

## ECE 4180 Lab 2 – Everything I/O with mbed - part 2

Section B Due Date: Feb 16 even groups – Feb 17 odd groups

Section A Due Date: Feb 18 even groups – Feb 19 odd groups

Names: \_\_\_\_\_ Sect \_\_\_\_\_ Sect \_\_\_\_\_

Item	Lab Demo	Extra Credit	Max EC Pts.
<b>4180 LAB 2</b>	n/a	n/a	n/a
Analog Input		n/a	n/a
Analog Output		n/a	n/a
I <sup>2</sup> C IMU		n/a	n/a
RS232 Serial		n/a	n/a
uLCD Graphics Bubble Level		n/a	n/a
uLCD Compass	n/a		<b>2 + 1%</b>
Ethernet Networking			<b>2 + 1%</b>
Relay and Transistor Driver		n/a	n/a
Servo, DC and Stepper motor		n/a	n/a
MEMs Microphone		n/a	n/a
MicroSD Card or USB File I/O		n/a	n/a
Speaker Class D Audio Output		n/a	n/a
Mbed Wi Fi IoT Web Control		n/a	n/a
Robot Dance	n/a		<b>2%</b>
USB device on mbed	n/a		<b>2%</b>
IR and RF data transmission	n/a		<b>2%</b>
Relay or Power Switch Tail	n/a		<b>1%</b>
Internet of Things LCD gadget	n/a		<b>2%</b>
Robot motor feedback control	n/a		<b>2%</b>

**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.

**Please Note:** The large white breadboards now have 5V 4A switching power supplies connected via the DB-9 connector on the back as seen below. The toggle switch on the top left can turn the connection to the power supply on/off along with a status LED. The power supply is connected to the binding posts on the bottom right of the board. This power supply can easily power the MBED with all of the components such as the motors and Wi-Fi module in this second lab. If you use both USB 5V for mbed and the second power supply for the motor, be sure to connect ground wires only between the two 5V supplies (and not the +5V). At home, a list of low cost power options can be found on the last page of this lab. New kits have a 5V 2A AC adapter and jack.



## ECE 4180 Laboratory Assignment 2

For lab 2, we will be using several modules with surface mount ICs mounted on small [breakout boards](#) that will allow them to plug into a protoboard. These boards are in the kit and the TA has some spares are stored in yellow bins in the lab cabinets at the front of the lab, if they are not out. You can check off parts one at a time or demo the entire lab using the big everything loop at the end, 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](http://www.sparkfun.com) and a few from [www.adafruit.com](http://www.adafruit.com). There are a few spares in the lab available for lab use only from the TAs.

### **Part 1- Using Analog Inputs and Outputs (10% total - 5% each)**

Send out a sine wave on the analog output pin. Use 20+ samples per cycle and send out two complete cycles. DC Bias and scale the analog output so that it uses the full analog output range (0 – 3.3V). It should output a complete cycles and repeat. Check the output with a scope (in lab) or MyDAQ and note the frequency.

**Check Off:** Capture the scope image and hand it in at checkout or show it to the TA.



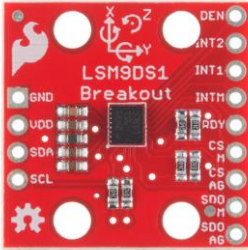
Read in analog data from a Sharp IR distance sensor. These have a strange three pin cable (small white JST connector) and you can plug the jumper wires into the end of the connector to connect to the protoboard. JST stands for Japan Solderless Terminal and you almost have to travel to Japan to find one outside of your kit! Red is 5V, Black is GND, and Yellow is the analog input (to mbed). A 10 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. With old kits, if you have problems plugging it into the breadboard strip, twist, and solder tin the leads (helps with small stranded wires). There is a wire stripper and soldering iron in the lab and the TA can help you with this, if you have never soldered before. If you give up, the TA should have some of the new JST cables in new kits that already have nice breadboard jumper pins. 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](http://sharp-world.com/products/device/lineup/data/pdf/datasheet/gp2y0a21yk_e.pdf). Output the range data from the Sharp IR sensor using a bar graph type display on the mbed’s four built-in LEDs (LED1-4, See *DigitalOut*). Write the C code and add it to the program loop. Check out the *AnalogOut* and *AnalogIn* APIs in the mbed handbook. You should be able to move your hand back and forth above the IR sensor and see the LED bar graph display change.

**Check-off:** Show the TA the IR sensor bar graph that uses the mbed’s built-in LEDs.

As an alternative to the IR sensor, you can also use one of the small Maxbotix EZ analog Sonar modules as seen in the photo below – it is not in your kit since it runs around \$25, but is it also real handy for distance measurements. Many robots use both sensors to increase the reliability of object detection (called “sensor fusion”). The Sonar modules have longer range but a slower update rate. It would tie into 3.3V for power (5V would saturate an mbed analog in pin that is looking for 3.3V max!). <http://www.maxbotix.com/uploads/LV-MaxSonar-EZ1-Datasheet.pdf> has the pin out info. The 3-wire small cable for it needs to be soldered to the PCB, if one is not already attached. With harsh use in the lab, the wires always seem to break off. Make sure wires are connected to the correct pins, and see if they need to be resoldered. See the TA if you need soldering help.



## Part 2 - I<sup>2</sup>C bus and USB virtual com port (10%)



<u>Mbed</u>	<u>IMU</u>
3.3V(Vout)	VDD
GND	GND
P9	SDA
P10	SCL

The LSM9DS1 (larger LSM9DS0 is in older 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/> (The older kit's LSM9DS0 IMU can be found at <http://developer.mbed.org/cookbook/LMS9DS0-IMU/>). This IMU should be in your 4180 parts kit, but we also have a very limited number of the older ones in the lab.

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! 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) is nice enough to include pull-up resistors on the chip-select, address, and I<sup>2</sup>C lines, so you only need to connect SDA and SCL (the I<sup>2</sup>C 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 I<sup>2</sup>C device to mbed without pullups, the mbed APIs lookup waiting forever a signal change that never occurs!

**Checkoff:** Show the IMU's data readings being printed to the USB COM port, updating once or twice a second. Import the more complex demo program **at the very end of the wiki page** that tries to calibrate the IMU for more accuracy into your mbed cloud compiler from <https://developer.mbed.org/components/LSM9DS1-IMU/>, **change the I<sup>2</sup>C 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.

(or in older kits with the LSM9DS0 [http://developer.mbed.org/users/aswild/code/4180\\_LSM9DS0\\_lab/](http://developer.mbed.org/users/aswild/code/4180_LSM9DS0_lab/) Note the different I<sup>2</sup>C pins used (p9 and p10) Add code and edit to the program to read the compass heading, accelerometer data, and temperature and print them out nicely to the USB serial port or LCD. You will get floating-point values, so please use fixed-width printf format strings to align the data nicely in columns.) You will find the API documentation at (LSM9DS0 <http://developer.mbed.org/users/aswild/code/LSM9DS0/> ) helpful (or take a look at the code yourself).

**Info on Terminal Application to monitor mbed USB output:** [Tera Term](#) (easier to use), or [Realtterm](#) (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. 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.

### Part 3 - Real RS232 Serial (10%)

Mbed	adapter
3.3V	VCC
GND	GND
RX	TX
TX	RX



Many I/O devices still require the older RS232 standard serial interface. In addition to the virtual serial com port, output the compass heading info from a “real” [RS232](#) serial port to the PC. 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 adapters from Sparkfun [http://www.sparkfun.com/commerce/product\\_info.php?products\\_id=449](http://www.sparkfun.com/commerce/product_info.php?products_id=449) are available in your kit and 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 use serial but with TTL logic levels. If it says “RS232” it is not TTL level and the voltage convertor will be needed!

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.**

**Check-off:** Show the same functionality in Part 2, but use the RS232 serial port instead of the USB virtual serial com port.

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 one 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. It would be too easy if they just marked which type of cable it is somehow! There is a gray tub of these PC serial cables in the back left corner of the lab.

If your laptop or PC at home does not have a serial port, we have some USB to serial 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 probably will not automatically do it. Windows 8 does not like the low-cost USB to serial converters in the lab – 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 are available for around \$15. Make sure it says that it “works with Windows 8,” 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 thousands of dollars to develop the Windows driver it is not surprising that they added driver code and starting checking for counterfeit ones.

#### Part 4 - 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

##### CAUTION

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.

**Lab Exercise:** 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 on the uLCD.

**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).

**(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.

*Graphics Tip:* Rather than clearing the screen between drawing each frame, it is better 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 of how the graphics and sensor should work.

**Check-off:** Show the bubble level working with your IMU and uLCD.

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



## Part 5 - Networking: Ethernet (10%)

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



Demonstrate one of the mbed's cookbook's Basic [Ethernet](#) projects. A small [Ethernet breakout module](#) is available in the lab and the mbed web site has some networking demo code. For debugging purposes you can detect packets with [Wireshark](#) or [Ethereal](#). Magjack connectors from different companies look the same, but have different pinouts! The ones we have are shown in the table. <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. 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. Mbed recently deprecated many of their networking libraries, so if you're running into a lot of 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!

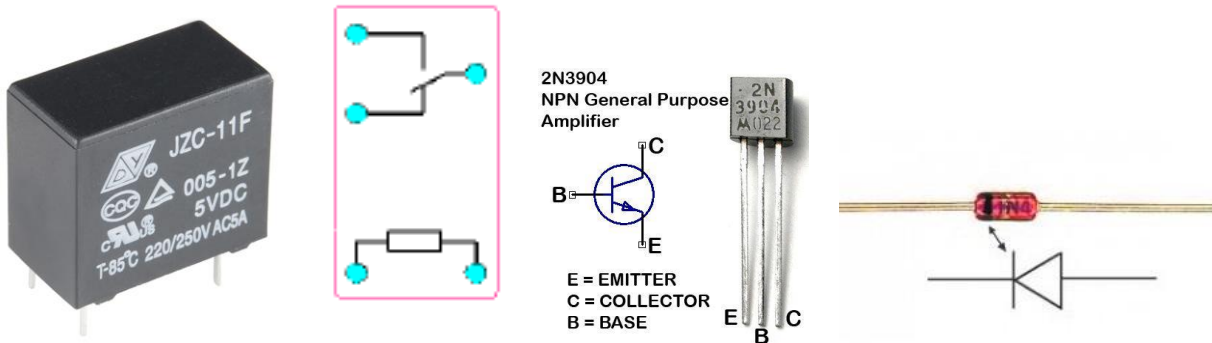
**Check-off:** Demonstrate an HTTP client executing a GET request. You can use the USB virtual com port to display useful connection information.

**Note:** For DHCP to work and get an IP address for mbed, the mbed module's NIC address must be registered with ECEs servers. If you grab a new one out of a box or bring your own from the parts kit it will likely get a timeout error during DHCP. 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. It takes a couple of days with email delays to computer support people to add a missing NIC address to the DHCP servers DHCP enable tables and it is just not a good solution for all of the new student owned mbeds. They are also running out of IP addresses in Van Leer and need new equipment to be able to fix it. But the new equipment would require dealing with the asbestos in the ceilings!

For people with the new mbed inventor's kit, in newer buildings like Klaus that come up with a GTLawn page for unknown MACs, it might be possible to use the MAC address to enable a temporary guest account or if your laptop connects with Wi-Fi, you might be able to bridge to network jack for internet access. Using GTother it is possible to setup a guest account for another device such as mbed. There is mbed code in the cookbook to print out your module's MAC address. 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.

## Part 6 - Driving high current devices with relays and/or transistors (10%)

There are two parts in your kit that are used to turn on and off high current devices. The first is the relay shown below 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.



SPDT 250VAC/30VDC 5A relay and pinout (top view), 2N3904 transistor, and 1N4148 diode

### Relays

The relay coil will require a driver circuit to provide enough current to control the coil. Use the 2N3904 small black transistor found in your basic mbed kit to turn the relay on and off. For the device or load to control, use one of the DC motors from the robot. For the motor, an **external power supply will be required** since the motor draws too much current. It will take an additional 5V power supply to power the motor like the one in the new kits.

Read the [driver wiki page](#) for additional background on driver circuits and relays how they work and how to hook them up. The same circuit setup is also used on the Sparkfun relay breakout board described in the wiki and the **schematic needed is found there** (yours does not need the add-on coil status LED). Per the wiki page, a [voltage suppression or snubber diode](#) should be used across the coil. The kit does not have a diode, but extra diodes are available in the lab (same gray parts cabinet as the resistors). In the special case of a relay, the power supply grounds do not need to have a common ground – the input control circuit can be electrically isolated from the output device and have different voltages.

### Adding Additional Power to a Breadboard

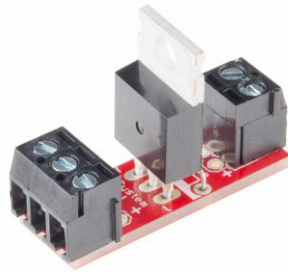
The 4180 parts kit has a black breadboard adapter for the 5V 2A AC adapter (or robot battery pack) barrel jack along with a small black breadboard power slide switch as seen below. These are handy to hookup and switch battery power to the breadboard. Don't use the tiny DIP switch for a power switch, the tiny metal contacts can't handle the current for very long. The robot's battery pack is also an option. On my 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 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 4180 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!**



### Power Transistor Driver

A relay is way too slow to ever dim LEDs or control motor speed. 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. The pen style 4-in-1 pocket screwdriver in your parts kit comes in handy for this.



Power MOSFET breakout board

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 to control it. This provides speed control for the DC motor, but the circuit cannot reverse it. To reverse the motor, four transistors are needed. They put this common circuit in an H-bridge. An H-bridge will be used in the next part of the lab to reverse the motor.

### Solenoids

Solenoids provide linear motion for mechanical devices and also require a relay or transistor driver circuit just like the DC motor used in this part of the lab. Read/review [An Introduction to Solenoids](#) to see a demo. 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. They are common in electronic door locks, electric water valves, and of course the classic pinball machine flipper.



Small Sparkfun Solenoid

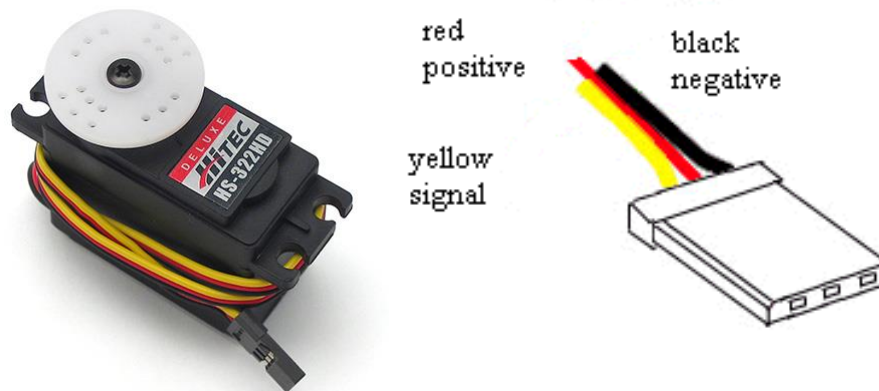


## Part 7 - PWM: Servo position control, DC motor speed control, and Stepper Motors (15%)

Many embedded devices require the use of both electronics (i.e., microprocessors with firmware) and mechanical devices. [Mechatronics](#) is a term often used to describe this engineering field. Many of the newer mechatronic devices also require networking. In this part of the lab, several of the more common electric devices used to provide mechanical motion will be controlled using the mbed module.

**Check-off:** In this part you will move an [RC servo](#) using [PWM](#) on the mbed, do a speed and direction control demo on a DC motor using PWM, and then move the small stepper motor. **They will all need the external power setup from Part 6!** 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](#), DC [motor](#), and [Stepper motor](#) code examples in the cookbook to use for each part that follows. In the handbook, check out the *PWMOut* API which these code examples call inside the motor and servo C++ class code. If you had to write the class code for one of these on the 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!

### RC Servos



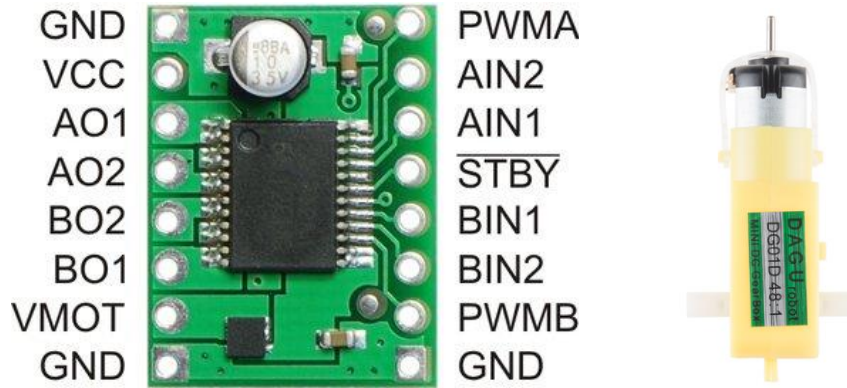
Standard RC servo and 3-pin connector

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 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** since a servo's internal DC motor or a regular DC motor has a large inrush 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 extra power 5V 4A supply that is available on the large breadboards in the lab or the 5VDC 2A adapter in your kit. Other ideas for power on the last page of this lab. There are also several large external DC power supplies out in the lab that could be used. Some people even use an AC wall wart left over from an old device at home that is no longer used. If it is anywhere from 4 to 6VDC and .5A or more, it will probably work for the servo and motor power supply (but not other ICs as they need very close to 5V DC with regulation). If you cut off the connector, the wires can be tied to the breadboard. The banana jack terminals on the larger breadboards are a handy way to hook up power. Some other power ideas are at the end of this lab.

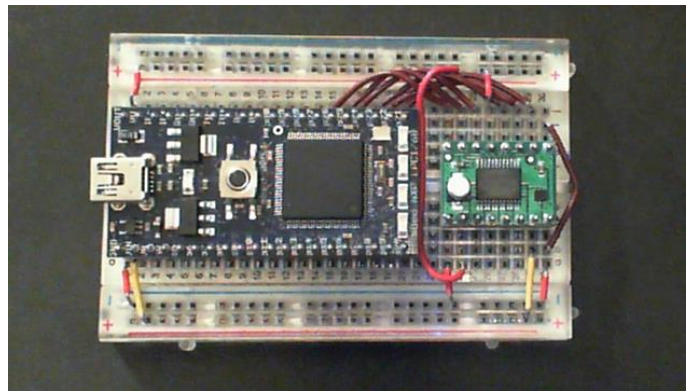
## DC Motor

To use a DC motor to demo speed control with PWM you will need to also add an H-bridge driver circuit. Use one of the 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. A and B) [H-bridge](#) driver module in the lab kit as seen below and a few spares in the lab. You only need to hookup one of A and B to a motor. To control both motors on your robot later will require both 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 Pololu.



H-bridge and a small geared DC motor from the robot kit

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.



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](http://www.sparkfun.com). The kit for 4180 now has the extra Hall Effect motor feedback sensors seen in the instructions and videos. 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's gray parts cabinet and the new kits, if you ever need to try this.

The position of the servo, or 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 Sparkfun 10K Trimpot will plug into a breadboard

**Once again, make sure you hook up the polarity for the H-Bridge correctly (motor supply in the image above). 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/](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 the motor turns on this is likely the problem. If you have an old AC wall wart around that outputs 4-6VDC, it could be used to power the DC motor only. 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 supply available in the lab on the protoboard.

## Stepper Motor



Small GM Automotive Stepper Motor

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. 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 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 may be needed for each coil.

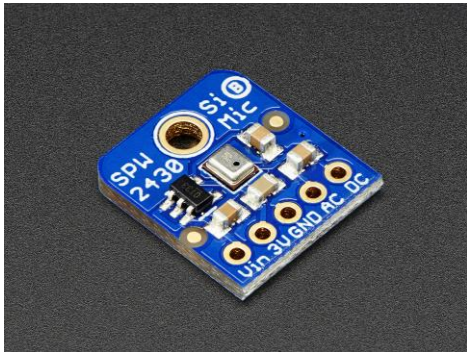
There are even special stepper motor driver ICs, which will automatically cycle through the necessary coil activation sequence using the correct time delays. Some of the newest ICs even use PWM on multiple coil circuits to try to step between steps (call microstepping). This demo does it all in software using the Dual H-bridge driver in the kit.

**Demo Checkoff:** Using the stepper motor in the kit, 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/>. Note my comment at the end of the Wiki page about some power pins missing from the H-bridge wiring table.

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 would probably not be a good idea long term for your mbed to drive a max current inductive load directly from the mbed pins (output pins might burn out over time from the motor's 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).

## Part 8 – MEMS Microphone (5%)



A microphone is used in many devices to detect sounds and to record or transmit human speech. There is a small MEMS microphone in the 4180 part kit. This small device (silver box in middle of board) is used in 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. For older kits, there are just a couple available from the lab TA or it might be faster to borrow one from someone with the new kit.

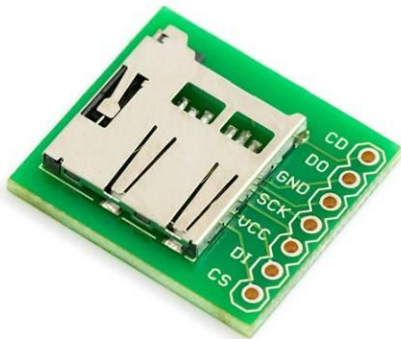
The LED audio level demo will actually work a bit better and be a bit more responsive using the changes using the AC coupled input as described in the [MPA example](#) on this Wiki page, since the DC bias level does not drift slowly when using the AC coupling output. For speech recording and audio transmission, a microphone pre-amp circuit with a gain of 50-100 is needed as outlined in the last example on the Wiki page. The TI 8pin DIP rail-to-rail op amp in the parts kit could be used to build the pre-amp. There are even special surface mount microphone pre-amp ICs for this purpose. Some even have automatic gain control (AGC).



### Part 9 - SPI bus: microSD card file system (5%)

Add a microSD or SD file system. SD cards are interfaced using an [SPI bus](#). MicroSD breakout boards are available from [Sparkfun](#) and a larger mbed [breakout setup](#) from the UK. The cookbook has [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 on the PC using the USB virtual com port. (Hint: the local file system only supports 8.3 filenames -- [http://en.wikipedia.org/wiki/8.3\\_filename](http://en.wikipedia.org/wiki/8.3_filename)). Leave the SD 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. The new mbed kit also has a small white USB SD card adapter. If you already feel that you understand the SD card interface from 2035 or 2036, you can hookup a USB Flash drive instead for this part, or do both for extra credit. See extra credit at the end, for the USB flash drive details.

**Check-off:** Show the TA the working file I/O program.

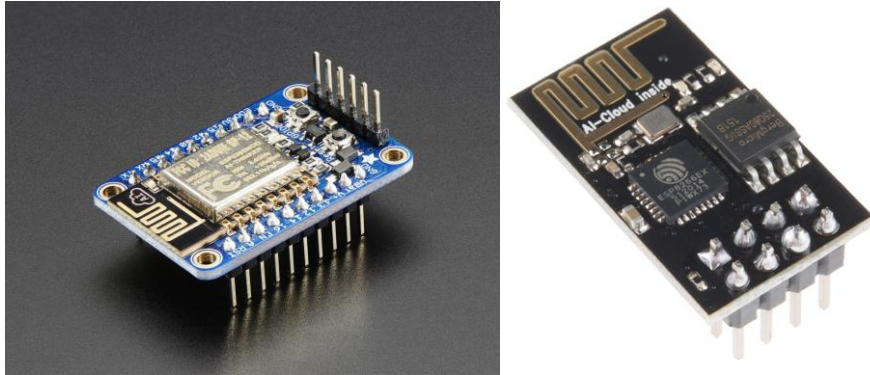


### Part 10 - Add the speaker for audio output (5%)



**Check-off:** Use a speaker and `playnote()` to play the first 17 notes of the Rambling Wreck Song, or play an SD 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. You must use the Class D audio amp board in the 4180 kit to drive the speaker (instead of the basic driver transistor you may have used in other classes to provide more volume). Older kits don't have one, but the TAs have a few from Adafruit that can be used in the lab. The wiki pages for both of the 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 would save time to leave it hooked up on the breadboard.

## Part 11: Wi-Fi networking and IoT (10%)



ESP8266 Wi Fi SOCs – left Adafruit Huzzah in newer kits and right 8266 module in older kits

There is an ESP8266 Wi-Fi SOC in your parts kit. 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.

**Checkoff:** 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!

Older kits: Hookup the device, run the two programs at the first wiki page and use a web browser on another device to control and report mbed I/O status over the internet. This type of device control is called IoT (Internet of Things). IoT is even appearing in new light bulbs and kitchen appliances!

Most of the older modules seem to need a power supply decoupling capacitor and these are available in the gray parts cabinet in the lab. Per the wiki page instructions, you also need six special M/F jumper wires to connect it to a breadboard– get these parts from the lab TA before starting this portion of the lab. Both chips need up to 500MA of power during an RF transmission burst, so the **external supply must be used** for reliable operation. The older module needs 3.3V power, so the 3.3V regulator in the parts kit will be needed (3-pins looks like a power transistor). The newer module has an on-board 3.3V regulator.



Other Parts needed for ESP8266 setup

It will probably be easier to get this working first on a Wi-Fi setup at home. With GTWiFi you have to run a web browser to enter your GT 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. To connect mbed Wi-Fi on campus, you will definitely

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>. Do this first and then try to connect the device. There is different Wi Fi hardware around in different buildings and it is not clear that they will all work the same. GTwifi/GTother seems to change setups and security settings all of time, faster than I can ever figure out the details everywhere on campus to have bulletproof instructions for this part of the lab. They also lock down some network services other than HTTP on GT Lawn, 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 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 device 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.

## 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 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 4 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 outline in the IMU wiki page.

(2%) Build a scanning IR 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 (now in the 4180 kit). 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 kit

(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](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 kit for 4180 now has the extra Hall Effect motor feedback sensors seem in the instructions and videos. 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.

(2%) Demo one of the mbed cookbook's two USB projects (to demo reading from and writing to a USB device) or the one that can be found at: [http://mbed.org/users/chris/programs/MSCUsbHost\\_FULL/5ymp5/docs/files.html](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](http://en.wikipedia.org/wiki/8.3_filename) - update: this may have been fixed now)

(2%) Demo IR and RF data transmission using the setup found at: [http://mbed.org/users/4180\\_1/notebook/ir-and-rf-remote-controls/](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/](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) -- <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.

(2%) Extra credit for the first two teams to complete the lab using the optional parts kit for the IMU, Stepper motor, and Wi Fi parts-- since the instructions are all new for these options.

## Some 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) 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 lab's 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 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.



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](#).