ECE 4180 Lab 2 – Advanced I/O with mbed

Section B Due Date: Sept 24 even groups – Sept 25 odd groups
Section A Due Date: Sept 26 even groups – Sept 27 odd groups

Names:______________________________ **Sect**______ __________________________________**Sect**______

Item	Lab Demo	Extra Credit	Max EC Pts.
4180 LAB 2	n/a	n/a	n/a
I^2C IMU and USB Virtual Com Port		n/a	n/a
Legacy RS232 Serial	n/a		**2%**
Serial uLCD Graphics Bubble Level		n/a	n/a
uLCD Compass	n/a		**2 %**
Ethernet Networking		n/a	n/a
Relay Driver Circuit for DC motor		n/a	n/a
MOSFET DC motor speed control		n/a	n/a
Solenoid Driver Circuit	n/a		**1%**
Reversible DC motor with speed control		n/a	n/a
Stepper motor control	n/a		**2%**
RC Servo with PWM control		n/a	n/a
MEMs Microphone Audio Input		n/a	n/a
MicroSD Card and USB File I/O		n/a	n/a
Speaker with Class D Audio Amp		n/a	n/a
Mbed Wi Fi and IoT Web Control			**1%**
ToF Distance Sensor		n/a	n/a
Extra Credit Options at End of Lab			
RTC display on LCD	n/a		**1%**
Electronic Compass + switch & correction	n/a		**2 + 1%**
Scanner using Servo and ToF Sensor	n/a		**2%**
Robot Dance	n/a		**2%**
HTTP mbed web server + image	n/a		**2 + 1%**
USB device on mbed	n/a		**1%**
IR and RF data transmission	n/a		**2%**
Relay Board or Power Switch Tail	n/a		**1%**
Internet of Things LCD gadget	n/a		**2%**
Add Robot motor feedback control	n/a		**2%**
Early Bird Bonus	n/a		**2%**

Final TA Signoff: ______________________________________ **Score:**____________________

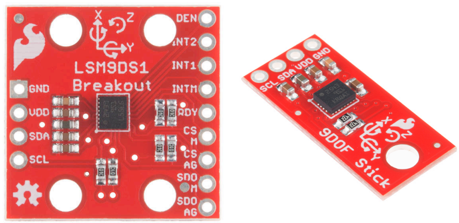
Note: Try to keep the stations organized. If you checkout a part, once you are done with a part return it to the TA. If you make a mess**, clean it up**!

ECE 4180 Laboratory Assignment 2

Most new ICs are only available in small surface mount packages that will not fit into a student breadboard. We will be using several modules with surface mount ICs mounted on small PCBs (printed circuit boards) called breakout boards with 1/10 inch pin spacing that will allow them to plug directly into a breadboard. Many of these boards are in the kit and those that are not are available for checkout in the lab. You can check off lab parts one at a time or demo the entire lab to the TA, assuming you still have everything hooked up and have a big breadboard. If you are missing some parts or kill one off, most come from www.sparkfun.com and a few from www.adafruit.com. There are a few spares available for lab use only that are available for checkouts from the TAs.

Soldering: Currently the **Blue Adafruit boards** in the 4180 parts kit will need to have the header pins soldered on. If you need soldering equipment or help from TAs, soldering stations are available in VL C356 and the sr. design lab in VL E356. They also have some soldering classes in there for those that are interested, and here is a soldering tutorial.

Part 1 - I^2C bus IMU and USB virtual com port (10%)



Mbed LPC1768	IMU
3.3V(Vout)	VDD
GND	GND
P9	SDA
P10	SCL

LSM9DS1 IMU breakout boards and wiring

Lab Demo: Show the IMU's data readings being printed to the USB COM port, updating once or twice a second.

Additional Details: The LSM9DS1 (new lower cost smaller breakout board on the right is in new kits) is a 9 degrees-of-freedom Inertial Measurement Unit which contains a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer as well as a temperature sensor on a single chip. Until recently this took several ICs and cost nearly $100. These sensors can be used to determine a very precise orientation and are useful for applications such as small aircraft/drone autopilots and robot position, direction, tilt, and balance. More information about the LSM9DS1 can be found at https://developer.mbed.org/components/LSM9DS1-IMU/, MEMs IMUs sensor data has a bit more noise than the high price IMUs that can cost several thousand dollars. They need to be calibrated, sampled at a high rate, and filtered a bit to obtain the maximum level of accuracy possible. Kalman filters for gyro and accelerometer readings are sometimes used in IMUs and the filter design is a bit tricky for IMUs! Mahoney and Magwick IMU filters require a bit fewer computations. In the end, it takes about a page of C float code. Filters can sometimes slow down the response time too much in some applications. Keep this device several inches away from magnets or coils (including DC motors) or the magnetometer readings will be way off. There are also special algorithms with some trig calculations to get the correct compass heading from a 3-axis magnetometer that may not be perfectly horizontal. Accelerometer readings can detect the gravity vector when a device is not moving.

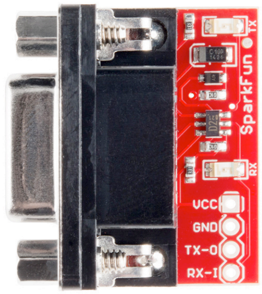
The breakout board from SparkFun (see https://www.sparkfun.com/products/12636 for datasheets and schematics) includes pull-up resistors on the chip-select, address, and I^2C lines, so you only need to connect SDA and SCL (the I2C lines) to appropriate pins on the mbed, as well as Vdd (3.3V) and ground. **Be very careful** – a 3.3V device tied to 5.0V will **instantly fry the chip** and these two pins are next to each other on mbed. If you ever hookup an I2C device to mbed without pullups, the mbed APIs lookup waiting forever a signal change that never occurs!

Import the more complex demo program **at the very end of the wiki page** that tries to calibrate the IMU for more accuracy and scales the all readings into meaningful units into your mbed cloud compiler from https://developer.mbed.org/components/LSM9DS1-IMU/, **change the I2C pins from p28,p27 to p9,p10 in the code,** compile and run it on the mbed. Start a terminal application window on the PC to monitor the output from mbed. Run the demo code, follow the calibration instructions and wait for it to start printing readings. When you tilt or turn the IMU, the readings should change. Leave the IMU connected on your breadboard/ It is used again in the next two parts of the lab.

Info on Terminal Application to monitor mbed USB output: Tera Term (easier to use), or Realterm (a lot more hardware debug features that are very handy when serial things don't work) which should be on the lab computers already, are free terminal programs for Windows which let you connect to the mbed's COM port and send/receive serial data via USB. Get one or both for your PC, if working outside of the lab. Create a new connection, and select the mbed's virtual COM port from the list. If your line endings are causing text to go diagonally across the screen, click *Setup->Terminal*... in the menu, and change Receive New-Lines to Auto rather than CR. If you're using your own PC, you will need to install the mbed usb serial port drivers first before the mbed com port appears. Most mbed serial code uses the default baud rate of 9600 – but some code uses other settings to speed up data. The baud rate must be correct or you will see only garbage characters or perhaps nothing at all. Mac setup info is also at the mbed Wiki. If an IMU supports both SPI and I2C, SPI is probably a better choice since it is a bit faster.

Part 2 - Legacy RS232 Serial (Extra Credit 2%)

Mbed	RS232 adapter
3.3V	VCC
GND	GND
RX	TX
TX	RX



An RS232 voltage level conversion breakout board and a null modem adapter.

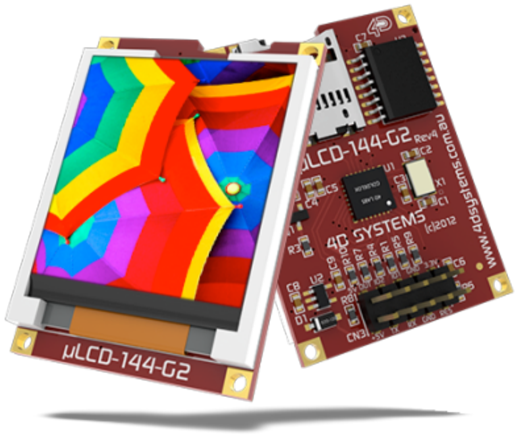
Lab Demo: Show the same functionality in Part 1, but use the RS232 serial port instead of the USB virtual serial com port to display the IMU data.

Additional Details: Many I/O devices still require the older legacy EIA RS232 standard serial interface that requires different voltage outputs. The two parts above and a PC serial cable are available for checkout from a lab TA. In addition to the virtual serial com port, output the compass heading info from a "real" RS232 serial port to the PC's COM port. Keep in mind that it is **not using the USB cable for data transfer** unlike the previous hookup (even though the USB cable may still be plugged in, it is only providing power)! Many devices do not have a USB cable. Attach a serial cable to the PC and display it using a terminal application program running on a real COM port (i.e., not the USB virtual com port). RS232 voltage level conversion (i.e., +3 and –3 voltage) is required and some very handy small RS232 SMD breakout board level conversion adapters from Sparkfun http://www.sparkfun.com/commerce/product_info.php?products_id=449 are available in the lab. If these were not available, you would need to use an IC like a MAX232 and some caps on the protoboard to adjust the voltage levels. Some devices on the same PCB use serial, but with TTL logic levels. If it says "RS232" it is not TLL level and the serial voltage convertor will be needed! Most RS232 devices use the DB9 connectors like those above.

Be aware of possible null modem serial cable issues. There are some small null modem adapters and gender changers in the kit and the lab as seen in the image above on the right, if you need one. Add this output code to your C code to the program loop. There are "real" serial port code examples in the handbook and cookbook and check out the *Serial* API. You cannot just switch the wires on the mbed's *TX* and *RX* pins to make a null modem connection (i.e., a wire is bidirectional, but the voltage conversion circuit is not). **If you are outputting from the MBED the red RX light should blink, while if you are reading the green TX light should blink on the Sparkfun adapter**.

The PCs in the labs have a com port and serial cables are available in the lab. If you want to work at home and do not have a PC serial cable, the TA can let you check out and borrow a serial cable from the lab. Gender changers are also available in kit and the lab, if needed. A few PC cables already swap TX and RX (i.e., null modem cable) and do not need a null modem adapter. If would be too easy if they just always marked on the cable which type it is someplace! If your laptop or PC at home does not have a serial port, we have some USB to serial cable converters you can borrow in the lab, but most don't like Windows 8 or 10? These need a new device driver installed and it may not automatically do it. Windows 8 and higher does not like the older low-cost USB to serial converters – according to the web blogs, most have a counterfeit USB to serial chip from China and the new Windows 8-10 drivers check for it and stop working. If you have Windows 8 or 10, and still want to work at home, real USB to serial cables that work with Windows 8 and 10 are available for around $15. Make sure it says that it "works with Windows 8 or higher," on the box or web page. Be careful, the web blogs make it sound like there are more counterfeit than real ones out there! The two companies (Prolific and FTDI) that make the "real" USB to Serial chips and that paid to develop the drivers have web sites with links to drivers and some places to buy one of their "real" cables. After they paid tens of thousands of dollars to develop the Windows drivers it is not surprising that they added driver code and starting checking for counterfeit ones. The newer blue USB to serial converter cables in the lab (from Adafruit) will work with Windows 10.

Part 3 - Serial TTL Level Color Graphics LCD display (10%)



Mbed	uLCD
Vu	+5V
p27	TX
p28	RX
GND	GND
p30	RES

Color Graphics LCD Module – Serial Interface with TTL Levels

Lab Demo: Leave your IMU from parts 2 and 3 connected, and add the uLCD wired as in the table above. It will be good if the X axis of the IMU points the same direction as the X axis of the uLCD. Using data from the IMU, draw a bubble level animation on the uLCD using data from the IMU.

Additional Details: Power on pin 1 (+5V) on the Color LCD in your parts kit **uses only the 5V mbed pin VU** (i.e., **not 3.3V – mbed pin Vout**) and pin 7 (GND) is connected to the mbed GND pin. Always double check power pin connections before turning on power for the first time – if you get them wrong it might burn out the device! See the Color LCD wiki page for additional help using the Color LCD with mbed. If you want to plug the LCD directly in a breadboard, it is necessary to carefully bend over one pin as shown on the wiki page. The LCD requires several wires and if it is placed near the mbed pins used, the breadboard setup will be easier. Note that the mbed pins used (27, 28, 29) are on the right side of the mbed module and these pins are a bit different than the wiki page example (LCD code can use any of mbed's three *Serial* outputs just by changing constructor pin number arguments), but use them to be compatible with the embedded systems you will build in labs that follow to have space for all of the parts added later. It turns out that it is more common to have more parts and the left side of the mbed module and this setup will leave space for them.

Bubble Level: When the IMU is flat, its acceleration vector points straight down (in the Z direction) due to gravity. When tilted, this vector gains non-zero X and Y components which can be used to measure how much and which direction the IMU is tilted. Draw a circle outline in the center of the screen, and a smaller filled circle which "floats" according to how much the breadboard is tilted (e.g. if you lift the right side up, the circle should "float" up to the right). There are bubble level and compass apps for smartphones, if you need to see how they work.

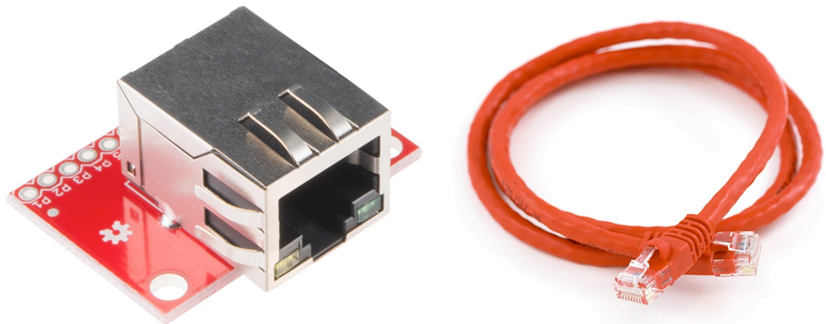
Graphics Tips: Rather than clearing the screen between drawing each frame, it is faster for your function to keep track of the position of the line/circle in the previous frame and draw over it in black before drawing the object in its new position. An IMU/LCD video demo can be seen at https://www.youtube.com/watch?v=2ydGgNEQ0oY that will give you an idea or how the graphics and sensor should work. The animation will look a bit better if a small time delay is added (i.e., *wait()*) so that the circle is visible a bit longer than it is erased. Be sure to jack up the baud rate on the LCD's serial port to have the fastest possible graphics update rate.

(2% Extra Credit) Compass: Using the compass heading from the IMU's magnetometer, display a compass needle centered on the screen, and pointing in the correct angle towards North. Also display the numeric heading in the corner of the screen. The lcd.line(…) function will be useful, as might lcd.circle(…). Without additional calibration to correct for nearby magnetic objects, magnetometer readings will likely be off a bit, but this is OK for the demo.

The uLCD is so handy, some people just always leave it hooked up for a new project.

Part 4 - Networking: Ethernet (10%)

Mbed LPC1768	Magjack adapter
TD+	P1
TD-	P2
RD+	P7
RD-	P8



An Ethernet Magjack Breakout Board and an Ethernet Cable

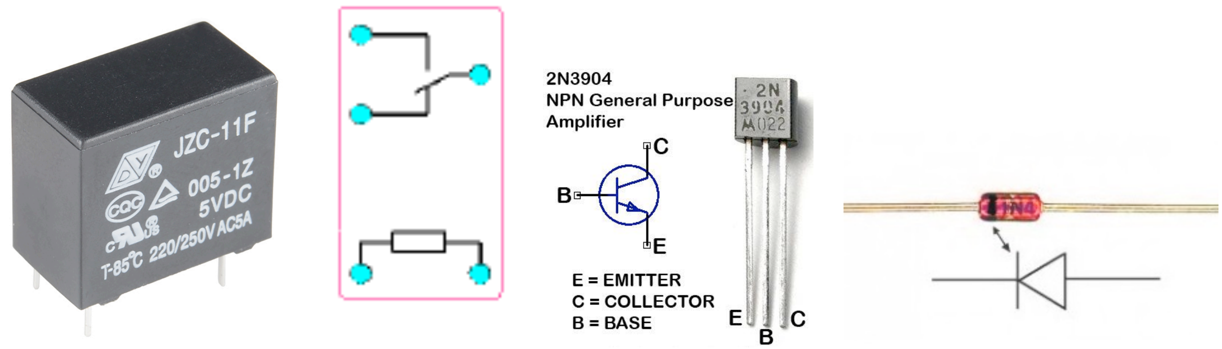
Lab Demo: Demonstrate an HTTP client executing a GET request to a server. It should use the USB virtual com port to display useful connection information.

Additional Details: The mbed LPC1768 contains an Ethernet controller and physical layer driver chip, but a magjack adapter must be added using a breakout board. http://mbed.org/handbook/Ethernet is probably a good test project to try first to see if your connection to the network jack is correct. It just displays network packets in a terminal application window. Magjack connectors from different companies look the same, but have different pinouts! The ones we have are shown in the table above. To learn more about networking on the mbed, http://mbed.org/handbook/Networking has a lot of useful information and includes the most recent officially supported libraries and network protocols. Mbed recently deprecated many of their networking libraries, so if you're running into a lot of compile errors, this might be your problem. The newest network software uses the RTOS and is more robust. The two networking libraries (old and new) have the same names and a ton of compile errors will occur if you somehow mix the various old and new libraries!

Note: Before mbed can connect to the Internet it needs a valid IP address. For DCHP to work and get an IP address for mbed, the mbed module's Mac or NIC address often needs be registered with the organization's servers to enable DHCP. If this is an issue it will likely get a timeout error during DHCP since it is not assigned an IP address. If you have a laptop with WiFi and a network jack you could setup a network bridge connection for the mbed module to get around this. Some network setups require a logon process using a web browser to get an IP address. There is not enough RAM on mbed to support a browser and usually another process is supported to enable small embedded devices to use the network. There is mbed code available to print out your module's MAC address to the USB virtual com port, if it is needed for network security setup. Also, it is possible to setup a network bridge on a laptop with both WiFi and an Ethernet jack. A network bridge is the recommended method for mbeds without registered MAC addresses running on a laptop.

While Wi Fi may be a more popular way to connect small devices to the network, Ethernet is still more reliable and provides higher bandwidth. An Ethernet Magjack connector actually contains tiny transformers and inductors for impedance matching and isolation along with two status LEDs (not used in lab code). See the circuit on the second page at https://www.sparkfun.com/datasheets/Prototyping/MagJack.pdf . If needed for debugging purposes, you can also display network packets with the free Wireshark application.

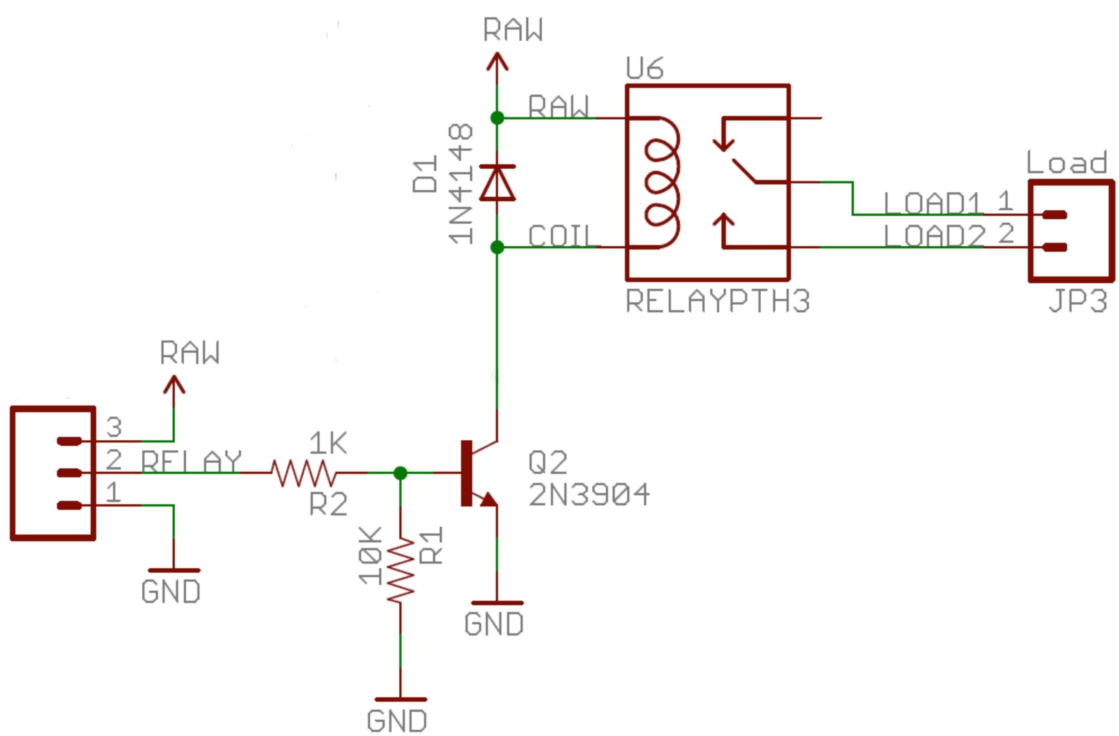
Part 6 - Driving high current devices with a relay (10%)



Parts Used from left to right: SPDT 250VAC/30VDC 5A relay, relay pinout (top view – coil is two pins on bottom, three pins on top are relay switch contacts), 2N3904 NPN transistor, and 1N4148 diode

Lab Demo: Turn a DC motor "on" and "off" using a relay

Additional Details: There are two parts in your kit that are used to turn "on" and "off" high current devices. The first is the relay shown above that will plug into the breadboard (only a few rare ones would ever fit – most need a breakout board or PCB). The pin schematic seen below next to the relay is a top view (pins are hidden below). The square box at the bottom is the coil used to control the relay contacts and the top shows the SPDT switch contacts that the coil controls. The NC normally closed (with coil off) switch pin connection is also shown (center switch pin connects to lower pin). When the coil is driven the switch pin connects to the upper pin.



Above: Schematic of a typical relay control board, Boxes at left and right are screw terminal blocks for wires. Breadboard jumper wires will only handle around 1.5 amps, screw terminal blocks allow hookup of larger gauge wires.

The relay coil will require a driver circuit to provide enough current to activate the relay's contacts. Use the 2N3904 small black transistor found in your basic mbed kit to turn the relay on and off. In the schematic above, *RAW* would be the 5V external power supply, *Relay* is the mbed DigitalOut control bit, and *Load 1,2* is a switch only that goes in series the DC motor leads (motor in series with relay switch contacts must be hooked between gnd and the external 5V DC power supply). Read the driver wiki page for additional background on driver circuits and relays how they work and how to hook them up. For the device or load to control with the relay, use one of the DC motors from the robot. For the motor, an **external power supply will be required** since the relay coil and DC motor draws too much current. It will take an additional 5V power supply to power both the motor and the relay driver circuit like the one in the new kits. Per the wiki page, a voltage suppression or snubber diode should be used across the coil. Connect the external power supply gnd to the mbed gnd. The kit does not have a diode, but extra diodes are available in the lab. Do not tie the unused NC relay switch contact pin to gnd, that can short the power supply. The 10K resistor is optional for the lab demo, it prevents a floating control input from turning on the relay.

Adding Additional Power to a Breadboard

The parts kit has a black breadboard barrel jack adapter for the 5V 2A AC adapter (or robot battery pack) barrel plug along with a small black breadboard power slide switch as seen below. These are handy to hookup and switch battery power to the breadboard. Don't use the tiny DIP switch for a power switch, the tiny metal contacts can't handle the current for very long and will burn out. The robot's battery pack is also a power option. On the battery pack, the inner jack conductor was +6VDC and the outer one was gnd. Before plugging in the AC adapter or adding batteries and turning things on, check all of your battery power connections to your breadboard with a voltmeter setup for ohms measurement! You can find datasheets for everything at Sparkfun.com, but it is just as easy to figure it out with the ohm meter. The barrel jack will fit the AC adapter in the kit. It should plug in and have a snug fit.

Caution: A reversed polarity battery/power supply hookup could burn out most of your chips!

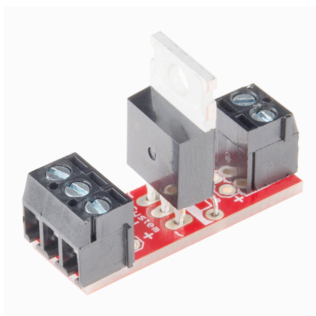
When external power is needed, use the 5VDC 2A AC adapter, the barrel jack adapter, and optional power switch.

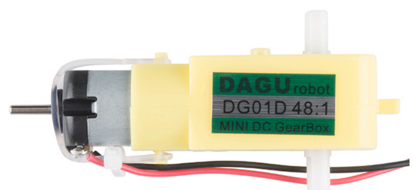
In the special case of a relay, the power supplies on the input and output side do not need to have a common ground. The input control circuit can be electrically isolated from the output device and have different voltages than the output. The output side can even switch a high voltage AC supply. In the lab demo, the grounds are connected since the driver circuit also needs more power and its logic control input signal from mbed needs a common ground reference.

Part 7 - Power Transistor Driver for DC motor speed control with PWM (one direction only) (5%)

Lab Demo: Use the power MOSFET board to control speed on the DC Motor

Additional Details: A relay switches far too slow to ever dim LEDs or control motor speed with PWM. For that, a transistor driver is needed. In the kit there is a 60V 30A power MOSFET breakout board as seen below. It has screw terminal leads since the max current levels it can support can be greater than a breadboard jumper wire can handle (>1A or so). The leads on the power MOSFET are also too large to plug directly in a breadboard. It could be used to drive the relay coil, but it would be engineering overkill since the small low-cost transistor can handle it. The schematic is available at Sparkfun and there is a section on it in the driver wiki page. Jumper wires can connect to the screw terminals. This MOSFET has a built-in diode, like many power MOSFETs. Use a screwdriver to attach jumper wires from the breadboard and motor leads to the breakout board. A pen style 4-in-1 pocket screwdriver or other small screwdriver comes in handy for this.



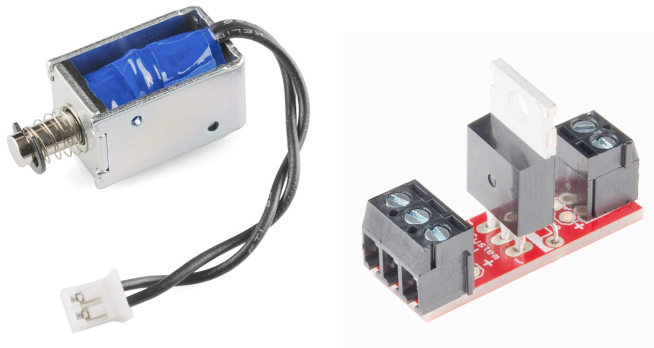


Power MOSFET breakout board and small geared DC motor

Use this board with a PWM control output, to control the speed on the DC motor. Sweep through the PWM values slowly (0.0-1.0) in a simple FOR loop or use the small pot hooked to an analog input to control it. This provides speed control for the DC motor, but a single transistor driver circuit cannot reverse it. To reverse the motor, four transistors are needed. They put this common circuit in an H-bridge IC. An H-bridge will be used later in the lab to reverse the DC motor and provide speed control.

Part 8 Control a Solenoid (Extra Credit 1%)

Lab Demo: Control a Solenoid using the MOSFET driver



Small 5V 1A Sparkfun Solenoid and Power MOSFET driver

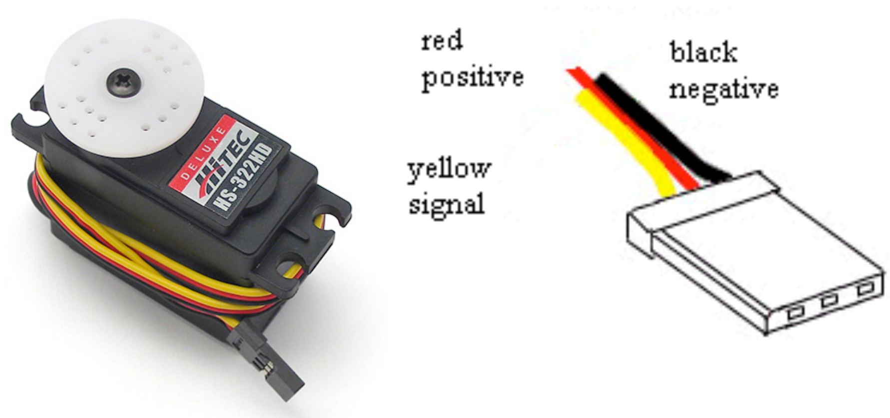
Additional Details: Solenoids provide linear motion for mechanical devices. They also require a relay or transistor driver circuit and external power supply just like the DC motor used earlier in the lab. Read/review An Introduction to Solenoids to see the setup for your demo. The first basic demo (without automatic timeouts) can be used for the lab. There is not a solenoid in the parts kit, but the TA has a few small ones in the lab, if you ever want to try one or need one for a design project.

Solenoids are common in electronic door locks, electric water valves, fuel injectors, and of course the classic pinball machine flipper. Even this tiny one needs 1amp! The typically require higher voltages such as 12 or 24 VDC at several amps, and some operate using AC power. Most solenoids are rated for intermittent duty which means that they will overheat and burn out the wires in the coil, if left on for long periods of time.

Part 9 – RC Servo position control using PWM (10%)

Lab Demo: Control a standard RC Servo motor using the small potentiometer for user input

Additional Details: An RC servo contains a DC motor with an internal driver circuit. In this part you will move an RC servo using PWM on the mbed. **Servos will need the external power setup from Part 6 and remember to connect only the grounds when using multiple DC power supplies!** A new servo wiki page is available in the cookbook and read this to understand how servos are used and how they work. There are servo code examples in the cookbook**.** In the handbook, check out the *PWMOut* API which these code examples call inside the new servo C++ class code. If you had to write the class code for one of these on a test could you figure it out and exactly how to hookup the pins? You should ask this question on all of the code examples used in lab – so take a couple minutes to look around, read, and understand the code (even though it is provided!). There will be problems like this on tests and you will need to supply this code!



A Standard RC servo and 3-pin servo connector

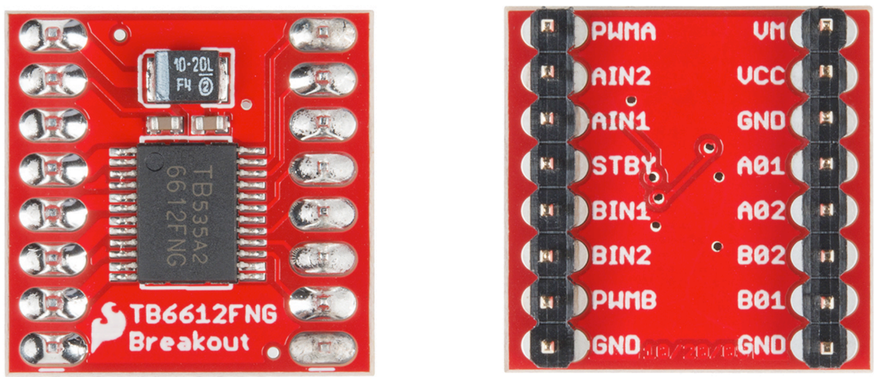
For some background on using servos, read/review an introduction to servos. Servos have a three wire connector with 4.8-6V DC, GND and the PWM control signal as seen in the servo image above. All of the RC servos have the three wire connectors and 4.8-6V DC is typically the middle pin, but the different brands of servos can move the three pins around and use different wire color codes. Each company does this to try to lock users into their RC equipment. The info on which pin is which can be hard to find, but is on the servo wiki page. There are even companies that make conversion and extension cables for servos!

The mbed module has built-in power protection that trips at 460MA (mbed chip uses around 200MA), so an **external power supply or battery for servos or a DC motor must be used.** A DC motor inside a servo or a regular DC motor has an even larger in-rush current at startup (or if it ever stalls). Don't forget to tie the two power supply grounds together at one point (but not 5V!). If you don't have a power supply and are working outside of the lab, you could use the robot battery pack for the servo and DC motor. Recall the earlier comments and photos at the start of this lab about the 5VDC 2A adapter in your kit.

Part 10: Reversible DC Motor with Speed Control using an H-Bridge (10%)

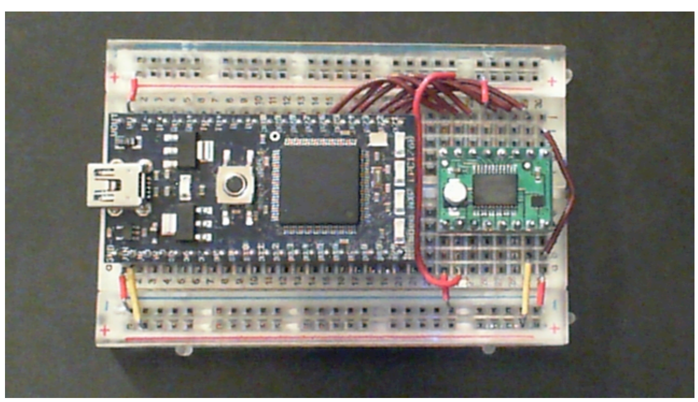
Lab Demo: Use the H-bridge breakout board to control direction and speed on a DC motor.

Additional Details: To use a DC motor with speed control using PWM and reverse it when needed, an H-bridge driver circuit is required. Use one of the small geared DC robot kit motors for this section. Small DC motors tend to rotate too fast with too little torque for most robotic applications, so they are typically geared down to solve this problem. This motor has a 48 to 1 internal gear ratio. There is a dual (i.e. two circuits A and B) H-bridge driver breakout board in the lab kit as seen below and a few spares in the lab. You only need to hookup one of the A or B circuits to a motor. To control both motors on your robot later will require both of the A and B circuits. Some basic H-bridge circuits will burn out, if you ever set forward and reverse on at the same time! This new smaller MOSFET H–bridge module is from Sparkfun. Pololu also makes a (green) breakout board that looks the same, but a few pins are in different locations.



Dual H-bridge breakout (top and bottom view showing pin names)

Wiring: AIN1,2 is the forward and reverse control inputs. PWMA should be a PWM speed control input. VMOT is the motor power supply + lead. VCC is the logic supply (3.3V on mbed). AO1,2 are the motor leads. Pull the /STBY pin high (chip has internal pulldowns on all control pins). It is easy to overheat and burn out the H-bridge with a short on the motor or power leads – so be careful. Use the cookbook motor code example that is already setup for an H-bridge. To control the robot in later labs, the H-bridge is small enough fit on one of the small breadboards below the mbed chip as seen below. There is a special mbed API for motors with an H-Bridge in the cookbook.

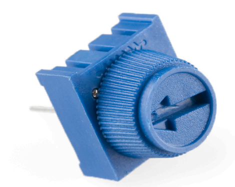


The H-bridge will fit on the robot later using one of the small breadboards.

The robot can be used in some extra credit options at the end. You don't have to assemble the robot at this point in the lab, but if you want some hints for use with mbed are available and links to the Shadow robot assembly instructions and videos are available at Sparkfun.com. Sparkfun sells about 90% of the parts in your kit, in case you ever need a replacement. The instructions show adding a few more sensor parts that you do not have in the kit (would add another $40 or so) and a different processor board (not an mbed). They do show how to snap the robot chassis parts and motors together. If you are into robots and want to do a robot design project, you might want some of the sensors seen in the video. They are available at www.sparkfun.com. The kit does not have the extra Hall Effect motor feedback sensors seem in the instructions and videos but they are a nice addition for extra credit. Put those on when assembling the robot, it may break a plastic tab, if you try to take it apart to install those later.

An external decoupling capacitor of a few hundred uf might be a good idea near the H-bridge motor power pins to smooth out the voltage drop when motors switch on (i.e., a short duration large inrush current spike occurs on DC motors, it can even drop the supply voltage so low that a micro or I/O chip on the same supply can crash). This breadboard and power supply setup worked without it, but others have needed it. An instant reverse of a fast moving motor is typically the worst case on this (called Back EMF). There are some larger caps in the lab, if you ever need to try this.

The DC motor's speed and direction must be controlled using a potentiometer setup as a voltage divider circuit connected to an mbed analog in pin. The potentiometer from the parts kit is shown below and the center pin is the wiper (output of voltage divider). 3.3V and gnd tie to outer pins. Swap outer pins to switch from CW to CCW rotation to increase the voltage output from the pot. The input value from the pot will be in the range from 0 to 1, but the motor class needs a value from -1 to 1 ("-" is reverse), so the pot's input value will need to be scaled in the program.



The Sparkfun 10K Trimpot will plug into a breadboard

Once again, make sure you hook up the polarity for the H-Bridge correctly (Vmot - motor supply in the H-Bridge image). If the polarity is reversed, the H-Bridge will be permanently damaged.

Use one of the small DC motors in the robot kit and the dual H-bridge driver breakout in your kit. It is not required to hookup both motors or assemble the full robot kit unless you want to. If you want to work at home and do not have the potentiometer in the new kit, two pushbuttons can be used to increase and decrease motor speed and some are available in the lab from the TA. With a bit more work, the touch keypad in the kit could also be used. A code example for the touch keypad can be found at http://mbed.org/users/4180_1/notebook/mpr121-i2c-capacitive-touch-sensor/.

Once again, the power supply is a bit limited over the USB cable and you will probably need a bit more current. If you notice the mbed power LED blink when things turn on this is likely the problem. Use the 5V 2A AC adapter in the parts kit with the breadboard friendly barrel jack. Don't forget to hook up the external power supply to mbed's ground (but not the two 5V pins). Another solution would be to use the 4AA battery pack in the robot kit to power your motor or the power supplies available in the lab on the protoboard.

Part 11: Stepper Motor Control with a Dual H-Bridge (Extra Credit 2%)

Small GM Automotive Stepper Motor used in Car Dashboards

Lab Demo: Control a small stepper motor using the dual H-bridge

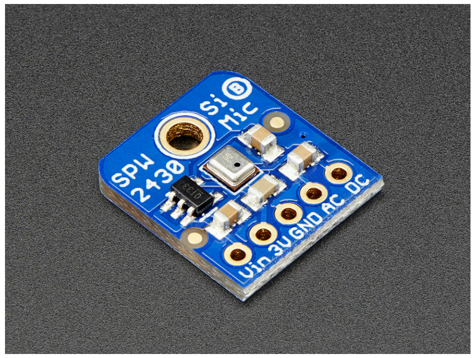
Additional Details: A [stepper motor]() provides accurate position control without the additional expense of a feedback sensor and control system which would be required to use a geared DC motor. Stepper motors are available for checkout in the lab or are available in older kits. They are a relatively inexpensive way to achieve precise position control of a mechanical device and are widely used in many products. You probably have one in your inkjet printer and perhaps your car's speedometer and other gauges. This one is mass produced for GM car speedometers and other dashboard gauges making it one of the smallest and lowest cost stepper motors available. Stepper motors have more than one coil. When the coils are activated in the correct sequence, the motor moves a few degrees and stops (assuming the motor is not overloaded and sufficient time delays are provided between steps). This one moves .5 degrees per step. Some stepper motors (called bipolar) also require reversing the current in the motor coils, so an H-bridge driver circuit may be needed for each coil. There is a mechanical stop inside the motor that prevents it from rotating more the about ¾ of a revolution. The mechanical stop allows initial calibration of the indicator needle location without additional sensors. Many stepper motors do not have the internal mechanical stop.

Hookup the stepper motor per the Wiki page wiring table and run the demo program found at https://developer.mbed.org/users/nhimani3/notebook/automotive-gauge-stepper-motor/ . It uses both of the two H-bridge circuits on the Dual H-bridge module in the parts kit. It is necessary to cut off the two long white plastic pins on the backside (per Wiki page photo and instructions) to get it to fit in a breadboard. Small diagonal wire cutters work well and the TA has some available in the lab, if you don't have one handy. It would probably be a good idea to place the motor on the breadboard first, and then carefully press on the needle indicator (so that the motor is held very steady when applying force). It does not need to be pressed on all of the way, just enough to hold steady.

This small stepper will run without an external power supply, if nothing else is attached to your mbed. It might even run directly off of the mbed PWM pins, but it would not be a good idea long term for your mbed to drive a max current inductive load directly from the mbed pins (the mbed's internal output pin drivers might burn out over time from the motor's brief back EMF voltage spikes). Almost all stepper motors will be a lot larger and need an external H-bridge driver circuit (this very tiny one is a bit of an exception).

There are even special stepper motor driver ICs, which will automatically cycle through the necessary coil activation sequence using the correct time delays (the optimal time delay would actually change based on a velocity and acceleration model of the motor and mechanical setup). This simple demo does it all in software using the Dual H-bridge driver in the kit by using only a constant time delay (a bit slower than optimal). To move as fast as possible the time delay should change to ramp up at a constant acceleration, level off at a constant maximum velocity, and then start decelerating at just the right point to stop at the desired point. Some of the newest stepper motor driver ICs even use PWM on multiple coil circuits to try to step between steps (call microstepping).

Part 12 – MEMS Microphone for Audio Input (5%)



A MEMs Microphone Breakout Board

Lab Demo: Use a MEMS microphone to detect audio levels and display them on LEDs

Additional Details: A microphone is used in many devices to detect sounds and to record or transmit human speech. Read the Using a Microphone for audio input wiki page. There is a small MEMS microphone breakout board in the parts kit similar to those found in cell phones. This small device (silver box in middle of board) is used in almost all cellphones.

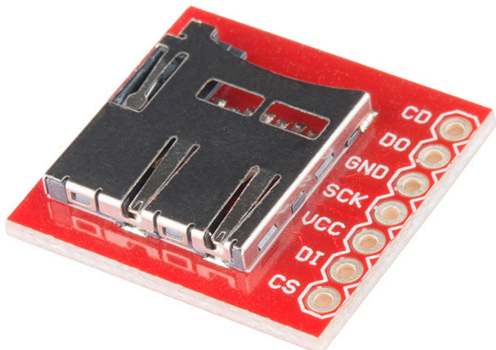
For your checkoff, hookup the first example circuit at https://developer.mbed.org/components/Adafruit-MEMS-Microphone-Breakout-SPW243/ that uses the basic DC coupled input to display a crude audio level on mbed's four built-in LEDs. Some burglar alarms use a similar idea to detect sound.

The LED audio level demo will actually work a bit better and be a bit more responsive using the changes using the AC coupled input as described in the MPA example on this Wiki page, since the DC bias level does not drift slowly when using the AC coupling output.

For speech recording and audio transmission, a microphone pre-amp circuit with a gain of 50-100 is needed as outlined in the last example on the Wiki page. An 8pin DIP rail-to-rail dual op amp could be used to build the pre-amp with a low pass filter on a breadboard. There are even special surface mount microphone pre-amp ICs designed just for this purpose. Some even have automatic gain control (AGC). Some more expensive microphone breakout boards include a microphone preamp IC and a low pass filter.

Part 13 - SPI bus: microSD card file system (5%)

Lab Demo: Write and read a file from a uSD card



A uSD card breakout board and a 4G uSD card

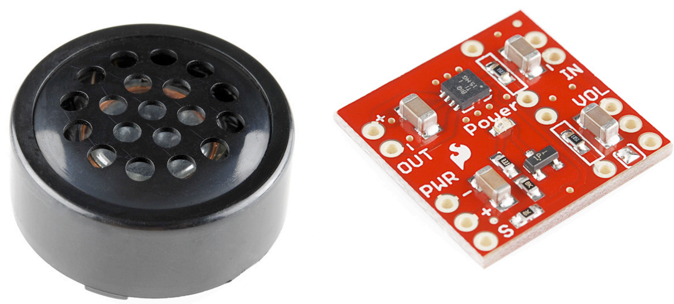
Additional Details: SD cards are interfaced using an SPI bus. MicroSD breakout boards are available from Sparkfun. The cookbook has SD file system code examples. Note that you will probably need to use a microSD card brand and size that has been noted as working OK in the mbed cookbook and forums (2G to 32G?). Write a program to create and write a file that contains “Hello SD file world” closes it, and then opens it and reads the file back and displays it on the PC using the USB virtual com port. Leave the uSD card setup for possible use in the next part. Many laptops have a built-in SD card reader that will work with the SD to uSD adapter that comes with most uSD cards. The new mbed kit also has a small white USB uSD card adapter. These are handy to transfer files between a PC and mbed. SPI is used instead of I2C for SD cards, since it can provide more bandwidth for file transfers. Leave the SD card on the breadboard, it is needed for one option on the next part.

Caution: Double check power pin: uSD cards operate off of 3.3V – 5V power will blow up the uSD card!

A USB Flash drive can also be used for file I/O. See this extra credit option at the end, for the USB flash drive details. The file system driver code uses several C++ virtual functions. By supplying the required R/W virtual functions, filesystems can be ported to any media with some additional work.

Part 14 - Use the speaker for audio output (10%)

Lab Demo: Play Audio on the Speaker using a Class D Audio Amplifier

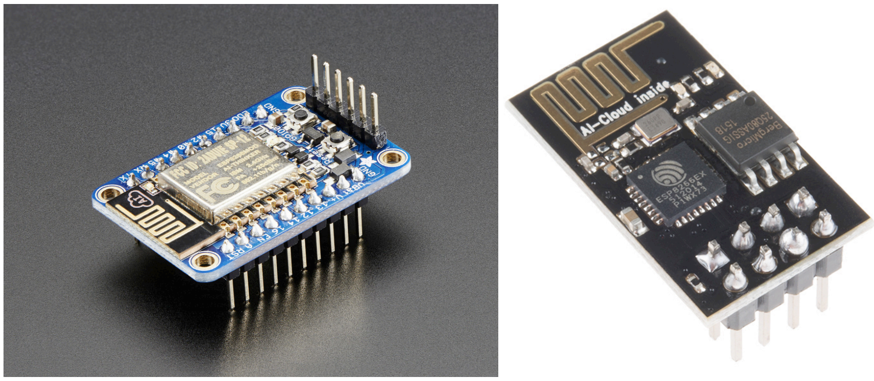


A small breadboard friendly speaker and a Class D Audio Amplifier breakout board.

Additional Details: Use a speaker and playnote() to play the first 17 notes of the Rambling Wreck Song, or play an uSD wave file of SiFi theme music or a well known SiFi sound effect. See the speaker wiki page for details on how to do this. Use the Class D audio amp board in the kit to drive the speaker (instead of the basic driver transistor to provide more volume and fidelity). The wiki pages for Class D amps are in the mbed component pages under sound. The SD card, amp, and speaker will be used in lab 3, so it will save time later to leave it hooked up on the breadboard.

Class D audio amplifiers use PWM internally for higher energy efficiency in the output driver circuit. The input signal can be analog or PWM. The PWM frequency is typically around 10X the highest audio frequency. This device runs PWM at 250,000Hz. Many have an H-bridge in the output driver circuit for the speaker. Almost all new audio designs use Class D Amps.

Part 15: Wi-Fi networking and IoT (10%)



ESP8266 Wi Fi SOCs – left Adafruit Huzzah breakout board and right the ESP 8266 module used on it

Lab Demo: Create a Wi Fi web page server

Additional Details: There are Huzzah ESP8266 Wi-Fi SOC modules available in the lab for checkout. It can be interfaced to mbed and provide Wi Fi for mbed projects. An ESP8266 mbed LPC1768 cookbook Wiki page is available with the details. There are also numerous Huzzah example projects from students available on the mbed Wiki pages.

There is a simple hello world demo program for the new Huzzah at https://developer.mbed.org/teams/Ece-4180-team-who/wiki/Using-Adafruit-ESP8266-Huzzah. Hookup the Huzzah wiring to mbed and the external 5V power. Find and change the Wi-Fi SSID and password in main.cpp for your setup, compile, download, and run. Start a terminal application program at 9600 baud to monitor the USB serial output from mbed. Reset the mbed and watch the status messages. After several seconds, once it connects to your network, note the first IP address it returns. Open a web browser on the PC using this IP address (not a domain name!) and a web page should appear from the Huzzah code. Change the code so that your name is displayed on the web page. Show the TA the web page server working. Registering a domain name to use (instead of an IP address) costs big bucks!

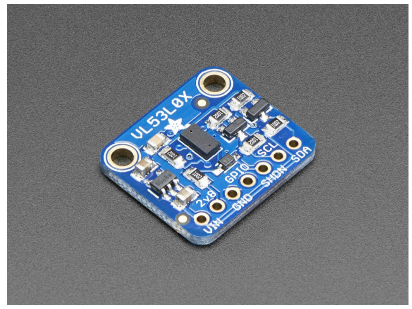
Extra Credit (1%) Using a web browser on your PC, demo the IoT web page mbed LED control program found at https://developer.mbed.org/users/ausdong/notebook/using-the-adafruit-huzzah-esp8266-to-add-wi-fi-to-/ This type of device control is called IoT (Internet of Things). IoT is even appearing in new light bulbs and kitchen appliances! Wi Fi chips need up to 500MA of power during an RF transmission burst, so the **external 5V supply must be used** for the Wi Fi chip's power supply for reliable operation. It will probably be easier to get this working first on a Wi-Fi setup at home since many schools require a special Wi Fi logon procedure. With many you have to run a web browser to enter your password to enable DHCP (i.e. get an IP address) and it tends to be locked down with security settings a lot more than a typical home Wi Fi setup. They also lock down some network services other than HTTP on many Wi Fi networks, but a typical HTTP web page transfer works and that is what is used in this demo since firewalls typically allow this port (80).

If you can't ever get it working on campus, you can show the TA videos or a couple screen captures where you have changed a couple things on the web page that it displays such as the name or title. You may also have a PC or Phone that could setup a mobile PC Wi Fi hotspot or MAC Wi Fi hotspot that you could use connect your mbed to on campus. This might turn out to be the best option for many.

To connect mbed Wi-Fi on the GT campus, you will most likely need to use GTother and enter the Wi Fi modules MAC address that is displayed in the first demo program. Read these instructions first and note the "SSID" and "password/key needed" for GTother. The link for entering and enabling the MAC address to get an IP is https://auth.lawn.gatech.edu/index.php. It needs to be reenabled every 30 minutes. Do this first and then try to connect the device. GTwifi/GTother seems to change setups and security settings often, faster than anyone can ever figure out the details everywhere on campus to have bulletproof instructions on how to use it.

Part 16: LIDAR TOF Distance Sensor (5%)

Mbed	VL53L0X
3.3V	VIN
GND	GND
P28	SDA
P27	SCL
P26	SHDN



Lab Demo: Measure and print the distance to the nearest object using the VL53L0X LIDAR TOF senor.

Additional Details: Many embedded devices need to measure distance. Distance sensors are used for gesture detection on tablets and phones, and are also found on many robots to detect and avoid obstacles. New ToF (time of flight) sensors measure the speed of light delay from a solid state laser (LIDAR - the term lidar is a portmanteau of "light" and "radar".) to detect distance. It uses a SPAD array (Single Photon Avalanche Diodes) detector and an IR Laser with a 940nm VCSEL emitter (Vertical Cavity Surface-Emitting Laser) controlled by an internal microprocessor all in a module smaller than an pencil eraser. Older IR reflective sensors use the angle of the reflected light from an IR LED to determine distance. Sonar is also used for longer range distance measurements using a reflected ultrasonic signal.
The parts kit contains an I^2C VL53L0X TOF distance sensor. A code example for mbed for the ToF sensor is available at: https://developer.mbed.org/users/4180_1/code/HelloWorld_VL53L0X_LPC1768/
This demo code prints out measurements on the USB virtual com port. Open a terminal application window to the mbed USB com port to see the range measurements. Ignore the startup message about missing left and right sensors, the demo code only uses the center sensor. Moving your hand towards and away from the sensor should change the range measurements. Since it uses a pin point IR Laser beam, it has a narrow beam width. It measures objects up to 1000mm away and has a longer detection range on reflective objects. The device contains a microcontroller and has a somewhat involved initialization and calibration procedure. Longer range LIDAR sensors are also available, but they are a bit too expensive for a student parts kit.

Older/used parts kits: Use the older technology Sharp IR distance sensor in the kit that has an analog output or a VL53L0X can be checked out from the lab. The Sharp analog IR distance sensors are found in many robotics applications. Read in analog data from a Sharp IR distance sensor. For info on the sensor pin out and the distance and voltage relationship, see the Sharp IR sensor's data sheet at http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf . Output the range data from the Sharp IR sensor to the USB virtual com port.

Sharp IR distance sensor and JST cable

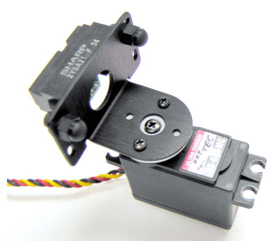
The Sharp IR sensors have a strange three pin cable (small white JST connector) and you can plug the jumper wires into the end of the connector to connect to the protoboard. JST stands for Japan Solderless Terminal and you almost have to travel to Japan to find one outside of your kit! Red is 5V, Black is GND, and Yellow is the analog input (to mbed). With a 5V supply, the maximum analog output is <3.1V. A 10-100 uf or so capacitor across the power supply lines helps reduce analog signal noise a bit (**WARNING -** polarity matters on large electrolytic caps! - the "+" lead goes to the higher voltage). There are some caps in the gray parts cabinet in the lab and in the new kits.

Extra Credit Options for Lab 2 (some may require additional parts from lab TA)

(1%) Display the current time from the mbed's real-time clock on the LCD used in Part 4 using larger red 2X characters in the middle of the LCD. See the *time* link in the handbook. Update the time every 200 mS. The mbed RTC clock does not run until the clock is set.

(2%) Complete the compass option for part 3 and include a push-button or DIP switch to toggle between modes. (**+1% more add on**) Implement the somewhat complex 3-axis accelerometer correction math and trig operations to give a true magnetic compass heading for the compass demo, when the IMU chip is not level. Articles on this can be found on the web with the equations and code examples for other chips. It is about a page or two of C code. The accelerometer detects the 3-axis gravity vector to correct and convert the 3-axis magnetometer readings to a horizontal vector. This is sometimes called an "electronic compass". Don't expect a lot of accuracy without calibration of the magnetometer as outlined in the IMU wiki page.

(2%) Build a scanning IR or LIDAR distance sensor using the setup seen in the image below. Write code to move the IR sensors with the servo taking readings every few degrees looking for the minimum distance value. After a full 180 degree scan, reset back to the angle that had the minimum distance value and stop. The servo can be mounted in a servo hole already on the robot kit later and used to rotate the IR sensor and/or a SONAR module, but it will require a mounting bracket as seen in the image below. You might find this handy for a robot orientated design project later. The servo does not need to be mounted on the robot chassis for this option.



Servo with IR sensor using mounting bracket from lab

(2%) Make a Dancing Robot. If you have the optional parts kit**,** build the robot kit. Using one of the small breadboards that come with the basic mbed kit mounted on top of the robot hookup both DC motors to the H-bridge and make a demo that makes the robot move around in a pattern such as forward a foot, backwards a foot, a full rotation CW and then CCW and repeat. For those without the optional kit, some older robot kits are also available in the lab. https://learn.sparkfun.com/tutorials/assembly-guide-for-redbot-with-shadow-chassis?_ga=1.167965857.1619691232.144081192 has links to more assembly instructions than can be found in the kit. There is also a fast pace video at https://www.youtube.com/watch?v=aJRYTqZu5OE. The extra Hall Effect motor feedback sensors seem in the instructions and videos can be checked out in the lab. Put those on when assembling the robot, it may break a plastic tab, if you try to take it apart to install those later. A second battery pack just for motors may be required to avoid crashing your mbed chip when the motors turn on.

(2%) Demo the HTTP Web Server using the mbed's flash file system for web page storage. **(+1% more add on)** include at least one image file on the web page.

(1%) Demo one of the mbed handbook's USB host projects https://developer.mbed.org/handbook/USBHost other than the mouse (to demo reading from and writing to a USB device such as a flash drive or keyboard) or the older one that can be found at: http://mbed.org/users/chris/programs/MSCUsbHost_FULL/5ymp5/docs/files.html. (Hint: the local file system only supports 8.3 char filenames -- http://en.wikipedia.org/wiki/8.3_filename - update: this has been fixed now in new library versions)

(2%) Demo low-bandwidth IR and RF data transmission techniques used in TV remotes and car key fobs using the low-cost setup found at: http://mbed.org/users/4180_1/notebook/ir-and-rf-remote-controls/ . The TA has these parts in the lab.

(1%) Demo one of the lab's large Sparkfun relay module breakout boards or a Power Switch Tail -- http://mbed.org/users/4180_1/notebook/relays1/. The TA has these parts in the lab.

(2%) Demo one of the Internet of Things LCD gadgets similar to these at Nokia Color Graphics LCD panel -- http://mbed.org/cookbook/Internet-of-Things---LCD-Gadgets, but using the newer Color Graphics uLCD used earlier in the lab. The Google weather API not free anymore, so that one would require finding another weather web API such as Yahoo and rewriting the code. There is a student design project posted in the cookbook's student project pages that already has this code for the clock and one even has the code for web weather info.

(**2%**) Assemble the robot kit with the Hall Effect motor feedback sensors and the H-bridge to drive the motors. There is an example program at https://developer.mbed.org/users/electromotivated/notebook/wifi-pid-redbot-robot-webserver/ that shows how to add feedback control (the web portion is not required for checkoff). Demo the robot moving in a straight line with and without feedback. It should move a bit straighter with feedback. See part 7 for links with more robot assembly info. A second battery pack just for motors may be required to avoid crashing your mbed chip when the motors turn on. There are a few sensor kits in the lab and they were in older 4180 kits. Sparkfun also sells the hall effect wheel encoder kits.

(**2%**) Extra credit for the first two teams to complete the lab with all extra credit options.

Some Additional Low Cost Ideas for Extra Breadboard Power Outside of the Lab

If you are working at home and need more power for the breadboard (servo, DC motor, Wi Fi portion of this lab, or a lot of devices attached on a big project and the mbed power light blinks off later) and do not have the 5V DC AC adapter here are some ideas:

The new parts kit contains a 5V 2A AC adapter. The robot battery pack with 4 AAs will also power the servo or DC motor. Use the breadboard friendly barrel jack power connector in the kit for this labs servo and motor power. They will both tolerate 6V and don't need a regulated 5V supply (but most other parts like ICs do). Mbed can connect to the battery pack directly for use later on the robot using the Vin pin (it has a 3.3V on board regulator). For this lab, leave the mbed attached to USB power and just use the battery or AC adapter's 5V power for the servo, motors and Wi-Fi module.

5VDC 2A AC wall adapter, the 4AA 6V battery pack from the robot kit, and barrel jack for breadboard power hookup

For old kits, if you have an old/spare 5VDC AC wall wart around from something no longer used, the connector could be cutoff and you could attach the wires to the binding post on the breadboard. Banana plugs could also be added so that it plugs into the breadboard power jack. Banana plugs come in different colors and versions that do not require soldering but cost a couple dollars each (almost as much as a new AC adapter!).

Sparkfun sells a nice low cost 5VDC 2A wall wart used in the new kit and the TAs have some in the lab with the breadboard barrel jack. They also make a barrel jack to screw terminal adapter that could be used to attach it to the breadboard without cutting the wire, or cut the wire and use banana plugs. Don't cut the wire on the lab's adapters!

Don't forget – When using multiple power supplies always tie the grounds together on the supplies (but not the 5V outputs!)

If you want a green renewable energy source for mbed and need some exercise, check out this video.