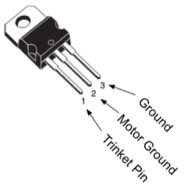
Circuit Assembly

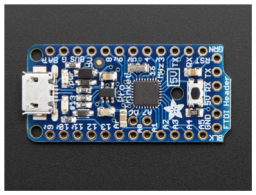
Important Circuit Parts

MOSFET
Trinket



Infrared (IR) Sensor

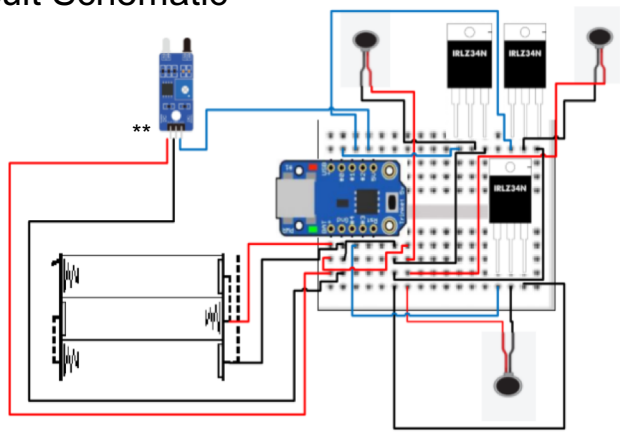
Arduino



Circuit Assembly

1) As you can see most of the circuit is already assembled
   a) The MOSFET and IR Sensor are the two components left to add to the circuit
2) Using the two MOSFETs already wired, connect the last one in the open quadrant of the breadboard
   a) The left is the wire that goes to the Arduino sensor Pin 4
   b) The center is the wire that goes to the motors ground wire
   c) The right pin is the wire that goes to the ground node
3) The IR Sensor has 3 wires
   a) Power, which connects to the pin labelled 'Bat' on the Arduino
   b) Ground, which connects to the pin labelled 'Gnd' on the Arduino
   c) Out, which connects to Pin 2 on the Arduino

Circuit Schematic



** Note: For some IR sensors, the blue and red traces from the diagram will be flipped. Make sure to read the breakout board pinouts, and ask for help if needed!

Loading Software on the Robot

What's in the provided starter code?

void setup()

Every program needs a setup() function to tell the Trinket what each pin will be doing

1) This is done with pinMode(variableName, INPUT/OUTPUT);
2) For more complex programs this setup() section is also used to set up an output window, declare initial motor spinning direction, declare if some devices are connected
3) IMPORTANT: This part of the code only runs one time

Motor Movement

The next command we will be using is 'analogWrite()'

1) analogWrite() simply writes/outputs a value rather than reading/inputting one
   a) In order to output a value we must first know what the receiver expects
      i) In this application we are writing/outputting the speed of the motor
      ii) This is done by using a PWM (Pulse Width Modulation) value (which ranges from 0 (off) to 255(full power))
2) To use analogWrite type: analogWrite()
   a) Then type the Pin# or variableName followed by a comma and the PWM value in the command
   b) You may also use digitalWrite() for this application but that limits you to either turning the motor on or off
      i) By typing in LOW or HIGH instead of the PWM number from 0 to 255

Note: Programming is capitalization dependent

Uploading the code

1) To verify the code does not have any typos select 'Verify' in the top left corner
2) To transfer the program you are writing to the robot we must first connect the Trinket to the computer using the micro-USB cord
   a) Make sure red LED on the trinket is blinking before hitting upload
   b) If not blinking unplug it and plug it back in
3) Then select 'Upload' in the top left corner to transfer the code
4) When it is completed the you should see 'Done Uploading' near the bottom of the screen
   a) If there is an error first try to unplug the micro-USB and plug it back in
5) Turn on the battery pack with the switch
6) If it starts to vibrate you have successfully uploaded code

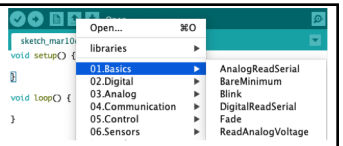
Using the Robot's Sensor

Reading from IR Sensor

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

1) In Arduino programming there are two main functions to every program
   a) The setup() function where the board is configured to what you will be using it for
   b) The loop() function where the bulk of the program goes
      i) This is where you write the code that you want the robot to perform
2) In order to read from the sensor we will use a command called: 'digitalRead()'
   a) This command collects data from the sensor and either returns/outputs a value of 0 or 1
   b) digitalRead(Pin number or Variable name of the pin)
   c) The 'digital' part means it can only output a value of 0 or 1
3) We can also store the output (0 or 1) of the sensor in a variable so we do not need to use 'digitalRead()' every time we want to use the sensor

Creating Global Variables



4) To do this create an 'int' global variable
   a) At the top of the program type 'int'
      i) 'int' means it is a whole number (0 and 1 do not have any decimals)
   b) Next type the name you would like to call the variable (Ex. SensorValue, sVal, detect, etc.)
   c) Lastly press the equals sign and the number of the pin the wire is connected to
5) If statements change allow us to only run certain code if a condition is met
   a) For this application we will simply write "if (Variable == Value) {Do certain code}"
6) Initially we can set the condition to be if the sensor equals a value it turns a built in LED on
   a) To do this we will use digitalWrite(LED_BUILTIN, HIGH or LOW) inside the if statement
   b) The example 'Blink' is an example of how to turn on and off the LED on a timer

Hint: Remember that the sensor can only output two values

Hand Following

Linking the Motor and the Sensor

Up to this point there has been no connection between reading the sensor value and turning the motors on

1) For this application (Hand Following) we want to write code that moves forward when the sensor sees an obstacle but stops if it does not see one
   a) To do this we can also use the previously introduced "if statement"
2) As each vibration motor and MOSFET are slightly different you may notice your robot likes to drift/turn in one direction
   a) To counteract this, vary the PWM/motor speed values of each leg
   b) You may also move the legs around themselves but make sure that the base is still flat and balanced

You may also need to switch the value of the if statement condition if the robot is running initially and stops when it detects something

Obstacle Avoidance

Converting from Following to Avoidance

1) The simplest way to convert from following to avoidance is to reverse the
   if statement
   a) To do this replace the 0 with a 1, or the 1 with a 0
1) You can also add a second if statement for the other sensor value that pulses the motors to make the robot rotate away from the obstacle

Tips

- using "//" to make comments is very useful to keep your code organized
- Not ever motor needs to be at the same PWM value
- Vary which motors are on and the power they are at. Observe the related movement for each of these
- There are 22 possible leg placements on the bottom of the robot, placing the legs/motors in different locations will cause the robot to move differently
- Always remember to end each line with a semicolon
  - This tells the computer that that line is done and you are starting a new command
- Remember to upload your code when you want to test the robot again

Feel free to ask your neighbors/group for questions and we will also be walking around to help if anyone needs it