SourceBots Electronics Labs

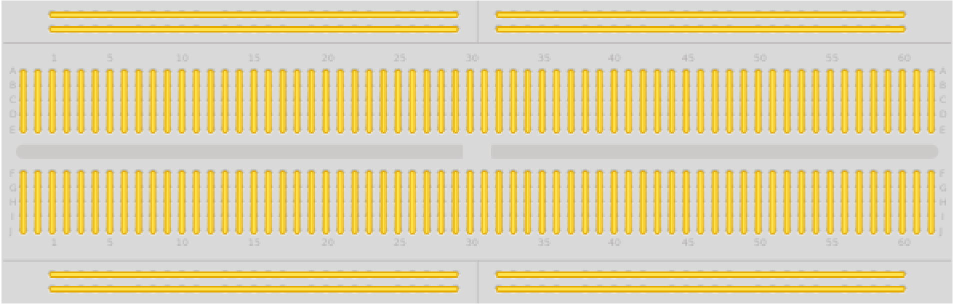
Summer 2019

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 protoboard) 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 protoypes, or small batches of simple circuits. Stripboards are arranged in a similar way to breadboards. Place the components legs through from the non-copper (FR4) 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 labled "+" 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 lose 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

2.1 Outcomes

2.1.1 Robot Progress

- Creation of collision/bump detectors

2.1.2 Knowledge

- Basic electronics concepts
- Switches and the need for pull-up/pull-down resistors

2.1.3 Skills

- Breadboard prototyping
- Basic soldering

2.2 Basic Theory

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. Since it is the *difference* between two points, all voltages are measured relative to another point. This is usually the *0V* or *ground* line. Potential difference is measured in **Volts (V)**.

Resistance is defined as the ratio of potential difference to current. You can think of it as a way of measuring how difficult it is for current to flow through something. It is calculated as:

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

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

or, rearranging:

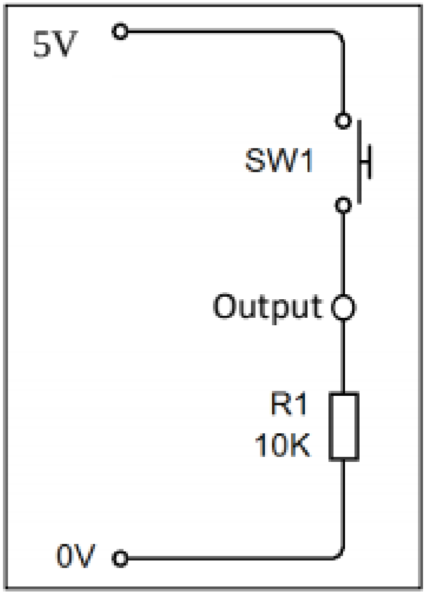
$$V = IR$$

Resistance is measured in Ohms (Ω).

Now that we know some theory, we can do something useful. You can use the chart in the appendix to work out the resistor's value from it's colours.

2.3 Building a simple sensor

Take a look at the following circuit:



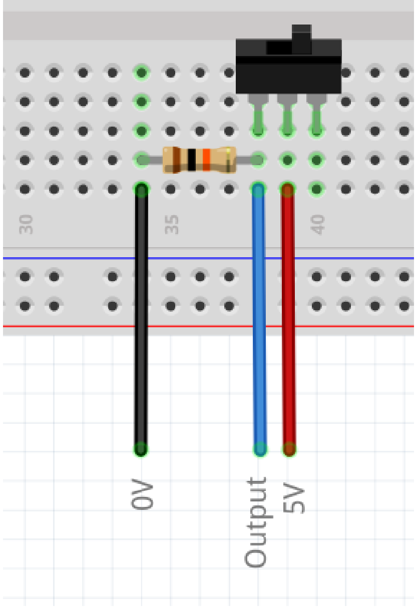
When the switch is closed (conducting), the output terminal is directly connected to the 5V line so the output is at 5V. When the switch is open (not conducting), there is no current in the resistor. The voltage drop across a resistor is proportional to the current through it ($V = IR$), so if there is no current there will be no voltage drop across the resistor, and 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 anything. This is called a *floating line*. The voltage between this point and the Arduino 0V line will be unpredictable, picking up random signals from the surroundings. Connecting the output to 0V through a resistor like as shown above makes sure the voltage at the output is 0V when the switch is not pressed; we call this a *pull-down resistor* which "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.



2.3.1 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

3.1 Outcomes

3.1.1 Robot Progress

- Creation of tripwire sensors
- Creation of reflection sensors

3.1.2 Knowledge

- Basic understanding of LED and transistor operation
- Calculating and selecting suitable resistors for a circuit

3.1.3 Skills

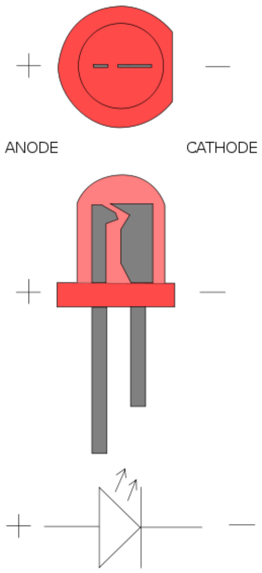
- Breadboard prototyping
- More complex soldering
- Using a multimeter to take measurements

3.2 Introduction

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: 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.

3.3 Diodes and LEDs

LEDs are a form of diode. A diode is an electronic component which will only conduct in one direction. The terminals are called the *cathode* and *anode*. The diode will only conduct from anode to cathode.



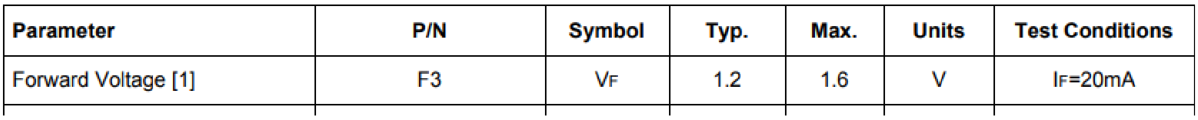
LEDs have a high *forward voltage* (the potential difference needed for it to conduct); roughly 1.2V for the IR LEDs that you will be using here. Each Coulomb of charge passing through it will lose 1.2J of energy, which is converted to Infrared radiation.

3.4 Wiring up the LED

LEDs are polarized devices. If you plug them in backwards they will not work. 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.

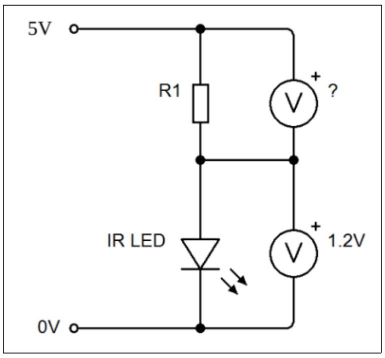
A useful way to remember this is **short to nought**.

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:



We can now figure out the value of R needed using R = V/I. V is the voltage drop across the resistor (5 – 1.2 = 3.8V) 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!

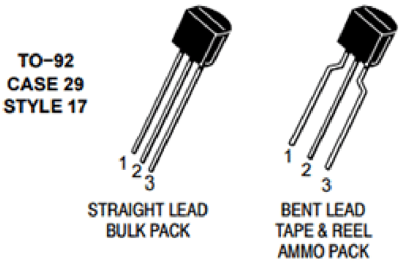
Wire this circuit up on a breadboard. A smart phone camera or webcam can see IR light so you can use it to test if your circuit works. (Ask if you need help).

3.5 Transistors and Phototransistors

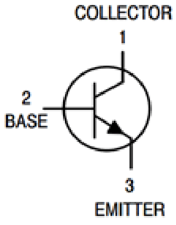
Transistors can be used to make very complicated devices, but for our purposes they will be used as an infrared detector. A transistor is a three terminal device. It has a collector, an emitter and a base. The key feature of the transistor is that the current that flows into the base determines the current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

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

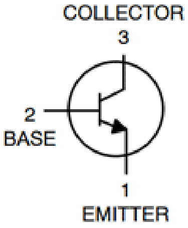
A typical transistor will have a gain of 100. Below are a couple of examples of common transistors.



Pin Diagram



Symbol of BC547A



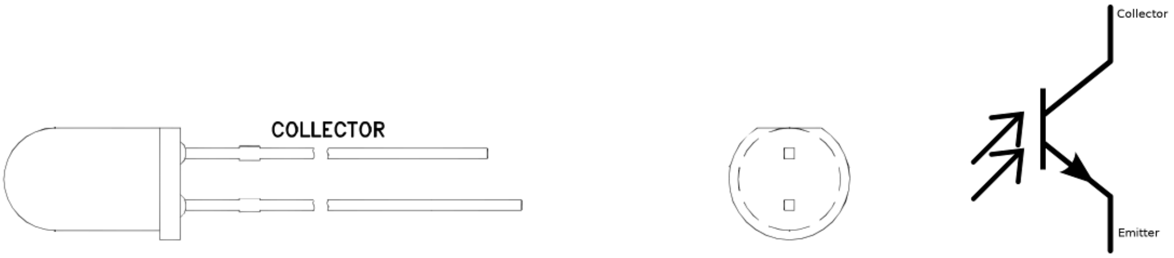
Symbol of PN2222A

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, with base current instead of a physical force.

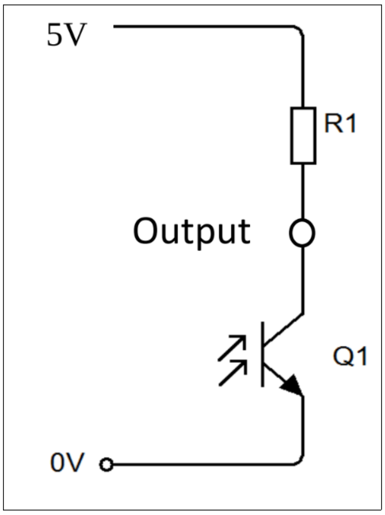
A phototransistor behaves in the same way, only instead of having a base wire phototransistors are triggered by light. When light shines on the phototransistor a current is allowed to flow from the collector through the emitter. More light gives a bigger current.

3.6 Wiring up the phototransistor

The phototransistor we are using is the L-53P3C.

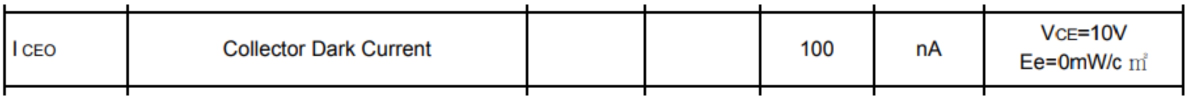


In the collision tutorial, we used a pull-down resistor to help us read the state of the switch. In this case we'll use a pull-up resistor. The voltage across the phototransistor will tell us if IR light is shining on it.



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 Ee=0mW/c㎡

So the current will be around 100nA when the beam is blocked.

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. In digital electronics values are relative to a threshold voltage, where if they are higher they read as "1", and lower as "0", even if they are not exactly 5V or 0V.

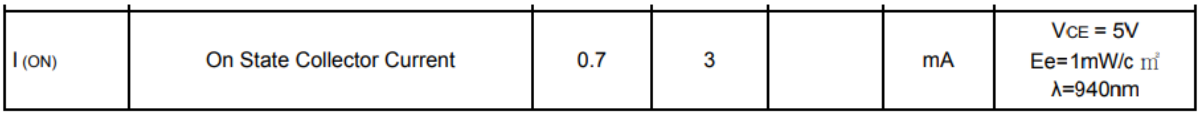
The Arduino will read a digital input as *high* or "1" if there is at least 4V across the phototransistor, so 1V across the resistor. When the current through the transistor increases, a larger current also flows through R1 so the voltage drop across R1 is increased (as $V = IR$).

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

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

This means that in theory a $10M\Omega$ resistor would be suitable. In practice, you can probably use a smaller value than this.

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	VCE = 5V Ee=1mW/c㎡ λ=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:

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

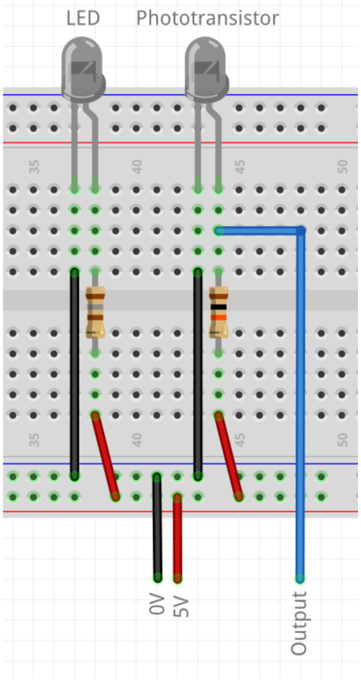
Now that you have a maximum and minimum, you can experiment to find what resistor value gives you the best high and low level outputs. Higher resistor values will make the detector more sensitive.

3.6.1 Reflection sensor

The tokens this year are covered in reflective foil. This gives you lots of options for using your IR LEDs in creative ways. You could point the IR LED and the phototransistor at a token and see if the light is reflected. If it is, your robot must be approaching the token at right angles, otherwise the reflected light would miss the detector. You could even work out which direction to turn based on the reflection if you had a few phototransistors on your robot.

3.6.2 Task - Make an infrared tripwire sensor for your robot

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 test the circuit using a multimeter. Try different resistors to change the sensitivity. Once you are happy with the values, solder the circuits onto stripboard. Afterward, see if you can turn your tripwire sensor into a reflection-style sensor. Discuss with your team how you might use theses sensors on your robot.



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?
- The IR sensors can be used with the analogue inputs of the Arduino too. Think about how you could make use of this.

4 Ultrasound Sensors

4.1 Outcomes

4.1.1 Robot Progress

- Implementing ultrasonic distance sensors

4.1.2 Knowledge

- Understanding pin labels and operation of the HC-SR04 sensor
- Understanding the advantages and limitations of ultrasound sensors

4.1.3 Skills

- Use of screw terminals
- Use of oscilloscopes
- Use of signal generator

4.2 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 for a time proportional 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.

4.2.1 Task 1 - 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
- 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.
- Connect one oscilliscope channel to the trigger pin
- Connect the second oscilloscope channel to the echo pin
- Connect both their ground connections to the ground of the ultrasound.

4.2.2 Task 2 - Checking the sensor and estimating the speed of sound

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

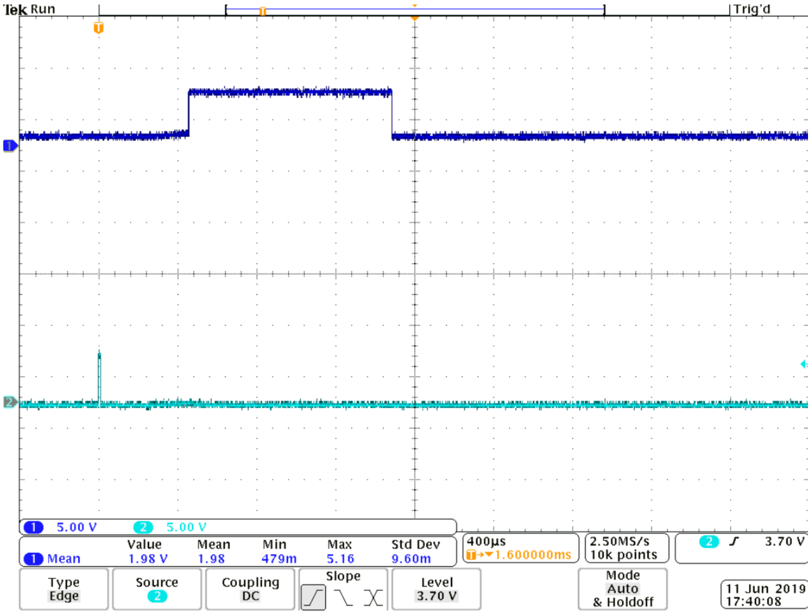
$$velocity = \frac{distance}{time}$$

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 10μs. You can verify this by looking at the value of each square on the oscilloscope's screen.

After 10μs 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
- Set 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).



1 5.00 V 2 5.00 V

	Value	Mean	Min	Max	Std Dev
1 Mean	1.98 V	1.98	479m	5.16	9.60m

400µs T→1.600000ms
2.50MS/s 10k points
2 ∫ 3.70 V

Type Edge
Source 2
Coupling DC
Slope
Level 3.70 V
Mode Auto & Holdoff
11 Jun 2019 17:40:08

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 detect the reflected ultrasound. **Since the ultrasound signal has to go outward and then return to the sensor, divide this time by 2.**

- Use your sensor to calculate the speed of sound. How close is this to the expected value?

4.2.3 Task 3 - Testing the sensor

- Increase the distance between your sensor and the target object until it is no longer detected. What is the maximum range of your sensor? Does the reliability change when you increase the distance?
- Set the distance to 50cm and try rotating the target object. How does this affect the results?

Things to consider:

- How could you use this sensor on your robot?
- How could you improve the reliability of your measurements?

4.3 Using the sensor on your robot

The Arduino board in your robot kit includes code to automatically calculate the distance measured by the sensor.

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.

5 Analogue Inputs and Potential Dividers

5.1 Outcomes

5.1.1 Robot Progress

- Creation of analogue force sensors

5.1.2 Knowledge

- Potential divider circuit theory
- Analogue sensor operation
- Basic understanding of analogue to digital conversion

5.1.3 Skills

- Constructing a simple force sensor
- Using a multimeter or oscilloscope for testing

5.2 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. For example, mass is an analogue quantity. Digital signals have **discrete** values, meaning that it can only have a predetermined set of possible values. For example, shoe size is a discrete quantity. In the case of a digital signal, the allowed values are either "1" or "0".

5.3 Analogue inputs for Arduino

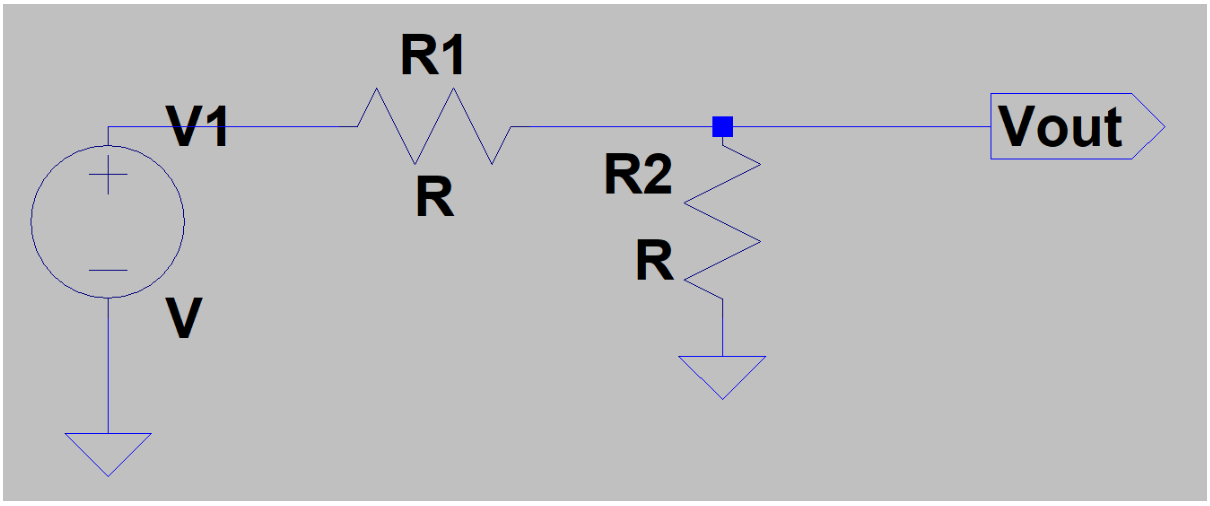
The Arduino you have been given for your robot has some analogue input pins. These pins are connected to the input of an Analogue-to-Digital Converter (ADC), which takes a sample of the input at a point in time and converts it to a binary value suitable for a microcontroller to handle. The Arduino has 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 (although A4 and A5 are reserved for I2C communication so you can't use them). 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.

5.4 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, typically 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 (Vout) is given by:

$$\text{Vout} = \text{V1} \times \frac{\text{R2}}{\text{R1+R2}}$$

If one of the resistors changes value the output voltage will also change. This means potential dividers can be used as part of a sensor circuit. For example, a light sensor can be made if one of the resistors is a light dependent resistor.

5.5 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 to 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 transfered 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). The resistance is calculated by the equation:

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

If you place your foam sensor 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.

5.5.1 Task 1 - Build an analogue force sensor

- Take a small piece of conductive foam and insert two wires into it. Make sure to strip the ends of the wire first!
- Use a multimeter to measure the resistance between the two terminals. You should see the resistance drop significantly when you press on the sensor. The harder you press, the further it should drop.

- Cover the foam and a small part of the wires with electrical tape to make sure it holds together nicely.
- Optionally add a piece of rigid material on either side of the foam to more evenly distribute the force applied.

5.5.2 Task 2 - Experiment with your sensor

- Connect the sensor in a potential divider with another resistor and record the range of voltages when a force is applied.
- Test a few other values of fixed resistor. How does this affect the range of voltage?

Things to consider:

- Which value of fixed resistor gives the greatest variation? You might be able to investigate mathematically.
- Is greater sensitivity always better?
- How could this type of sensor be useful for your robot?

6 Use of Operational Amplifiers (Op Amps)

6.1 Outcomes

6.1.1 Robot Progress

- Implementing a comparator system

6.1.2 Knowledge

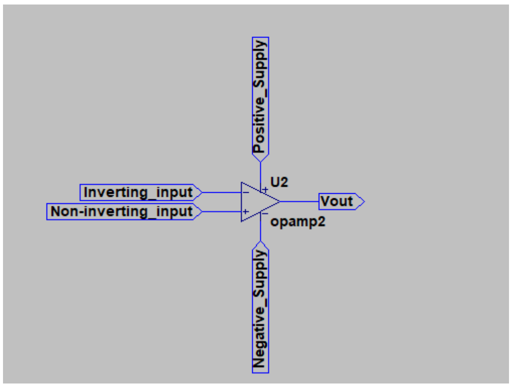
- Op-amp operation
- Configuring op-amps as comparators

6.1.3 Skills

- Breadboard prototyping
- Basic soldering

6.2 Theory

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.

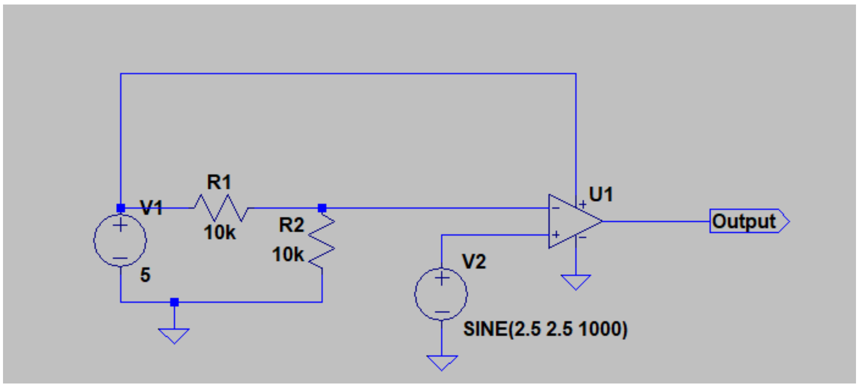


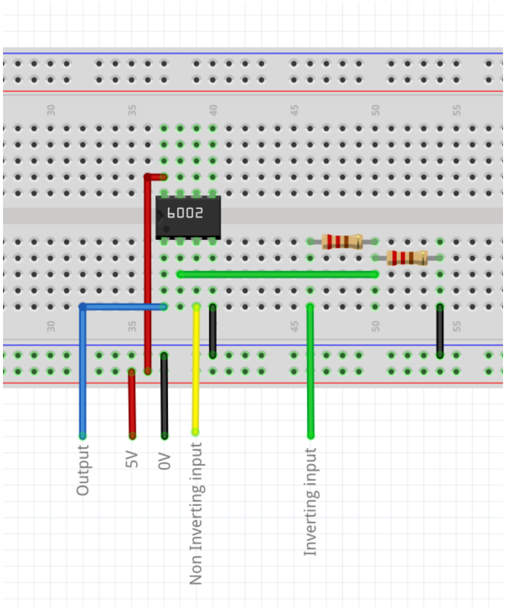
An op-amp requires two different supply voltages to function. Op-amps can be either single supply, which require a positive supply and ground, or dual supply, which require a positive and negative supply along with a ground connection.

These supplies determine the maximum and minimum output voltages of the op-amp as it will be unable to exceed these voltages, regardless of its gain. When the output is at a maximum or minimum we say the op-amp is *saturated*. For a real op-amp the saturation output is usually slightly lower than the power supply voltage.

The open-loop 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.

The simplest use of an op-amp is a *comparator* circuit, as shown below.





In this configuration the output will be high if the non-inverting output is higher than the inverting input. Otherwise the output will be low.

6.3 Task - Building a comparator circuit

- Build 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 happens if you connect your phototransistor in place of the signal generator?

Things to consider:

- How could a comparator circuit be useful for your robot?
- What is the advantage of using this circuit instead of analogue readings from the Arduino?

7 Metal Object Sensor

7.0.1 Robot Progress

- Creation of a metal tape detector

7.0.2 Knowledge

- Need for a pull-up resistor
- Switch bounce and how to avoid it

7.0.3 Skills

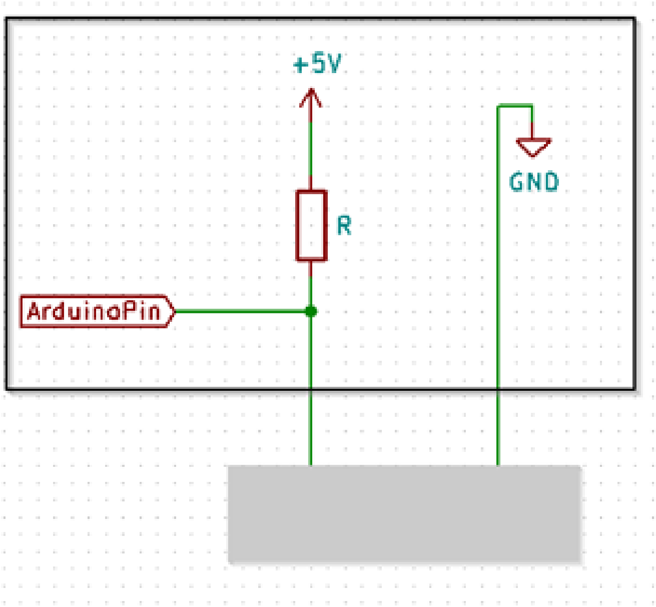
- Breadboard prototyping
- Component selection
- Basic soldering

7.1 Theory

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.

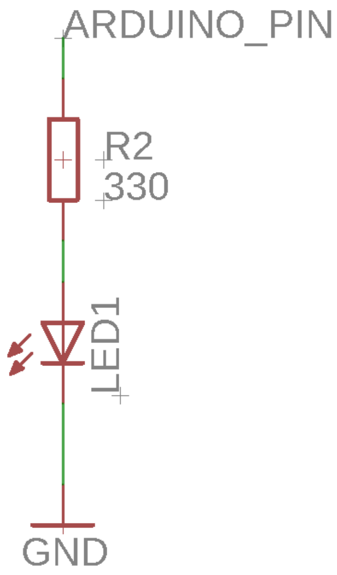
The schematic below illustrates the basic circuit setup. Anything inside the black border is in your robot:



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 metalic 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).

7.2 Task 1

- Build a metal object sensor using the theory above
- 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.



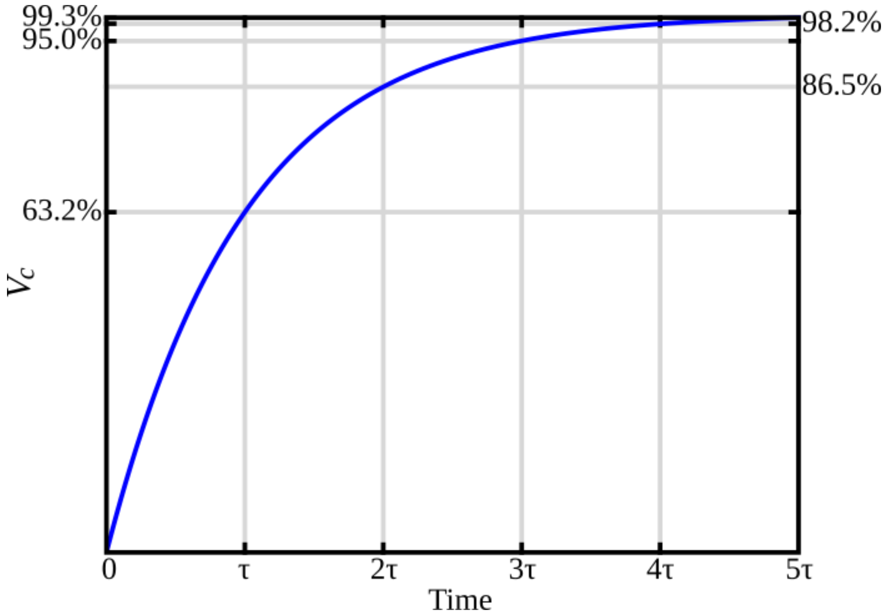
- Write a program that keeps track of how many times you have completed the circuit and make it iluminate an LED after the circuit has been completed three times. Connect the metal contact into pin 5.

7.3 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 will usually '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 time a contact is made. To make your counter more reliable you need to make sure it only counts once each time.

7.3.1 *Debouncing* a switch

- You can avoid switch bounce in software by programming the robot to wait for a few milliseconds after detecting a connection has been made. This means it is spending time doing nothing - including controlling your robot - so we need to find a "non-blocking" way of dealing with this instead.
- You can also solve it in hardware, by adding a capacitor to your circuit. The potential difference across a capacitor takes time to change, so it will *smooth* the signal from the switch.

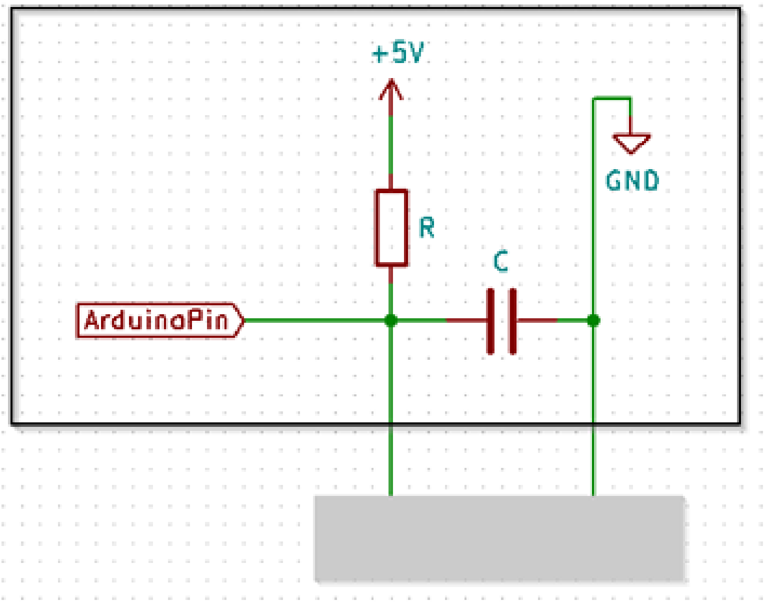


The voltage rise is exponential, described by the time constant $\tau = RC$. Think about what capacitance values might be suitable.

- There are also far 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?

7.4 Task 2 - Debouncing

- Implement a debouncing solution for your metal strip detector.
- If you decide to use a capacitor, your circuit should be:

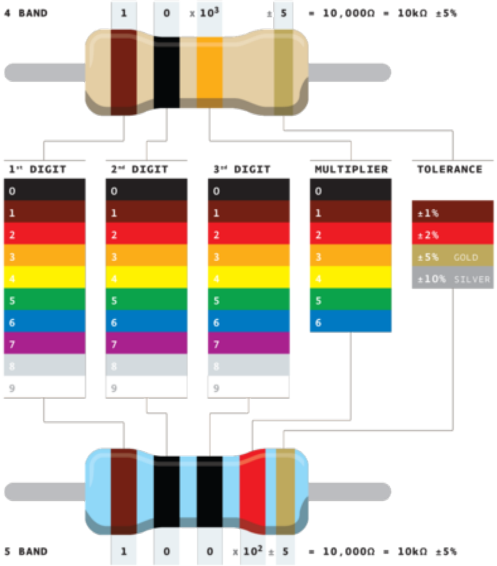


Things to consider:

- How can you mount this sensor on your robot?
- How many sensors would you need?

8 Appendix

Resistor colour codes


4 BAND
1
0
x 10³
± 5
= 10,000Ω = 10kΩ ±5%
1st DIGIT
0
1
2
3
4
5
6
7
8
9
2nd DIGIT
0
1
2
3
4
5
6
7
8
9
3rd DIGIT
0
1
2
3
4
5
6
7
8
9
MULTIPLIER
0
1
2
3
4
5
6
TOLERANCE
±1%
±2%
±5% GOLD
±10% SILVER
5 BAND
1
0
0
x 10²
± 5
= 10,000Ω = 10kΩ ±5%