ECE 4180 Lab 1 – Basic I/O with mbed

(dates might change later based on final kit distribution dates)

Section A Due Date: Feb 1 odd groups – Feb 2 even groups

Section B Due Date: Feb 3 odd groups – Feb 4 even groups

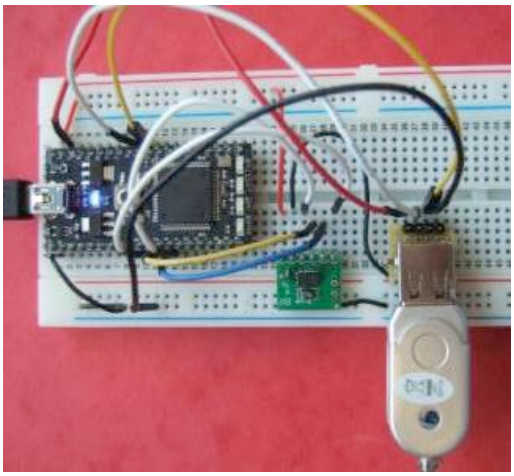
Names:_____**Sect:**_____ _____**Sect:**_____

Item	Lab Demo	Extra Credit	Extra Credit Pts
4180 LAB 1			
Digital I/O	yes	n/a	n/a
Digital I/O using PWM	yes	yes	1%
Digital I/O with Port Expander			1%
Navigation Switch (Joystick)	yes	n/a	n/a
Touch Keypad		n/a	n/a
WatchDog Timer	n/a		2.5%
ARM Assembly Language	n/a		4%
RGB LED	n/a	yes	2.5%
Power Management	n/a		4%
Two MCP23S17s on one Bus	n/a		2.5%
Early Bird Checkoff	n/a		2%

Even/Odd group names rule: Add the ASCII value of the first character of last names for each student in the lab team and do % 2 to determine even (0) or odd (1). Using the official last name found on the class roll. In the case of students in a lab group that are in different sections, the first due date applies

Final TA Signoff: _____ **Total Points** _____

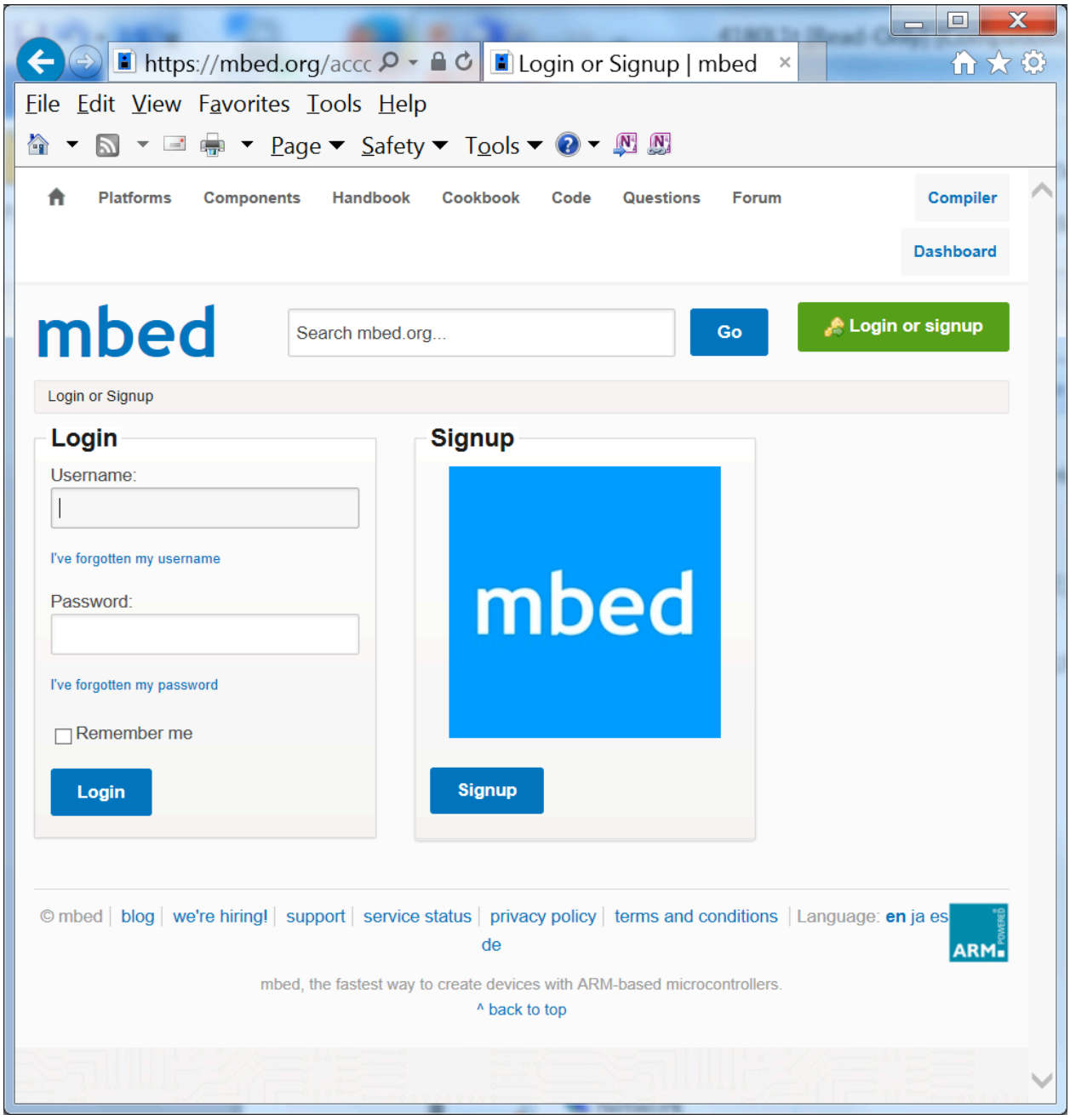
Note: Bring your student breadboard to the lab and any jumper wires you may have. We have around a dozen large breadboards in the lab that can be used. It will not be possible to leave your setup using the breadboards in the lab, if you don't finish once you start with the large numbers of students in this class.



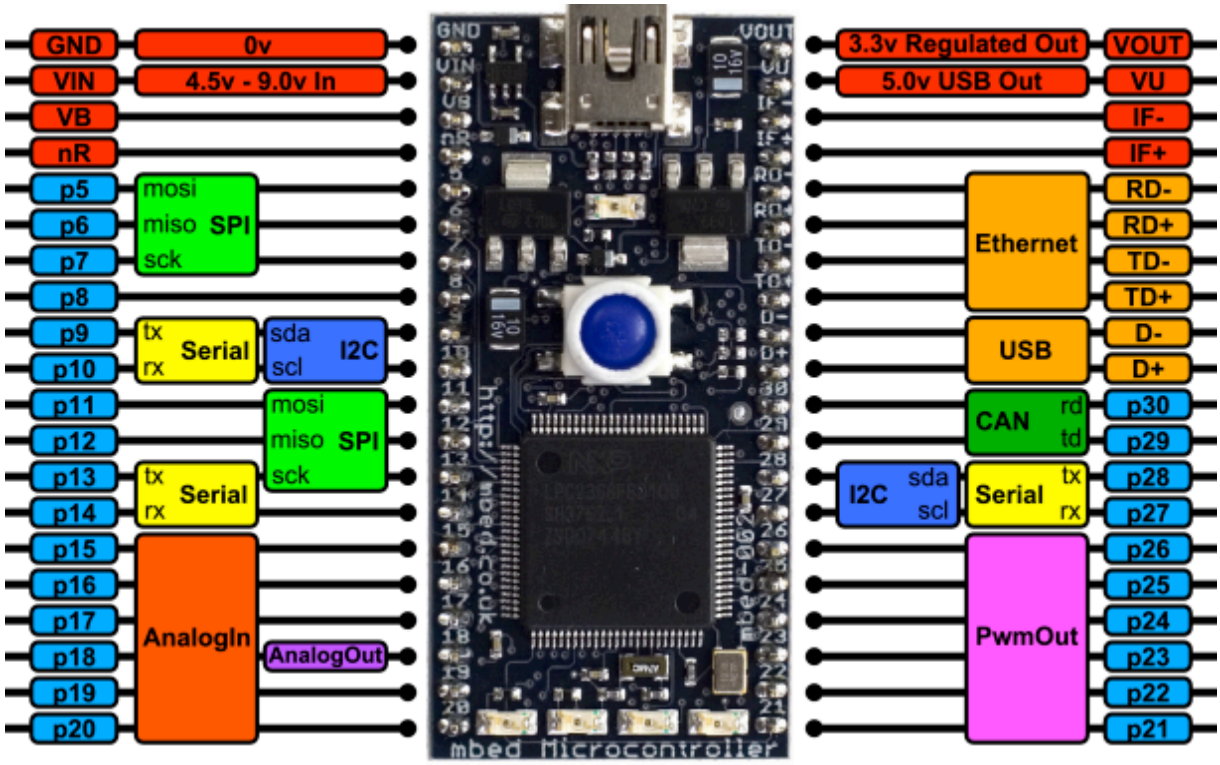
The purpose of this lab is to build and demo several types of widely used I/O interfaces to a small microcontroller. Since this lab uses several input devices in your kit, it would be helpful to read and/or review the wiki page on common user input devices at: https://developer.mbed.org/users/4180_1/notebook/user-input-devices/

Getting Started with the mbed module

The easy to use mbed LPC1768 modules www.mbed.org have an ARM processor, use a web-based C++ compiler, and work like a USB flash drive. To get started, **plug the mbed USB cable into the PC** and **click the *MBED.HTM* link** on the flash drive that appears. The mbed must be attached to signup for a new account. **Click signup,** and create an new account (if you don't already have one). You will also save your source files on the cloud server using this code. You will need to setup and logon this account to compile on the web. The signup page is something like the one seen below (might have some minor changes).

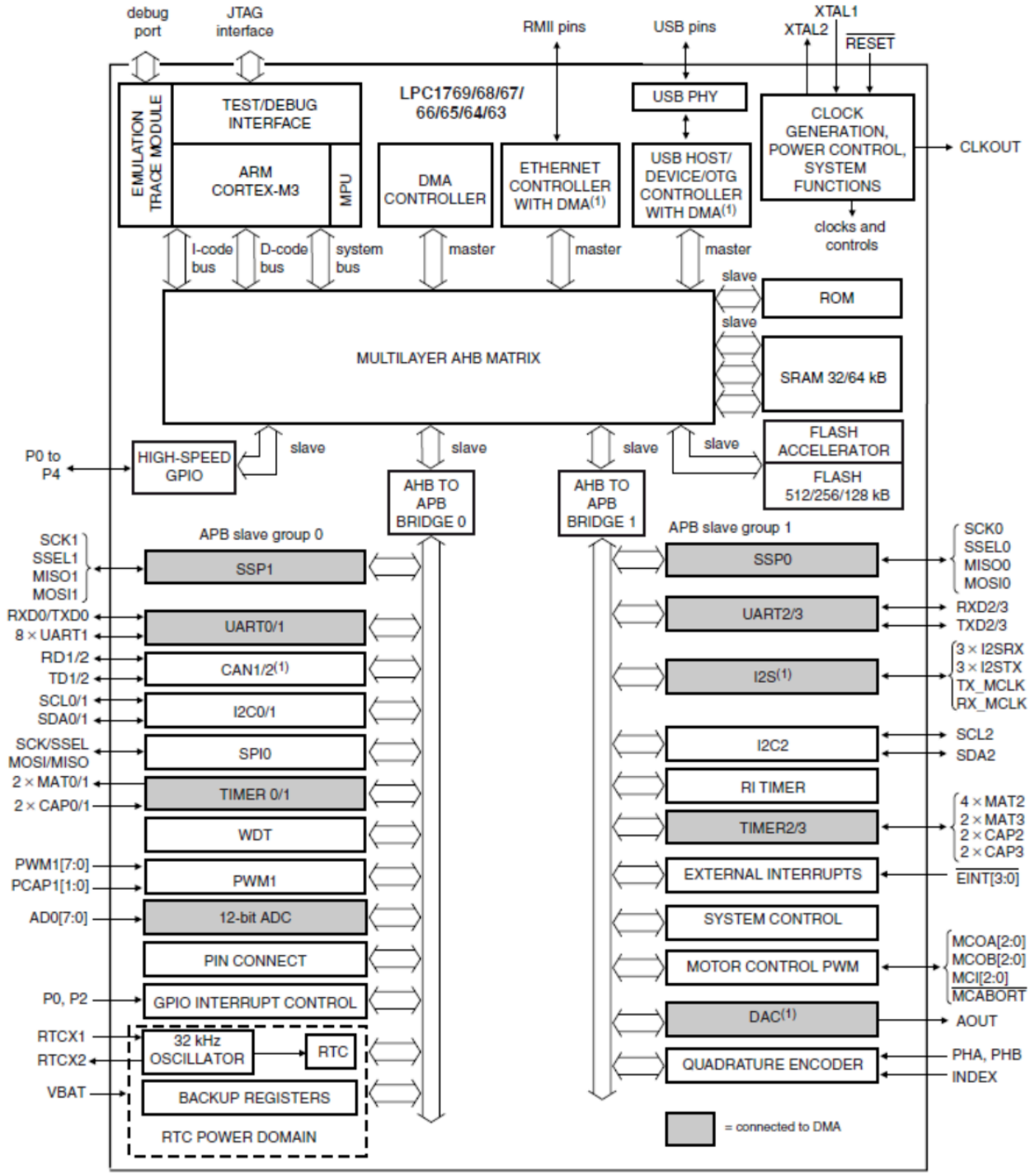


After you signup and create your account, run the *HelloWorld* example on the mbed board, save the program to the mbed flash drive, and push the center black/blue button on the mbed board to reset. This will start running the most recently downloaded program. The program will flash a blue LED once it runs. Then switch to the compiler window to write code. Read through the *HelloWorld* C code and note the use of *DigitalOut* and *LED1*. If you cannot find HelloWorld in your Program Workspace in the online compiler, click on "New" and enter "HelloWorld" for the Program Name and click "OK" to create a new project with the demo code.
If you want your own or an extra mbed LPC1768 modules from the kit, they are available at www.sparkfun.com or www.digikey.com for $49. The mbed handbook has info and examples using the built-in I/O APIs. The mbed cookbook has more complex project examples. The TAs have a few of the mbed modules that can be borrowed for use in the lab only, for those still waiting on kit delivery. If you ever need to remove one from the breadboard, remember the cautionary notes on the back of the next page.



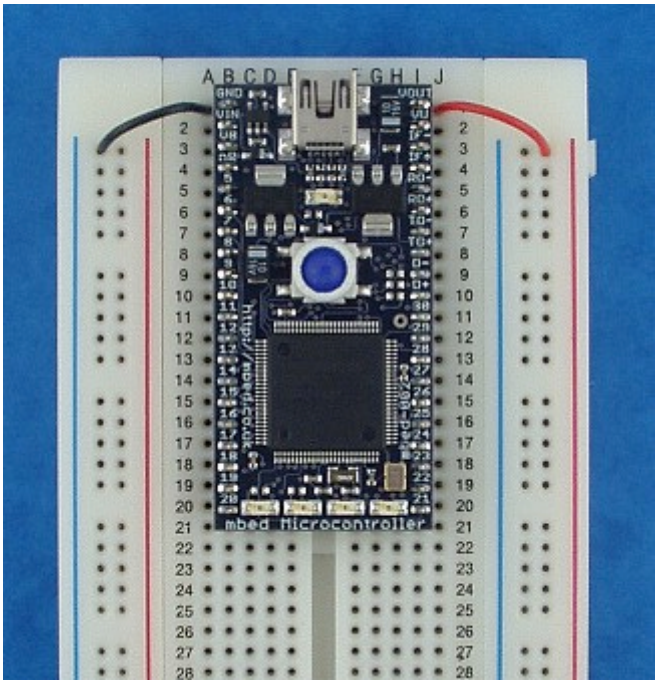
Here is a handy image and pin out of the mbed board. In C++ code, you can use the pin's name to read or send a value to the pin. Note that power (3.3V and 5V up to 460MA) is available for the protoboard from the upper pins. On chip power circuits produce this voltage using power from the USB cable (500MA max). *Vin* is for an external power source like a battery – don't connect anything to it while using the USB cable. *VB* is for a real-time clock battery backup, so skip it. There are four blue user LEDs at the bottom of the board (*LED1 – LED4*). They are handy for status displays. Over the USB cable, you can also set it up to send serial data to the PC using *printf*.
This mbed Microcontroller is based on an ARM Cortex-M3 Core running at 96MHz, with 512KB FLASH, 64KB RAM and a load of interfaces including Ethernet, USB Device, CAN, SPI, I2C and other I/O. RS232 serial, CAN, USB, and Ethernet outputs require a breakout board with a connector. Many of these are available from Sparkfun and we have an assortment available in the lab in addition to those in your kits.

A schematic of mbed is available at http://mbed.org/media/uploads/chris/mbed-005.1.pdf and the 32-bit NXP ARM Cortex-M3 LPC 1768 processor data sheet and a user manual is also available at http://ics.nxp.com/products/lpc1000/datasheet/lpc1763.lpc1764.lpc1765.lpc1766.lpc1767.lpc1768.lpc1769.pdf , but you will likely never need it. The various devices have I/O registers on the machine, but the handy built-in C++ API calls for I/O hide the nasty 78 pages of I/O register details with another layer of abstractions. The LEDs will flash on the bottom of the mbed board if you ever have a compiler run-time error. The most common way to get this is to try to setup a pin to do something that is not supported in hardware (that function is not on color coded mbed card!)



NXP LPC 1768 Block Diagram

Note the wide range of I/O features built into mbed's ARM Cortex–M3 processor chip. They include GPIO, ADC, DAC, Serial (UART), SPI (SSP), I2C, PWM, USB, and Ethernet. We will be using all of these in the lab exercises that follow.



Inserting an mbed into a breadboard

Note: The mbed Microcontroller is fragile, and inserting it in to breadboard will require some force; ensure you line up the holes, then press evenly until it goes in to the board fully. Mess up and pins will break!

Removing an mbed from a breadboard – Use Caution!

Never try and remove the mbed Microcontroller by pulling on the USB lead or USB connector. You'll just 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, be patient and do it carefully. Use a small jewelers screwdriver or similar, and lever it a bit at a time 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 microcontroller (which might damage the components on the underside).

So remember..

- Don't pull on the lead/connector (it will just pull off the connector; you should avoid there being force on the connector at all times)
- Don't pull it out from one side (you will bend and break the pins!)
- Don't lever the device where it touches the bottom of the microcontroller (there are lots of components there, and you will damage them!)

Turning Power Off

You can push the button on the mbed to reboot, but to turn power off on the mbed board disconnect the USB cable. The small USB connector on the mbed module is a bit fragile, so use the larger USB connector that plugs into the PC. If the blue LED is on you have power to the mbed. **Always Remove power** when changing wiring on the breadboard!

ECE 4180 Laboratory Assignment 1

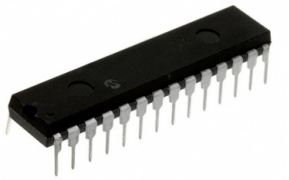
Part 1 - Basic Digital Input and Output (30%)

Any numbered pin, *Px,* can be used for general purpose digital I/O (GPIO) but try to stay clear of the analog, serial, SPI, networking, and I^2C pins, they will be used later. Attach an external switch input using a DIP switch module and 2.2K ohm pull-up resistor and at least one LED using a discrete LED or an LED DIP module. Don't forget the resistor in series with the 2.2V LED to drop the voltage a bit. 100 ohm seems to work well on blue LEDs. Note that the mbed takes power from USB and has power pins to power external devices. Use 3.3V by default since the chips I/O pins use 3.3V and not 5V level logic. Write C++ code to read in the DIP switch and output to the LED and loop forever. http://mbed.org/handbook/Homepage has a handy list of built-in C I/O API calls. Check out *DigitalIn* and *DigitalOut* in the mbed handbook. Use the external switch pullup resistor for this first lab assignment, but there is a way to avoid it later and even debounce the pushbuttons that is explained in the pushbutton tutorial at http://mbed.org/users/4180_1/notebook/pushbuttons/. Don't forget that an LED should have a voltage dropping (100-220 ohm) resistor in series with a digital logic output pin that controls it as they typically want to see only around 2 volts. https://developer.mbed.org/users/4180_1/notebook/rgb-leds/ has more info on dropping resistors and exactly how this value is calculated. Extra switches, LEDs, pushbuttons, and resistors are in a gray parts cabinet in the lab on a scope cart (left center back of lab near TA desk).

Part 2 - Using PWM with Digital Output (30%)

Add code to dim and brighten an LED using Pulse Width Modulation (PWM) using additional switches or pushbuttons for control. Check out the *PwmOut* API in the mbed handbook. Note that it is more energy efficient to control devices using PWM instead of an analog output. LEDs also emit light only in a very limited voltage range (2-3V depending on color). Using analog out will dissipate more heat in the transistor driver circuits than PWM since the transistors are not driven to saturation. As an example, if an analog driver circuit is turned on half way, this means that half of the power is dissipated as heat in the driver transistor.
Extra Credit (1%) Monitor the PWM output and duty cycle using an oscilloscope in the lab or a MyDAQ. Capture the scope image show it to the TA or show a copy of a screen image at checkout.

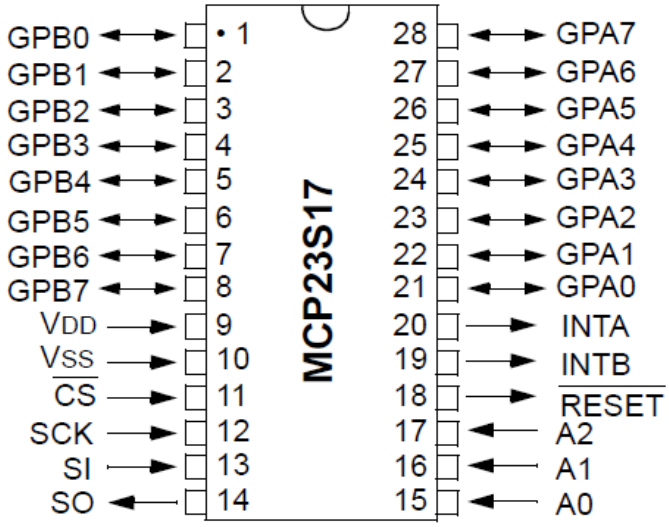
Part 3 - Expanding Digital I/O ports (20%)



When extra digital I/O pins are needed in small microcontrollers, I/O expander chips are typically used. The MCP23S17 IC is an addressable 16-Bit I/O Expander with an SPI interface. SPI is a low cost 1-bit serial bus with a clock that works just like a shift register. As seen in the image above, is available in a 28-pin DIP package that will plug directly into a protoboard. The mbed cookbook has a link for a MCP23S17 and a library code example using this chip at http://mbed.org/users/romilly/notebook/mcp23s17-addressable-16-bit-io-expander-with-spi/ with test demo code at http://mbed.org/users/4180_1/programs/MCP23S17_Basic_IO_Demo/llprau . The library code uses the mbed compiler's *SPI* API function for low-level operations and adds a layer of functions to configure, read, and write the MCP23S17 chip using mbed's built-in hardware SPI controller. A datasheet for the MCP23SA17 chip can be found at http://ww1.microchip.com/downloads/en/DeviceDoc/21952b.pdf. There is one of these ICs in the optional kit and we have some available in the lab.
Be sure to correctly configure the reset pin on the MCP23S17 chip and note the wiring suggestions on the next page. Reset cannot be just left floating high on this chip per the data sheet.

Repeat Part 1 using the I/O expander IC by attaching an LED to a bit on an output port, a switch to a bit on an input port, and writing code to read the switch and output to the LED. The MCP23O17 is an I^2C bus version of this IC and we also have some of them in the lab for use later in design projects. Make sure you don't get the wrong chip from the parts bin!

Extra Credit (1%) Capture an input and output SPI bus cycle (clock and data lines) using a logic analyzer or the mixed signal oscilloscope in the lab or a MyDAQ if you have one. Be sure to demo it on the scope or LA to the GTA, or capture the image and hand it in later at checkout. . If you need to review how to use a logic analyzer a tutorial is available.



mbed	MCP23S17
GND	A0, A1, A2
P20	/CS
P14	/RESET
P5, P6, P7	SI, SO, SCK
3.3V	V_{DD}
GND	V_{SS}

Parts 4 & 5 Demo the other two input devices from your kit

Part 4 - Hookup and run the navigation switch demo program. The mbed's built-in LEDs show which switches have been activated. **(10%)**

Details on wiring and code is provided on this mbed cookbook wiki page:

https://developer.mbed.org/users/4180_1/notebook/using-a-navigation-switch-digital-joystick/

Part 5 - Hookup and run the touch keypad demo program. The mbed's built-in LEDs show a binary number for the key has been touched. **(10%)**

Details on wiring and code is provided on this mbed cookbook wiki page:

https://developer.mbed.org/users/4180_1/notebook/mpr121-i2c-capacitive-touch-sensor/

Don't forget the **required I2C pullups** (per instructions on wiki page)!

Additional Extra Credit Options for Lab 1

(2.5%) Add a WatchDog timer example to your Part 1 demo. Include a simulated fault that occurs after a few seconds. See http://mbed.org/cookbook/WatchDog-Timer for more info and examples.

(4%) Do Part 1 by using only ARM assembly language. See http://mbed.org/cookbook/Assembly-Language for more info and examples.

(2.5%) Like Part 2, write code for the new RGB LED or the RGB LED on the ShiftBrite Module (in older mbed kits), to vary the RGB output color using the blue pot found in the optional parts kit. The pot can be setup as a voltage divider to produce an analog input. Analog inputs can be read using the mbed AnalogIn API. See https://developer.mbed.org/users/4180_1/notebook/rgb-leds/ and/or http://mbed.org/users/4180_1/notebook/shiftbrite1/ for more info and examples. https://developer.mbed.org/users/4180_1/notebook/led-lighting-effects-for-modelers/ has an mbed example of how to use the pot as a voltage divider (first code example on page).

(4%) Use Power Management to minimize the power consumption on your Part 2 demo. Hook up an external 5V power supply to V_{in} (with USB cable not connected!) and monitor the current saved. A couple of the lab power supplies display current or a Voltmeter could be used. See http://mbed.org/cookbook/Power-Management for more info and examples. There are new power management APIs coming for mbed, but they are not available yet.

(2.5%) Connect two MCP23S17 devices to a single SPI bus without toggling chip select. Do the Part 3 LED and switch demo with the LED on one MCP23S17 IC and the switch on the other MCP23S17 IC.

Early Bird Bonus (2%) extra credit to first four groups to demo the regular lab 1 assignment including the extra credit options. The first couple of lab teams to finish it will likely wind up doing a bit of extra work blazing a path forward for everyone else. So they will get extra credit.