ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

*«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

**Курсовой проект**

по дисциплине «Объектно-ориентированное программирование»

**«Разработка системы автоматического мониторинга вертикальной скорости и высоты на бвзе платформы esp32»**

|  |  |  |
| --- | --- | --- |
| Выполнил  студент гр. 3331506/20401 | *(подпись)* | Травин А.Ю. |
| Работу принял | *(подпись)* | Ананьевский М.С. |

Санкт-Петербург

2025

# Техническое задание

**Тема:**  
Разработка системы автоматического мониторинга вертикальной скорости и высоты на бвзе платформы esp32**.**

**Цель проекта:**  
Создать программно-аппаратный комплекс, способный в реальном времени отслеживать высоту и вертикальную скорость, а также производить визуальную индикацию высоты по средством дисплея, а скорости по средствам звукового сигнала.

## Задачи:

* Реализовать Электрическую схему устройства.
* Реализовать связь по низкоуровневым протоколам между всеми компонентами устройства.
* Реализовать алгоритм фильтрации данных, поступающих с датчика давления.
* Обеспечить индикацию полученных данных.

**Ожидаемый результат:**  
Прибор должен автономно работать и определять свою высоту и вертикальную скорость.

# Введение

Современное приборостроение активно развивается в направлении автономных систем малого размера, способных взаимодействовать с человеком как визуально, так и акустически для обеспечения большей наглядности поступающих данных.

В данной работе рассматривается разработка системы автономного мониторинга высоты, применяемого в авиации (барометр - вариометр) В качестве технической основы используется микроконтроллера: ESP32 C6.

Актуальность проекта обусловлена необходимостью создания простого и недорогого вариометра на распространённых компонентах.

# Теоретическая часть

Основной проблемой при обработке данных высоты являются значительные шумы. В связи с этим перед расчётом высот и скоростей был реализован алгоритм фильтрации входных данных.  
В качестве возможных вариантов предложенных в литературе [1,2] алгоритмов были рассмотрены:

- Растянутая выборка

- Бегущее среднее

- Экспоненциальное бегущее среднее

- Фильтр Калмана

В качестве наиболее подходящего был выбран фильтр Калмана. Его реализацию можно посмотреть в колде (см приложение).

# Описание аппаратной части

В данном проекте для создания системы автоматического трекинга используется роботизированная плата «esp xiao», и дополнительные компоненты:

* Модуль oled дисплея;
* Модульдатчика давления;
* Пьезо динамик;
* Юатарея питания;
* Пассивные компоненты.

Фото компонентов и фото в спянм виде изображены на рисунках 1, 2.

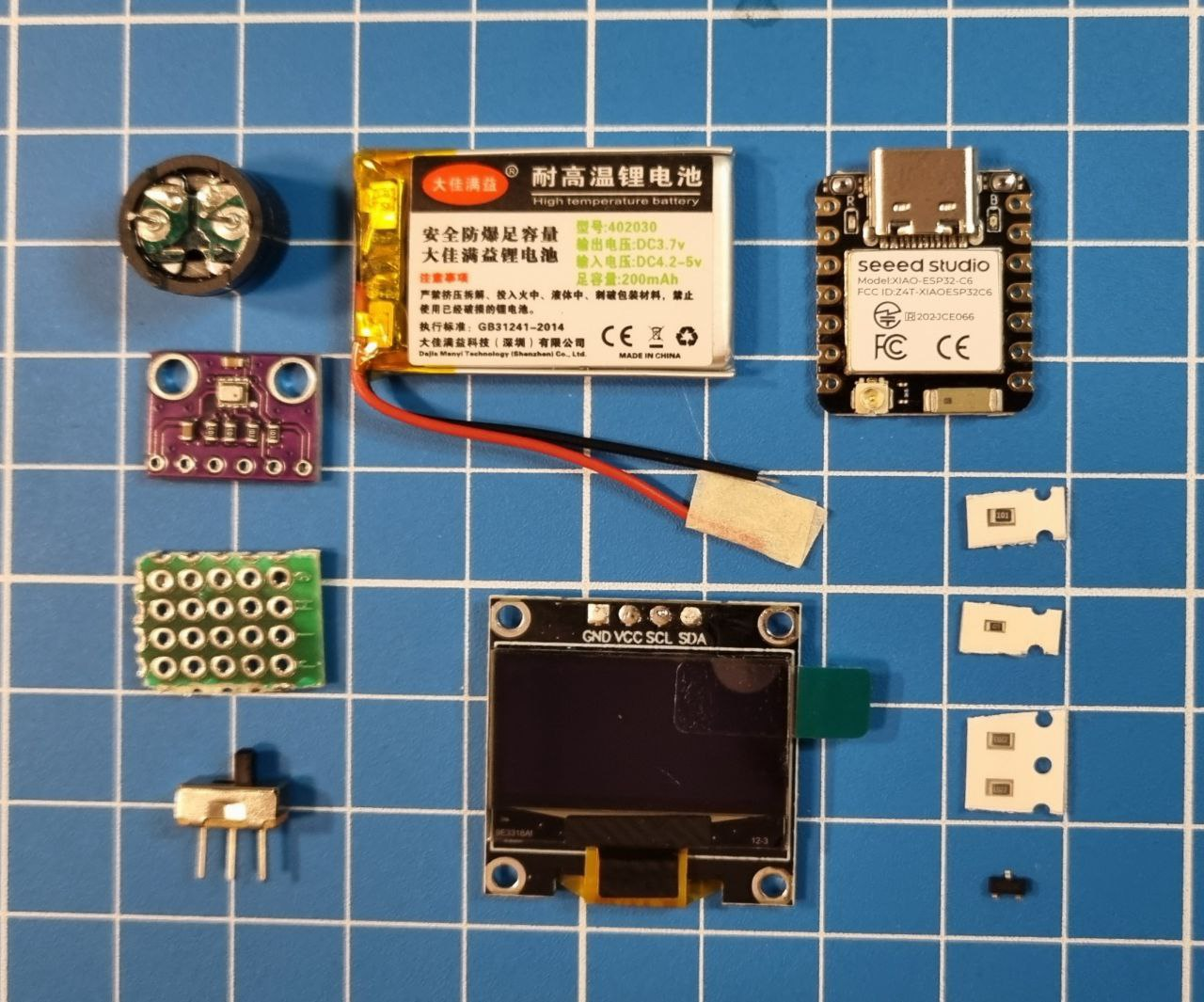


Рисунок 1 – компоненты устройства

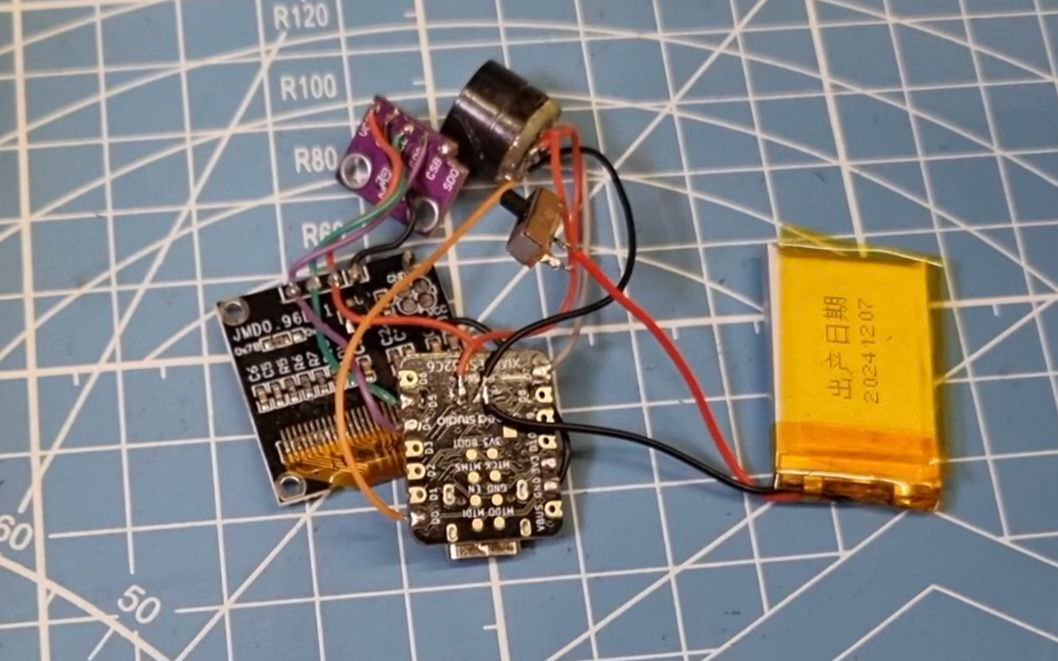


Рисунок 2 – компоненты в спаянном виде.

# Результат

В результате была разработана система, реализована электрическая схема устройства, связь по низкоуровневым протоколам между всеми компонентами устройства, написан алгоритм фильтрации данных, поступающих с датчика давления, обеспечена индикация полученных данных.

Физически реализованный прибор изображен на рисунке 3

.



Рисунок 3 – реализованный вариометр.

# Заключение

В процессе выполнения проекта мы получили опыт в области обработки данных и микроконтроллерного управления. Научились работать с библиотекой несколькими низкоуровневыми библиотекамина языке C++, разбираться в принципах фильтрации

В дальнейшем планируются следующие улучшения для более стабильной работы системы:

* Добавление функции ухода в режим энергосбережения при заряде ментше 15%;
* Добавление кнопок и меню для ручной настройки без использования компютера;

**Список литературы**

1. В. Г. Коберниченко Основы цифровой обработки сигналов Учебное пособие – URL:

https://elar.urfu.ru/bitstream/10995/65261/1/978-5-7996-2464-4\_2018.pdf

1. Ерофеев А.А. Теория автоматического управления : учебник для вузов / А.А. Ерофеев. — Санкт-Петербург : Политехника, 2021. — 416 с. — ISBN 978-5-7325-1204-3.
2. Полный справочник по С++ 4-е издание Издательский дом «Вильямс» https://sharpened.ucoz.ru/lib/polnyj\_spravochnik\_po\_c-gerbert\_shildt-2006.pdf
3. Документация на библиотеку использования дисплея Gyver Oled

<https://github.com/GyverLibs/GyverOLED>

1. Электронные проекты на основе ESP8266 и ESP32: Нил Кэмерон ISBN978-5-93700-141-2

# Приложения

Приложение 1. Программа управления вариометра

#include <Wire.h>

#include <GyverOLED.h>

#include <Adafruit\_BMP085.h>

GyverOLED<SSD1306\_128x64, OLED\_NO\_BUFFER> oled;

chars and 2 line display

float vario\_down = -1.2; // Установка СНИЖЕНИЯ

float vario\_up = 0.45;

float alt[51];

float tim[51];

float beep;

float Beep\_period;

float mux;

float Altitude;

const float p0 = 101325;

float old\_semple = 0;

float old\_altitude = 0;

long Pressure = 101325;

unsigned long bounseInput4P = 0UL;

int samples=40;

int maxsamples=50;

int countPressVal = 0;

bool tmp1 = 0;

bool countPress = 0;

bool bounseInput4S = 0;

bool bounseInput4O = 0;

boolean thermalling = false;

int p = 3;

Adafruit\_BMP085 bmp;

void setup()

{

pinMode(D7, OUTPUT);

bmp.begin();

oled.init();

oled.clear();

}

void loop()

{

float tempo=millis();

if ((millis() - old\_semple)> 400){

oled.setScale(10);

old\_altitude = bmp.readAltitude();

oled.invertText(false);

oled.setCursor(8, 2);

oled.print(old\_altitude);

old\_semple = millis() ;

};

float vario=0;

float N1=0;

float N2=0;

float N3=0;

float D1=0;

float D2=0;

for(int cc=1;cc<=maxsamples;cc++){ // Фильтр

alt[(cc-1)]=alt[cc];

tim[(cc-1)]=tim[cc];

};

alt[maxsamples]=bmp.readAltitude();

tim[maxsamples]=tempo;

float stime=tim[maxsamples-samples];

for(int cc=(maxsamples-samples);cc<maxsamples;cc++){

N1+=(tim[cc]-stime)\*alt[cc];

N2+=(tim[cc]-stime);

N3+=(alt[cc]);

D1+=(tim[cc]-stime)\*(tim[cc]-stime);

D2+=(tim[cc]-stime);

};

vario=1000\*((samples\*N1)-N2\*N3)/(samples\*D1-D2\*D2); // рачёт звук

if ((tempo-beep)>Beep\_period)

{

beep=tempo;

if (vario>vario\_up && vario<15 )

{

Beep\_period= 450-(vario\*5);

tone(D7,(1500+(120\*vario)),300-(vario\*6)); // звук на подъёме

//tone(D7,(1003+(100\*vario)),300-(vario\*5));

thermalling = true;

}

else if ((vario < 0.1 ) && (thermalling == true))

{

thermalling = false;

// tone(200, 800); // звук предпоток (по-желанию )

noTone(D7);

}

else if (vario< vario\_down){ // звук на сливе

// Beep\_period=200;

// tone(D7,(300-(vario)),340);

// tone(D7,(303-(vario)),340);

noTone(D7);

thermalling = false;

}

}ы

}