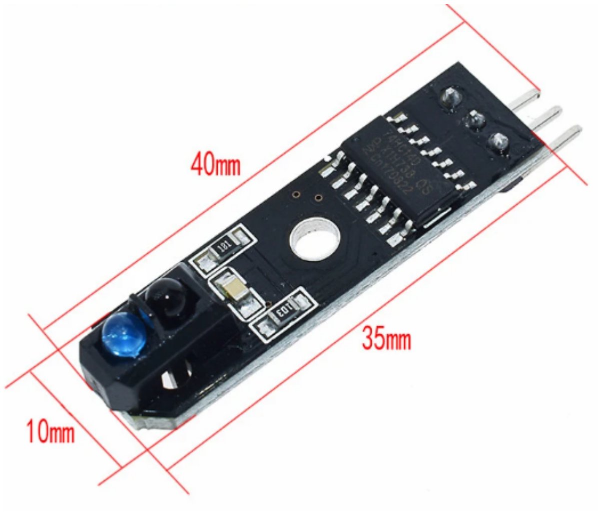
Infrared Sensor

This sensor is sometimes called the line sensor and in our use, it is going to help us detect when our bots are at the edge of the ring or arena. The arenas are white with black outlines and so we need sensors that can detect the differences in these colors.



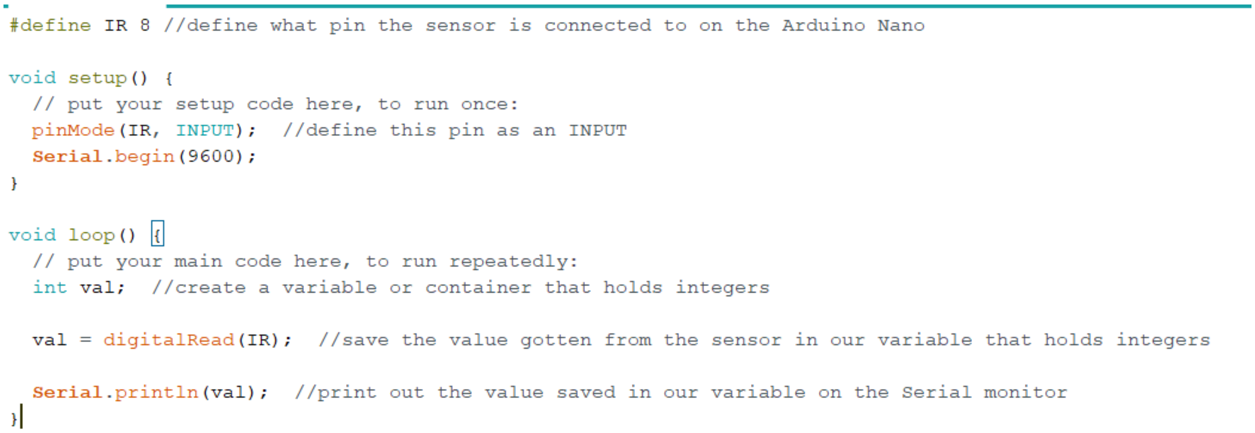
KY-033 infrared sensor

This sensor is very simple to use and it has only three pins. The pins and connections to the Arduino are shown below:

Arduino Pin	Pin on the module
D8	OUT
3V3	VCC
GND	GND

The working of this particular module is easy. When the sensor (LEDs) are against a surface that is white, they give out a HIGH output, when they are against a surface that is darker, they give out a LOW output. They also give out a LOW output if you randomly hold them up in the air. A HIGH state corresponds to ‘1’ and a LOW state corresponds to ‘0’.

Code:



```
#define IR 8 //define what pin the sensor is connected to on the Arduino Nano

void setup() {
  // put your setup code here, to run once:
  pinMode(IR, INPUT);  //define this pin as an INPUT
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  int val;  //create a variable or container that holds integers

  val = digitalRead(IR);  //save the value gotten from the sensor in our variable that holds integers

  Serial.println(val);  //print out the value saved in our variable on the Serial monitor
}
```

The complete code can be found in (github.com/McMasterSumobot/Sensors).

In this code we first start by defining the pin where the sensor is connected. After which we go on to define this pin as an input pin and set the baud rate for our serial monitor (read more on baud rates later, for now just copy the code to understand the working of the sensor).

We then go on to the loop function and declare or create a variable of integer type called “val”. What this line of code does is it tells the Arduino that we want to create a variable or container that holds integer values.

Using the “digitalRead” function, we read the input on the IR pin and save the value to the val variable we created initially.

Lastly, we tell the Arduino IDE to display the values stored in the variable on the serial monitor. The serial monitor can be accessed using this icon:

*note to open the serial monitor after uploaded the code to the Arduino and it is running. Also make sure the baud rate written in the setup function is the same on the serial monitor. Check the drop down menu on the serial monitor to make sure they are the same.