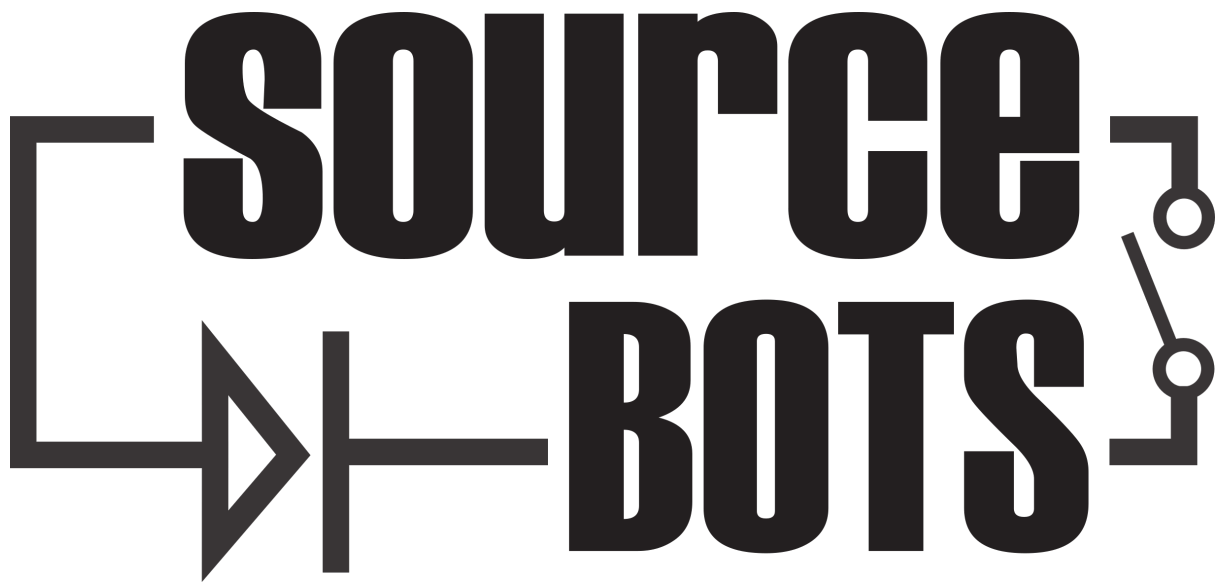


SourceBots Electronics Labs

Summer 2019

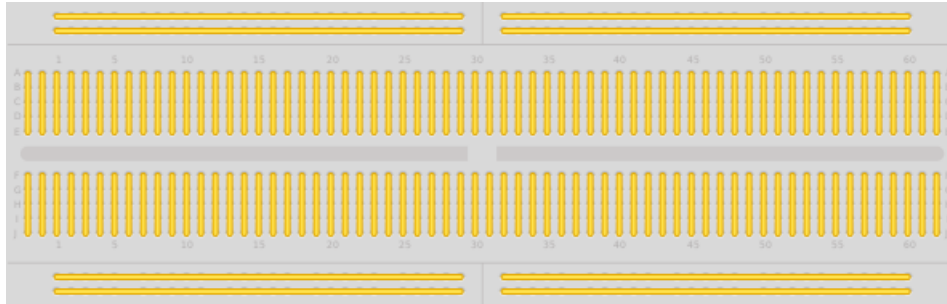


1 Introduction to Circuit Construction and Testing

This section of the lab aims to equip you with the skills you need to construct circuits for your robot. In its most simple form, a circuit provides a path for current to flow, usually doing some useful work on the way.

1.1 Constructing Circuits

1.1.1 Breadboards



A breadboard is a way of quickly prototyping circuits. A breadboard consists of a piece of plastic with small holes arranged in a grid. Underneath the plastic surface, each column has a metal component. This metal piece has two purposes. Firstly, it provides an electrical connection between each of the components in a row, and secondly, clamping onto the leg of a component to hold it in place. While this provides a fast, easy way to test whether your circuit will work, the components are not held in place strongly enough for this to be a long term solution.

Along with the connections in each column of the breadboard, which are often split into two sides along a centre line, which is marked by a deep channel, the breadboard has two long rows along the top and bottom. These are split along the middle (marked by a W), and are useful for providing supplies to your circuit, for example 5V and Ground.

1.1.2 Stripboard

As mentioned previously, breadboard is often not suitable for creating a circuit permanently. This is what stripboard (also known as proto-board) is used for. Stripboard allows you to solder your components into place for a more permanent solution. While it is more permanent, stripboard does still have issues which lead to it generally only being used for prototypes, or small batches of simple circuits. Stripboards are arranged in the same way as breadboards. Place the components legs through from the plastic coated side, and then solder the legs to the exposed copper on the bottom. One area where stripboard is more flexible than breadboard is the ability to break the copper trace. This allows multiple components to be placed in the same column without connecting them. A specialist tool is used to do this.

1.2 Testing Circuits

1.2.1 Powering your Circuit

The best way to initially test your circuit is to use one of the power supplies on your desk. These come in two forms in our lab, 30V,2A and 15V,5A. These power supplies come with adjustable voltage (in both coarse and fine increments) and current limit. The current limit should always be used as it prevents excessive short circuit currents. This can prevent damage if it is set appropriately. Generally, you should calculate how much current your circuit should draw and set the current limit slightly above that. In order to use these supplies, they must first be turned on using the switch to the left, which will allow you to set voltage and current limits (this should be done before connecting your circuit, as the power supply may have been left enabled by the previous user, and may be set to voltages and currents too high for your circuit to handle), and then the output must be enabled by pressing the button to the right. The output being enabled is indicated by a small LED. This can be connected to your circuit by using the screw terminals labeled "+" and "-" using wire (the banana connectors unscrew to allow you to clamp wire behind them).

1.2.2 Taking Measurements from Circuits

In the lab, you will find that your desk has both an oscilloscope and a digital multimeter. These are both useful tools for investigating issues within your circuit, but have different functions.

Multimeters A Digital Multimeter (DMM) is a useful tool for general debugging of a circuit, being able to measure a number of different physical properties. On the front panel of the device there are buttons to choose what the DMM will measure. This allows you to take readings of resistance, capacitance, current, voltage, diode polarity, and continuity. While these are extremely useful for identifying components, and checking circuit information such as supply voltages, they are often lacking when it comes to observing rapidly changing signals. DMMs are generally used with the red and black test leads found on your desk.

Oscilloscopes Oscilloscopes are far more specialist than DMMs, generally providing only a measure of voltage or current. However, where these devices excel is in providing a visual representation of the signal. This can be much more useful than the information that a DMM would provide in instances where a signal is only present for a short time. These use probes rather than test leads to take measurements, and generally require more setup than a DMM to provide any usable data.

In order to monitor a signal, there are 3 main things that require set up: The channel that the oscilloscope is reading from (this can be multiple channels simultaneously, and unused ones can be turned off if they are getting in the way), along with both the horizontal and vertical scales. The horizontal scale represents time, while the vertical scale represents signal magnitude. The selected timescale is used for all channels, while the vertical scale is set per channel.

The oscilloscopes in the lab also have an autoset function. This is normally worth trying first, however, it is not always suitable, depending on the signal contents.

1.3 General Circuit Debugging Tips

- Visually inspect your circuit. Are there any solder bridges between components? Does something look like it isn't soldered in or is loose in the breadboard.
- Is your circuit drawing the amount of current that you expect it to?
- Check your supply rails, are they at the voltages that you expect them to be?
- Is the power supply showing the voltage you set it at? If not, this usually indicates either a short circuit (normally this would result in lower than the set voltage between the terminals, and more current than expected being drawn) or the current limit being set too low. Think about how much current you expected.
- Does anything in your circuit feel hot?
- Think about what you would expect your signals to be doing. Use the oscilloscope to verify this is happening. Start from the beginning of your circuit, and work forward. Find where you stop seeing things that you are expecting. This is generally where the fault lies.

2 Collision Sensors

Before you can do any practical electronics, you need to have some understanding of the following concepts:

- Charge
- Current
- Potential Difference (Voltage)
- Resistance

Charge is a fundamental property of matter. This means it is something we can observe but not explain in other terms. In this case we can observe that some particles attract each other, some repel, and some experience no force. There are three possible states of charge: positive, negative and neutral (no charge). Charge is measured in **Coulombs (C)**.

Current is the **rate of flow** of charge. This means the amount of charge flowing in one second. In the case of a wire **electrons will flow**. Current is measured in **Amperes (A)**. If a wire carries a current of 1 Amp, 1 Coulomb of charge (6.25×10^{18} electrons) moves through that wire every second!

Potential difference (Voltage) is the amount of work done (in Joules) to move 1 Coulomb of charge from one point to another. There isn't a way to measure voltage as an absolute quantity, so all voltages are measured relative to another point. Potential difference is measured in **Volts (V)**.

Resistance is a way of measuring how difficult it is for current to flow through something. You might be familiar with the following equation:

$$\text{Resistance} = \frac{\text{Potential Difference}}{\text{Current}}$$

$$R = \frac{V}{I}$$

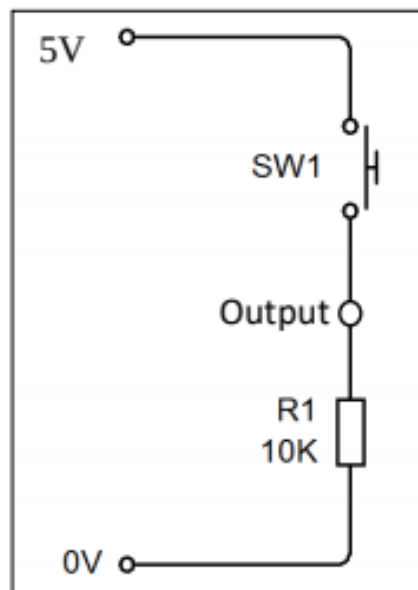
or, rearranging:

$$V = IR$$

Resistance is measured in Ohms (Ω).

Now that we know some theory, we can do something useful.

Take a look at the following circuit:



What is the voltage between the points labeled “Output” and “0V” when the switch is open (not conducting) and closed (is conducting)? When the switch is closed, the output terminal is directly connected to the 5V line, so the output is at 5V. When the switch is open, the output is at 0V *because of the resistor*. When the switch is open, there is no current in the resistor. The voltage drop across a resistor is proportional to the current through it ($V = IR$), so if there is no current there will be no voltage drop across the resistor, hence the output is at 0V. **Important note!** You might wonder why you can’t just connect a switch between 5V and the output. If you do this, when the switch is open the output will not be connected to either 5V or 0V. This is called a *floating line*, and there is no way of knowing what potential it is at. Remember that potential *difference* has to be measured between two points - a floating line isn’t connected to the circuit, so there is no way of measuring it’s potential relative to the 0V of your Arduino. Connecting the output to 0V through a resistor like is shown above makes sure the voltage at the output is 0V when the switch is not pressed; we say the resistor is a “pull-down resistor” and that it “pulls the output low” when the switch is open. We choose a resistor with a fairly high value of about $10k\Omega$ because we want to avoid drawing a large amount current when the switch is closed.

This circuit can be used as an input to a digital system. Your Arduino board can read the state of this input using the digital read command. See the programming docs for more detail.

In order to wire up the above circuit, three connections would be required:

- A connection between the ground (0V) of the Arduino and the ground of the sensor.
- A connection between a digital input on the Arduino and the output between the switch and the resistor.
- A connection between the 5V pin on the Arduino and the switch.

Task: Make a collision detector for your robot

Now that you know how to make a switch work as a digital input, try assembling some switches that could be used on your robot. Start by assembling it on a breadboard. When you're sure it works, you can solder it onto stripboard (ask a SourceBots volunteer for help).

You'll find some switches, prototyping board, resistors and wire in the lab.

Some things to consider are:

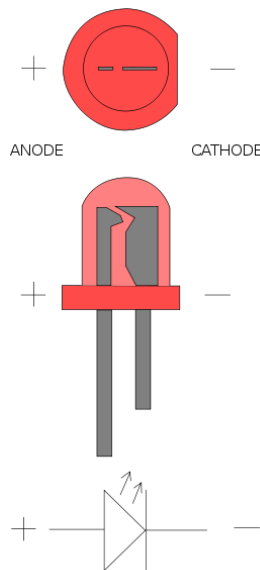
- How could your robot use a digital collision sensor?
- How will you mount the switch to your robot?
- Could you use more than one sensor?

3 Infrared Tripwires

Many electronic systems can be used to produce digital inputs. In the first tutorial, we considered physical switches. In this tutorial we're going to look at another source of a digital input. Something which robots can make good use of is tripwires. This can tell a robot whether or not it has collected an item. Tripwires can be made by generating a beam of light and then detecting that beam of light with a sensor. If we use visible light for the tripwire it will have a lot of **interference** from the arena lights, so it would be better to use something less common. For short ranges Infrared is a good choice. The IR tripwire circuit contains four components that you would need to wire to your robot. Two resistors, an IR LED, and an IR phototransistor. The IR LED is used to create the IR beam, and the phototransistor detects this beam.

3.1 Diodes and LEDs

LEDs are a form of diode. A diode is an electronic component which will only conduct in one direction. A diode has a cathode and an anode (see below). The diode will only conduct from anode to cathode.



LEDs have a particularly high forward voltage; roughly 1.2V for the IR LEDs that you will be using here. Each Coulomb of charge passing through it will lose 1.2J of energy, which is converted to Infrared radiation.

3.2 Wiring up the LED

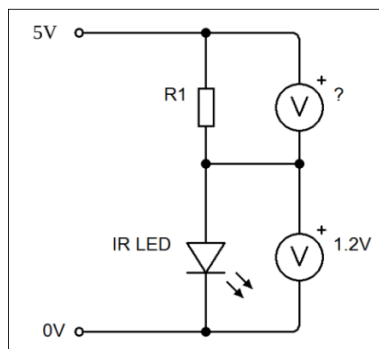
LEDs are polarized devices. If you plug them in backwards they may break. LEDs have two features to help you tell anode from cathode. Firstly, LEDs have one long and one short leg. The long leg is the anode (+). The other feature is that LEDs will have a round side and a flat edge. The flat edge shows you where the cathode (-) is.

A useful way to remember this is **short to nought**.

Our IR LED is an L-53F3C. From the datasheet we can find that the LED uses a forward voltage of 1.2V and a forward current of 20mA. So we will need a resistor to limit the current.

Parameter	P/N	Symbol	Typ.	Max.	Units	Test Conditions
Forward Voltage [1]	F3	V _F	1.2	1.6	V	I _F =20mA

This is the circuit that we will be using, and using the diagram we can work out what value the resistor must be.



We can now figure out the value of R needed by using $R = V/I$. V is the voltage drop across the resistor ($5 - 1.2$) and I is the current flowing through it (20mA). This gives us a value for R of:

$$R = \frac{3.8}{0.02} = 190\Omega$$

We could make this up from a 180Ω resistor and a 10Ω resistor in series, but according to the datasheet the absolute maximum current is 50mA, so a 180Ω will be close enough!

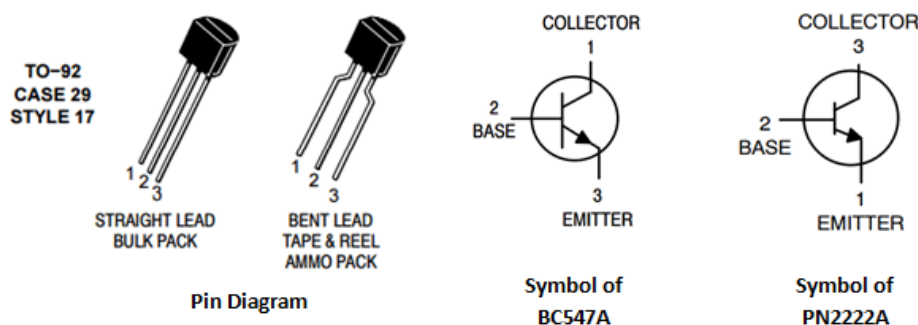
Wire this circuit up on a breadboard. A smart phone camera or webcam can see IR light so you can use it to test if your circuit works. (Ask if you need help).

3.3 Transistors and Phototransistors

Transistors can be used to make very complicated devices, but for our purposes they will be used as an infrared detector. A transistor is a three terminal device. It has a collector, an emitter and a base. The key feature of the transistor is that the current that flows into the base determines the current that flows into the collector.

$$I_{Collector} = \text{Gain} \times I_{Base}$$

A typical transistor will have a gain of 100.

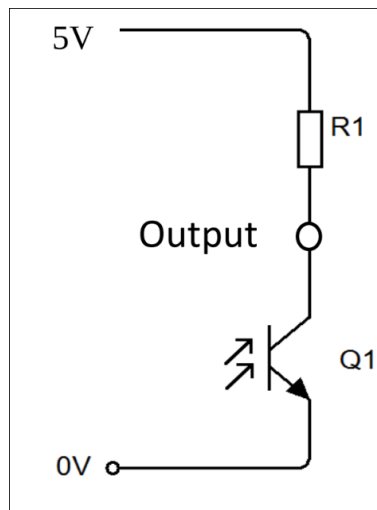


When no current flows into the base, the transistor acts like a huge resistor and very little current can flow through it. When a small current flows into the base the transistor acts like a resistor with a small value, so a lot of current can flow through it. This is essentially the same as a switch; only a small current base current is used instead of a physical force.

A phototransistor behaves in the same way, only instead of having a base wire phototransistors are triggered by light. When light shines on the phototransistor a current is allowed to flow from the collector through the emitter. More light gives a bigger current.

3.4 Wiring up the phototransistor

In the collision tutorial, we used a pull-down resistor to help us read the state of the switch. In this case we'll use a pull-up resistor. The voltage across the phototransistor will tell us whether or not light is shining on it.



Choosing a value for R is slightly harder than before. Let's start with the datasheet for the L-53P3C phototransistor.

When the beam is blocked, there will be a very small current in the circuit. From the datasheet:

I_{CEO}	Collector Dark Current			100	nA	$V_{CE}=10V$ $E_e=0mW/cm^2$
-----------	------------------------	--	--	-----	----	--------------------------------

So the current will be around 100nA when the beam is blocked. For the Arduino to read a high input we will need at least 4V across the phototransistor, so 1V across the resistor. Here, as the current through the transistor increases, a larger current also flows through R1, and hence, the voltage drop across R1 is increased. In an ideal case, you would read exactly 5V when no light reached the transistor, and 0V at the output when maximum light reached the transistor. An interesting property of digital circuits is the reason why this still works, even with non-ideal components. In digital electronics, values are relative to a threshold voltage, where if they are above, they read as "1", and below as "0", even if they are not exactly 5V or 0V.

To calculate the maximum resistor size you can use the usual equation:

$$R = \frac{1}{100 \times 10^{-9}} = 10000000\Omega$$

So a 10MΩ resistor would be suitable.

When the beam is not blocked a larger current will flow. From the datasheet the maximum allowed current is 3mA.

$I_{(ON)}$	On State Collector Current	0.7	3		mA	$V_{CE} = 5V$ $E_e=1mW/cm^2$ $\lambda=940nm$
------------	----------------------------	-----	---	--	----	--

So for the on state we want a current of no more than 3mA, and a voltage across the phototransistor of 1V. This means we want 4V across the resistor.

Using the equation again:

$$R = \frac{4}{3 \times 10^{-3}} = 1333\Omega$$

Now that you have a maximum and minimum, you can experiment to find what resistor value gives you the best high and low level outputs. Higher resistor values will make the detector more sensitive.

Task: Try assembling the circuit that has been described. All the parts you need are available in the lab. If you need help, remember that you can always ask a mentor. Start on breadboard and once you're happy, solder it onto a stripboard.

Things to consider:

- Is a more sensitive detector necessarily better?
- Will the distance between the LED and phototransistor make a difference? Why?
- How could you make the IR beam from the LED narrower? Why might this be useful?

4 Ultrasound Sensor

4.1 About the sensor

Each team has been provided with two HC-SR04 ultrasonic distance sensors.

The sensor has four screw terminals, labelled **Vcc**, **Trig**, **Echo** and **GND**.

- **Vcc** is the positive voltage pin and must be connected to 5V.
- **GND** is the ground pin and must be connected to the Arduino GND.
- **Trig** is used to start a measurement. When it is pulled HIGH(5V) the sensor will send out a pulse of ultrasound.
- **Echo** will send out a pulse proportional in length to the distance measured by the sensor.

The Robot library will automatically measure the time between the trigger and when the echo is received, then calculate the distance. It can detect an object up to around 4m away.

4.2 Setting up the sensor

Using the provided PCB, connect the ultrasound sensor to the female header on one side of the board. On the other side of the board, there is a screw terminal. Connect the VCC pin to 5V from a bench power supply and connect the ground pin to the ground connection on the power supply. Connect the trigger to the red wire from the signal generator and the black wire to ground. Then using two channels on the oscilloscope, connect one channel to the trigger pin, and connect a second to the echo pin, connect both their ground connections to the ground of the ultrasound.

4.3 Calculating the speed of sound

To use the sensor, you will need to do some wiring.

1. Cut and strip 4 wires (one red, one black, 2 in colours of your choice)
2. Screw the red wire into VCC
3. Screw the black wire into GND
4. Screw the other two wires into Trig and Echo.
5. Connect the red wire to 5V on the Arduino and the black wire to GND on the Arduino.
6. Connect the **Trig** and **Echo** wires to any digital IO lines on the Arduino.
7. See the docs for information on using the sensor.

In order to estimate range using your ultrasound, you will need to calculate the speed of sound. Since velocity is given by:

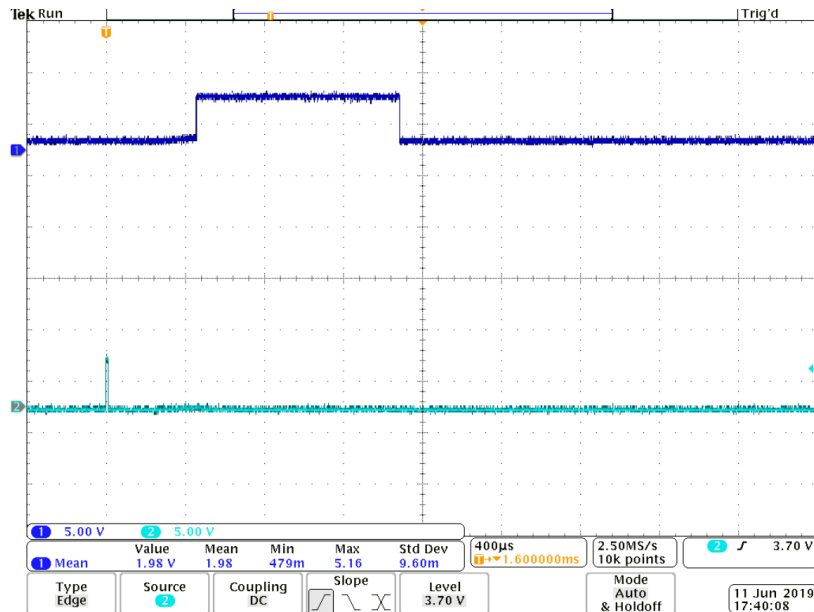
$$velocity = \frac{distance}{time}$$

You can calculate the speed of sound by taking a known distance, and seeing how long it takes for a response from the ultrasound.

Set the signal generator to produce a square wave with a frequency of 20Hz, with a low level of 0V, and a high level of 5V. Adjust the duty cycle of your square wave until the 5V period is approximately 10 μ s. You can verify this by looking at the value of each square on the oscilloscope's screen.

After 10 μ s the ultrasound sensor will then transmit. In order to get an accurate reading, you may need to adjust your oscilloscope's "Trigger" settings. Press the trigger menu button towards the right of the oscilloscope, and set your triggering on the same channel as the trigger pin of your ultrasound sensor, along

with setting the coupling to DC and the trigger type to edge. You can also adjust the trigger level, which is the threshold that will cause your oscilloscope to take a reading. See if you can find a section where the echo signal goes high (You may need to adjust your time scale to do this).



You should now see something like the image above. Using the oscilloscope, find the length in seconds of the high section of the echo signal. This is the length of time required to get receive a signal back. Since the ultrasound signal has to go outward and then return to the sensor, divide this time by 2.

- What is your measured speed of sound? Is this close to the expected value?
- How could you integrate this into your robot? How would you modify your code to measure range?

5 Analogue Inputs and Potential Dividers

5.1 Analogue and Digital Signals

An analogue signal is one which is continuous in time and in value, meaning that it can have any possible value at any possible point in time. This is in contrast to a digital signal, which is discrete in value, meaning that it can only have a predetermined set of possible values, in the case of a digital signal, either “1” or “0”.

5.1.1 Analogue inputs for Arduino

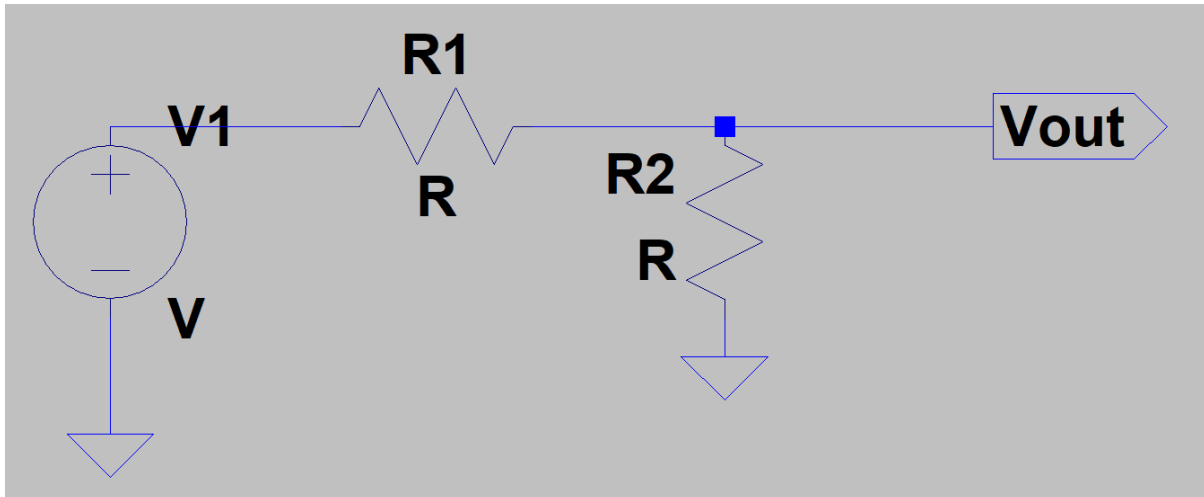
The Arduino you have been given for your robot has some analogue input pins. In the case of a microcontroller, these pins are connected to the input of an Analogue-to-Digital Converter (ADC), which takes a sample of the input at a point in time and converts it to a binary value suitable for a microcontroller to handle. The Arduino has a 10 bit ADC, which divides the range between 0V and 5V into 1024 steps. The change required in the analogue signal required for a 1 bit change in the value produced by the ADC is known as its “resolution”. The resolution of the ADC can be calculated as follows:

$$\text{Resolution} = \frac{5}{1024}$$

This gives the resolution of the ADC as 4.9mV. This function of the Arduino can be accessed using the **analogread(analogpin)** function on anything connected to the Arduino’s dedicated analogue pins. Your Arduino has 6 such pins, noted by A0, A1, A2, A3, A4 and A5. This may be useful for some of your sensors, as some of them do not return a digital value, and instead return a range of analogue values, particularly if you need more detail from the sensors than simply on or off.

5.2 Potential Dividers

A potential divider is a circuit that consists of 2 or more resistors. This is useful for bringing a voltage that is out of range of your Arduino down to a lower voltage which is in range, for your robot, this will typically be reducing a signal from 5V to 3.3V. Here, Ohm's law causes a drop in voltage across the first resistor (R1) which is proportional to its resistance, This leaves the second resistor (R2) to drop the remaining voltage.



Above is an example of a potential divider, where the output voltage (V_{out}) is given by:

$$V_{out} = V1 \times \frac{R2}{R1+R2}$$

5.3 Force Sensitive Resistors (FSR)

You may find at some point that you not only need to know if your robot is touching something, but how hard it is pressing against something. Force sensitive resistors provide one way in which you could do this. You can make your own FSR out of some of the conductive foam that you can find in the lab, and two sheets of any non conducting material, ideally something that does not compress very much, so that as much of the force as possible is transferred to the foam, rather than just deforming the material on the outside.

The resistance of any object is defined by three factors, the object's length (L), it's cross-sectional area (A), and a constant known as resistivity (ρ), this gives the resistance equation as:

$$R = \rho \times \frac{L}{A}$$

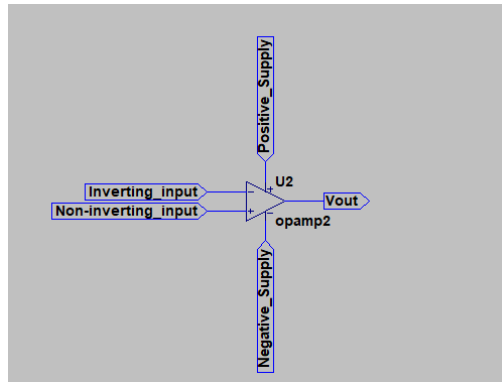
If you place this in a potential divider configuration with a resistor of a known value, you can work out the resistance of your sensor relative to the other resistor by the voltage across each of them.

- Can you build one of these sensors?
- What value should your other resistor be? Are there any advantages or disadvantages for lesser/greater resistances?

6 Use of Operational Amplifiers (Op Amps)

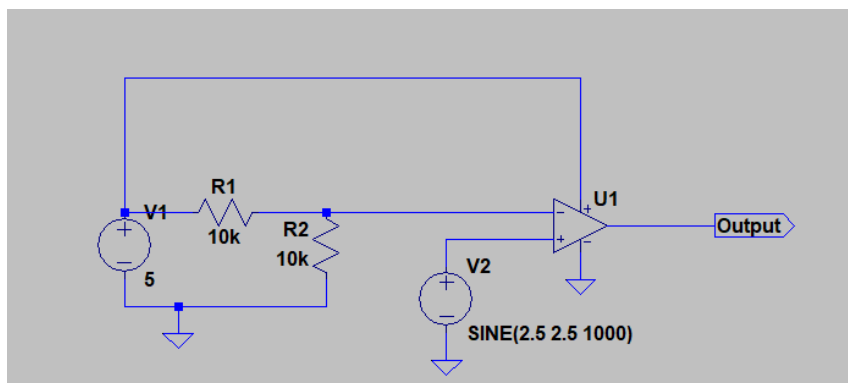
An operational amplifier is a device that takes 2 inputs, measures the difference between them, then outputs this voltage increased by a factor known as "gain". The two inputs of are normally identified as the "inverting" and "non-inverting", indicated on the schematic by "-" and "+" respectively.

An op-amp also requires two different supply voltages to function. Op-amps can be characterised by the power supplies they require, which divides them into 2 groups, single supply, which require a positive supply and ground, and dual supply, which require a positive and negative supply, along with a ground connection. These supplies determine the output range of an op-amp, as it will be unable to exceed these voltages, regardless of its gain. The gain of an op-amp is typically on the order of 100,000. In other words, if the supply were large enough, a 1 mV difference between the inputs would result in a 100V output.



In an analogue circuit, a gain of this magnitude is generally not useful, as the op-amp will saturate for extremely small signals. In an ideal op-amp, saturation means that the output is equal to the supply voltage. However, in the case of a real op-amp, this is normally slightly lower than the supply voltage. A common solution to this problem is to build a “feedback loop”. This is where the output of the op-amp is connected back to one of the inputs through a resistor. Several configurations of this are possible, however, in all cases the feedback is used to reduce the gain.

However, In circuits where analogue and digital signals are both present, the saturation of an “open loop” op-amp can sometimes be useful. An “open loop” configuration has no connection between its input and its output. This is the condition under which an op-amps gain is usually measured.



Try building the circuit shown above, connecting the signal generator in place of voltage supply V2. Set the signal generator to produce a sinusoidal output with a peak to peak amplitude of 5 volts and a DC offset of 2.5 volts.

- What happens if you change the values of R1 and R2?
- What if you were to connect your phototransistor in place of the signal generator?

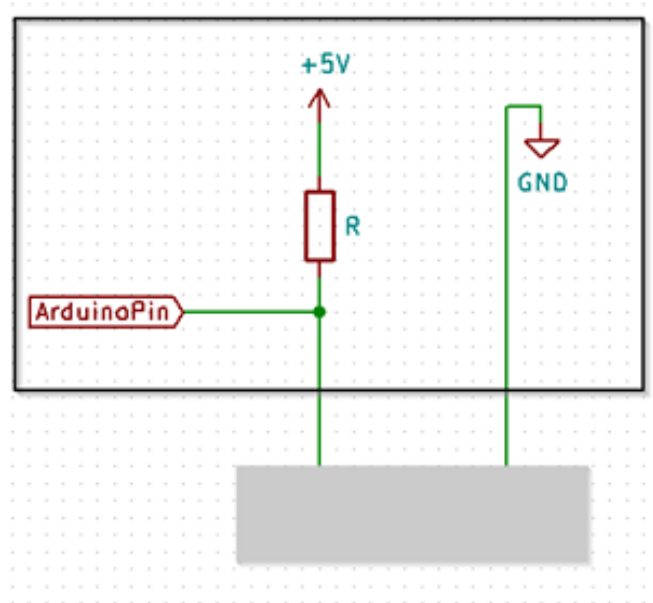
What do you think is happening here? Do you think that you could use it for your robot?

7 Metal Object Sensor

For this year’s game the scoring zones will be marked out with metal tape.

Using what you have learned in previous labs about switches, you can place metal contacts on your robot so you can detect when you have passed one of these strips.

The schematic below illustrates the basic circuit setup. Anything inside of the black border is in your robot:

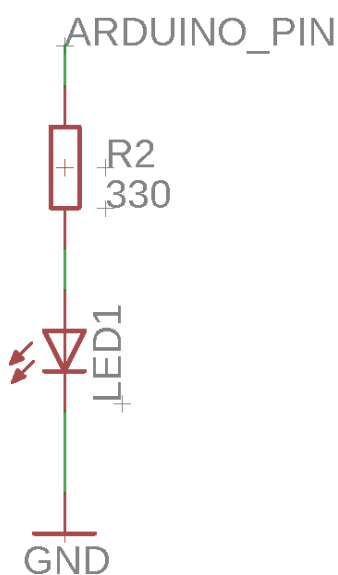


In the above figure, the grey box is a metal object. When the contacts are not on a metal object the Arduino pin is pulled **HIGH** by **R1**, but when the contacts are connected by a metallic object, the pin is pulled **LOW** through the object (The object and **R1** actually form a potential divider where the resistance across the object is *very low*, less than 1Ω . What is the voltage between the Arduino pin and 0V in this potential divider?).

7.1 Tasks

- Using the theory learned above, build a circuit that will implement the above functionality and think about how you could use this for your robot.
- Connect an LED to pin 2 on the Arduino and make it turn on when the circuit is completed. You will need to put a resistor in series with the diode to avoid damaging it. We'd recommend a value of 330Ω .

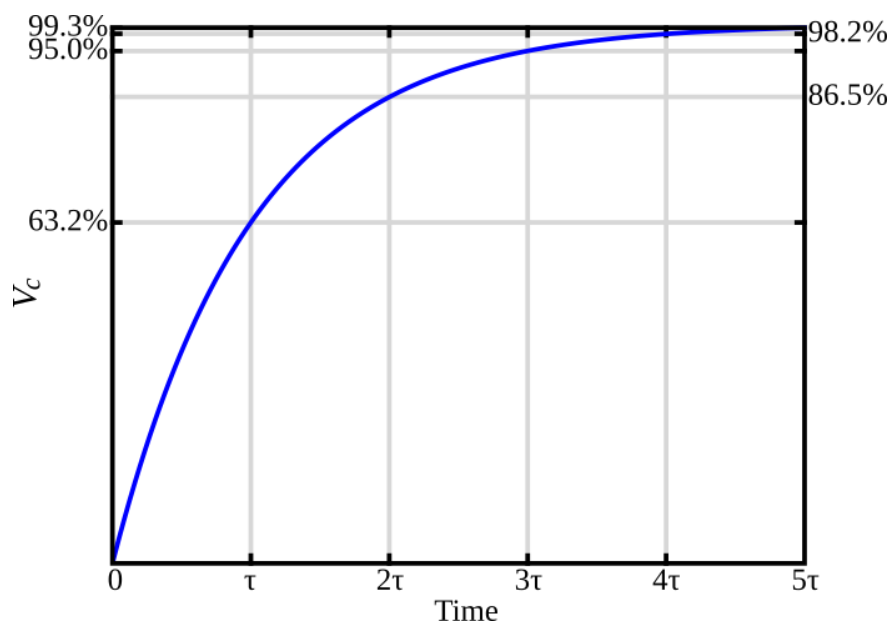
The diagram below illustrates the required circuit for above:



- Write a program that keeps track of how many times you have completed the circuit and make it illuminate an LED after the circuit has been completed three times. Connect the metal contact into pin 5.

7.1.1 Switch bounce

You might find that your counting circuit from the previous task is quite unreliable and seems to register lots of contacts each time you touch something. This is because the contacts can easily 'bounce' when they are connected. This effect happens very quickly, so you won't be able to see it but your Arduino can read it's input pins fast enough for it to count each small bounce as a new time a connection has been made. To make your counter more reliable you need to make sure it only counts once each time you connect the metal contacts. - One way of doing this is to force the Arduino to wait for a few milliseconds after detecting a connection has been made so that all the 'bouncing' has stopped before it carries on running it's code. This means it is spending time doing nothing though - including not controlling your robot so we need to find a "non-blocking" way of dealing with this instead. - It is possible to find out how many milliseconds have passed since the Arduino turned on (ask for help if you need it). You could use this to keep track of when each connection is detected and only count the ones that aren't caused by bounces. - You can also solve it in hardware, by adding a capacitor to your circuit. The behaviour of a capacitor is dependent on *the rate of change of voltage* across it. This means it will smooth out voltage spikes, including getting rid of the bouncing.



When the voltage across the capacitor changes (like the step input above) the voltage across the capacitor takes time to reach the supply. That rise is exponential, described by the time constant $\tau = RC$. Think about what capacitance values might be suitable. - There are also more complicated circuits you could use for this that would work better (if you are interested, have a look at Schmitt triggers). However like most engineering problems, you need to weigh up the pros and cons of each option based on:

- Performance
- Cost
- Simplicity
- Reliability

as well as other factors based on your specific project. Which method do you think is the best for your robot?

- Try to improve your counting circuit / code to deal with switch bounce. If you use a capacitor, this is where it should go:

