Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 15.03.06 Мехатроника и робототехника

**О Т Ч Е Т**

**по учебной исследовательской работе**

Выполнил студент гр. МИР-21-2б

Плешков Михаил Вячеславович

(фамилия, имя, отчество)

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

(оценка) (подпись)

(дата)

**Пермь 2022**

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 15.03.06 Мехатроника и робототехника

**О Т Ч Е Т**

**по учебной исследовательской работе**

Выполнил студент гр. МИР-21-2б

Поздняков Егор Сергеевич

(фамилия, имя, отчество)

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

(оценка) (подпись)

(дата)

**Пермь 2022**

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 15.03.06 Мехатроника и робототехника

**О Т Ч Е Т**

**по учебной исследовательской работе**

Выполнил студент гр. МИР-21-2б

Некрасов Сергей Андреевич

(фамилия, имя, отчество)

(подпись)

Проверил:

ст. преподаватель Д.А. Карлов

(должность, Ф.И.О. руководителя по практической подготовке от кафедры)

(оценка) (подпись)

(дата)

**Пермь 2022**

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc122895510)

[1. Подключение датчика MPU6050 4](#_Toc122895511)

[2. Получение и обработка данных с датчиков. 6](#_Toc122895512)

[3. Калибровка датчиков, расчет ошибки 9](#_Toc122895513)

[ЗАКЛЮЧЕНИЕ 10](#_Toc122895514)

# **ВВЕДЕНИЕ**

Мы решили сделать электрический стабилизатор для камеры (стедикам), на основе датчика MPU6050, который имеет 3-осевой акселерометр и 3-осевой гироскоп.

Задачи проекта:

1. Изучить принцип работы датчика MPU6050.
2. Получить данные с акселерометра и гироскопа, обработать их для дальнейшего использования в проекте.
3. С помощью данных с датчика рассчитать углы наклона и угловое ускорение по трем осям.
4. Отфильтровать полученные значения для корректной работы устройства.

# **1. Подключение датчика MPU6050**

Датчик будет связываться с Ардуино через I2C интерфейс.   
I2C представляет собой протокол синхронной связи – это значит что оба устройства, которые обмениваются информацией с помощью данного протокола должны использовать общий сигнал синхронизации. Поскольку в этом протоколе используются всего 2 линии (провода), то по одной из них должен передаваться сигнал синхронизации, а по другой – полезная информация.   
Serial Clock (SCL): по ней передается общий сигнал синхронизации, генерируемый ведущим устройством (master);

Serial Data (SDA): по ней осуществляется передача данных между ведущим и ведомым.

Схема подключения датчика к плате Ардуино показана на рисунке 1.

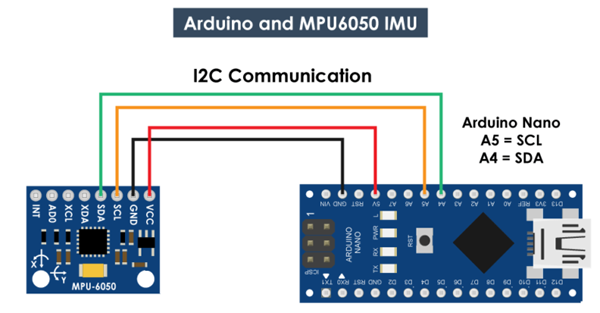


Рисунок 1 – подключение MPU6050 к Arduino

Для управления датчика через порт I2C необходимо подключить библиотеку <Wire.h>. В разделе setup() нужно инициализировать библиотеку подключиться к датчику и сбросить его через регистр управления. Для этого необходимо посмотреть назначение регистров датчика в datasheet. Регистр 0х68 используется для сброса аналоговых и цифровых сигналов датчиков, регистр 0х6В отвечает за настройку режима сна и настройку тактирования устройства. Код для подключения и сброса датчика показан на рисунке 2.

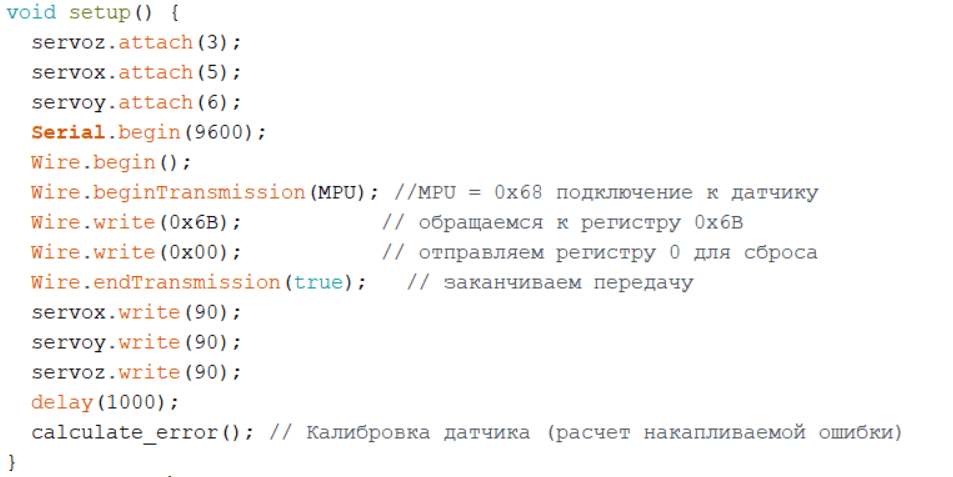


Рисунок 2 – подключение и сброс датчика

Также после сброса датчика, его необходимо откалибровать для корректной работы. Для этого напишем специальную функцию расчета накапливаемой ошибки. Работа данной функции будет показана далее.

# **2. Получение и обработка данных с датчиков.**

В разделе loop() начинаем считывать данные с датчика. Данные для каждой оси хранятся в двух байтах или в двух регистрах, адреса регистров для получения данных с акселерометра показаны на рисунке 3.

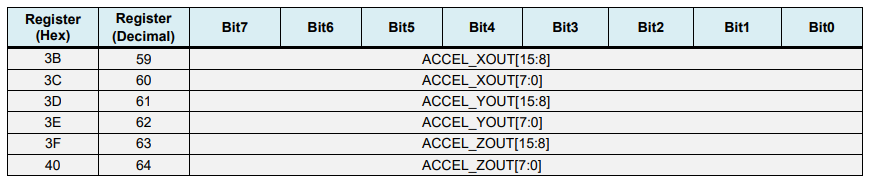


Рисунок 3 – адреса регистров акселерометра

Что бы получить входные значения от -1 до +1, подходящие для расчета углов, мы делим выходной сигнал с предварительно выбранной чувствительностью, показано на рисунке 4.

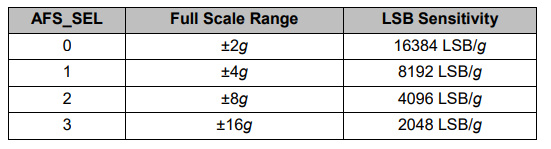


Рисунок 4 – диапазоны чувствительности датчика

После получения и обработки данных вычислим углы крена и тангажа используя две формулы, показано на рисунке 5.

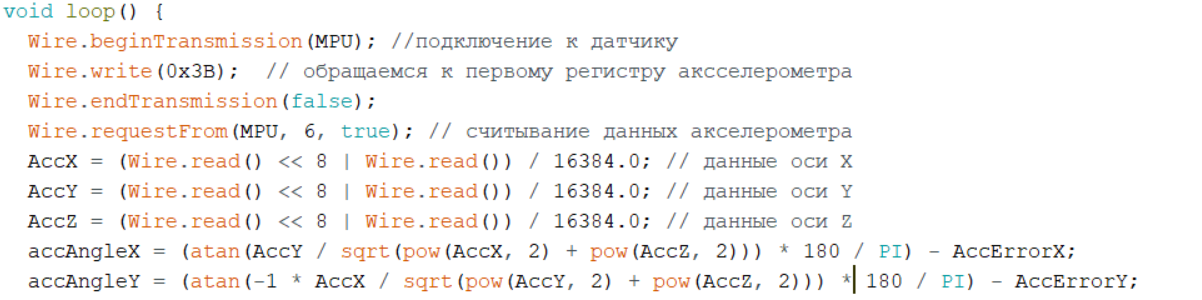


Рисунок 5 – получение и обработка данных акселерометра

Далее получим и обработаем данные с гироскопа по тому же принципу. Обращаемся к регистру гироскопа для получения данных, делим их на выбранную чувствительность, чтобы получить значения в диапазоне ±250 градусов в секунду, адреса регистров гироскопа показаны на рисунке 6.

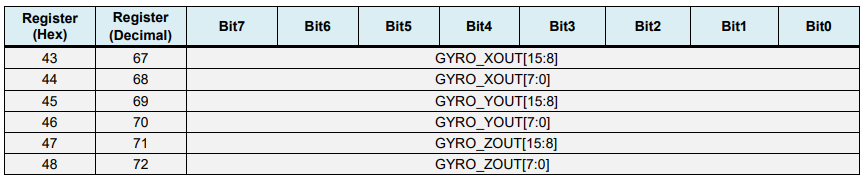


Рисунок 6 – адреса регистров гироскопа

После получения значений с гироскопа корректируем их на величину ошибки полученную при калибровке датчика. Для перевода значений с гироскопа в градусы необходимо поделить их на время, значение времени фиксируется перед каждой итерацией чтения с помощью функции millis(), показано на рисунке 7.

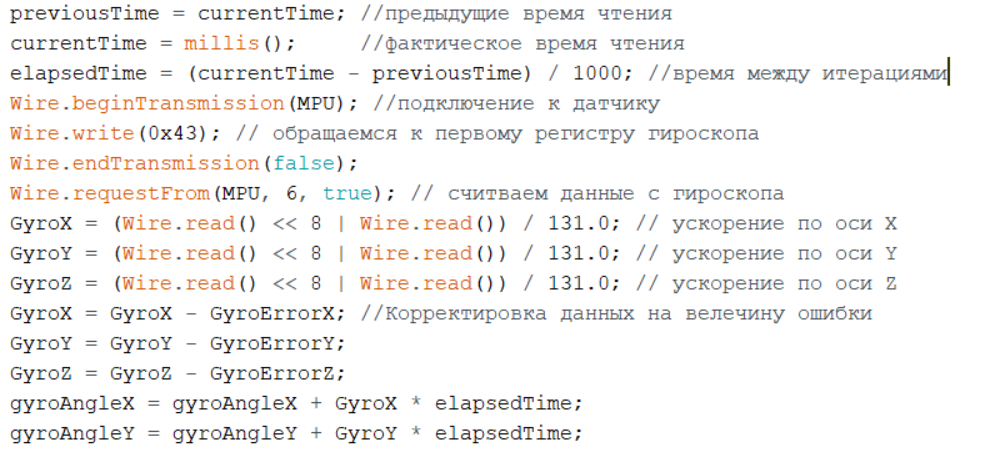


Рисунок 7 – получение и обработка данных с гироскопа

После получения значений с акселерометра и гироскопа объединяем их с помощью дополнительного фильтра. Далее преобразуем полученные данные, из диапазона значений ±90 сделаем диапазон от 0 до 180, это необходимо для корректного управления сервоприводами. Фрагмент кода для объединения значений фильтром и управления сервоприводами показан на рисунке 8.

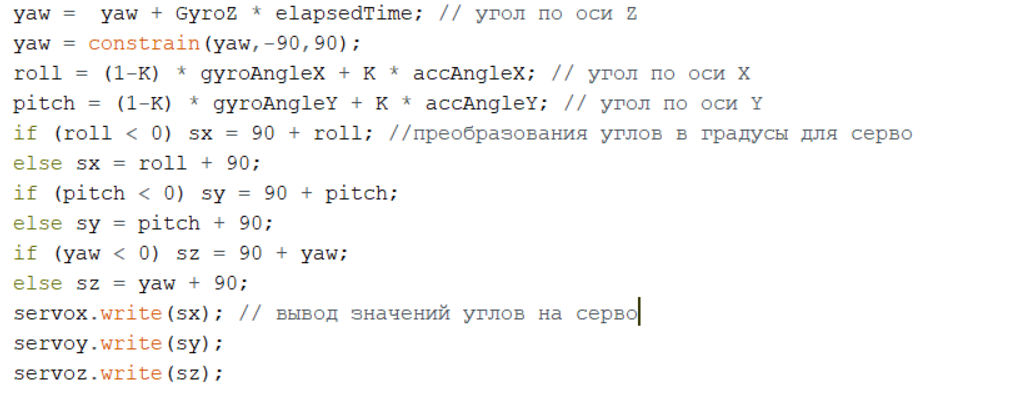


Рисунок 8 – Комплементарный фильтр и управление серво-приводами

# **3. Калибровка датчиков, расчет ошибки**

Главной проблемой гироскопа является то, что он дрейфует или вносит ошибку в выходной сигнал с течением времени. Для устранения этой проблемы необходимо откалибровать датчик перед его использованием или рассчитать его ошибку. Создадим отдельную функцию для расчета этой ошибки, которую будем вызывать в начале запуска программы. В этой функции производим 200 измерений для всех выводов с датчиков, суммируем их и делим на 200. При выполнении калибровки датчика, он должен быть в горизонтальном неподвижном положении, таким образом мы получим средние значение ошибки датчика. Код функции для расчета ошибки датчика показан на рисунке 9.

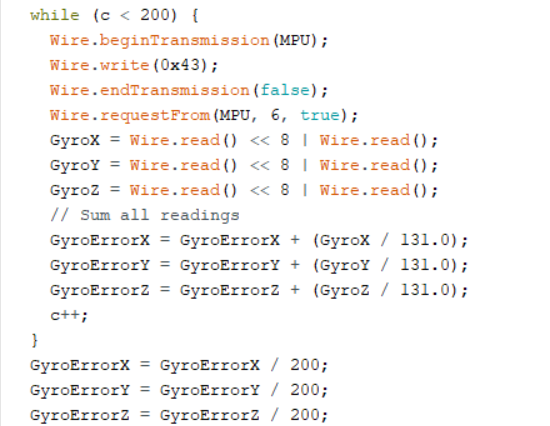


Рисунок 9 – функция для получения средней ошибки датчика

# **ЗАКЛЮЧЕНИЕ**

Была проведена большая работа по изучению гироскопа и акселерометра. В ходе создания устройства выяснилось, что качественно фиксировать положение горизонта объекта в пространстве не возможно, если не использовать одновременно гироскоп и акселерометр, так как им свойственно дополнять друг друга для более точного позиционирования объекта в пространстве.