МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА ИНФОРМАЦИОННО-СЕТЕВЫХ ТЕХНОЛОГИЙ

КУРСОВАЯ РАБОТА (ПРОЕКТ)   
ЗАЩИЩЕНА С ОЦЕНКОЙ

РУКОВОДИТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ст. препод, канд. техн. наук |  |  |  | Т. В. Кротова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ |
| РЕЛЯЦИОННАЯ БАЗА ДАННЫХ «РЕГИСТРАЦИЯ БОЛЬНЫХ В ПОЛИКЛИНИКЕ» В СРЕДЕ ACCESS |
| по дисциплине: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Д. В. Самарин, Никита |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

Оглавление

[Задание на курсовое проектирование 3](#_Toc185195779)

[Техническое задание на прибор 4](#_Toc185195780)

[Диаграмма вариантов использования 5](#_Toc185195781)

[Основные требования 5](#_Toc185195782)

[Схемы и настройка проекта 7](#_Toc185195783)

[Описание алгоритмов 14](#_Toc185195784)

[Тестирование 21](#_Toc185195785)

[Слежение за объектом 21](#_Toc185195786)

[Вывод тепловой карты на экран 21](#_Toc185195787)

[Итог 21](#_Toc185195788)

[Экономическая оценка 21](#_Toc185195789)

[Заключение 22](#_Toc185195790)

[Список использованной литературы 23](#_Toc185195791)

[Приложение А. Код программы 25](#_Toc185195792)

# Задание на курсовое проектирование

Реализовать систему автоматического обнаружения и наведения на горячие объекты на основе тепловизора AMG8833 и микроконтроллера Arduino. Устройство должно считывать тепловую матрицу, определять координаты наиболее горячего объекта и автоматически регулировать положение камеры с помощью сервоприводов. Результаты работы системы отображаются на LCD-дисплее в виде теплового следа.

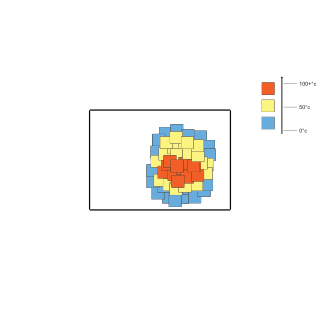
Основные требования к системе:

* Тепловизор AMG8833: матрица 8×8 пикселей, диапазон температуры от 0°C до 80°C.
* Сервоприводы: управление положением камеры в двух плоскостях.
* LCD-дисплей: отображение тепловой карты в режиме реального времени.
* Программное обеспечение: микропрограмма для Arduino, обеспечивающая обработку данных тепловизора, управление сервоприводами и вывод информации на дисплей.

Функциональность:

* Считывание тепловой матрицы с помощью AMG8833.
* Обнаружение горячих объектов и определение их координат.
* Управление сервоприводами для автоматической навигации на горячую точку.

Отображение текущей тепловой карты и данных на LCD-дисплее:



# Техническое задание на прибор

Цель: разработка устройства для автоматического определения и наведения на горячие объекты.

Состав устройства:

* Сервопривод (управляет направлением тепловизора);
* Тепловизор AMG8833 (с матрицей 8×8 пикселей);
* Микроконтроллер Arduino (обработка данных и управление);
* TFT-дисплей для отображения данных о температуре и положении.

В рамках разрабатываемой системы автоматического обнаружения и наведения на горячие объекты выделяются следующие исполнители (актеры):

* Пользователь (наблюдает за результатами работы);
* Микроконтроллер (обрабатывает данные и управляет устройством);
* Тепловизор (снимает тепловую матрицу);
* Сервоприводы (осуществляют перемещение камеры).

Основные функции системы:

* В режиме бездействия, когда пользователь не взаимодействует с прибором, на LCD-дисплее отображается тепловая карта в реальном времени;
* Регулярное считывание тепловой матрицы тепловизором AMG8833;
* Обнаружение наиболее горячих объектов в поле зрения тепловизора;
* Автоматическое наведение камеры на обнаруженный объект;
* Вывод тепловой карты на дисплей;

## Диаграмма вариантов использования

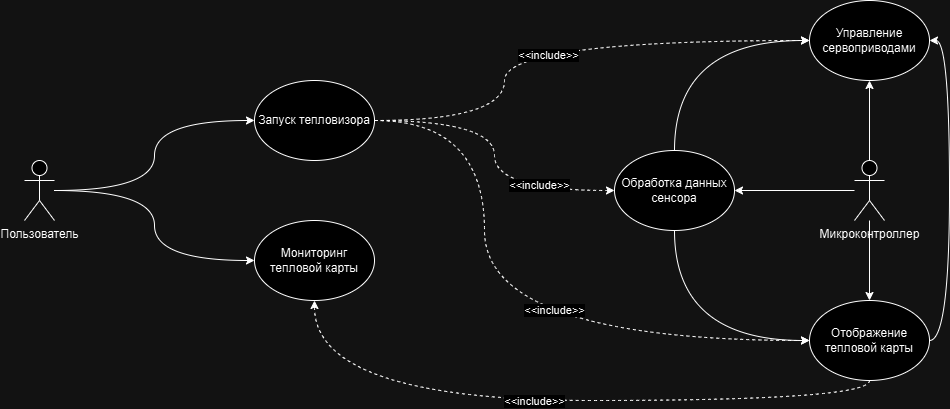


Рисунок 1. Диаграмма вариантов использования

## Основные требования

Входы

* Тепловизор AMG8833: возвращает матрицу 8x8, содержащую значения температуры в градусах Цельсия;
* Сервоприводы (2 шт.): обеспечивают повороты устройства в горизонтальной и вертикальной плоскостях;
* Контроллер Arduino: управляет сбором данных и движением сервоприводов.

Выходы

* TFT дисплей (128x160): отображает тепловую карту в виде цветной матрицы 8x8, где цвет каждого пикселя соответствует температуре.

Функции

1. Обработка матрицы температур для поиска максимального значения;
2. Управление сервоприводами для поворота в сторону горячей точки;
3. Визуализация теплового изображения на дисплее;
4. Отображение теплового изображения на LCD-дисплее.

Особенности

* Визуализация данных в реальном времени;
* Плавное движение сервоприводов с шагом 2°;
* Использование алгоритма обработки матрицы для управления движением;
* Работа от источника питания 5 В.

Питание

* Питание от USB или батареи 5 В.

Размер и вес

* Компактная конструкция, подходящая для использования в портативных устройствах.

Стоимость производства:

1. Arduino — 300 руб.;
2. AMG8833 — 1700 руб.;
3. Сервоприводы — 200 руб.;
4. LCD-дисплей — 170 руб.

В проекте используется тепловизор AMG8833, который передает данные в формате матрицы 8x8. Каждая ячейка матрицы содержит значение температуры в градусах Цельсия. Сенсор имеет высокую чувствительность и способен фиксировать изменения температуры с точностью до 0,25 °C. Для отображения данных используется дисплей, который позволяет визуализировать температурную карту с помощью цветового градиента.

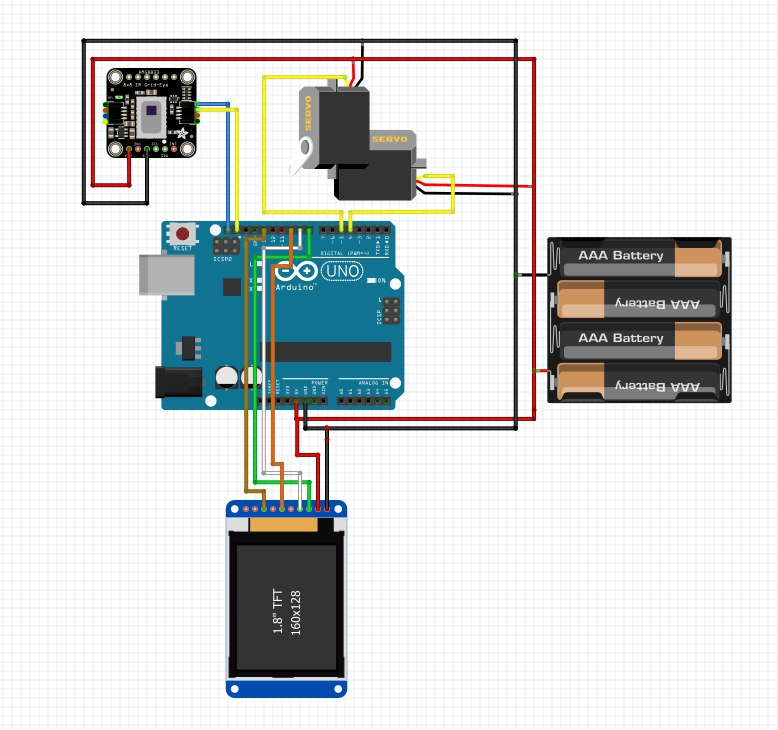


Рисунок 2. Схема взаимодействия с контроллером

# Схемы и настройка проекта

В проекте используется TFT-дисплей с диагональю 1.8" и разрешением 160x128 пикселей, построенный на базе контроллера ST7735. Этот дисплей поддерживает 16-битный цвет и оснащен последовательным интерфейсом SPI, что делает его подходящим для использования с Arduino Uno.

Для работы с дисплеем была выбрана библиотека **Adafruit GFX** в сочетании с **Adafruit ST7735**. Эта библиотека предоставляет функции для рисования примитивов, текста и изображений. В проекте используется аппаратный SPI-интерфейс Arduino Uno для подключения дисплея.

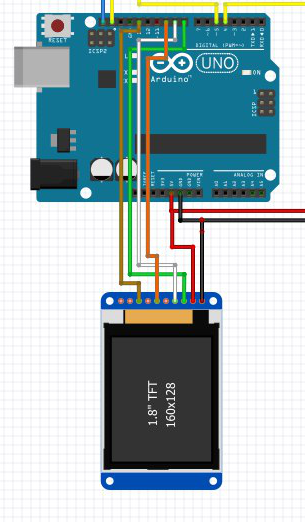


Рисунок 3. Схема подключения дисплея к Arduino

Схема подключения дисплея к Arduino Uno

* GND (дисплей) -> GND (Arduino Uno)
* VCC (дисплей) -> 5V (Arduino Uno)
* SCL (дисплей) -> D13 (Arduino Uno)
* SDA (дисплей) -> D11 (Arduino Uno)
* RES (дисплей) -> D9 (Arduino Uno)
* DC (дисплей) -> D8 (Arduino Uno)
* CS (дисплей) -> D10 (Arduino Uno)

Используемый тепловизор AMG8833 подключен к Arduino Uno через интерфейс I2C:

* SDA (тепловизор) -> A4 (Arduino Uno)
* SCL (тепловизор) -> A5 (Arduino Uno)

Для инициализации дисплея используется следующая последовательность команд:

1. Инициализация SPI-интерфейса через библиотеку SPI.
2. Настройка пинов для RES, DC и CS.
3. Вызов функции для инициализации дисплея.

Для работы с AMG8833 используется библиотека Adafruit AMG88xx. Она позволяет легко считывать данные с тепловизора в виде массива температур.

Более подробную информацию можно найти на сайте библиотек/github:

* <https://github.com/adafruit/Adafruit-GFX-Library>
* <https://github.com/mikaelpatel/Arduino-SPI>

Для работы с TFT-дисплеем и тепловизором в данном проекте были реализованы следующие ключевые функции:

* Функция передачи данных на дисплей:

Обеспечивает отображение тепловой карты в виде цветной матрицы. Для этого использована библиотека Adafruit ST7735, предоставляющая удобные методы для управления пикселями и рисования графики. Основная задача — преобразование значений температуры в цветовой градиент и вывод его на экран.

* Функция считывания данных с тепловизора:

Позволяет получить массив температур 8x8 с сенсора AMG8833 через интерфейс I2C. Для этого используется библиотека Adafruit AMG88xx, предоставляющая доступ к методам чтения массива температур и настройкам сенсора.

* Алгоритм обработки данных:
  + Преобразование температурной матрицы в цветовую палитру;
  + Определение горячей точки (максимальная температура);
  + Управление сервоприводами для поворота в сторону этой точки;
  + Обновление тепловой карты в реальном времени.

Эти функции были объединены в одну систему, где каждый компонент взаимодействует с микроконтроллером Arduino для обмена данными, выполнения команд и визуализации результатов.

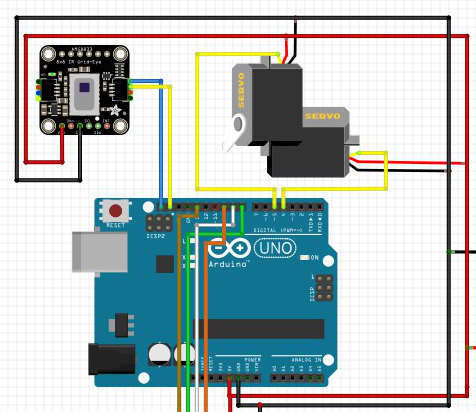


Рисунок 4. Схема подключения AMG8833

Все время выполнения программы тепловизор считывает данные с матрицы 8x8, а микроконтроллер ожидает изменений в температурном поле. Как только обнаруживается горячая точка (максимальная температура в матрице), программа запускает обработчик данных, который выполняет следующие действия:

* Поворот сервоприводов в направлении горячей точки на основании её координат в матрице;
* Обновление тепловой карты на дисплее с визуализацией данных в реальном времени;
* Отображение текущего состояния системы, включая температуру горячей точки и её положение.

После выполнения действий микроконтроллер возвращается в состояние ожидания новых данных с тепловизора, повторяя цикл измерений и обработки.

В зависимости от текущего состояния устройства возможны следующие действия:

* Обновление данных матрицы тепловизора и визуализация на дисплее;
* Корректировка положения сервоприводов для более точного слежения за горячей точкой;
* Уведомление об ошибке, если тепловизор не отвечает или выходные данные находятся вне допустимого диапазона.

Эта логика работы позволяет обеспечивать непрерывный контроль теплового поля и реагирование на изменения температуры в зоне покрытия.

Для более глубокого понимания предлагаю рассмотреть диаграмму состояний (рис. 5).

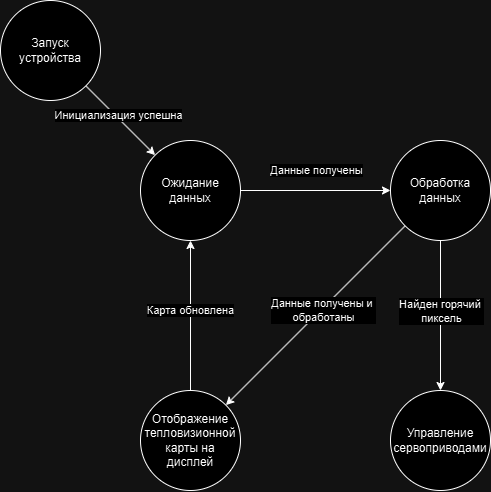


Рисунок 5. Диаграмма состояний

В контексте нашего проекта с тепловизором важно понимать, как различные компоненты взаимодействуют друг с другом, от инициализации до завершения работы системы. Предлагаю ознакомиться с диаграммой последовательности (рис. 6) и выделить ключевые моменты.

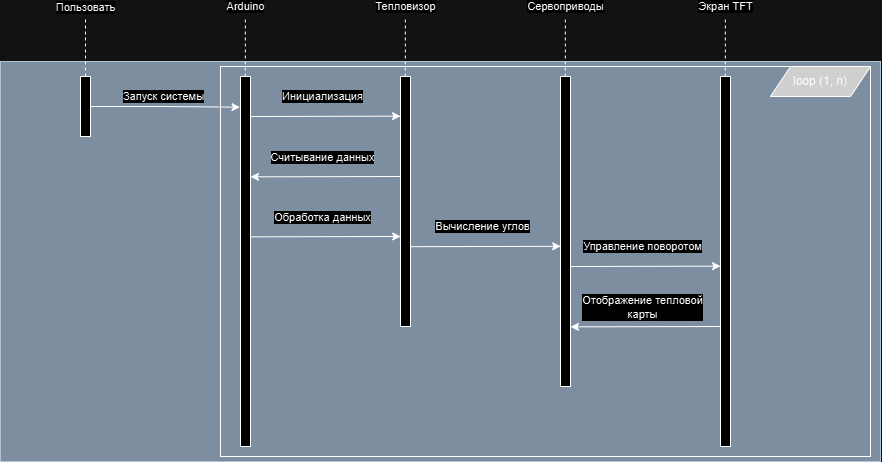


Рисунок 6. Диаграмма последовательности взаимодействия

Из диаграммы последовательности мы видим, что сначала пользователь инициирует запуск системы, что приводит к инициализации всех устройств (сенсора, сервоприводов и экрана). Затем тепловизор считывает данные температуры с помощью сенсора и передает их на Arduino, который анализирует эти данные. В свою очередь Arduino вычисляет, в каком направлении должны поворачиваться сервоприводы, чтобы сфокусироваться на самом горячем пикселе, и управляет их движением, параллельно отображая тепловую карту на TFT экране.

Цикличность взаимодействий позволяет следить за движущимся объектом в реальном времени.

1. Управление TFT-дисплеем

Для взаимодействия с TFT-дисплеем используется библиотека TFT, которая управляет экраном через интерфейс SPI.

Подключение осуществляется через три линии:

* CS (Chip Select) – выбор устройства для взаимодействия, назначен пин 10.
* DC (Data/Command) – определяет, передаются ли данные или команды, назначен пин 9.
* RST (Reset) – сброс экрана, назначен пин 8.

2. Работа с тепловизором AMG8833

Тепловизор AMG8833 подключается через интерфейс I²C. Используются линии SDA и SCL, которые взаимодействуют с библиотекой Adafruit\_AMG88xx.

Основная задача тепловизора — считывать температуру с матрицы 8×8 пикселей. Данные записываются в массив.

4. Управление сервоприводами

Сервоприводы используются для перемещения тепловизора в направлении горячей точки.

* Реализовано горизонтальное и вертикальное перемещения.

# Описание алгоритмов

Для более детального понимания того, как работает наш проект, следует ознакомиться с более детальным разбором алгоритмов некоторых функций.

Для начала детально разберем функцию передачи данных на дисплей. Ранее мы уже описали то, как она работает. Упростим вид подачи информации, сделав для неё диаграмму (рис. 7).



Рисунок 7. Алгоритм передачи изображения

Процесс повторяется, обеспечивая постоянное обновление данных и корректировку позиции сервоприводов.

Алгоритм работы функции считывания данных с тепловизора AMG8833 будет включать все этапы от инициализации сенсора до получения и обработки данных. Рассмотрим его наглядно (рис. 8).

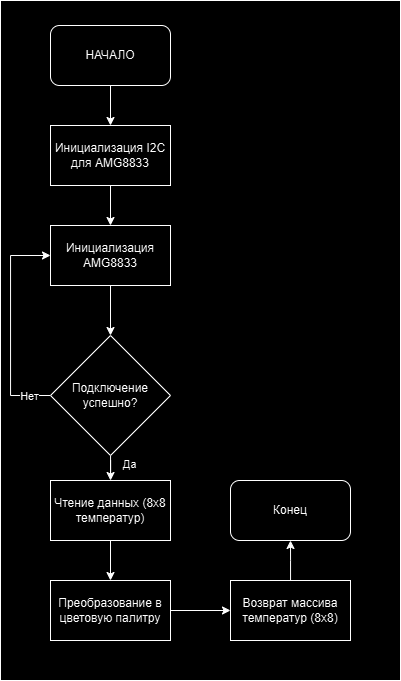


Рисунок 8. Алгоритм функции считывания данных

Благодаря преобразованию температуры в цвет в процессе считывания, на выходе мы получим цветное изображение, которое будет видно на TFT дисплее.

Выделим следующие основные управляющие структуры:

Функция setup выполняется один раз при старте программы. Она инициализирует TFT-дисплей, термодатчик AMG8833, устанавливает начальные углы для сервоприводов и отображает на экране статус датчика (успешная инициализация или ошибка).

Функция loop выполняется бесконечно в цикле и обеспечивает основной функционал программы:

* Считывание данных тепловой карты с термодатчика;
* Обработка данных для поиска пикселя с наивысшей температурой;
* Вычисление целевых углов для управления сервоприводами на основе позиции самого горячего пикселя;
* Постепенное изменение углов сервоприводов для достижения целевого положения;
* Визуализация тепловой карты на экране.

Функция getColorFromTemperature («Преобразование температуры в цвет») принимает значение температуры и возвращает 16-битное представление цвета для отображения пикселя на тепловой карте. Это обеспечивает визуализацию распределения температур в удобном для восприятия формате.

Функция rgbTo565 («Преобразование RGB в 16-битный формат») используется для перевода значений цветов из формата RGB в формат, поддерживаемый TFT-дисплеем.

Алгоритм плавного движения сервоприводов внутри функции loop реализован через сравнение текущего угла каждого сервопривода с целевым значением. Углы изменяются по небольшому шагу, что обеспечивает плавное движение.

Обработка данных тепловой карты: считывание массива из 64 температурных значений (8x8 матрица) и их преобразование для визуализации и управления.

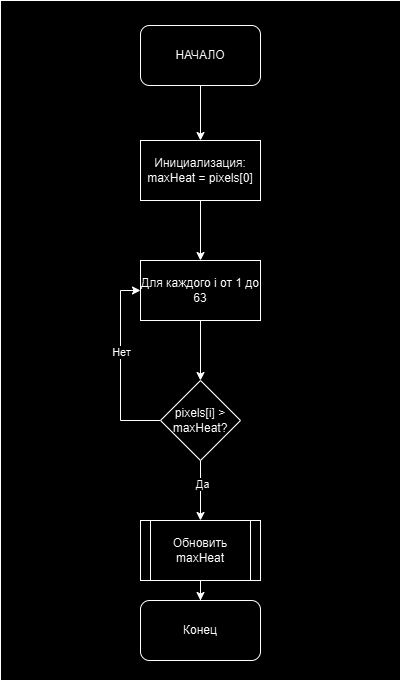


Рисунок 9. Алгоритм поиска самого горячего пикселя

Этот алгоритм определяет пиксель с максимальной температурой из массива данных тепловой карты, полученного от термодатчика AMG8833.

Описание процесса:

* Термодатчик AMG8833 передает массив температурных данных для каждого из 64 пикселей (матрица 8×8).
* Алгоритм проходит по каждому элементу массива, сравнивая его значение с текущим максимальным значением температуры.
* Если текущая температура больше максимальной, обновляется максимальное значение и индекс пикселя.

Результат работы:

* Получены индекс самого горячего пикселя в массиве, а также его значение температуры.
* Индекс используется для вычисления позиции этого пикселя в матрице (ряд и колонка).

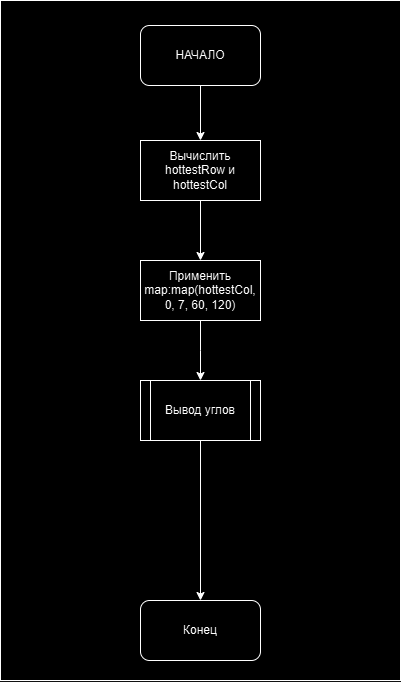


Рисунок 10. Алгоритм привязки положения пикселя к углам сервоприводов

Этот алгоритм преобразует координаты самого горячего пикселя (ряд и колонка в матрице 8×8) в целевые углы для управления сервоприводами.

Описание процесса:

* Координаты пикселя в матрице:
  + Ряд (от 0 до 7) преобразуется в угол наклона вертикального сервопривода.
  + Колонка (от 0 до 7) преобразуется в угол горизонтального сервопривода.
* Для преобразования используется функция map, которая масштабирует значения из одного диапазона (0-7) в другой (например, 60-120 градусов).

Результат работы:

* Вычислены целевые углы для обоих сервоприводов, чтобы система могла направиться на пиксель с максимальной температурой.

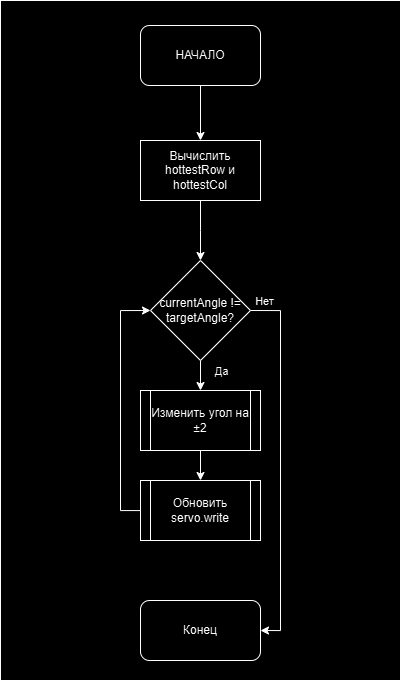


Рисунок 11. Алгоритм плавного движения сервоприводов

Этот алгоритм управляет сервоприводами, постепенно изменяя их углы, чтобы они достигли целевого значения.

Описание процесса:

* Для каждого сервопривода сравниваются текущий угол и целевой угол.
* Если текущий угол меньше целевого, он увеличивается на фиксированный шаг (например, 2 градуса).
* Если текущий угол больше целевого, он уменьшается на тот же шаг.
* После каждого изменения угла вызывается команда servo.write, которая обновляет положение сервопривода.

Особенности:

* Плавное изменение углов предотвращает резкие движения, повышая точность и надежность системы.
* Обновление углов выполняется в каждом цикле loop, обеспечивая постоянный контроль.

Результат работы:

* Сервоприводы постепенно поворачиваются в направлении целевого угла, ориентируя систему на горячий пиксель.

# Тестирование

Тесты для контроля соответствия прибора техническому заданию:

## Слежение за объектом

Агу агу

## Вывод тепловой карты на экран

Агу агу

## Итог

В ходе тестирования не было выявлено критических ошибок, влияющих на корректную работу прибора. Алгоритмы, прописанные на программном уровне, отработали правильно, тепловизор видит объект и следит за ним.

Однако, при более комплексном тестировании проекта есть вероятность обнаружить незначительную ошибку в работе прибора. Для того, чтобы прибор работал идеально, требуется проверить его на стрессовые/неформальные ситуации, чтобы исправить возможные ошибки.

# Экономическая оценка

Стоимость производства:

|  |  |
| --- | --- |
| Компонент | Цена |
| Arduino | 300 руб. |
| AMG8833 | 1700 руб. |
| Сервоприводы | 200 руб. |
| TFT-дисплей | 170 руб. |
| Итог | 2370 |

При массовом производстве, цена на компоненты будет ниже из-за оптовых покупок, что является одним из путей снижения цены.

# Заключение

В ходе выполнения курсового проекта была разработана и реализована система автоматического обнаружения и наведения на горячие объекты с использованием тепловизора AMG8833 и микроконтроллера Arduino. Основными функциями устройства стали: считывание тепловой матрицы, определение координат самого горячего объекта, управление сервоприводами для точного позиционирования камеры, а также визуализация тепловых данных на TFT-дисплее.

Разработанная система продемонстрировала способность в реальном времени анализировать тепловое поле, находить горячие области и динамически управлять положением камеры. Использование библиотеки Adafruit AMG88xx упростило работу с тепловизором, а библиотека Adafruit GFX обеспечила удобное отображение данных на дисплее.

Результаты проделанной работы подтверждают функциональность и эффективность разработанной системы. Она может быть применена в задачах мониторинга температуры, безопасности, автоматизированного контроля технологических процессов, а также в исследовательских целях.

Перспективы дальнейшего развития проекта включают:

* Повышение разрешения тепловых изображений с помощью интерполяции данных;
* Расширение функционала за счёт добавления обработки динамических объектов;
* Интеграцию системы с беспроводными технологиями для удалённого мониторинга;
* Улучшение интерфейса и расширение возможностей визуализации.

Таким образом, выполненное курсовое проектирование продемонстрировало успешное применение современных технологий для решения задачи автоматического обнаружения и наведения на горячие объекты.

# Список использованной литературы

1. “Arduino Cookbook" by Michael Margolis” — это руководство содержит большое количество рецептов для работы с Arduino, включая работу с сенсорами, экранами, сервоприводами и другими компонентами. Оно поможет понять, как правильно интегрировать различные устройства в одном проекте.
2. "Programming Arduino: Getting Started with Sketches" by Simon Monk — хорошая книга для начинающих, где объясняется основное программирование на Arduino, а также примеры работы с сенсорами и сервоприводами.
3. Официальная [документация Adafruit AMG8833](https://learn.adafruit.com/adafruit-amg8833-8x8-thermal-camera-sensor)
4. Adafruit AMG8833: 8x8 Thermal Camera Sensor Документация содержит подробности о подключении и использовании тепловизора AMG8833, а также примеры кода для считывания температурных данных с сенсора.
5. Официальная [документация библиотеки TFT для Arduino](https://www.arduino.cc/en/Reference/TFT)
6. TFT Library for Arduino Описание библиотеки для работы с TFT экранами, что поможет вам понять, как управлять экраном и отображать графику и текст.
7. Официальная [документация библиотеки Servo для Arduino](https://www.arduino.cc/en/Reference/Servo)
8. Servo Library for Arduino - руководство по использованию библиотеки для сервоприводов, которая позволяет управлять углами и движениями сервомоторов.
9. [Color Mapping for Heatmaps](https://en.wikipedia.org/wiki/Heat_map) — статья на Википедии, где рассмотрены различные способы отображения температурных значений с использованием цветовых градиентов, которые можно использовать для создания тепловых карт.
10. "[Arduino Robotics" by John-David Warren, Josh Adams, and Harald Molle](https://lib.hpu.edu.vn/handle/123456789/25863?show=full)” — это книга для тех, кто заинтересован в создании роботов и механических систем с использованием Arduino. В ней подробно объясняется, как управлять сервоприводами и сенсорами для создания сложных систем, таких как роботизированные камеры с тепловизорами.

# Приложение А. Код программы

Для удобства реализации проект расположен в открытом репозитории на GitHub.

Листинг кода:

|  |
| --- |
| #include <TFT.h>  #include <SPI.h>  #include <Adafruit\_AMG88xx.h>  #include <Wire.h>  #include <Servo.h> // Подключаем библиотеку для управления сервоприводами    #define cs   10  #define dc   9  #define rst  8    // Создаем объект для экрана TFT  TFT TFTscreen = TFT(cs, dc, rst);  Adafruit\_AMG88xx amg; // Создаем объект для тепловизора AMG8833    // Создаем объекты для управления сервоприводами  Servo horizontalServo; // Для горизонтального вращения  Servo verticalServo;   // Для вертикального наклона (не используется в этом коде)    // Текущие углы для сервоприводов  int currentHorizontalAngle = 90; // Начальный угол для горизонтального серво (по центру)  int currentVerticalAngle = 90;   // Начальный угол для вертикального серво (по центру)    // Функция для преобразования RGB в 16-битный цвет  uint16\_t rgbTo565(uint8\_t r, uint8\_t g, uint8\_t b) {      return (r >> 3) << 11 | (g >> 2) << 5 | (b >> 3);  }  // Функция для отображения температуры в виде цвета  uint16\_t getColorFromTemperature(float temperature, float mid\_temp) {      int k = 10;      // Если температура больше средней, то используем красные оттенки      if (temperature > mid\_temp) {          return rgbTo565(100 + mid\_temp \* k, 255 - temperature \* k, 25);      } else {          // Если температура ниже средней, то используем холодные оттенки          return rgbTo565(25, 25 + temperature \* k, 255 - mid\_temp \* k);      }  }  void setup() {      // Инициализация последовательного соединения      Serial.begin(9600);        // Инициализация экрана TFT      TFTscreen.begin();      TFTscreen.background(0, 0, 0); // Очищаем экран черным фоном      TFTscreen.stroke(255, 255, 255); // Устанавливаем белый цвет текста      TFTscreen.setTextSize(2); // Устанавливаем размер текста        // Инициализация тепловизора AMG8833      if (!amg.begin()) {          Serial.println("AMG ERROR");          TFTscreen.text("AMG ERROR", 12, 10);          while (1); // Останавливаем выполнение, если датчик не найден      } else {          Serial.println("AMG OK");          TFTscreen.text("AMG OK", 12, 10);      }        // Подключаем сервы к пинам      horizontalServo.attach(4); // Пин для горизонтального серво      // verticalServo.attach(5); // Пин для вертикального серво (не используется)      // Устанавливаем начальные положения серво      horizontalServo.write(currentHorizontalAngle);      // verticalServo.write(currentVerticalAngle);  }  int rectSize = 10; // Размер каждого прямоугольника (пикселя на экране)  int startX = (128 - (8 \* rectSize)) / 2; // Центрируем матрицу по горизонтали  int startY = (160 - (8 \* rectSize)) / 2; // Центрируем матрицу по вертикали    void loop() {      float pixels[64]; // Массив для хранения данных с тепловизора      amg.readPixels(pixels); // Считываем пиксели с датчика        // Определяем самый горячий пиксель      int hottestPixelIndex = 0;      float maxHeat = pixels[0];      for (int i = 1; i < 64; i++) {          if (pixels[i] > maxHeat) {              maxHeat = pixels[i];              hottestPixelIndex = i;          }      }      // Определяем строку и колонку самого горячего пикселя      int hottestRow = hottestPixelIndex / 8; // Ряд (0-7)      int hottestCol = hottestPixelIndex % 8; // Колонка (0-7)      // Преобразуем строку и колонку в углы для сервоприводов      int targetHorizontalAngle = map(hottestCol, 0, 7, 60, 120); // Угол для горизонтального серво      int targetVerticalAngle = map(hottestRow, 0, 7, 60, 120);   // Угол для вертикального серво      // Плавное движение горизонтального серво      if (currentHorizontalAngle != targetHorizontalAngle) {          if (currentHorizontalAngle < targetHorizontalAngle) {              currentHorizontalAngle += 2; // Увеличиваем угол для серво          } else {              currentHorizontalAngle -= 2; // Уменьшаем угол для серво          }          horizontalServo.write(currentHorizontalAngle); // Применяем новый угол      }      // Плавное движение вертикального серво (этот код не используется, так как вертикальный сервопривод закомментирован)      // if (currentVerticalAngle != targetVerticalAngle) {      //     if (currentVerticalAngle < targetVerticalAngle) {      //         currentVerticalAngle += 2;      //     } else {      //         currentVerticalAngle -= 2;      //     }      //     verticalServo.write(currentVerticalAngle);      // }      // Отображение тепловой карты на экране      for (int i = 0; i < 8; i++) {          for (int j = 0; j < 8; j++) {              int index = i \* 8 + j;              float temperature = pixels[index]; // Температура на этом пикселе              uint16\_t color = getColorFromTemperature(temperature, 19); // Получаем цвет для пикселя              TFTscreen.fillRect(startX + j \* rectSize, startY + i \* rectSize, rectSize, rectSize, color); // Рисуем прямоугольник с цветом          }      }        delay(50); // Задержка для плавности отображения  } |