МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Пензенский государственный технологический университет»

(ПензГТУ)

Факультет информационных и образовательных технологий

Кафедра «Информационные технологии и системы»

Дисциплина «Программное и аппаратное обеспечение

информационных систем»

ОТЧЕТ

по лабораторным работам

Выполнили: студенты гр. 15ИС3м (Группа №3)

Сивишкин А.О., Кирков А.О., Соломадина Е.А., Зелинская С.С., Глухова В.О.

Проверил: ст. преподаватель каф. ИТС Володин К.И.

Работа защищена с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пенза, 2016

Содержание

Раздел 1. Интерфейсы датчиков систем на базе ARM-процессоров

***Общие сведения.***

***ARM-процессоры.***

ARM — это целая архитектура для семейства 32- и 64-битных процессоров производства компании ARM Limited. В 2007 году в 98% мобильных телефонах мира стоял как минимум один процессор ARM. Эти процессоры обладают низким энергопотреблением, поэтому широко распространены на рынке портативной техники и особенно в сфере мобильных устройств: КПК, мобильные телефоны, цифровые носители и плееры, игровые консоли и периферия. У ARM много известных лицензиатов, которые делают собственные версии ядер на основе этой архитектуры: например, Qualcomm Shapdragon или Nvidia Tegra.

***Arduino.***

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе  Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

***Микросхема MPU6050*** содержит на борту как акселерометр, так и гироскоп, а помимо этого еще и температурный сенсор. MPU6050 является главным элементом модуля GY-531.

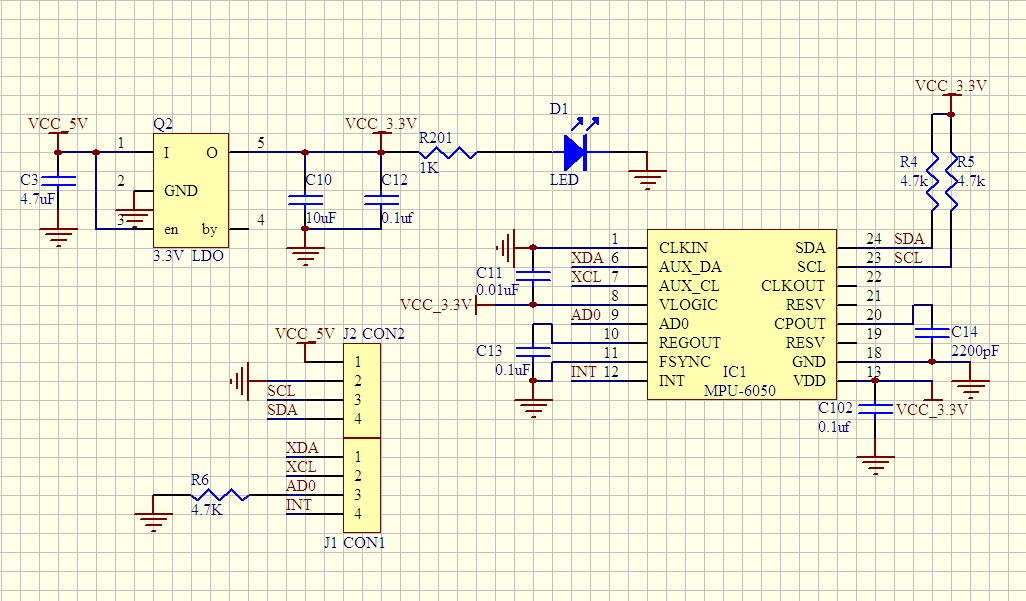


Рис. 1 – Схема датчика MPU6050

***Цель***

Научится снимать данные с датчика MPU6050 и выводить их через COM-порт в Терминал.

***Задача***

1. Снять данные с Акселерометра.

2. Обработать через библиотеку “Wire.h”

3. Вывод полученных данных на интерфейс Терминала.

Микросхема mpu6050 подключается к микроконтроллеру ArduinoMega 2560 с помощью макетной платы и соединительных проводов, как показано на рисунке 1. Подключение идет по 4-ем пинам:

1. VСС – питание микросхемы током 3.3 В;
2. GND – земля;
3. SCL – последовательная линия данных;
4. SDA – последовательная линия тактирования;

Питание на и передача данных ArduinoMega 2560 происходят через USBпорт и виртуальный COMпорт.

***Метод решения***

Для получения данных с Акселерометра была использована библиотека “Wire.h”. Используется мастером для запроса байта от ведомого устройства (слейва).  Байты могут быть получены с помощью методов available() и read().

***Описание лабораторной установки***

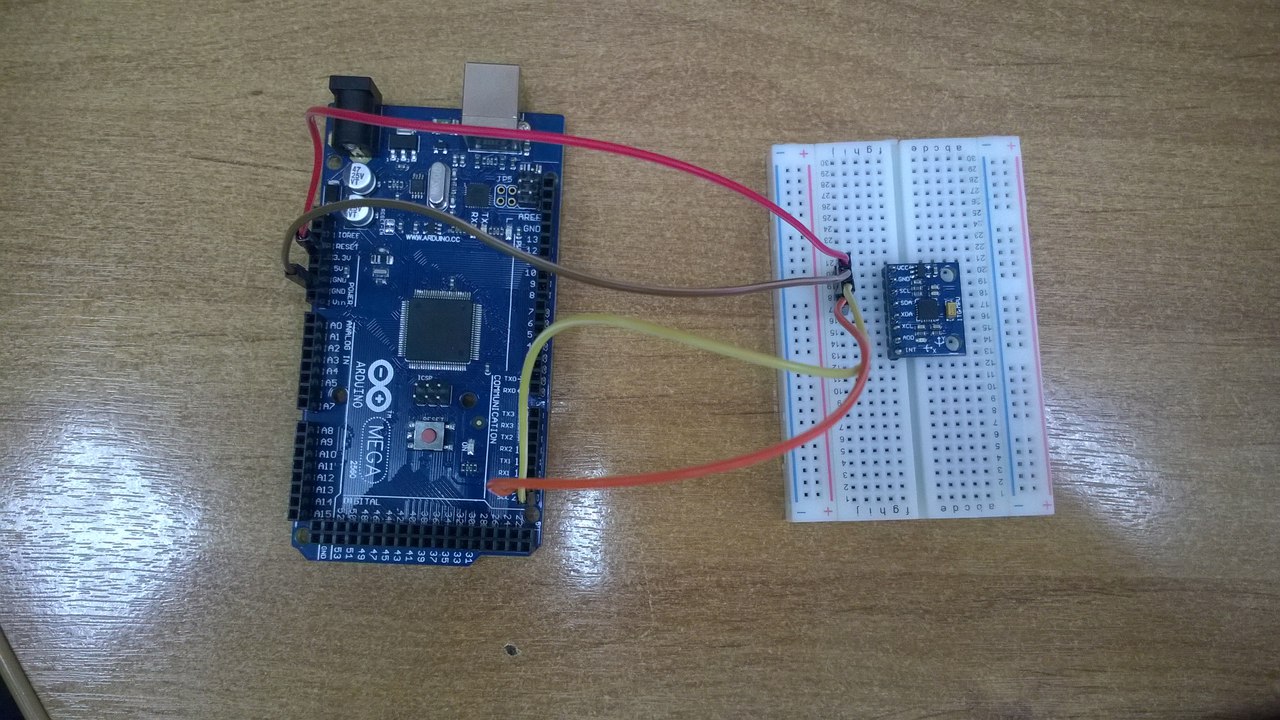


Рисунок 2 – Лабораторная установка

***Реализация***

В процессе работы было реализован программный код для считывания, обработки и вывода данных с датчика (см. Листинг 1). Микроконтроллер снимает показания 3х осей акселерометра, 3х осей гироскопа и температуру с микросхемы mpu6050.

***Листинг 1***

// MPU-6050 Short Example Sketch  
// By Arduino User JohnChi  
// August 17, 2014  
// Public Domain  
[#include](http://new.vk.com/feed?section=search&q=%23include)<Wire.h>  
const int MPU\_addr=0x68; // I2C address of the MPU-6050  
int16\_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;  
void setup(){  
Wire.begin();  
Wire.beginTransmission(MPU\_addr);  
Wire.write(0x6B); // PWR\_MGMT\_1 register  
Wire.write(0); // set to zero (wakes up the MPU-6050)  
Wire.endTransmission(true);  
Serial.begin(9600);  
}  
void loop(){  
Wire.beginTransmission(MPU\_addr);  
Wire.write(0x3B); // starting with register 0x3B (ACCEL\_XOUT\_H)  
Wire.endTransmission(false);  
Wire.requestFrom(MPU\_addr,14,true); // request a total of 14 registers  
AcX=Wire.read()«8|Wire.read(); // 0x3B (ACCEL\_XOUT\_H) & 0x3C (ACCEL\_XOUT\_L)   
AcY=Wire.read()«8|Wire.read(); // 0x3D (ACCEL\_YOUT\_H) & 0x3E (ACCEL\_YOUT\_L)  
AcZ=Wire.read()«8|Wire.read(); // 0x3F (ACCEL\_ZOUT\_H) & 0x40 (ACCEL\_ZOUT\_L)  
Tmp=Wire.read()«8|Wire.read(); // 0x41 (TEMP\_OUT\_H) & 0x42 (TEMP\_OUT\_L)  
GyX=Wire.read()«8|Wire.read(); // 0x43 (GYRO\_XOUT\_H) & 0x44 (GYRO\_XOUT\_L)  
GyY=Wire.read()«8|Wire.read(); // 0x45 (GYRO\_YOUT\_H) & 0x46 (GYRO\_YOUT\_L)  
GyZ=Wire.read()«8|Wire.read(); // 0x47 (GYRO\_ZOUT\_H) & 0x48 (GYRO\_ZOUT\_L)  
Serial.print("AcX = "); Serial.print(AcX);  
Serial.print(" | AcY = "); Serial.print(AcY);  
Serial.print(" | AcZ = "); Serial.print(AcZ);  
Serial.print(" | Tmp = "); Serial.print(Tmp/340.00+36.53); //equation for temperature in degrees C from datasheet  
Serial.print(" | GyX = "); Serial.print(GyX);  
Serial.print(" | GyY = "); Serial.print(GyY);  
Serial.print(" | GyZ = "); Serial.println(GyZ);  
delay(444);  
}

***Тестирование***

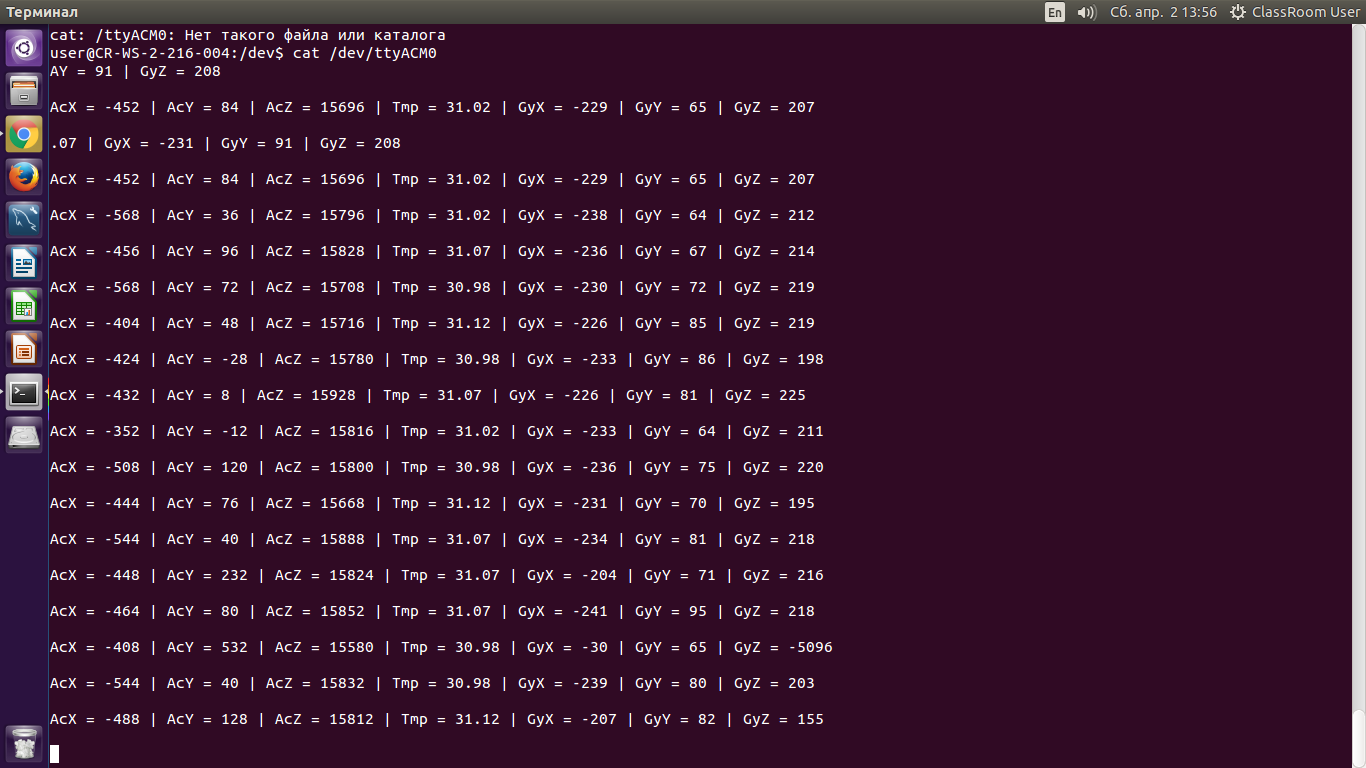


Рисунок 3 – Вывод результатов в Терминале

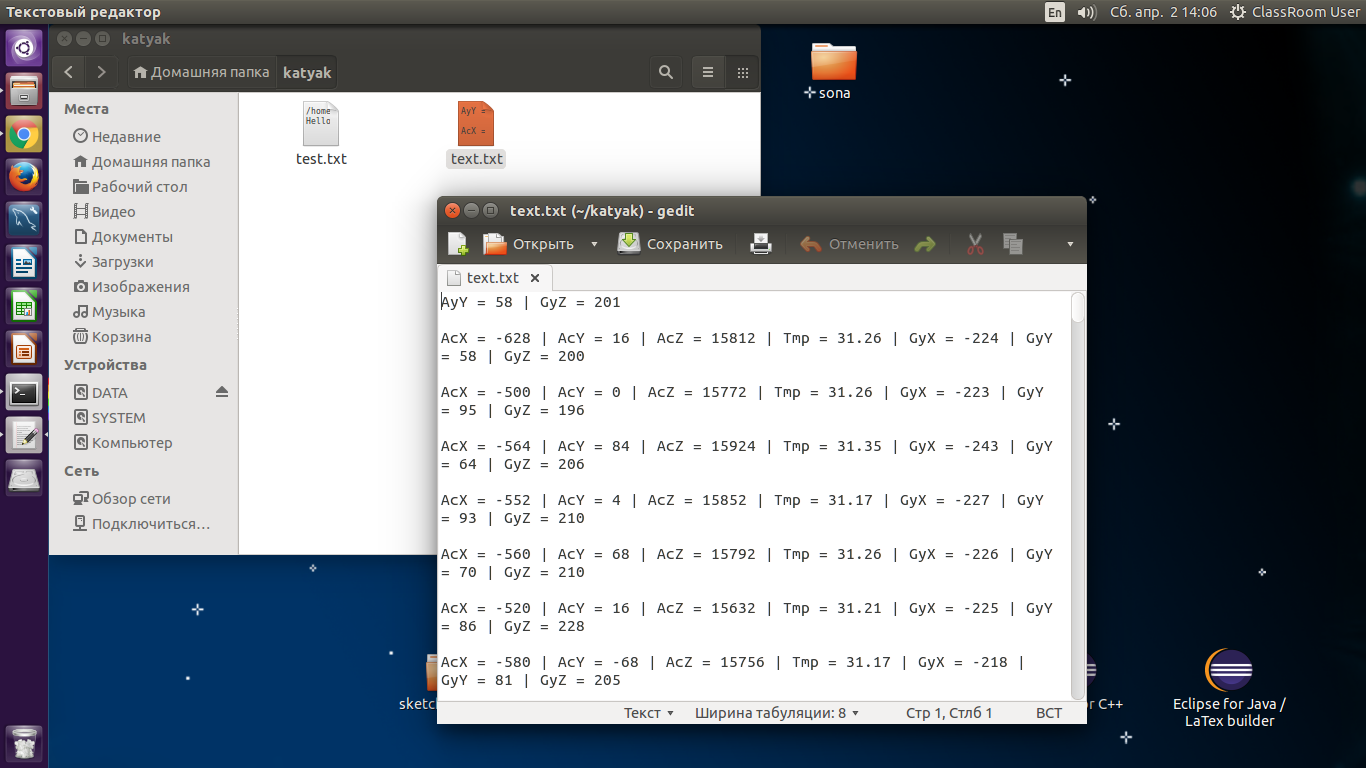


Рисунок 4 – Тестирование работоспособности лабораторной установки

***Анализ результатов***

Данные снимаются и выводятся в удобочитаемом состоянии в реальном времени. Были получены показания с акселерометра, гироскопа, термометра.

AX - Положения акселерометра относительно оси x.

AY- Положения акселерометра относительно оси y.

AZ - Положения акселерометра относительно оси z.

Tmp - Абсолютное значение температуры в градусах цельсия.

GX - Положения гироскопа относительно оси x.

GY - Положения гироскопа относительно оси y.

GZ - Положения гироскопа относительно оси z.

Полученные данные могут быть использованы для обработки.

***Выводы по разделу***

Цели достигнуты. Были приобретены навыки программирования микроконтроллера на базе Arduino в среде ArduinoIDE. Задачи выполнены. Все счастливы.

Раздел 2. Основы работы в Linux. Работа с портами и устройствами систем на базе ARM-процессоров

Общие сведения

***Терминал Linux***

Большинство задач в Linux можно выполнить в командной строке. И хотя для большинства программ существуют графические утилиты, иногда их просто недостаточно. Здесь и используется командная строка.

Терминал часто называют командной строкой или оболочкой. В далекие дни пользователи общались с компьютером именно так; и тем не менее пользователи Linux находят, что использование терминала может быть быстрее графического метода. Сейчас вы узнаете как пользоваться терминалом.

Изначально терминал использовали в качестве браузера файлов и он действительно все еще используется в этой роли. Вы можете использовать терминал в качестве браузера файлов, добраться до ваших файлов и отменить сделанные изменения.

***Цель***

Научиться писать через Терминал код на Си++.

Выводить данные в файл.

***Задача***

1. Написать программу на Си++ для считывания данных с порта.

2. Через Терминал создать документ для этих данных.

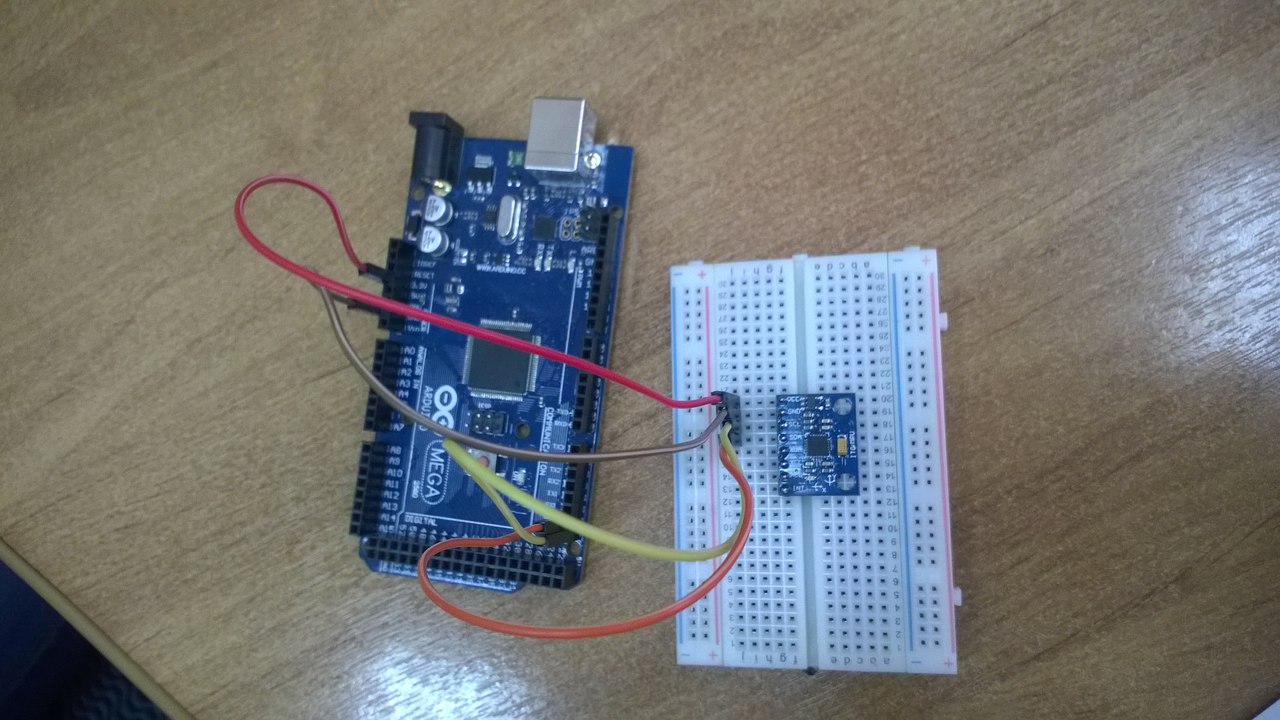
3. Записать данные в этот документ.

***Метод решения***

Использование основ Си++. Использование командной строки Linux.

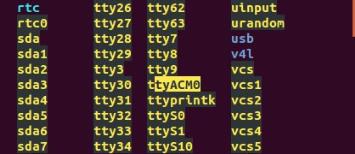
***Описание лабораторной установки***

Лабораторная установка состоит из микроконтроллера ArduinoMega 2560, макетной платы, микросхемы MPU6050, соединительных проводов (Рисунок 2) и компьютера с операционной системой Linux.



***Реализация***

В процессе работы с помощью языка Си++, которая выполняет считывание данных с порта и вывод их в терминал и в файл. При подключении микроконтроллера ArduinoMega 2560 к компьютеру с операционной системой Linux в каталоге /dev, можно увидеть имя подключенного устройства, в нашем случае это «ttyACM0».



Далее на языке Си был написан код программы для снятия данных с микроконтроллера и записи их в файл.

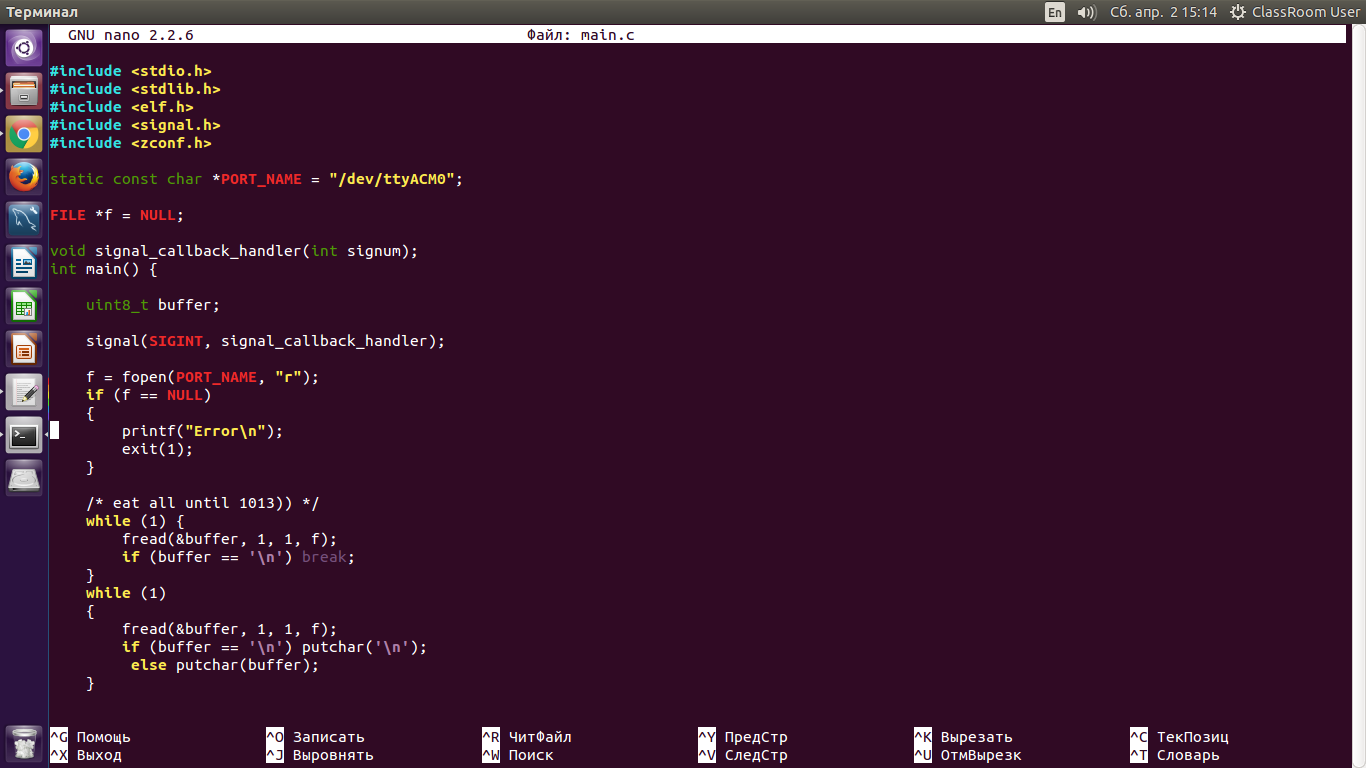


Рисунок 6 – Код на Си++ из Терминала

***Тестирование***

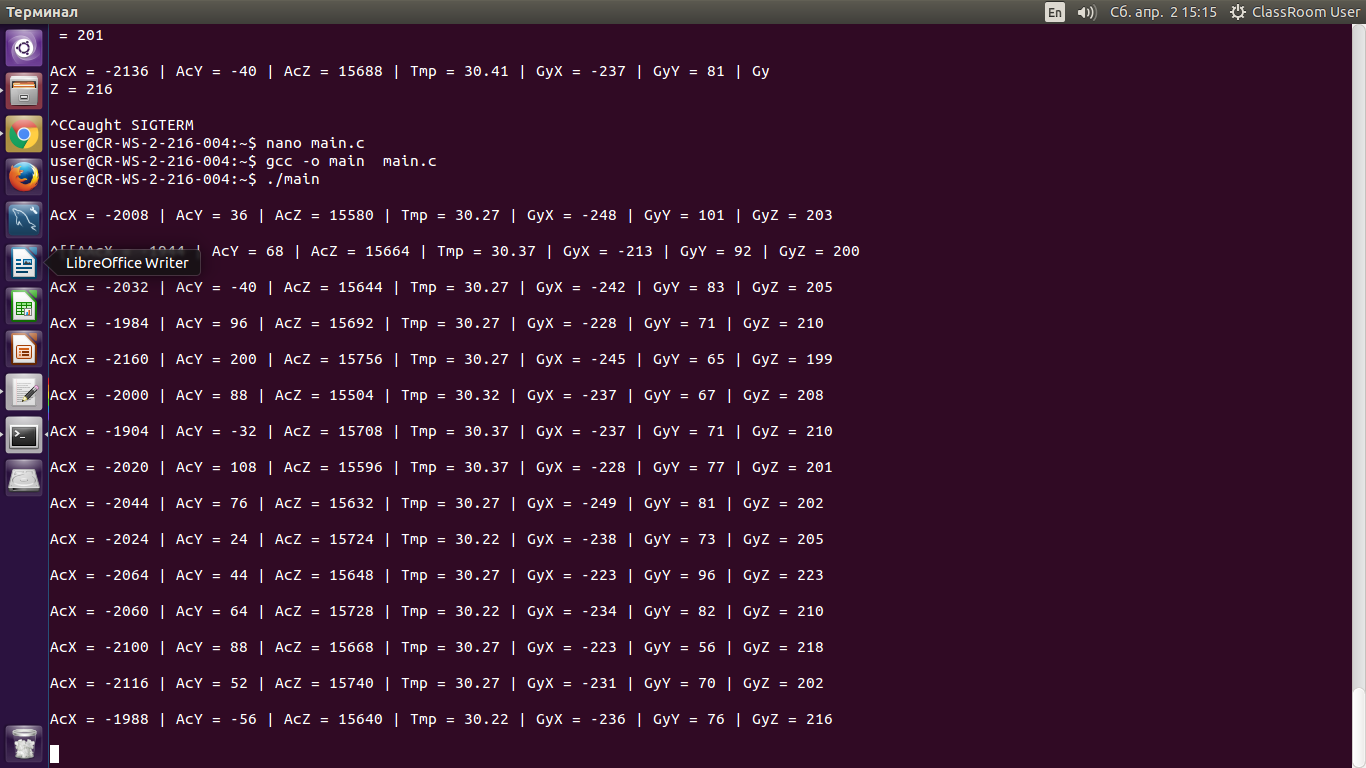
Открытие файла с записанными данными через Терминал. 

Рисунок 7 – Работа скомпилированного файла

***Анализ результатов***

Данные снимаются и выводятся в удобочитаемом состоянии в реальном времени. Разработанная программа позволяет считывать данные, записывать в файл.

***Выводы по разделу***

Цели достигнуты. Задачи выполнены. Были приобретены практические навыки работы в терминале Linux. Все счастливы.

Раздел 3. Обработка и визуализация данных в системах на базе ARM-процессоров

***Общие сведения***

**Python** — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра **Python** минималистичен.

***Цель***

Написать на Питоне программу, которая позволит снимать данные с порта, «парсить» их и визуализировать их в диаграмму.

***Задача***

1. Написать код-программу на Питоне
2. Визуализировать данные с датчика MPU6050.

***Метод решения***

Используется интегрированная среда разработки PyCharm, библиотеки для анализа математических данных, язык Python.

***Описание лабораторной установки***

Лабораторная установка состоит из микроконтроллера ArduinoMega 2560, макетной платы, микросхемы MPU6050,соединительных проводов, компьютера с операционной системой Linux и средой разработки JetBrains Pycharm.

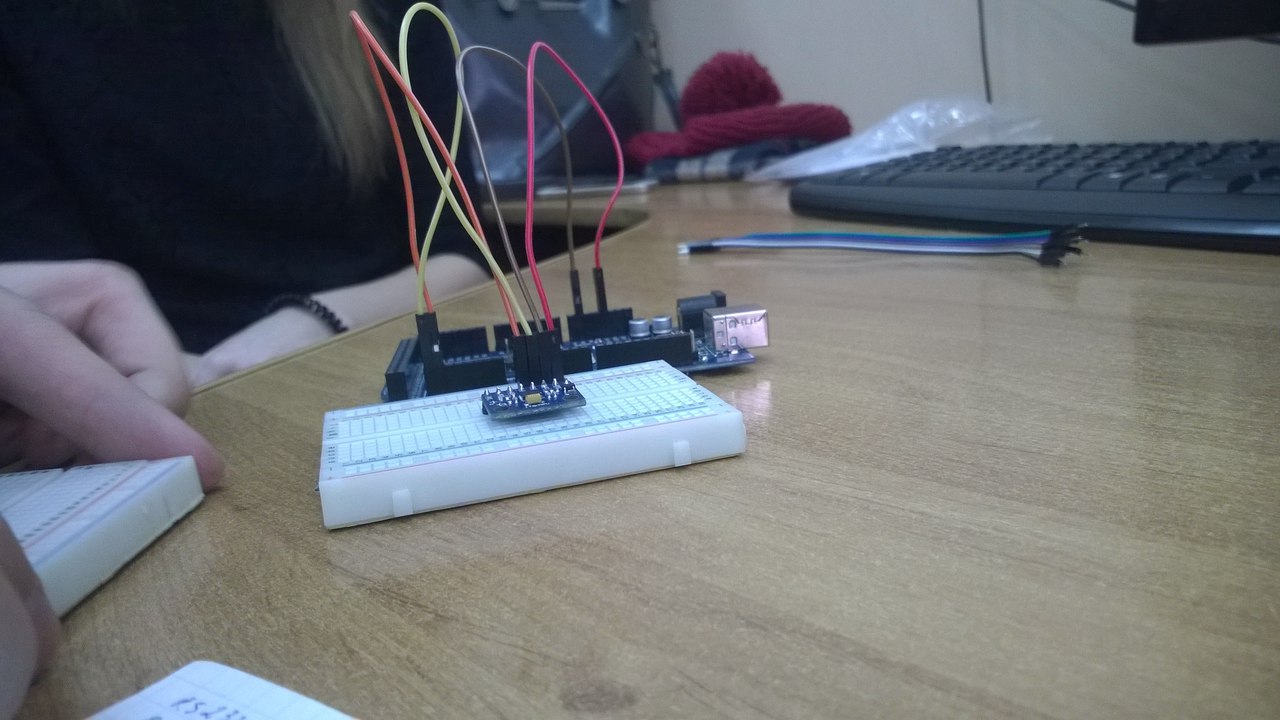


Рисунок 8 – Лабораторная установка.

***Реализация***

В процессе написания кода были выведены данные, которые парсились.

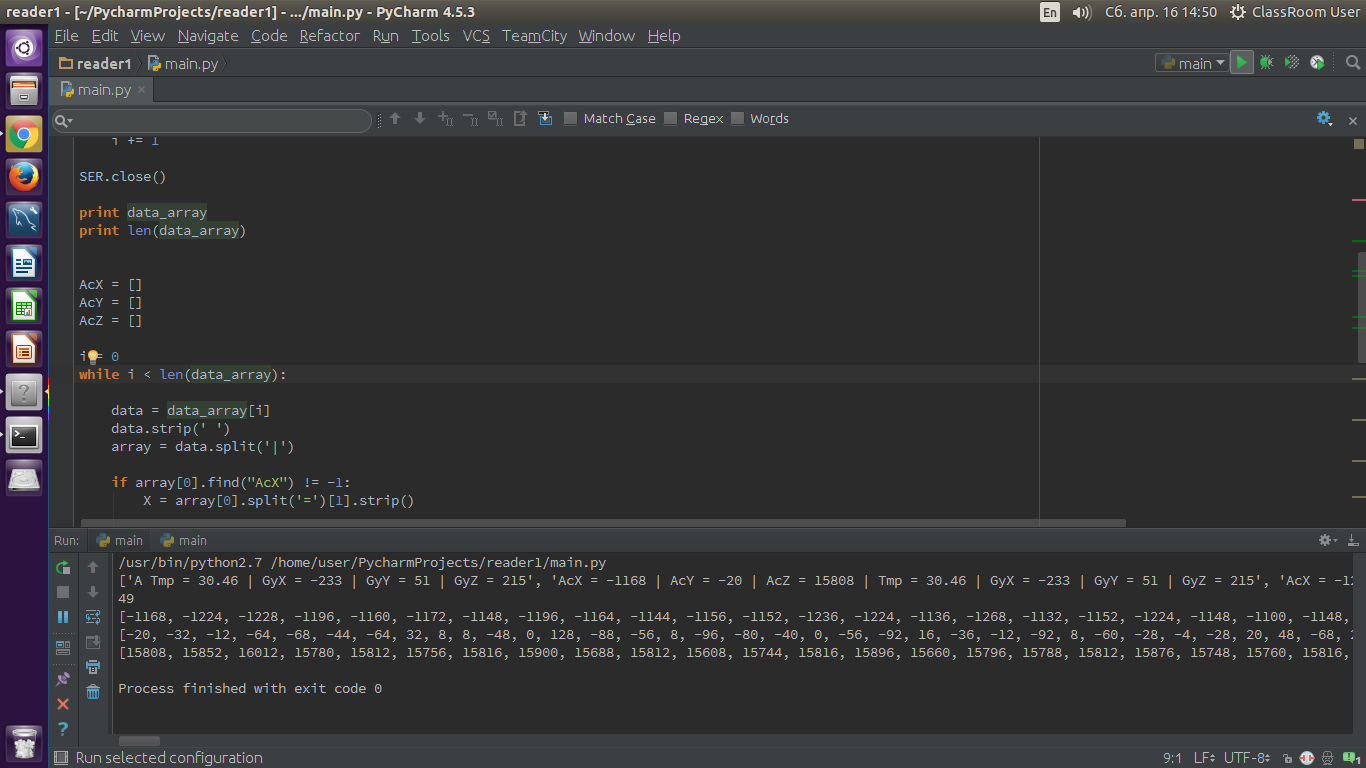


Рисунок 9 – Реализация программы

***Тестирование***

Построены диаграммы по данным, которые были предварительно распарсены. Следующим этапом исходный код был проверен средой JetBrains Pycharm на наличие ошибок и запущен. За счет библиотеки MATPLOTLIB данные с датчика MPU6050 были визуализированы.

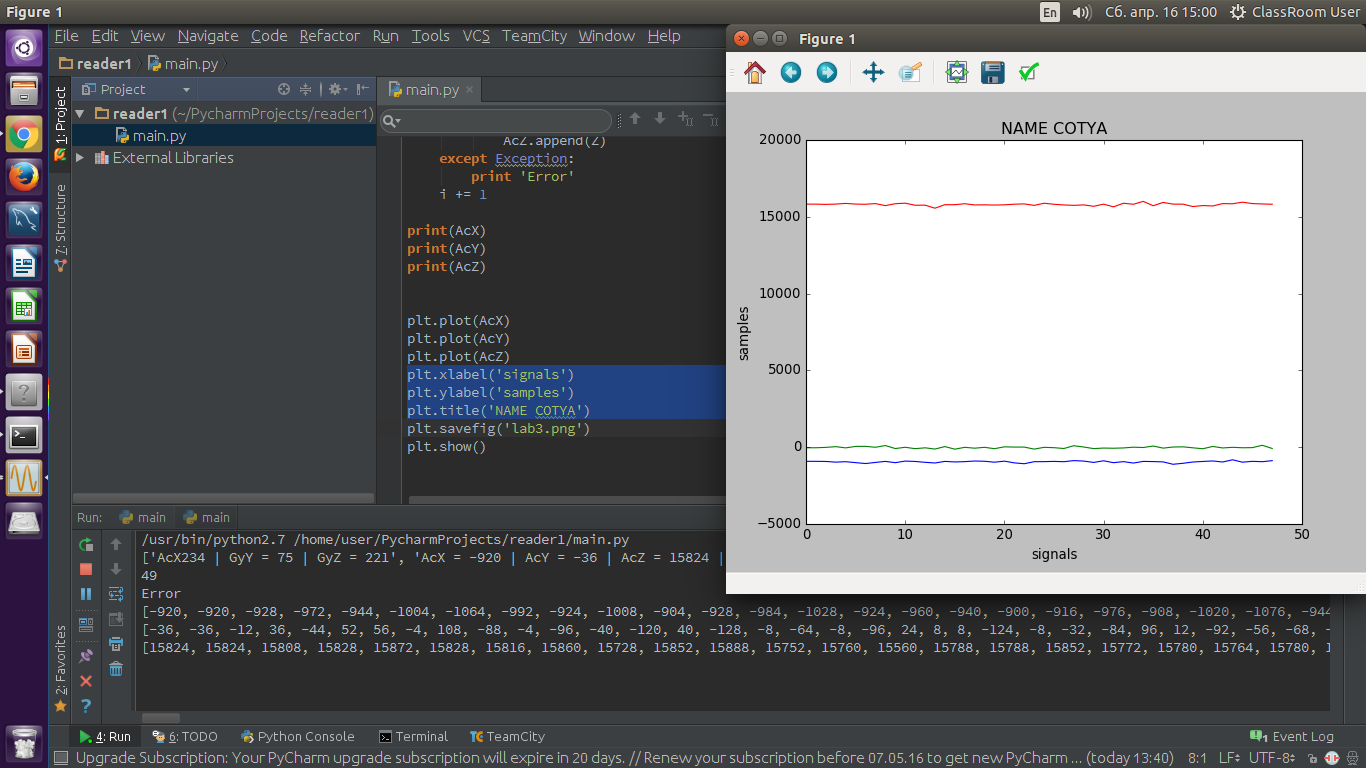


Рисунок 10 - Тестирование

***Анализ результатов***

Разработанная программа позволяет считывать данные, парсить их, строить по ним диаграмму.

***Выводы по разделу***

Цели достигнуты. Задачи выполнены. Были приобретены практические навыки обработки и визуализации данных. Приобретены практические навыки программирования на языке Python при помощи библиотеки matplotlib. Все счастливы.

***Список литературы***

1. <https://github.com/volodink/ishs-course>

2. <https://help.ubuntu.com/kubuntu/desktopguide/ru/terminals.html>

3. <http://hi-news.ru/tag/arm>

4. http://parsing.valemak.com/

5. https://www.linux.org.ru/