Let's build a Robot.

Our robot will use an ultrasonic sensor and servos to navigate its way around a maze. We will be making 2 voltage circuits:
- A 5 Volt – for our ultrasonic sensor, sound and lights powered by our Arduino.
- And a 6 Volt – to power our servo motors, direct from a battery pack with 4 x 1.5 Volt AA batteries.

LED: Stands for "Light-Emitting Diode." An LED is an electronic device that emits light when an electrical current is passed through it.
All LEDs are made with a maximum voltage they can connect to without being destroyed. This voltage is typically between 1.5 volts and 3.6 volts.

Resistor: A resistor is an electrical component that limits or regulates the flow of electrical current in an electronic circuit.
We use resistors in this case to stop too much current damaging our LED lights.
Let's start wiring!
1st project make a circuit to flash the lights.

Our led has a positive and a negative leg, the longer leg is positive.

We going to connect our led’s to row ‘E’, leaving one space between the two legs.

Step 1 : 1st LED
1. Connect the positive leg to E3 and the negative leg to E1
2. Connect our 220 ohm resistor from D1 to the blue negative rail
3. Connect light to the board, run a long signal/control wire from D3 to pin 6 on the Arduino.

2nd LED
1. Connect the positive leg to E30 and the negative leg to E28
2. Connect our 220 ohm resistor from D28 to the blue negative rail
3. Connect light to the board, run a long signal/control wire from D30 to pin 7 on the Arduino.

We will now power the breadboard, run a wire from the 5V pin on the arduino through to the red positive line on the breadboard.

We also need to run a ground to the board, run a long black wire from GND to the blue rail.

We are now ready to code the lights, open up mBlock and add the following code:
HIGH is Arduino for ON and LOW is Arduino for OFF. The wait allows us to see the light blink.

When you are ready, upload your code to the board:
1. Under Boards, select Arduino Uno
2. Under connect select Serial port and then the serial port with the highest number.

Click on Edit, tick Arduino mode and upload the code to Arduino.
Second project: Measuring distance by adding a sensor

Ultrasonic: Ultrasonic sensors generate high-frequency sound waves and evaluate the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance to an object. We can then get the lights to flash based on whether the distance is close or far.

Step 2 – Add the ultrasonic to the breadboard along pins A14-A17 with the sensors pointing away from the board.
Using a short wire, connect VCC (Power) from pin C14 to the red power rail on the breadboard. Using a short wire, connect GND (Ground) C17 to the blue ground rail on the breadboard. Using a long wire, connect the TRIG on C15 to pin 13 on the Arduino.
Using a long wire, connect the ECHO on C16 to pin 12 on the Arduino.  
Add the code to our Mblock programming.  
We will need to add a variable (distance) as the value for distance as read by the ultrasonic is always changing.  
We will also need to create an “if ... then” loop to our program.  
These are the building blocks for most code, “IF” something then do “action”.  
In this case we will set a minimum value for our distance variable, when it’s too close, the light on pin 6 will flash, otherwise the light on pin 7 will flash.

When you are ready upload the code using the previous instructions.
Project 3 – Adding sound with a Piezo

A Piezo is a speaker which converts the voltage passed through it to movement. The vibration of this movement becomes sound.
We have added a 10kilo Ohm resistor to our piezo.

Connect the black ground wire to E5 and the red power wire to E7.
To complete the circuit, we connect a 10K resistor from B5 to the blue ground rail and run a long control/signal wire to pin B7 on the Arduino.
Adding to our code a single block, we will change the note to A2 and the beat to quarter.
Assembly time!

Our robot should now look a bit like this.

Using the really long bolts put them through the Arduino on the top layer, securing them on the bottom of the chassis with the nuts.
Using blue tack, affix the breadboard to the the upper section of the chassis.
Add the battery pack and the 9V battery underneath the top layer but leave the wheels off for now.
A servo is a small gear motor that is designed for precise motion. Most Servos only have a range of 0-180 degrees, our servos have been altered so they turn all the way round (360 degrees). Let’s wire up the servos.

We create a new 6V circuit on the other side of the board. With the battery pack, put the screw terminal with the black wire connect to the blue ground rail at 14 and the pin connected to the red wire pushed into the positive red rail at 14. Each servo has a black ground wire, a red positive wire and a white signal/control wire. Using short wires, add in your wires to the servos, where possible connect red to red and black to black.

Connect the black wires from both servos to 28 and 29 on the blue rail.
Connect the red wires from both servos to 28 and 29 on the red rail.
Connect the control/signal wires from the left servo to pin 9
Connect the control/signal wires from the right servo to pin 8.

Finally add in a short black wire from the blue ground rail to pin GND above pin 13 on the board.

Save your code to the desktop, changing the file name to your name.
We are now going to test our servos.
Firstly, save your work to the desktop using your name as the filename.

Next make a new project (don’t worry, we will get back to the old project in a moment).
We are just going to make a program to test the wheels.
In the new project add the following code
Our servos have been adapted from servos that only go 180 degrees. That’s why to go one way the code is 0 and to go the other way is 180 (even though a circle is 360).

Full rotation one way = 0
Full rotation the other way = 180
Stop = 90.

Upload the code and change it until our wheels are going forward, once we know our wheels are going forward we can then determine how to turn the robot right, left and backwards.

If we determine that forward on the left servo (pin 9) is 0 and forward on the right servo (pin 8) is 180
Then turning right is a case of making the left servo(Pin 9) go forwards (0) and set the right wheel to stop (90)

```
set servo pin 9 angle as 0°
set servo pin 8 angle as 90°
```

To make the robot turn left, we make the right servo(pin8) go forward(180) and the left servo(pin9) to stop(90).
To make it reverse, we just set the values opposite to our forward motion.

```
set servo pin 9 angle as 90°
set servo pin 8 angle as 180°
```

Once we have worked out our forward motion, we can upload our old code and make any changes necessary.
The final code (before testing)

If we were speaking to our robot in english we would say something like:

If you see an obstacle:
- reverse
- flash the light on pin 7
- make a sound
- turn
otherwise
- flash the light on pin 6
- go forward
We are now ready to test our robot.
Tidy up the wires at the back of the robot with a pipe cleaner.
We will need to then hook up our 9 Volt battery to the robot, when you are ready, please let us know.
If we do this part wrong, we may break our robot.

What can we do to make our robot more responsive?

Question: How long does it take for the robot to take its next distance measurement?

Question: What happens if you change the distance variable from 20?

Question: There is another way for the robot to turn without using stop (90) can you figure it out?