**INSTRUCTIONS**

**OS Tested:** Windows 10 64 bits

1. **Setup Python environment**

* Install Python 3.7.3

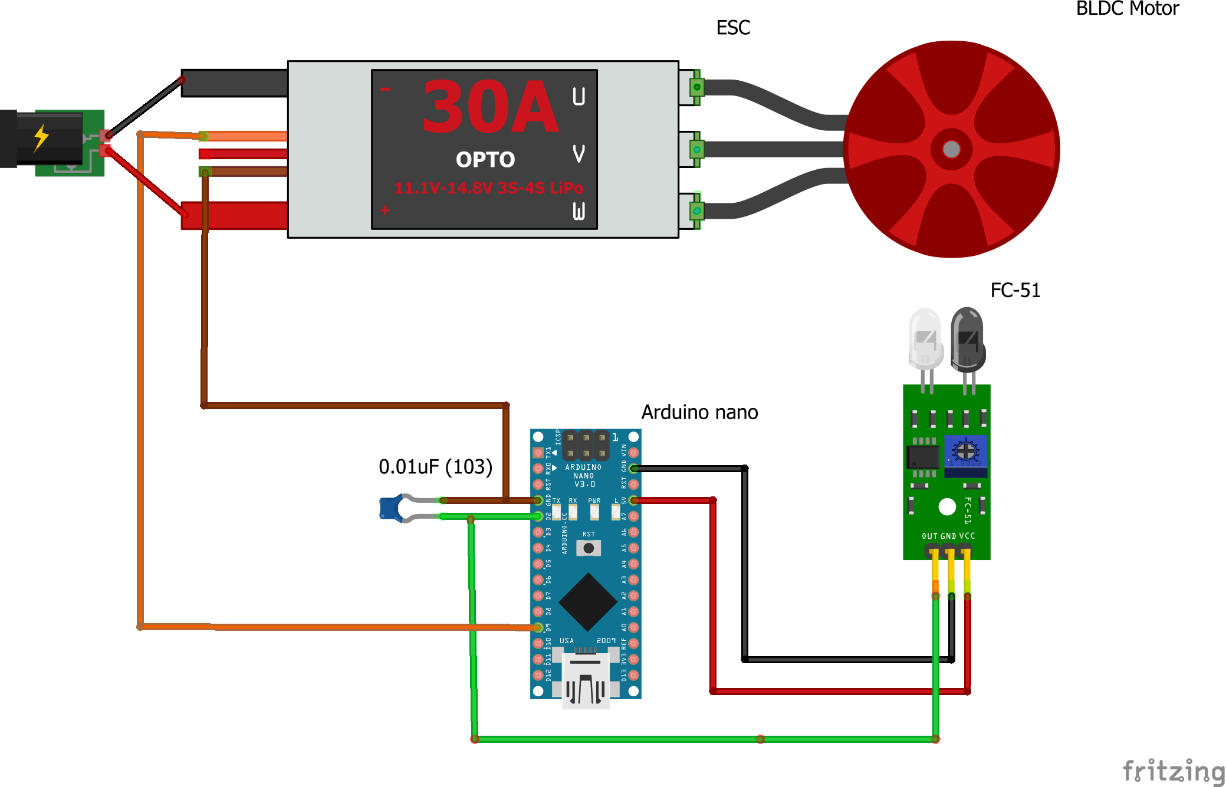
<https://www.python.org/downloads/release/python-373/>

* Add python 3.7 to Path of system
* Open a Command Prompt
* Run: cd path\BLDC\_Speed\_Control\Python\_DataLogger
* Run: pip install -r requirements.txt

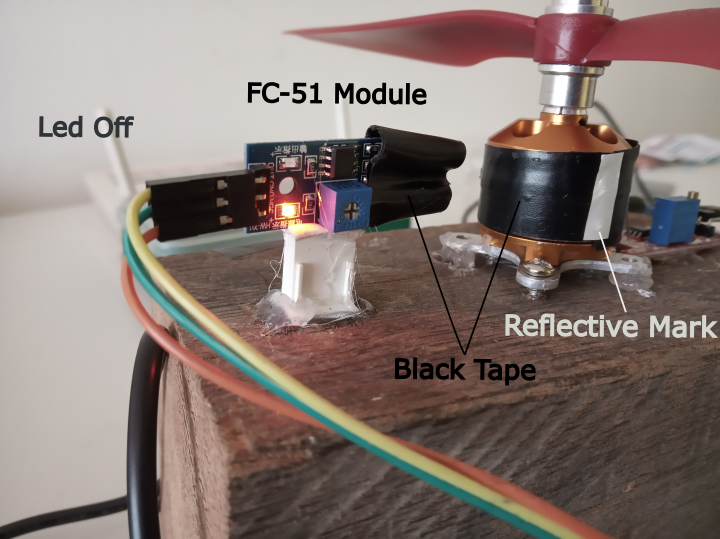
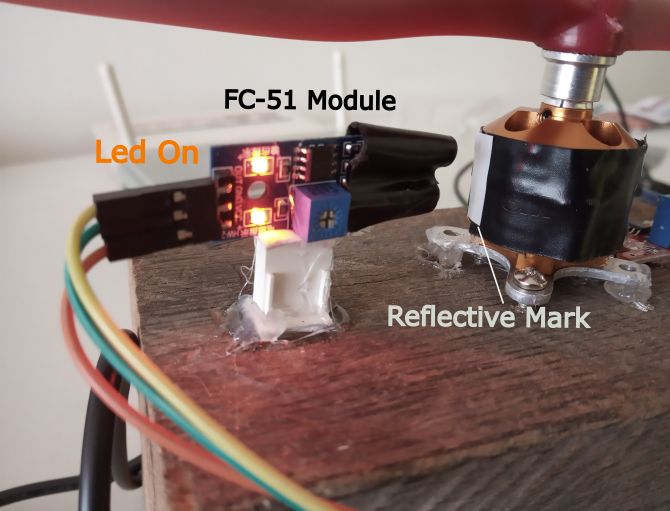
1. **Setup Arduino environment**

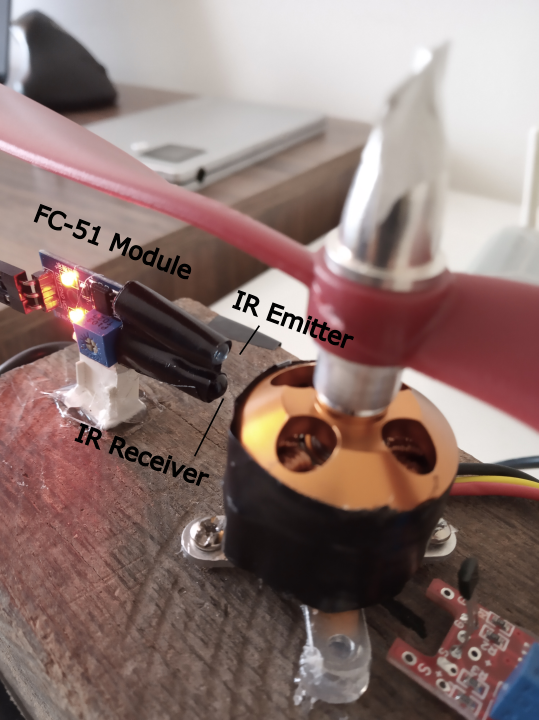
* Install Arduino IDE
* Upload Arduino script to Arduino board

1. **Using IR Sensor**
   1. **Circuit**

****

* 1. **Implementation**

** **

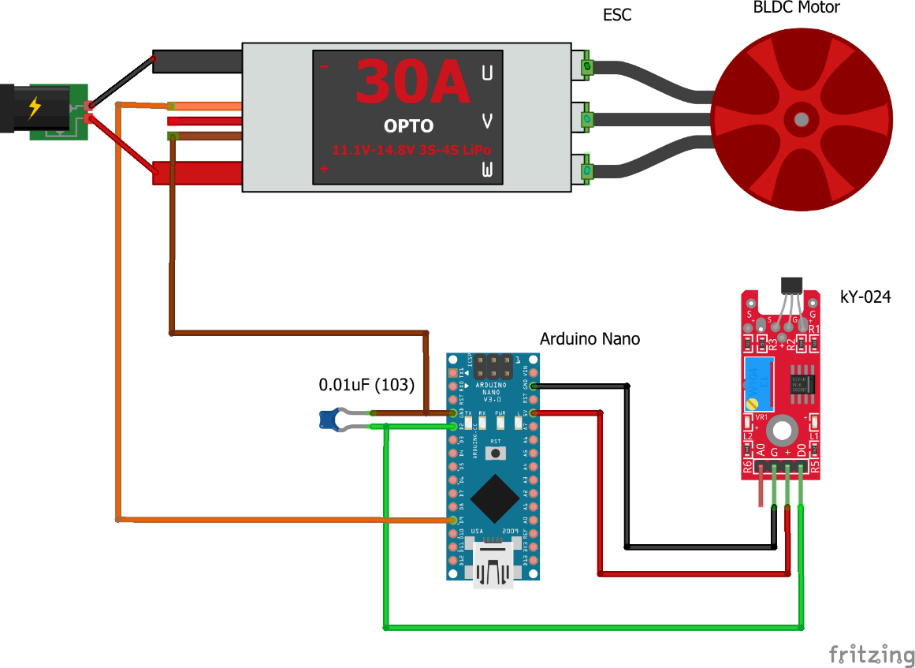
****

* 1. **Upload Arduino Sketch**
* Connect Arduino PC
* Open sketch *IR\_Sensor.ino* at path\BLDC\_Speed\_Control\IR\_Sensor
* Upload code to Arduino nano or Arduino Uno
  1. **Start motor**
* Disconnect ESC from energy source
* Connect Arduino PC
* Open a Command Prompt
* Run: cd path\BLDC\_Speed\_Control\Python\_DataLogger\src
* Run: python test.py
* Choose Arduino ID. Example 1, 2 and then enter
* Press 1 and enter
* Connect ESC to energy source and wait until it is calibrated
* Motor start to rotate to 1800 rmp
* An csv report is exported at path\BLDC\_Speed\_Control\Python\_DataLogger\csv
* For stopping motor, run: ctrl+c in Command Prompt

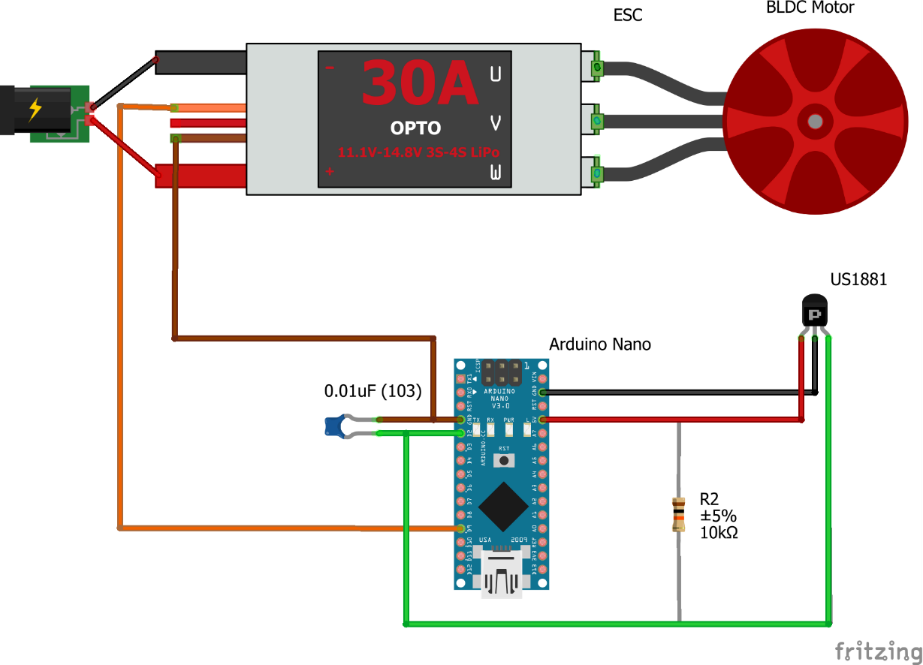
If you calibrated the motor and it does not be disconnected, you can choose normal operation.

Note: if the motor is disconnected from energy source or it shut down, it needs to be calibrated again.

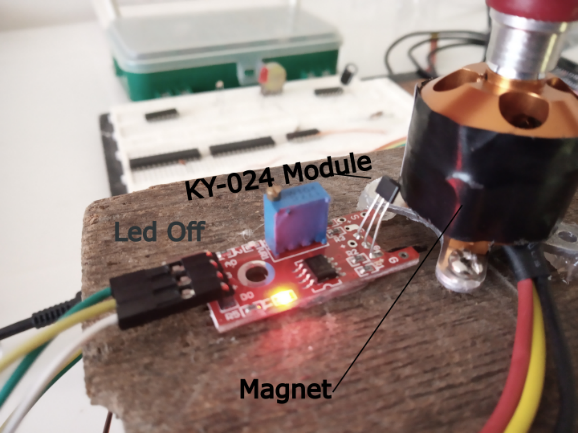
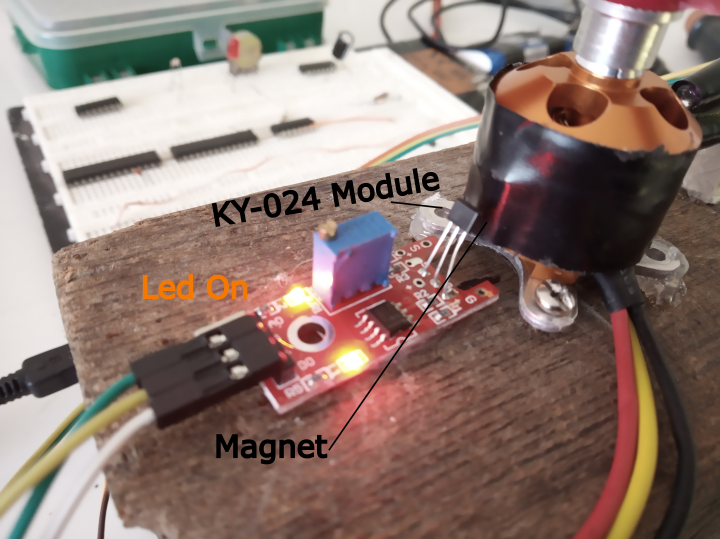
1. **Using Hall Sensor**
   1. **Circuit with KY-024 Module**

****

* 1. **Circuit with US1881 Hall sensor**

****

* 1. **Implementation**

** **

* 1. **Upload Arduino Sketch**
* Connect Arduino PC
* Open sketch *Hall\_Sensor.ino* at path\BLDC\_Speed\_Control\Hall\_Sensor
* Upload code to Arduino nano or Arduino Uno
  1. **Start motor**
* Disconnect ESC from energy source
* Connect Arduino PC
* Open a Command Prompt
* Run: cd path\BLDC\_Speed\_Control\Python\_DataLogger\src
* Run: python test.py
* Choose Arduino ID. Example 1, 2 and then enter
* Press 1 and enter
* Connect ESC to energy source and wait until it is calibrated
* Motor start to rotate to 1800 rmp
* An csv report is exported at path\BLDC\_Speed\_Control\Python\_DataLogger\csv
* For stop stopping, motor run: ctrl+c in Command Prompt

If you calibrated the motor and it does not be disconnected, you can choose normal operation.

Note: if the motor is disconnected from energy source or it shut down, it needs to be calibrated again.

1. **PID Calibration**

Both *Hall\_Sensor.ino* and *IR\_Sensor.ino* have the line:



The arguments are:

*PID (float Kp, float Ki, float Kd, int sampleT, int outMin, int outMax)*

*Kp:* proportional constant of PID

*Ki*: Integral constant of PID

*Kd:* derivative constant of PID

*sampleT:* Sampling time of PID controller

*outMin:* min output of controller (Should be kept to 1000 for this application)

*outMax:* max output of controller (The max value could be 2000, but 1200 is ok)

For this kind of system just Kp and Ki should be modified. The python script helps in tuning because it plots the speed measure for Arduino. Kp and Ki should be changed in very little rates.