IE6-BUL S23

Function and Characterization ofanInertial SensorCluster/ I2C bus

Authors: Soodeh Mousaviasl, Celestine Machuca.

Materials used

- Arduino nano microcontroller
- Breadboard for prototyping
- MPU-6050 inertial sensor

Connection diagram

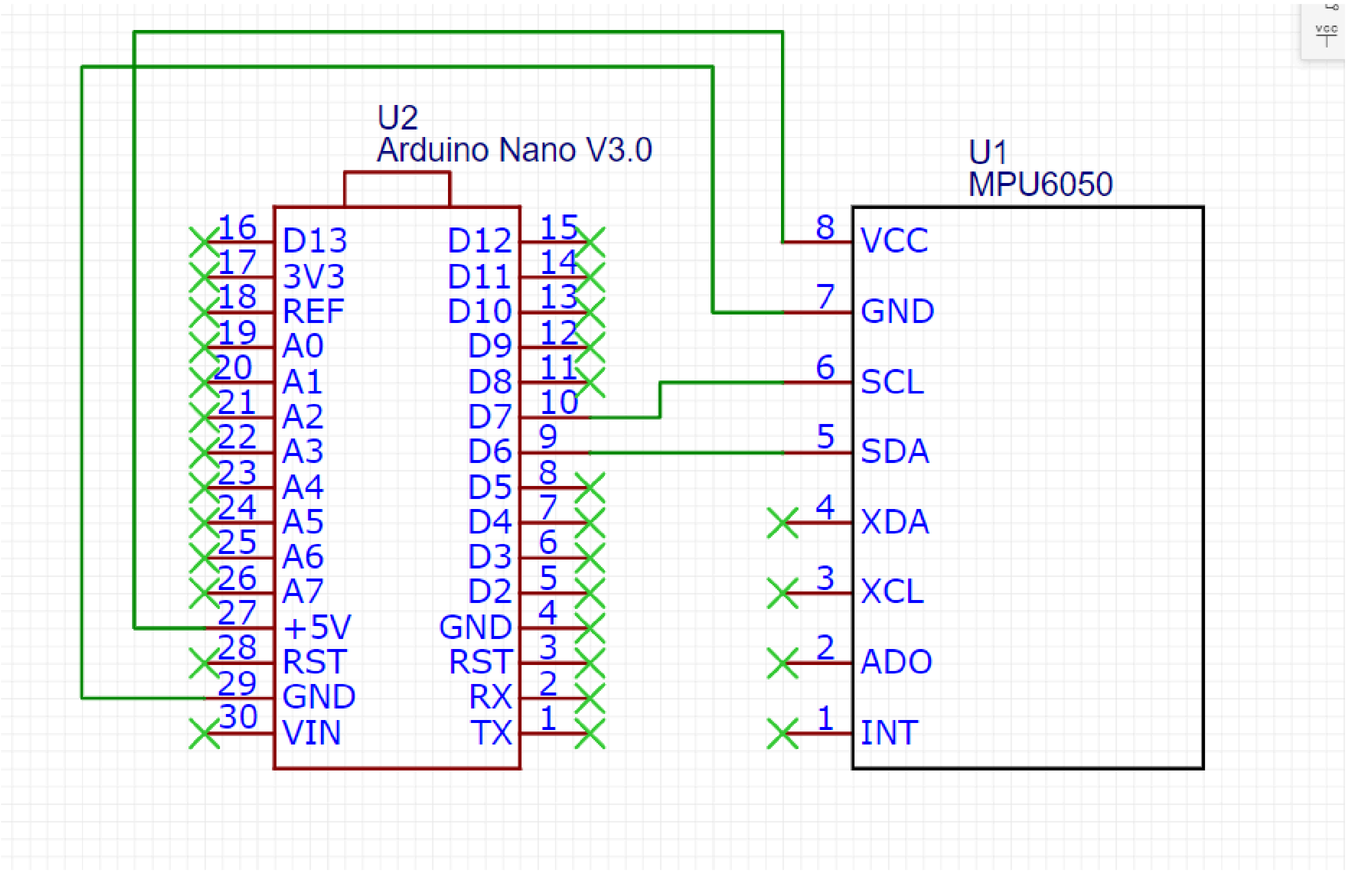

Fig 1 MPU-6050 connections

Setup Used

Setup

Fig 2 Setup

Lab Tasks

Part 1 Setup

- Which measurement data is recorded

From the code the data being recorded is:

```
// Accelerometer sensor
int acc_X = (((int)result[0]) << 8) | result[1];
int acc_Y = (((int)result[2]) << 8) | result[3];
int acc_Z = (((int)result[4]) << 8) | result[5];

// Temperatur sensor
int temp = (((int)result[6]) << 8) | result[7];

// Gyroscope sensor
int gyr_X = (((int)result[8]) << 8) | result[9];
int gyr_Y = (((int)result[10]) << 8) | result[11];
int gyr_Z = (((int)result[12]) << 8) | result[13];
```

Example output:

```
1482,3320,47558,64891,65331,63,62103
1478,3320,47552,64889,65331,62,62103
1478,3324,47550,64890,65331,63,62102
1482,3326,47550,64889,65333,63,62101
1484,3326,47546,64889,65332,62,62100
1480,3326,47550,64890,65332,63,62101
1474,3326,47540,64890,65331,63,62100
1468,3328,47548,64890,65331,62,62099
1468,3320,47556,64889,65332,62,62098
1468,3318,47562,64888,65331,63,62098
1474,3322,47562,64888,65331,62,62096
1470,3324,47562,64888,65331,62,62096
1468,3328,47558,64887,65332,62,62096
1468,3326,47556,64887,65332,61,62096
1464,3324,47552,64887,65332,62,62097
```

- how does the measurement data get to the laptop?

flow chart

Fig 3 Flow Chart

The MPU6050 sends data from its gyro, accelerometer, and temperature sensor to the Arduino via the I2C bus.
The Arduino processes the received data and formats it for transmission.
The Arduino sends the formatted data to the laptop via a USB cable using serial communication.
A serial terminal or custom software on the laptop reads and interprets the received data, allowing you to view and analyze the measurements.

Part 2 Configuration

- Find out how to set the sensor cluster to different bandwidths and measurement ranges by analyzing the data sheet?
- Try out different configurations for the measuring range of one channel of the accelerometer and measure the digital output values for a = -1 g; 0 ; +1 g.
- What is the resolution of each of these measurements?

Part 3 Oscilloscope measurements on the I2C bus

- Connect the oscilloscope to your measurement setup. Measure the voltage between SCL and GND with a probe (please adjust the square-wave signal first!)

and examine the data line SDA with a second probe.

- Compare your measurement result with the I2C data protocol from the lecture (or e.g. from https://de.wikipedia.org/wiki/I²C).
- Please answer the following questions in your lab report:What is the datarate? How many bits (raw) are transferred per second?Analyze a single I2C telegram based on your oscilloscope measurement.How does your measurement compare tothe physical layer of the ideal I2C?

Part 4 Measuring Noise on a acceleration sensor

- Analyze the noise performance of one of the three axes of the accelerometer for different bandwidths (e.g. 260 Hz vs. 5 Hz)
- To do this, keep the sensor vibration-free/still and carry out a long-term measurement, e.g. over 1000 values. Check the measurement in the time domain for outliers, filter them out if necessary,using a suitable filter(either on the Arduino or on your computer).Document the filter and include source code and explanation into your report.
- Create and compare the histograms for at least two different bandwidths and, if applicable, with or w/o filter. What are the reasons for the differences?

Part 5 Determination of the noise behavior of a channel of the angular rate sensor -->

- Analyze the noise behavior of one of the three axes of the angular rate sensor for different bandwidths (e.g. 260 Hz vs. 5 Hz).
- To do this, keep the sensor vibration-free/still and carry out a long-term measurement, e.g. over 1000 values. Check the measurement in the time domain for outliers, filter them out if necessary,using a suitable filter in the Arduinoor on your PC.Document the filter and include source code and explanation into your report.
- Create and compare the histograms for at least two different bandwidths and, if applicable, with or w/o filter. What are the reasons for the differences?
- How large is the offset of the yaw rate signal in the respective measurements?

Part 6 Visualization with “Processing”

- Switch the setupto evaluating the data with Processing –for this you need the program „ArduinoProcessingMPU6050“ on the Arduino and „ProcessingMPU6050“withinProcessing
- Analyze the program for the Arduino. How does it work? How do you find out the correction values that need to be entered into the program?
- Analyzethe program for Processing. What does this program do? How does it work?
- Try itout: Move the sensor and watch the screen. How do you know that your sensor is not yet perfectly calibrated?
- Document your results with a screendump in your lab report.