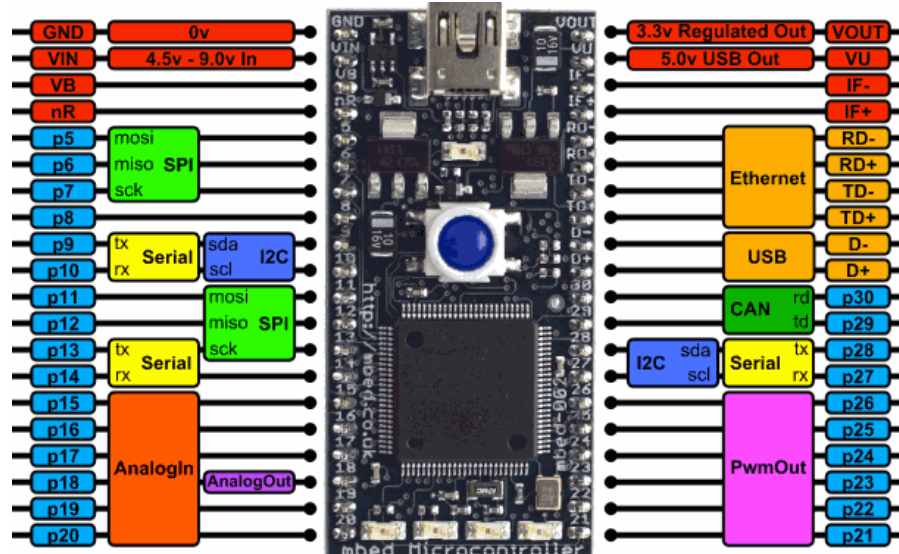
It is best to leave the mbed microcontroller in the board whenever possible. If you do need to remove it, patiently and carefully use a small jewelers screwdriver or similar, and lever it a bit at a time evenly from each of the four corners in turn, ensuring you pull up with the screwdriver so you are pressing against the breadboard with the tip and the edge of the microcontroller with the shaft, rather than the tip touching the bottom of the mbed (which might damage the components on the underside).

The mbed box contains a small plastic card that shows the pinout diagram for the mbed (also shown below). This gives pin names that you will use to read or send data values to the pins. Note that power (3.3V and 5V up to 460mA) is available for the breadboard from the upper pins. On-chip power circuits produce this voltage using power from the USB cable (500mA max). *V_{in}* is for an external power source like a battery – *don't connect anything to it while using the USB cable*. There are four blue user LEDs at the bottom of the board (LED1-- LED4). They are useful for providing status information. The LEDs will flash if you ever have a run-time error.

Turning power off: Pushing the button on the mbed causes it to reboot, but to turn power off on the mbed, disconnect the USB cable. The small USB connector on the mbed module is fragile,

so disconnect the larger USB connector that plugs into the PC. If the blue LED is on, you have power to the mbed. *Always turn off power when changing wiring on the breadboard!*

If you have never used a breadboard before or if you need to review its underlying connection patterns, please go to <https://mbed.org/handbook/Breadboard> for a summary.



LPC1768 Pinout Diagram

Task 3: Download and run a program

After you activate your account, go to the quick start guide for the Arm Mbed OS <https://os.mbed.com/docs/mbed-os/v5.11/quick-start/index.html> (choose the online compiler option) and step through the tasks: run the example on the mbed board by saving the program to the mbed flash drive and pushing the blue button on the mbed board to reset. The mbed will always start running the most recently downloaded program. To go directly to the compiler from the <http://developer.mbed.org/> website, select Compiler in the upper right.

Note that the libraries we use to interface the MBED with peripherals are written in C++, but for the purpose of this class, the game engine and related code are implemented strictly in C.

Task 4: Build the circuit and run test programs

In this task, you will create an embedded system in which the mbed microcontroller interfaces with a color micro-LCD display, an accelerometer, a microSD card reader, push-buttons and a small audio speaker. Then, you will download a few test programs to ensure that your system is wired correctly.

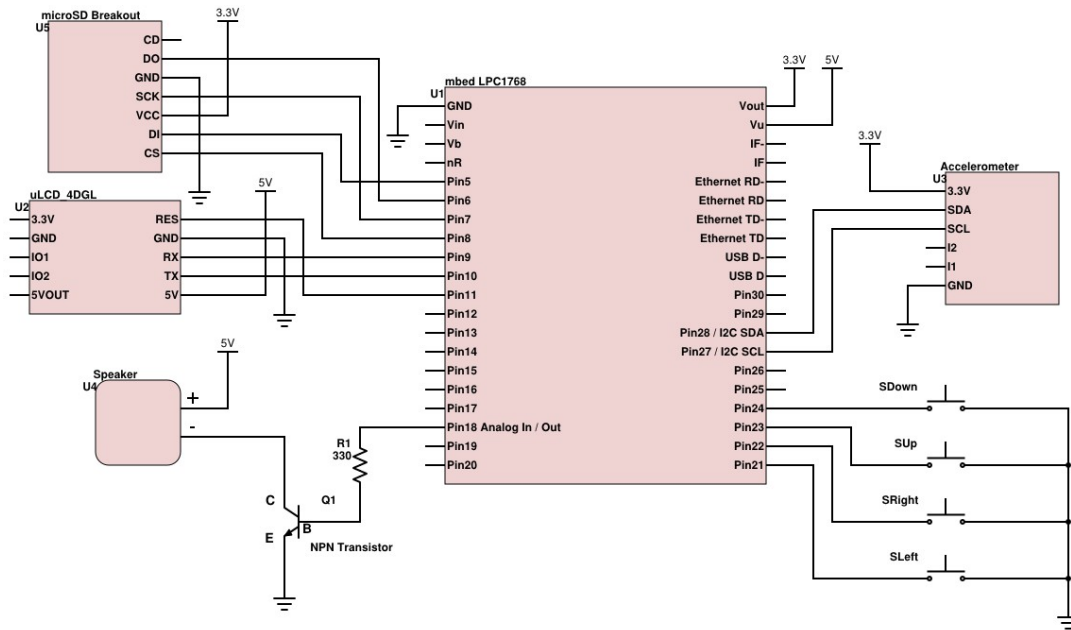


Figure 1: Wiring Diagram

Note the difference between VO and VU. Also be careful to properly orient the BJT transistor (see Fig. 3).

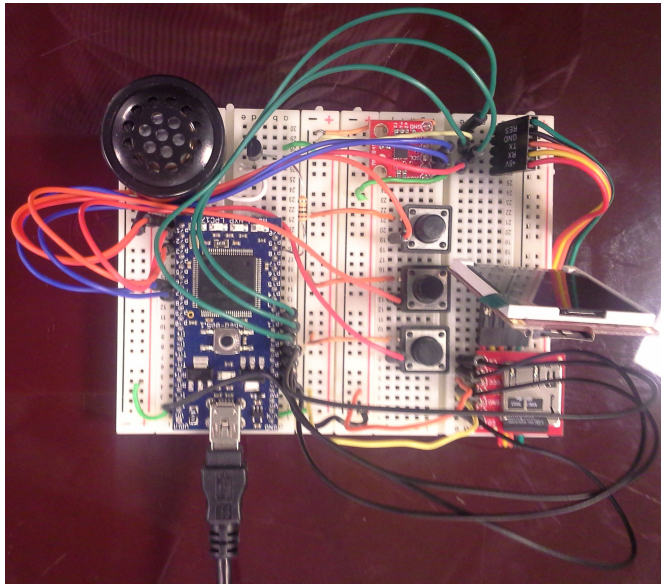
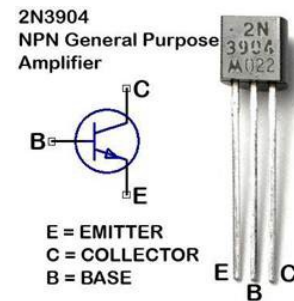


Figure 2: Completed Circuit

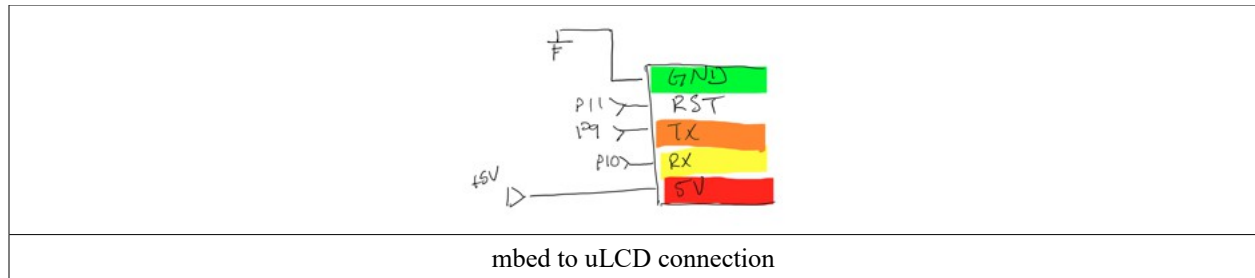


Please note the proper orientation of the BJT transistor.

Figure 3: BJT Transistor

Task 4A: The first step is to wire up the components that are provided with the mbed. Follow the diagram in Figure 1 to wire the mbed microcontroller to each of the components that we will be using in this project. **Note:** The speaker is connected to pin 18 in the schematic (via a BJT). Change the wire connect to pin 18 to pin 26 to use PWM audio in the audio test below in Task 4B. We have let the SD Card in the circuit schematic in case you wish to use it to store sound files, but you do not need to wire it in or test it unless you actually use it in P2-2. In that case, you would need to use pin 18 for audio.

An example of the completed circuit is shown in Figure 2.

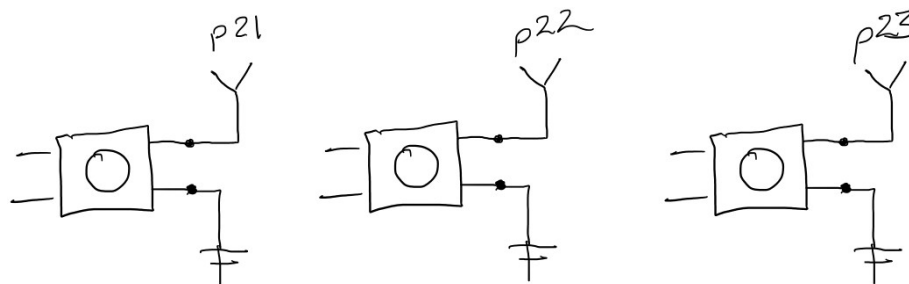


Also, note that the transmit (TX) pin of the mbed (pin 9) connects to the receive (RX) pin of the uLCD and the RX pin of the mbed (pin 10) connects to the TX pin of the uLCD. The ribbon cable that is used to connect the mbed to the uLCD has labels (RX and TX) that refer to the *mbed's* RX and TX pins, not the uLCD's pins. The ribbon cable is plugged into the bottom row of pins on the uLCD (the row closest to the edge).

Please see http://developer.mbed.org/users/4180_1/notebook/ulcd-144-g2-128-by-128-color-lcd/ for close up views of the uLCD pins and the uLCD ribbon cable with details of the proper wiring.

Task 4B. Run the following test programs:

1. 2035_P2_PB_LED: This test program uses the existing 4 LED lights on mbed, and 3 pushbuttons to create runway light effects.



Pushbuttons

Behavior

p23: increase the frequency of lights

p22: pause the runway lights

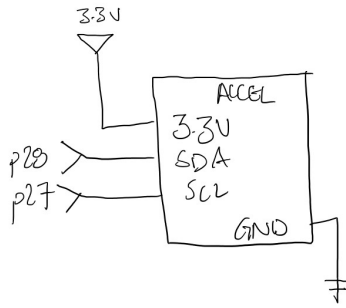
p21: decrease the frequency of lights

Reference: https://os.mbed.com/users/4180_1/notebook/led-lighting-effects-for-modelers/

2. 2035_P2_LCD: This test program displays a bouncing red ball through uLCD display.

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

3. 2035_P2_Accelerometer_LCD: The test program displays a red ball on the uLCD screen. Different from 2035_P2_LCD demo, the red ball moves on uLCD screen based on accelerometer readings. LCD and accelerometer connections should be the same as illustrated in the reference.



Accelerometer

References: <https://os.mbed.com/components/MMA8452Q-Triple-Axis-Accelerometer/>

<https://learn.sparkfun.com/tutorials/mbed-starter-kit-experiment-guide/experiment-4-accelerometer>

4. 2035_P2_Speaker_PWM:

The test program uses mbed's PWM hardware output to play 3 different tones triggered by pushbuttons. The speaker should be set up according to the reference.

Behavior

p21: 100Hz tone

p22: 300Hz tone

p23: 500Hz tone

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

Task 4C. Download **P2-baseline.bin** and play with the baseline version of P2.

This game is a simplified version of the Gameboy Advance versions of Fire Emblem. It is a two player turn based strategy game set in a 2D grid world with a top down view.

In this baseline implementation, there are two players that each control three characters, denoted with red and green squares, respectively.

You start out in free roam mode, in which you can freely navigate the world by tilting the mbed (making use of the accelerometer) to move your gray cursor. You can either use the action button on a character that you control to select them, or use the back button to end your turn.

Once a character has been selected, you can move them up to a certain distance away (configurable for each character). If the character has been moved next to an enemy character, they fight and deal damage to each other according to their stats (each character has an attack, defense, and hit points).

Once a character's hit points (HP) reaches zero, they die and are removed from the map.

The objective of the game is to kill all of your opponent's characters.

The red characters (controlled by player one) are viewable at the start of the game. The green characters, controlled by player two, can be found to the southeast.

Homework 3 Check-off

Once you have successfully created the circuit in Figure 1, and can successfully run the test programs, **demo your working system to a teaching assistant by the deadline** to get Homework 3 checked off for a grade.

References

1. A. Vance, "ARM Designs One of the World's Most-Used Products. So Where's the Money?" *Bloomberg Businessweek*, 4 February 2014, available online at: <http://www.bloomberg.com/bw/articles/2014-02-04/arm-chips-are-the-most-used-consumer-product-dot-where-s-the-money>
2. Mbed Pinout:
<http://developer.mbed.org/handbook/mbed-NXP-LPC2368>
3. Mbed-NXP Pinout:
<http://developer.mbed.org/users/synvox/notebook/lpc1768-pinout-with-labelled-mbed-pins/>
4. NXP-LPC1768 User's Manual:
http://www.nxp.com/documents/user_manual/UM10360.pdf