Part 2

lesson 13

IR Receiver Module

Overview

Using an IR Remote is a great way to have wireless control of your project.

Infrared remotes are simple and easy to use. In this tutorial we will be connecting the IR receiver to the UNO, and then use a Library that was designed for this particular sensor.

In our sketch we will have all the IR Hexadecimal codes that are available on this remote, we will also detect if the code was recognized and also if we are holding down a key.

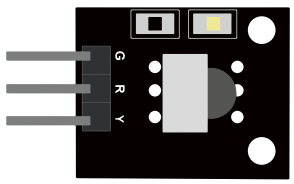
Component Required:

(1) x Elegoo Uno R3

(1) x IR receiver module

(1) x IR remote

(3) x F-M wires (Female to Male DuPont wires)



Component Introduction

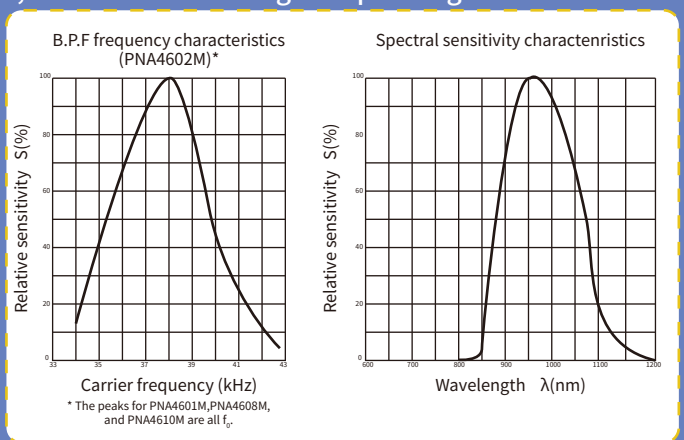
IR RECEIVER SENSOR:

IR detectors are little microchips with a photocell that are tuned to listen to infrared light. They are almost always used for remote control detection - every TV and DVD player has one of these in the front to listen for the IR signal from the clicker. Inside the remote control is a matching IR LED, which emits IR pulses to tell the TV to turn on, off or change channels. IR light is not visible to the human eye, which means it takes a little more work to test a setup.

There are a few difference between these and say a CdS Photocells:
IR detectors are specially filtered for IR light, they are not good at detecting visible light. On the other hand, photocells are good at detecting yellow/green visible light, and are not good at IR light.

IR detectors have a demodulator inside that looks for modulated IR at 38 KHz. Just shining an IR LED won't be detected, it has to be PWM blinking at 38KHz. Photocells do not have any sort of demodulator and can detect any frequency (including DC) within the response speed of the photocell (which is about 1KHz)

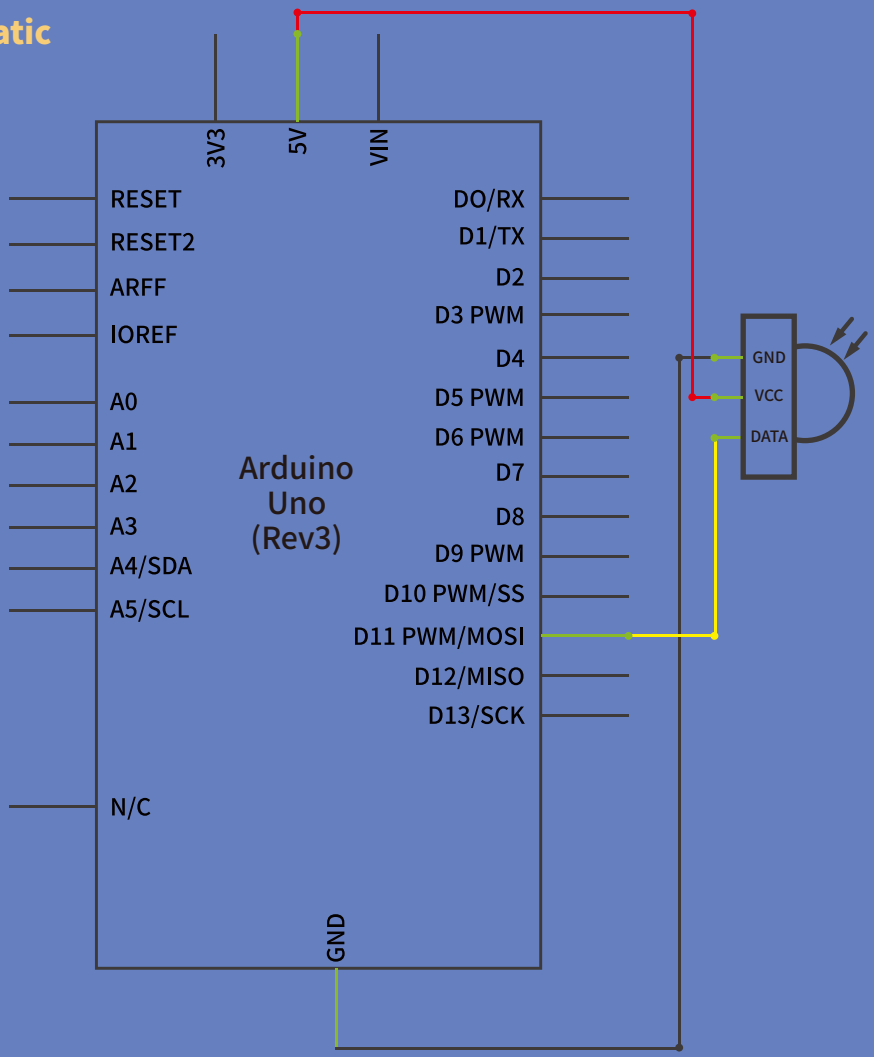
IR detectors are digital out - either they detect 38KHz IR signal and output low (0V) or they do not detect any and output high (5V). Photocells act like resistors, the resistance changes depending on how much light they are exposed to.

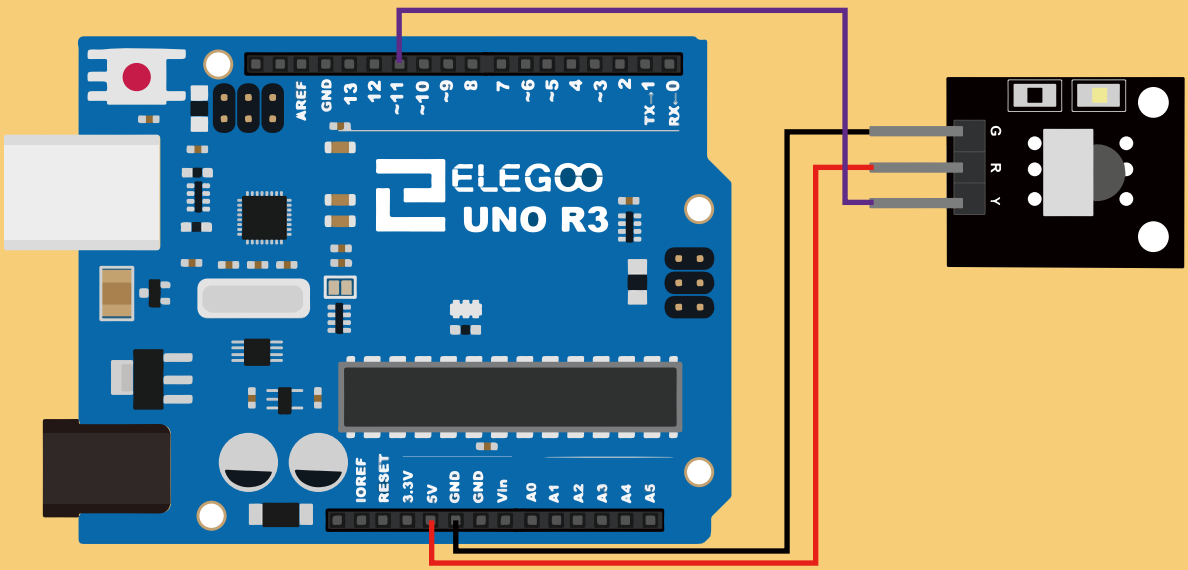


As you can see from these datasheet graphs, the peak frequency detection is at 38 KHz and the peak LED color is 940 nm. You can use from about 35 KHz to 41 KHz but the sensitivity will drop off so that it won't detect as well from afar. Likewise, you can use 850 to 1100 nm LEDs but they won't work as well as 900 to 1000nm so make sure to get matching LEDs! Check the datasheet for your IR LED to verify the wavelength.

Try to get a 940nm - remember that 940nm is not visible light!

Connection Schematic





There are 3 connections to the IR Receiver.
The connections are: Signal, Voltage and Ground.
The “G” is the Ground, “Y” is signal, and “R” is Voltage 5V.

Wiring diagram

Code

- After wiring, please open the program in the code folder- **IR_Receiver_Module** and click UPLOAD to upload the program. See Lesson 5 in part 1 for details about program uploading if there are any errors.
- Before you can run this, make sure that you have installed the < IRremote > library or re-install it, if necessary. Otherwise, your code won't work.
- For details about loading the library file, see Lesson 5 in part 1.
- Next we will move the <RobotIRremote> out of the Library folder, we do this because that library conflicts with the one we will be using. You can just drag it back inside the library folder once you have done programming your microcontroller.
- Once you have installed the Library, just go ahead and restart your IDE Software.

```
switch(results.value)

 {
case 0xFFA25D: Serial.println("POWER"); break;
case 0xFFE21D: Serial.println("FUNC/STOP"); break;
......
default:
Serial.println(" other button   ");

}// End Case
```

switch...case

[Control Structure]
Description

- Like if statements, switch case controls the flow of programs by allowing programmers to specify different code that should be executed in various conditions. In particular, a switch statement compares the value of a variable to the values specified in case statements. When a case statement is found whose value matches that of the variable, the code in that case statement is run.
- The break keyword exits the switch statement, and is typically used at the end of each case. Without a break statement, the switch statement will continue executing the following expressions ("falling-through") until a break, or the end of the switch statement is reached.

Syntax

```
switch (var) {
  case label1:
    // statements
    break;
  case label2:
    // statements
    break;
  default:
    // statements
    break;
}
```

Parameters

var: A variable whose value to compare with various cases. Allowed data types: int, char.
label1, label2: constants. Allowed data types: int, char.

Returns

Nothing