Homework 4 - Arduino Uno

Group Members: Omar Chedid, Anirudh Ganji, Athishay Kiran, Katherine Brown

The Arduino UNO is acting as an analog receiver in this system. It is connected to a 6050-MPU Accelerometer/Gyroscope sensor. It takes regular readings and reports them to the Raspberry Pi.

Requirements

Necessary Operating System, Packages, and Libraries

This project requires a working copy of the Arduino IDE installed on the computer that is uploading the sketch to the Arduino. On Fedora 27, the command for that installation is:

```
sudo dnf install arduino
```

Software Setup

The sketch is uploaded via the Arduino IDE via the following sequence:

- connect the Arduino via USB
- compile sketch
- Choose board to upload sketch onto

```
Tools -> Board -> Arduino Uno
```

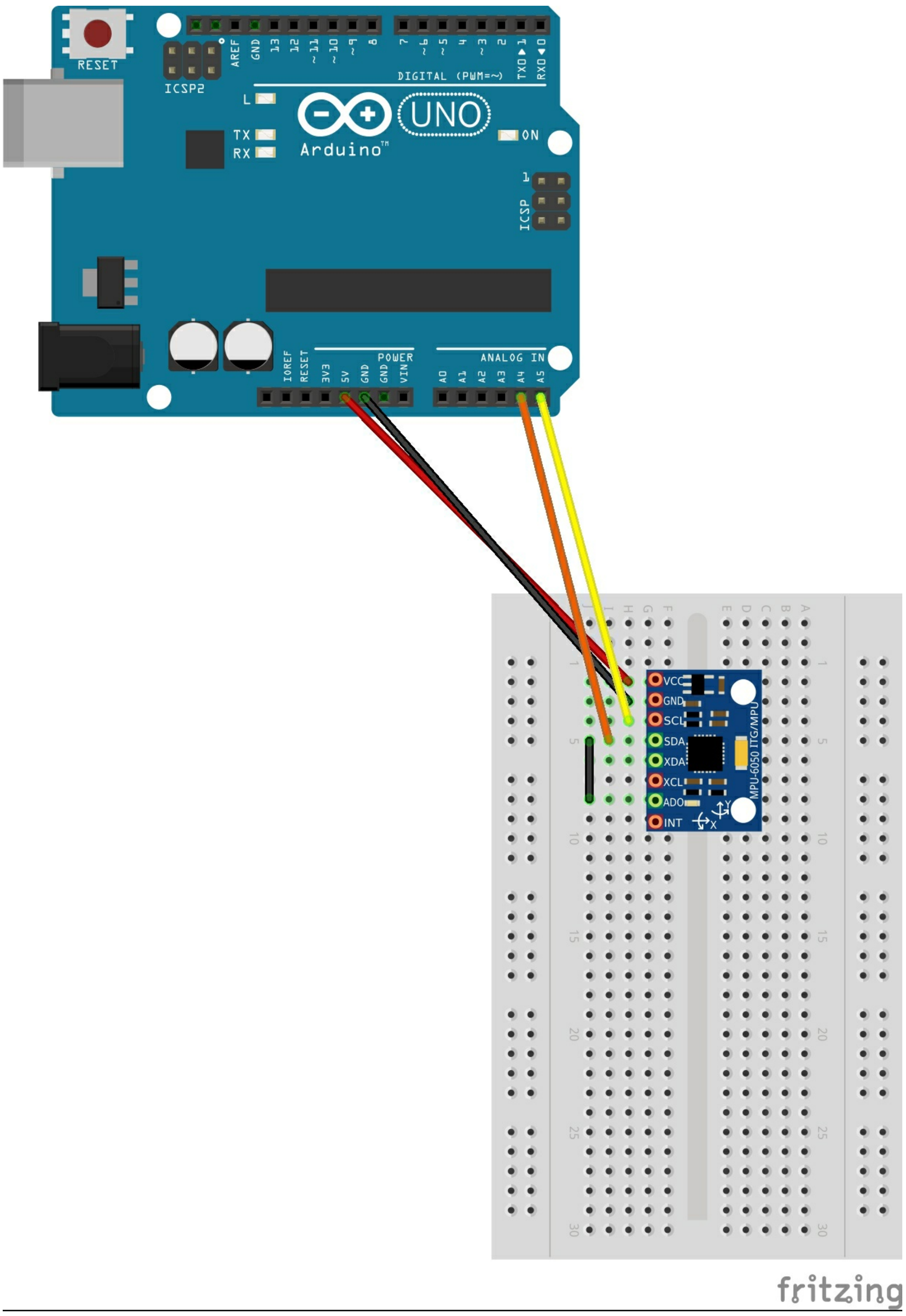
- select serial port

```
Tools -> Board -> Port
```

- Press upload button to send sketch to Arduino Board

As long as the board is powered on, the sketch will run its setup() and loop() functions.

Hardware Setup



Software Functionality

Functions

void setup()

- initializes serial communication at 2000000 bits/sec
- runs function to initialize MPU chip
  - sets register values and brings chip out of sleep mode

void loop()

- records data from Accelerometer registers
- records data from Gyroscope registers
- prints analog value read as formatted list of Strings
- delays for stability

Flow of Control

On powerup, the sketch will execute automatically. The setup() function runs, which establishes serial communication at 2000000 bits/sec and uses the Wire library to initialize the MPU registers. This sets the range of values that the Gyroscope reads as +/-250°/sec, the range that the Accelerometer reads as +/- 2g, and brings the chip out of sleep mode. The loop() function runs continuously, monitors the Accelerometer/Gyroscope, processes the values, and writes the values via the serial port over USB to the Raspberry Pi.

Role in Overall System

The Arduino acts as the analog receiver, monitoring the values recorded by the MPU-6050 sensor and reporting those values to the program running on Raspberry Pi.

References

Arduino script was derived from tutorial by EEEnthusiast hosted on YouTube at https://www.youtube.com/watch?v=M9lZ5Qy5S2s and templates available on Github https://github.com/VRomanov89/EEEnthusiast/tree/master/MPU-6050%20Implementation, as well as the MPU sensor manual and register guide found at https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf