

Alex Shah

Project 8 - Orientation Sensor Data

2024-04-07

EN.605.715.81.SP24

Table of Contents

Requirements.....	.1
Design.....	.1
Implementation.....	.2
Demo.....	.4
References.....	.4

Requirements

This project requires using an Adafruit BNO055 Absolute Orientation Sensor on an Arduino to report orientation data from the sensor. The implementation should use FreeRTOS to create a task that runs and communicates via I2C to capture orientation data. The Arduino should be connected to a Raspberry Pi, mounted to a flight capable drone. The data from the sensor should be read via WiFi to a remote device while the drone is in flight.

Design

I used a small breadboard to connect up the Arduino UNO and Adafruit BNO055 sensor with short wires.

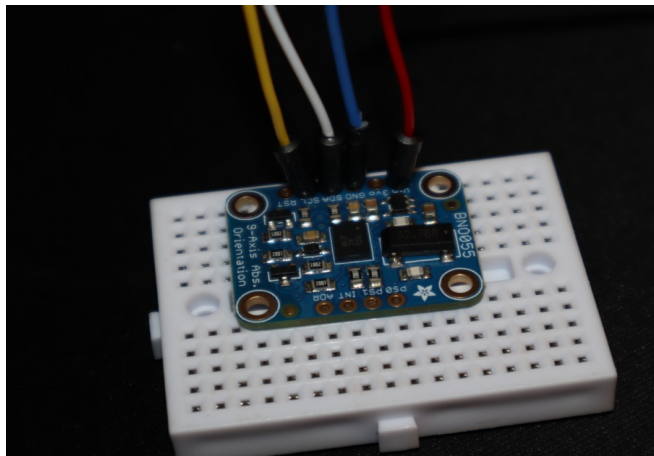


Figure 1. Adafruit BNO055

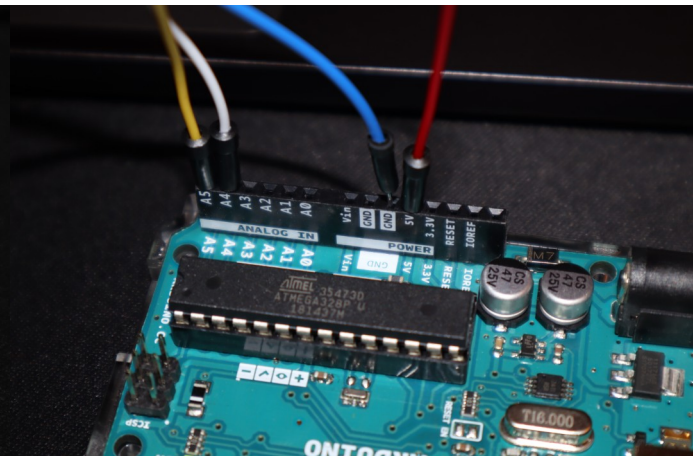


Figure 2. Arduino UNO

Arduino to Sensor:

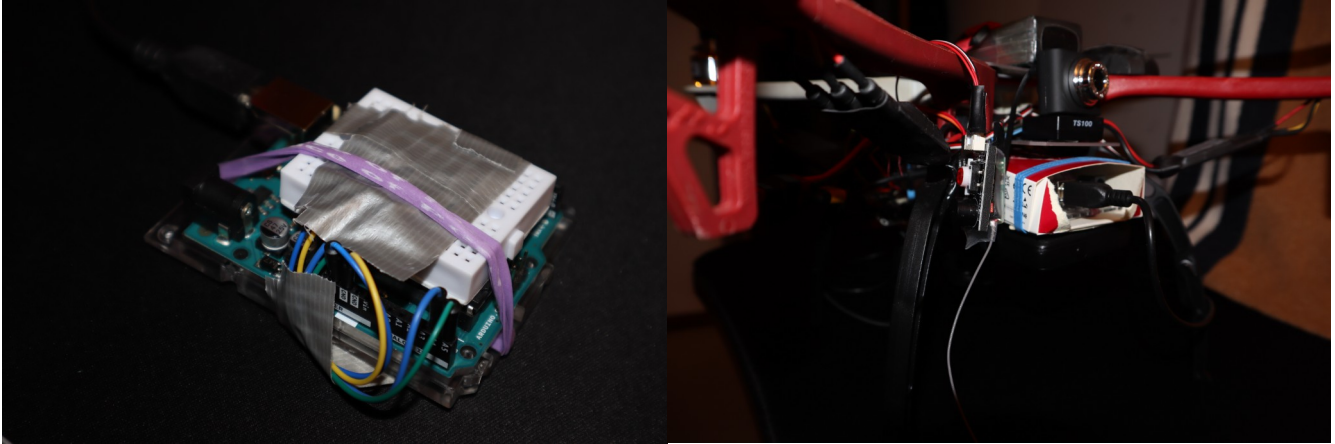
5v > Vin

GND > GND

A4 > SDA

A5 > SCL

The assembly is secured and put it in a small box and attached to the drone. The Arduino plugs into the Raspberry Pi by USB.



Implementation

In order to use the sensor, we need to install the Adafruit Sensor and BNO055 libraries in Arduino IDE. The sensor worked, so I continued on by adding the FreeRTOS library to use RTOS capabilities like interrupt routines by creating tasks.

...

```
1 #include <Arduino_FreeRTOS.h>
2 #include <Wire.h>
3 #include <Adafruit_Sensor.h>
4 #include <Adafruit_BNO055.h>
5 #include <utility/imumaths.h>
6
7 void TaskReadIMU(void *pvParameters);
8 #define BNO055_SAMPLERATE_DELAY_MS (100)
9 Adafruit_BNO055 bno = Adafruit_BNO055(55, 0x28, &Wire);
10
11 void setup(void) {
12     Serial.begin(9600);
13     while (!Serial)
14         ;
15
16     Serial.println("Orientation Sensor Test");
17     if (!bno.begin()) {
18         Serial.print("No BNO055 detected !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!");
```

```

19         while (1)
20             ;
21     }
22
23     // Create task
24     if (xTaskCreate(TaskReadIMU, "ReadIMU", 256, NULL, 3, NULL) != pdFAIL) {
25         Serial.println("Task created successfully.");
26         delay(500);
27     } else {
28         Serial.println("Failed to create task.");
29     }
30     bno.setExtCrystalUse(true);
31 }
32
33 void loop(void) {
34     // do tasks
35 }
36
37 void TaskReadIMU(void *pvParameters) {
38     (void)pvParameters;
39
40     for (;;) {
41         imu::Vector<3> euler = bno.getVector(Adafruit_BNO055::VECTOR_EULER);
42
43         Serial.print("X/ROLL: ");
44         Serial.println(euler.x());
45         Serial.print("Y/PITCH: ");
46         Serial.println(euler.y());
47         Serial.print("Z/YAW: ");
48         Serial.println(euler.z());
49
50         delay(BNO055_SAMPLERATE_DELAY_MS);
51     }
52 }
53
54

```

Remotely connect to pi e.g.

```
$ ssh user@IP
```

make note of tty address to verify the Arduino is connected

```
$ ls /dev/tty*
```

```
> /dev/ttyACM1 is present after plugging the Arduino into the Pi's USB port
```

```
$ cat /dev/ttyACM1
```

```
> Serial output from the Arduino showing orientation data (X/Y/Z or Pitch, Yaw, Roll)
```

Demo

This video contains a quick demonstration of the orientation sensor alone, then flying with the sensor mounted to the drone, reporting orientation data to my laptop over wifi.

NOTE: The XYZ Roll/Pitch/Yaw labels are flip flopped in the video but fixed in the code.

<https://drive.google.com/file/d/1ku1LYaIgcpg0NyLsr4TFWORH8FwcfTtZ/view?usp=sharing>

References

Adafruit documentation for BNO055

<https://learn.adafruit.com/adafruit-bno055-absolute-orientation-sensor>

<https://cdn-learn.adafruit.com/downloads/pdf/adafruit-bno055-absolute-orientation-sensor.pdf>

FreeRTOS Documentation

<https://www.freertos.org/features.html>