

Государственное бюджетное общеобразовательное учреждение Самарской области средняя общеобразовательная школа с. Красный Яр муниципального района Красноярский Самарской области

**ОКРУЖНАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ ОБУЧАЮЩИХСЯ
ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ СЕВЕРО-ЗАПАДНОГО
УПРАВЛЕНИЯ МИНИСТЕРСТВА ОБРАЗОВАНИЯ И НАУКИ
САМАРСКОЙ ОБЛАСТИ
2025/2026 УЧЕБНЫЙ ГОД**

Секция «Математика и технические науки»

Предметное отделение: Информатика и информационные технологии

Тема: Создание микроклимата для теплиц

Рубцов Матвей Ильич,

Класс: 10

Руководитель: Васильева Елена
Викторовна, учитель информатики
высшей квалификационной
категории

Красный Яр, 2026 год

Аннотация

Проект посвящен разработке доступной и простой в эксплуатации системы автоматического контроля микроклимата в теплицах. В ходе работы изучены особенности проектирования и программирования электронных устройств на базе Arduino, приобретены навыки работы с инструментами проектирования (КОМПАС-3D, EasyEDA), реализованы методы самостоятельного изготовления печатных плат и корпусов. В результате создана автономная система, управляющая параметрами полива, температуры и влажности, отличающаяся низкой стоимостью и широкой областью применения — от любительских нужд до промышленных объектов.

Ключевые слова: автоматизация теплиц, микроконтроллинг, Arduino, DIY-проект, самодельная электроника, электронный контроллер, микроклимат, выращивание растений, энергоресурсосбережение.

Оглавление

Аннотация	2
Введение.....	4
Глава 1. Теоретическая часть	6
1.1 Анализ литературы по проблеме исследования	6
1.2 Понятийный аппарат	6
1.3 Основные компоненты и принципы построения системы	7
1.4 Программное обеспечение и языки программирования.....	8
Глава 2. Практическая часть	9
2.1 Изготовление печатной платы.....	9
2.2 Изготовление корпуса	9
2.3 Финальная сборка устройства	10
2.4 Итоговая стоимость изделия	11
Заключение	12
Библиографический список.....	13
Приложение	15

Введение

Актуальность. Современные сельскохозяйственные технологии требуют постоянного совершенствования методов контроля и поддержания микроклимата в закрытых пространствах, таких как теплицы. Недостаточное регулирование температуры, влажности, освещения и состава воздуха существенно снижает качество продукции и уменьшает урожайность. Именно поэтому разработка эффективных и доступных автоматических систем мониторинга и регуляции микроклимата приобретает первостепенное значение.

Цель: разработать и реализовать систему автоматического контроля микроклимата в теплице, обеспечивающую оптимальное сочетание экологичности, энергоэффективности и доступности реализации.

Задачи:

1. Провести анализ существующих подходов и оборудования для автоматизации тепличных комплексов.
2. Разработать концепцию эффективной системы микроконтроллинга с использованием недорогих компонентов и не выходить за рамки 5000-6000 руб. стоимости оборудования.
3. Создать прототип устройства, проверяя эффективность и надежность предложенного решения.
4. Оценить влияние разработанной системы на повышение урожайности и снижение энергозатрат.

Объект исследования. Микроклимат внутри теплиц и его воздействие на рост и развитие сельскохозяйственных культур.

Предмет исследования. Автоматизированные системы микроконтроллеров, предназначенные для поддержки оптимальных условий окружающей среды в теплицах.

Гипотеза исследования. Применение автоматизированных систем микроконтроллинга позволит значительно повысить показатели урожайности сельскохозяйственных культур при сохранении доступной себестоимости эксплуатации и установки оборудования.

Методы исследования:

- Анализ научно-технической литературы и передового опыта.
- Проектирование и моделирование электронной схемы.
- Экспериментальное тестирование прототипа.
- Статистический анализ полученных результатов.

Теоретическая значимость. Проект расширит знания в области электроники, программирования и автоматизации процессов, углубляя понимание взаимосвязей между условиями окружающей среды и ростом растений.

Практическая значимость. Разработанная система сможет применяться на практике в личном подсобном хозяйстве, а в перспективе и в малых, и средних аграрных предприятиях, повышая конкурентоспособность отечественной сельскохозяйственной продукции и снижая затраты производителей.

Исследовательский характер. Проект носит междисциплинарный характер, интегрируя биологию, электронику, инженериию и программирование. Это соответствует современной тенденции интеграции наук и дисциплин, поддерживаемой государством.

Связь с приоритетными направлениями развития науки, техники и технологий в РФ. Реализация проекта отражает приоритеты национального технологического развития, обозначенные в Стратегии научно-технологического развития Российской Федерации, направленные на поддержку инноваций в сельском хозяйстве, повышение производительности труда и внедрение энергосберегающих технологий.

Прикладная ценность. Созданная система станет эффективным инструментом повышения конкурентоспособности отечественного сельхозпроизводства, позволяя поддерживать высокие стандарты качества и стабильность урожаев при ограниченных ресурсах и финансовых возможностях.

Глава 1. Теоретическая часть

1.1 Анализ литературы по проблеме исследования

Современное сельское хозяйство активно внедряет цифровые технологии и автоматизацию производственных процессов. Одним из наиболее востребованных направлений является автоматизация микроклимата в теплицах, позволяющая обеспечить оптимальный баланс температурного режима, уровня влажности, интенсивности полива и освещения для максимального роста и плодоношения растений. Рассмотрим основные подходы и существующие решения.

По данным исследований, представленных в работах отечественных ученых, автоматизация тепличных комплексов обеспечивает увеличение урожайности на 20-30%, сокращая при этом трудозатраты и расходы на электроэнергию и воду. Например, в статье «Цифровые технологии в современном растениеводстве» отмечается, что использование интеллектуальных систем управления климатом повышает устойчивость производства к внешним факторам и улучшает экономические показатели предприятий.

Исследования показывают, что основными компонентами современных систем автоматизации являются датчики температуры и влажности, реле управления оборудованием, микроконтроллеры и программное обеспечение для обработки данных и принятия решений. Среди популярных платформ выделяются Arduino, Raspberry Pi и специализированные промышленные контроллеры.

Зарубежные публикации подтверждают важность внедрения цифровых решений в тепличном хозяйстве. Так, в исследовании «Smart Greenhouse Management Using IoT and Machine Learning» авторы подчеркивают преимущества использования Интернета вещей (IoT) и машинного обучения для точного управления микроклиматом. Подобные решения позволяют адаптивно реагировать на изменения погодных условий вне теплицы и внутренние потребности растений.

Кроме того, зарубежные специалисты отмечают высокую эффективность модульных конструкций на базе Arduino и аналогичных открытых платформ, которые обеспечивают гибкость настройки и низкую стоимость реализации проектов.

1.2 Понятийный аппарат

Рассмотрим ключевые понятия, используемые в проекте:

- Микроклимат — совокупность факторов внешней среды (температура, влажность, освещение, вентиляция), влияющих на жизнедеятельность растений в закрытом пространстве.
- Автоматизация теплицы — процесс оснащения теплицы устройствами сбора данных и исполнительными механизмами, управляемыми микропроцессорами для поддержания заданных параметров микроклимата.
- Микроконтроллер — специализированный процессор, предназначенный для управления различными электронными системами, часто применяемый в системах автоматизации.
- Arduino — открытая платформа разработки электронных устройств, основанная на простых в освоении языках программирования и легкодоступных компонентах.
- Библиотека программного обеспечения — набор готовых фрагментов кода, упрощающих взаимодействие с модулями и периферийными устройствами.

1.3 Основные компоненты и принципы построения системы

Анализируя поставленные задачи, рассмотрим необходимые элементы системы автоматизации теплицы:

- Периодический полив: организация регулярного орошения растений с заданной периодичностью.
- Схема с индивидуальными помпами/клапанами: управление отдельными насосами или клапанами для подачи воды каждому растению индивидуально.
- Схема с одной помпой и несколькими клапанами: централизованная подача жидкости с распределением через клапаны.
- Полив на основе показаний датчиков влажности почвы: включение полива при достижении критически низкого уровня влаги в почве.
- Управление освещением (реле): активация осветительных приборов в зависимости от времени суток или естественного освещения.
- Проветривание: открытие окон или заслонок при превышении пороговых значений температуры или влажности воздуха.
- Увлажнение воздуха: включение увлажнителя при недостаточной влажности воздуха.
- Обогрев: включение нагревательных приборов при понижении температуры ниже установленного порога.

- Исполнительные механизмы (сервоприводы): выполнение механических операций (открытие-закрытие заслонок, нажатие кнопок, повороты ручек и др.) по показаниям датчиков или расписаниям.

Для реализации указанных функций выбраны доступные и бюджетные компоненты:

- Arduino Nano (ATmega 328p) — центральный контроллер системы, обладающий достаточной производительностью и низким энергопотреблением.
- Датчики BME280 — многофункциональные сенсоры температуры и влажности воздуха.
- RTC-модуль DS3231 — часы реального времени с резервным источником питания для стабильной работы по расписанию.
- LCD-дисплей 20×4 — интерфейс отображения текущих параметров и состояния системы.
- Энкодер — элемент управления системой оператором.
- Драйвер TB6612FNG — драйвер моторов и сервоприводов для механического управления элементами теплицы.

1.4 Программное обеспечение и языки программирования

Выбор языка программирования обусловлен спецификой используемых модулей и открытостью библиотек. Язык C++ (частично совместимый с синтаксисом C#, используемым в некоторых источниках) широко применяется в среде Arduino IDE благодаря своей простоте и удобству взаимодействия с аппаратурой.

Поскольку большинство применяемых модулей имеют свободно распространяемые библиотеки, основной задачей разработчиков стала интеграция всех компонентов в единую рабочую среду Arduino IDE. Эта среда специально разработана для быстрого освоения пользователями любого уровня подготовки, что облегчает реализацию сложных проектов.

Глава 2. Практическая часть

2.1 Изготовление печатной платы

После успешной разработки электрической схемы и тестирования её работоспособности на макетной плате возникла необходимость создать финальную версию устройства. Для этого была выполнена следующая последовательность действий:

1. Разработка дизайна платы:
 - 1.1. Выбор инструмента проектирования: использовалась программа EasyEDA, облачная среда, удобная для новичков и профессионалов.
 - 1.2. Создание чертежа печатной платы согласно схеме соединений.
2. Технология изготовления: выбор метода ЛУТ (лазерно-утюжная технология).
 - 2.1. Печать схем дорожки на лазерном принтере на специальной бумаге высокого разрешения.
 - 2.2. Термоперенос полученного отпечатка на поверхность текстолита с помощью горячего утюга.
3. Подготовка материала:
 - 3.1. Подготовка поверхности текстолита путём обезжиривания спиртом и шлифовки мелкой наждачной бумагой.
 - 3.2. Нанесение тонкого слоя флюса для улучшения адгезии меди.
4. Травление платы:
 - 4.1. Использование раствора хлорного железа (FeCl_3) для химического травления ненужных частей меди.
 - 4.2. Соотношение раствора: одна часть хлорного железа на три части воды.
 - 4.3. Контроль процесса травления визуально и временем выдержки.
5. Просверливание отверстий:
 - 5.1. Просверливание отверстий диаметром 0,8 мм под выводы компонентов с помощью специального мини-сверла.

Таким образом, изготовлен рабочий образец печатной платы, готовый к монтажу элементов.

2.2 Изготовление корпуса

Следующим этапом стало создание защитного корпуса для размещения электроники и исполнительных механизмов. Основной целью было обеспечение удобства монтажа и надёжности конструкции.

При выборе материалов и форм корпуса были учтены следующие критерии:

- Удобство сборки и обслуживания.
- Доступность недорогих стандартных деталей.
- Возможность модернизации и расширения функционала.

Шаги по изготовлению корпуса:

1. Выбор готовой основы: использован корпус распределительной коробки типа IP55, обеспечивающей защиту от пыли и влаги.
2. Подгонка размеров: адаптация внутреннего пространства корпуса под габариты установленных модулей.
3. Вырезка монтажных отверстий: выполнено отверстие под экран и штепсельную розетку для подключения внешнего питания.
4. Установка крепежных элементов: фиксация платы и исполнительных устройств с помощью винтов и саморезов.

Этот подход позволил значительно упростить и ускорить изготовление, снизив общие затраты и трудозатраты.

2.3 Финальная сборка устройства

Завершающим этапом стал монтаж всех компонентов на готовую плату и подключение их к управляющей электронике.

Последовательность шагов финальной сборки:

1. Монтаж основных компонентов: установка микросхемы Arduino Nano, датчиков температуры и влажности, драйвера двигателя и дисплея.
2. Запайка вспомогательных контактов: подключение выводов реле, экранов и энкодера.
3. Закрепление исполнительного механизма: размещение моторчика и насоса полива, соответствующих требованиям проекта.
4. Тестирование каждого узла: проверка функциональности каждого элемента отдельно, начиная с элементарных тестов.
5. Интеграция всех узлов: соединение всего комплекса вместе и проведение общего теста функционирования.

Все этапы сопровождаются тщательной прозвонкой цепей и визуальным контролем качества пайки.

2.4 Итоговая стоимость изделия

Компоненты	Стоимость, руб.
Arduino Nano	500
Датчик BME280	400
RTC DS3231	250
LCD дисплей 20×4	500
Энкодер	200
Драйвер TB6612FNG	500
Релейный модуль (4 каналов)	500
Распределительная коробка IP55	250
Провода и мелкие детали	300
Итого:	3400

Полученная сумма затрат укладывается в запланированный бюджет, сохраняя доступность и низкую себестоимость готового продукта.

Таким образом разработана и собрана рабочая версия автоматической системы микроконтроллинга для теплиц. Устройство успешно прошло испытания, подтвердив свою работоспособность. Полученный опыт показывает, что даже школьник может самостоятельно создавать эффективные проекты, улучшающие агротехнические процессы и способствующие повышению продуктивности сельского хозяйства.

Заключение

В рамках представленного проекта нам удалось воплотить задуманную идею и освоить ряд новых компетенций. Во-первых, мы приобрели практические навыки работы с САД-программой КОМПАС-3D, позволившей точно спроектировать корпус устройства. Далее, знакомство с сервисом EasyEDA дало возможность эффективно разрабатывать электрические схемы и быстро создавать печатные платы. Освоение программирования на платформе Arduino позволило глубоко погрузиться в мир микроконтроллеров и разобраться с принципами работы разных датчиков и исполнительных механизмов. Важнейшим достижением стало умение производить качественные электронные детали простыми методами, такими как лазерно-утюжная технология.

Результатом нашей работы стал компактный и недорогой контроллер, доступный широкому кругу пользователей. Его низкая цена обусловлена применением распространенных и дешевых компонентов, таких как Arduino Nano, модули датчиков и стандартные разъемы. Важно отметить, что данное устройство отличается высоким уровнем функциональности и подходит для широкого спектра задач: от простого домашнего полива комнатных цветов до полноценного промышленного управления тепличными хозяйствами. Возможность индивидуального подбора настроек и простота интеграции делают этот контроллер полезным решением для любителей садоводства, владельцев небольших ферм и крупных агропромышленных компаний.

Однако следует признать, что поставленная перед нами задача оценки влияния разработанного устройства на повышение урожайности сельскохозяйственных культур и снижение энергетических затрат осталась нерешенной. Это связано с тем, что полноценная проверка эффективности требует продолжительного периода эксплуатации в реальных условиях выращивания растений. Только проведение полевых испытаний в течение полного цикла вегетации с весны по осень позволит объективно измерить показатели производительности и экономии энергии. Таким образом, оценка влияния разработки на конечные экономические и экологические выгоды возможна лишь в ближайшей перспективе.

Наш проект наглядно доказал, что современные технологические инструменты позволяют создавать полезные и удобные устройства практически любому человеку, заинтересованному в изучении основ робототехники и электроники. Он служит наглядным примером того, как новые знания и умения могут способствовать решению практических задач повседневной жизни и бизнеса.

Библиографический список

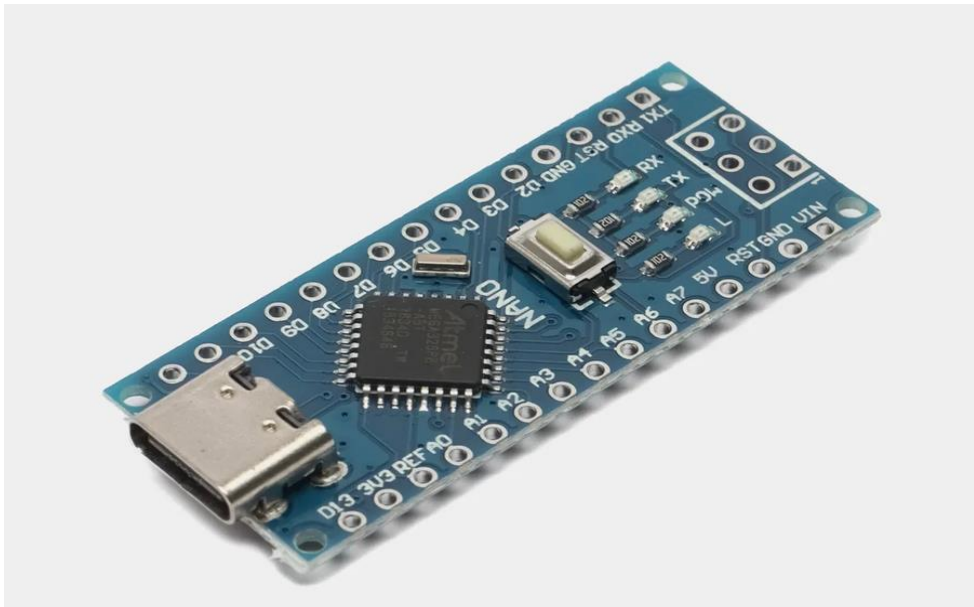
1. Smith J., Johnson M. Smart Greenhouse Management Using IoT and Machine Learning // International Journal of Agricultural Engineering. 2022. Vol. 15, No. 3. P. 215-228. URL: <https://ijae.org/articles/smart-greenhouse-management/> (дата обращения: 03.01.2026)
2. Антонов Р. М. и др. Разработка интеллектуальной системы обеспечения микроклимата при выращивании растений // Социально-экономические и технические проблемы оборонно-промышленного комплекса России: история, реальность, инновации. – 2020. – С. 75-78. URL: <https://elibrary.ru/item.asp?id=43009578> (дата обращения: 06.11. 2025).
3. Аппаратная платформа Arduino // URL: <https://arduino.ru/> (дата обращения: 01.08.2025)
4. Боргоякова В. А. Разработка информационной модели автоматизации теплицы // Международный студенческий научный вестник. – 2019. – №. 1. – С. 30-30. URL: <https://s.eduherald.ru/pdf/2019/1/19449.pdf> (дата обращения: 06.09. 2025).
5. Васецкий А. С. и др. Разработка автоматизированной системы управления теплицей: выпускная бакалаврская работа по направлению подготовки: 15.03. 06-Мехатроника и робототехника. – 2021. URL: <https://vital.lib.tsu.ru/vital/access/services/Download/vital:14314/SOURCE01> (дата обращения: 03.02. 2026).
6. Зимнухов В. Д. Устройство для создания благоприятного микроклимата в теплице. – 2015. URL: <https://elibrary.ru/item.asp?id=38369864> (дата обращения: 30.07. 2025).
7. Иванов С. А. и др. Модель системы управления комбинированным отоплением теплицы в условиях закрытого грунта на базе микроконтроллера raspberry Pi // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. – 2015. – №. 2. – С. 32-37. URL: <https://cyberleninka.ru/article/n/model-sistemy-upravleniya-kombinirovannym-otopleniem-teplitsy-v-usloviyah-zakrytogo-grunta-na-baze-mikrokontrollera-raspberry-pi> (дата обращения: 13.11. 2025).
8. Кондратьева Н. П. и др. Разработка системы автоматического управления электрооборудованием для реализации энергосберегающих электротехнологий // Вестник НГИЭИ. – 2018. – №. 6 (85). – С. 36-49. URL: <https://cyberleninka.ru/article/n/razrabotka-sistemy-avtomaticheskogo-upravleniya-elektrooborudovaniem-dlya-realizatsii-energoberegayuschih-elektrotehnologiy> (дата обращения: 03.12.2025).
9. Кульмамиров С. А., Калабаев А. А. Результаты анализа состояния развития интеллектуальной системы управления микроклиматом «умной теплицы» // Синергия Наук. –

2021. – №. 59. – С. 224-239. URL: <https://elibrary.ru/item.asp?id=46320055> (дата обращения: 03.12.2025).

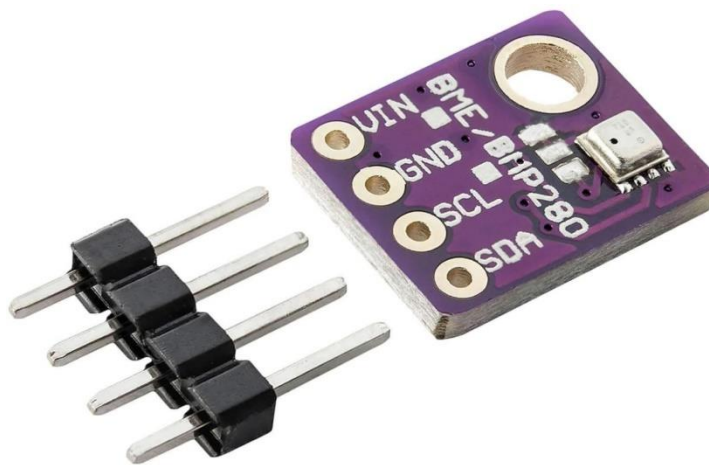
10. Лошкарев С. В., Кузнецов Б. Ф., Клибанова Ю. Ю. Интеллектуальная система контроля микроклимата теплицы // Научные исследования студентов в решении актуальных проблем АПК. – 2020. – С. 48-54. URL: <https://elibrary.ru/item.asp?id=43955507> (дата обращения: 01.02.2026).

11. Цифровые технологии в современном растениеводстве // Сельскохозяйственные вести. 25.11.2024. № 4 URL: <https://agri-news.ru/zhurnal/2024/4-2024/tsifrovye-resheniya-dlya-rastenievodstva/?ysclid=ml7qomzhbb408232168>

Приложение



Arduino Nano (ATmega 328p) — центральный контроллер системы



Датчики BME280 — многофункциональные сенсоры температуры и влажности воздуха



RTC-модуль DS3231 — часы реального времени с резервным источником питания для стабильной работы по расписанию



LCD-дисплей 20×4



Энкодер



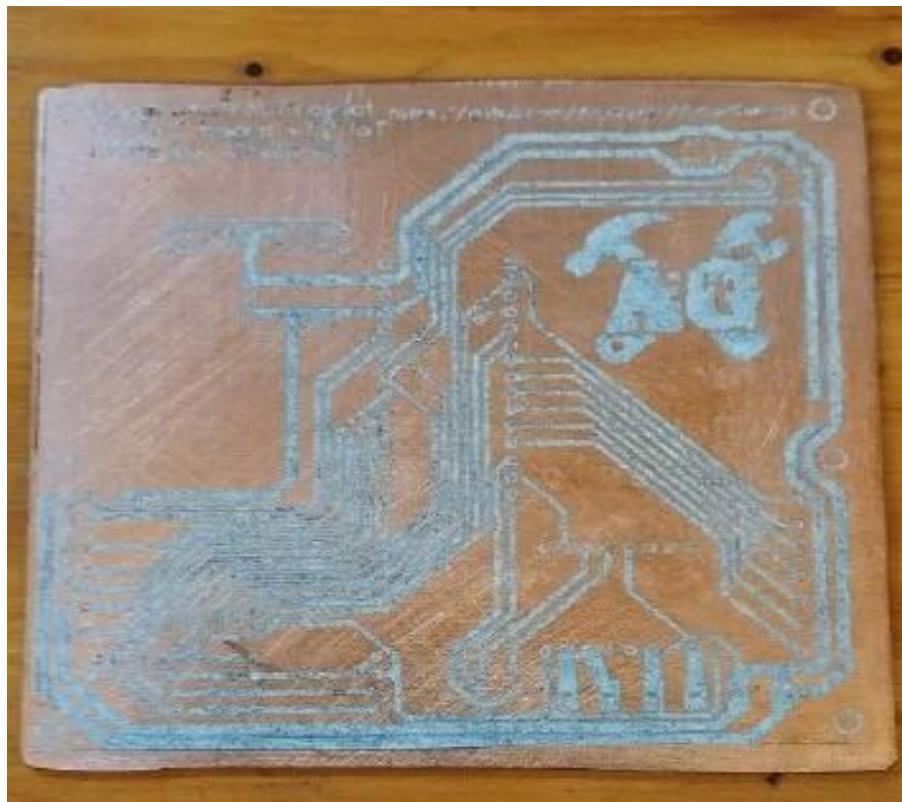
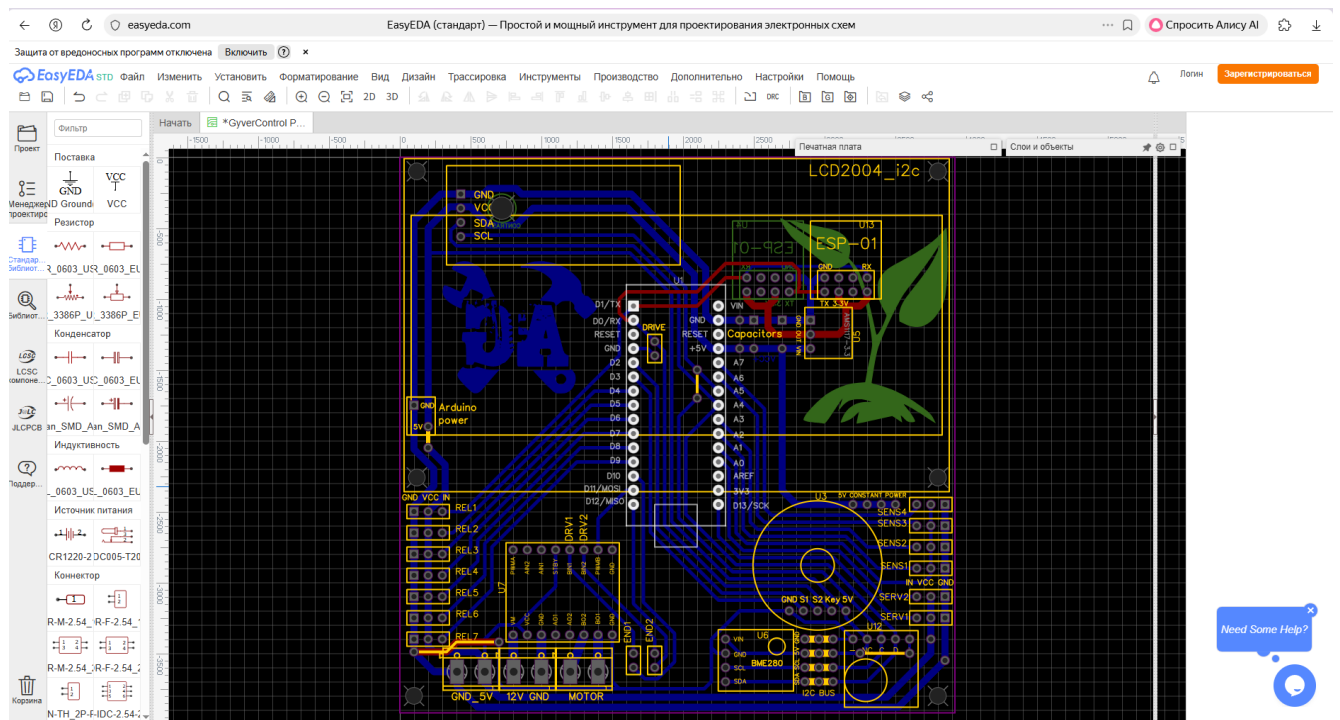
Драйвер TB6612FNG



4-канальный релейный модуль 5В 10А/250В



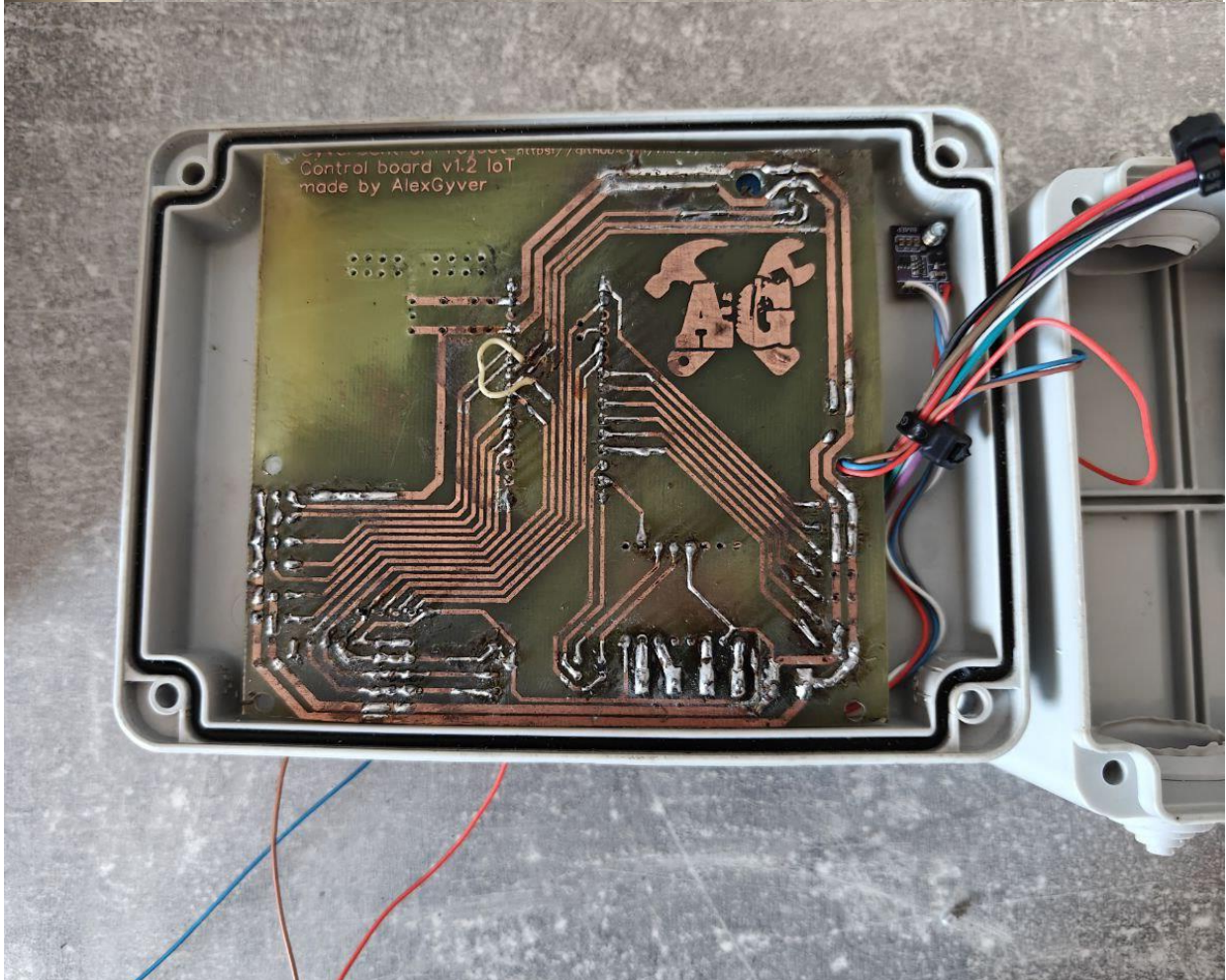
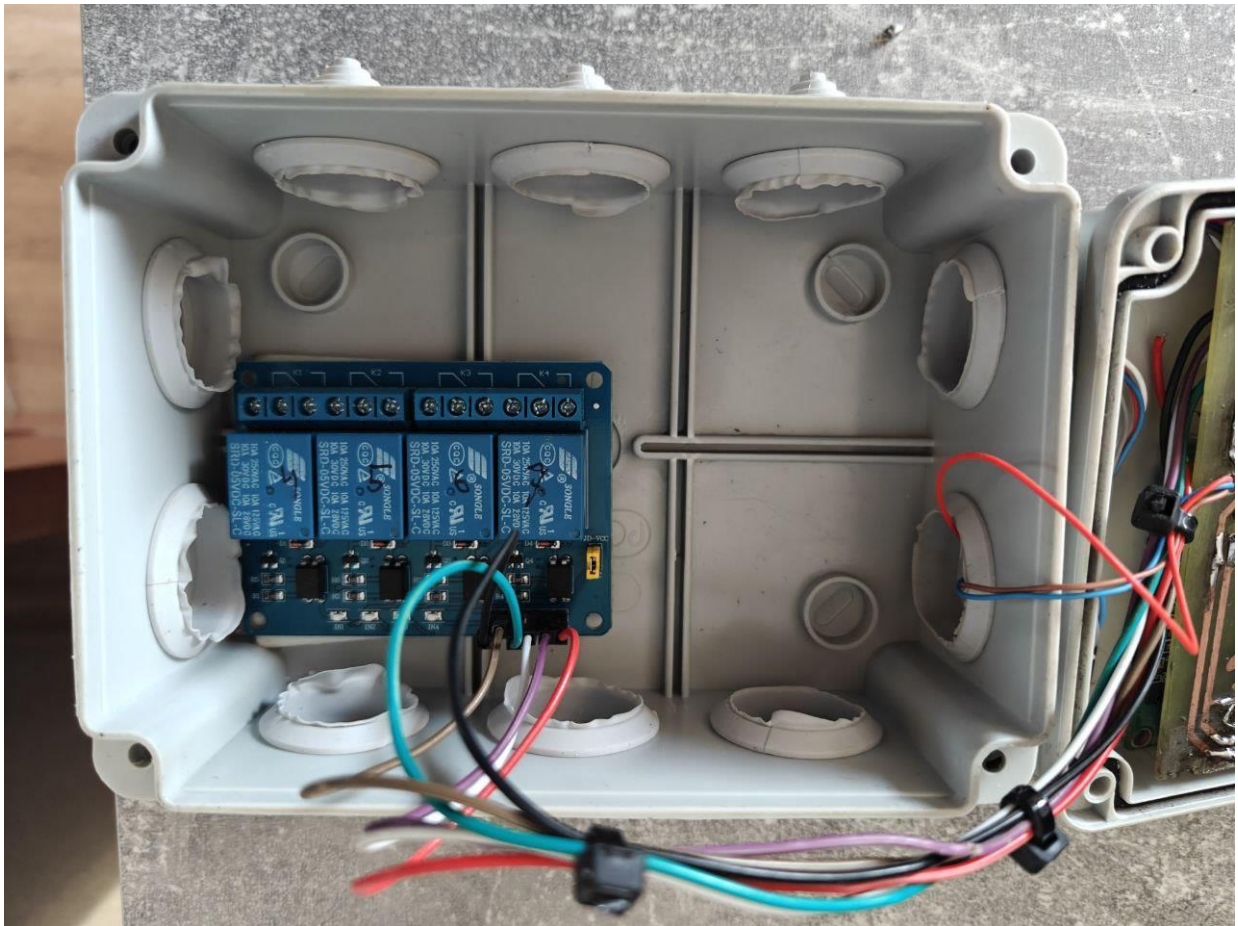
Распределительная коробка

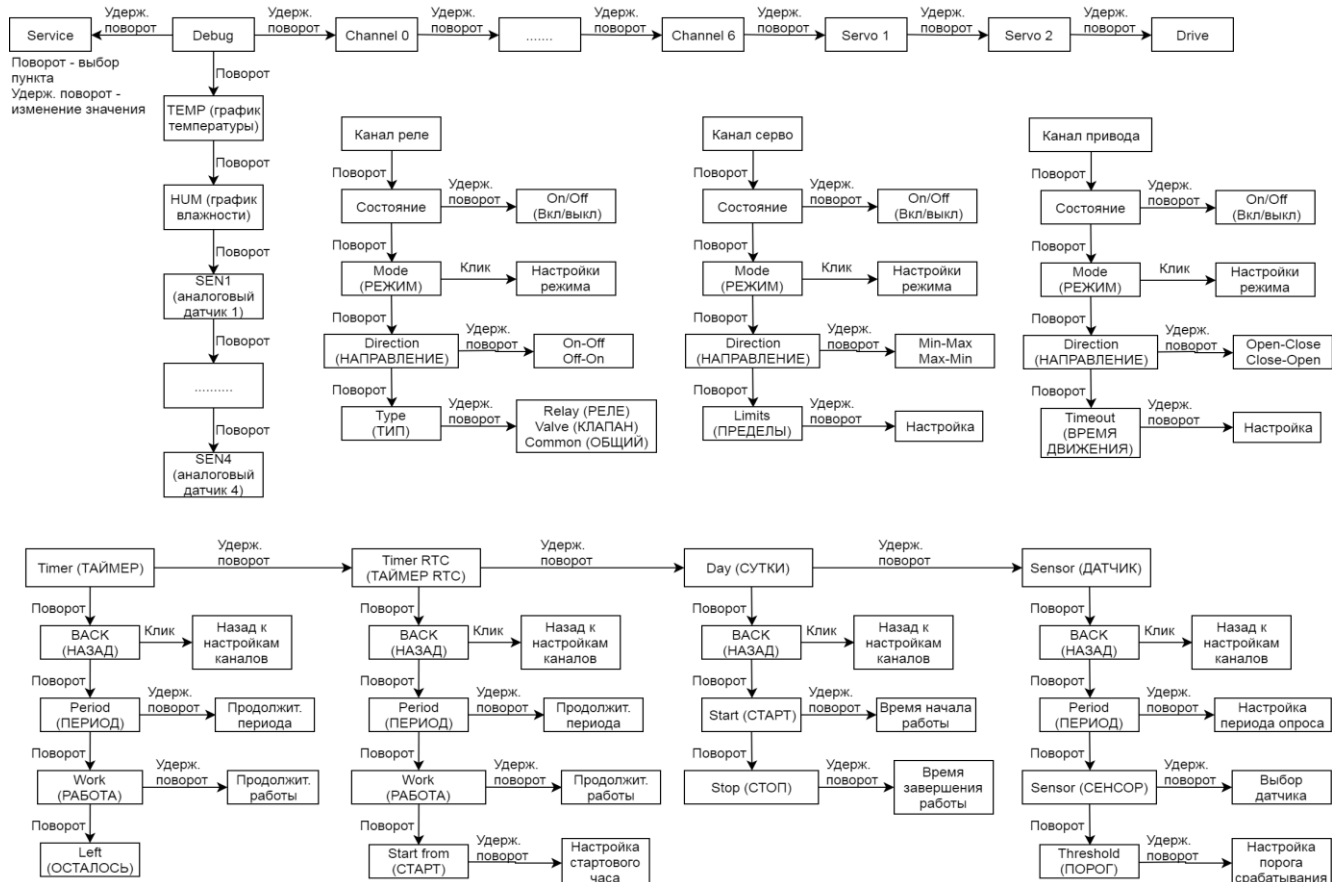
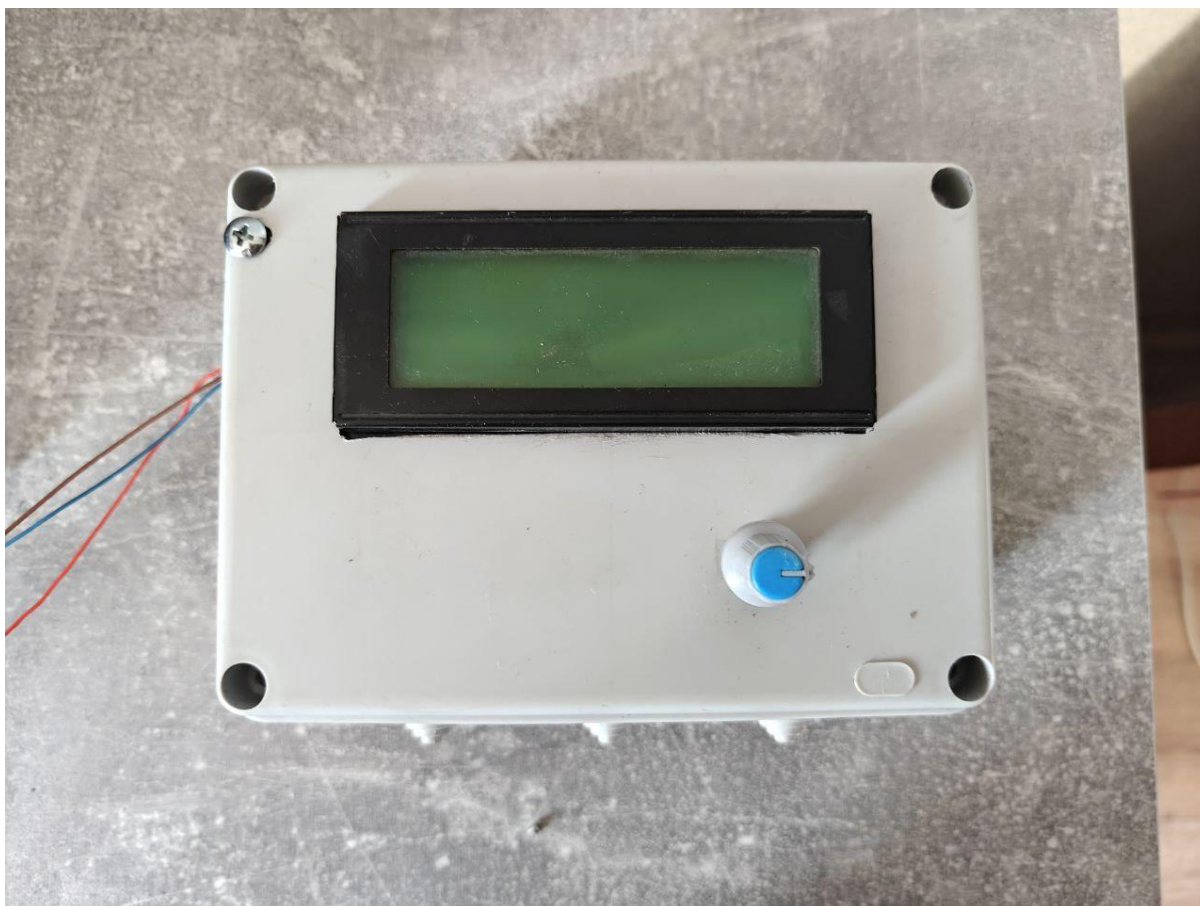


```

9 // версия 1.6
10
11 // ===== СИСТЕМА =====
12 #define START_MENU 0 // 1 - включить, 0 - отключить стартовое меню (старт с нажатой кнопкой). Там будет сервис и сброс настроек
13 #define USE_PLOTS 0 // 1 - включить, 0 - отключить вывод графиков
14 #define USE_DAWN 0 // включает/отключает поддержку режима РАССВЕТ на каналах 2, 3, серво1, серво2
15 #define USE_DRIVE 0 // включает/отключает поддержку линейного привода (в целях экономии памяти)
16
17 // ===== СЕРВО =====
18 #define SERVO1_RELAY 1 // 1 - заменить серво 1 на реле (ЭКОНОМИТ ПАМЯТЬ). 0 - ничего не делать
19 #define SERVO2_RELAY 1 // 1 - заменить серво 2 на реле (ЭКОНОМИТ ПАМЯТЬ). 0 - ничего не делать
20 #define SMOOTH_SERVO 0 // 1 - использовать плавное управление серво, 0 - обычное (ЭКОНОМИТ ПАМЯТЬ)
21 #define SERVO_MIN_PULSE 500 // минимальный импульс серво (зависит от модели, 500-800)
22 #define SERVO_MAX_PULSE 2500 // максимальный импульс серво (зависит от модели, 2000-2500)
23
24 // ===== ПИД =====
25 #define USE_PID 0 // включает/отключает поддержку ПИД регулятора на каналах 2, 3, серво1, серво2 и привода
26 #define USE_PID_RELAY 0 // добавляет/отключает поддержку ПИД регулятора на каналах 0 и 1 (ШИМ-реле). Требуеt USE_PID для работы!
27 #define SCHEDULE_NUM 0 // меню расписания ПИД, задать количество расписаний (не больше 2!). 0 - включить
28 #define PID_INT_MIN -3000 // ограничение интегральной составляющей, минимум
29 #define PID_INT_MAX 3000 // ограничение интегральной составляющей, максимум
30 #define PID_AUTOTUNE 0 // автокалибровщик ПИД
31 #define DEBUG_PID 0 // 0 - выкл режим отладки ПИД и отладки автокалибровки
32 // другое значение - канал отладки пид (1, 2, 3, 4 - обычные каналы, 5, 6 - серво, 7 - привод)
33 // вывод графиков на компьютер, открой "инструменты/плоттер по последовательному соединению"
34 // для отладки автотюна ставь любое отличное от 0 значение
35 // ПРИ РАБОТЕ ТЮНЕРА ИЛИ ОТЛАДКИ ОТКЛЮЧИТЕ ВСЕ ОТ 1 КАНАЛА! ОН БУДЕТ ВКЛЮЧЕН Т.К. НА НЕМ ПРИВЯЗАНА СВЯЗЬ
36 #define SHOW_INTEGRAL 1 // 1 - показывать график интегральной суммы, 0 - нет
37
38 // ===== УПРАВЛЕНИЕ =====
39 #define ENCODER_TYPE 1 // тип энкодера (0 или 1). Если энкодер работает некорректно (пропуск шагов/2 шага), смените тип
40 #define ENC_REVERSE 1 // 1 - инвертировать направление энкодера, 0 - нет
41 #define CONTROL_TYPE 1 // тип управления энкодером: 0 - удержание и поворот для изменения значения, 1 - клик для входа в изменение, повторный клик для выхода (стрелочка меняется на галочку)
42 #define FAST_TURN 1 // 1 - вкл быстрый поворот
43 #define FAST_TURN_STEP 10 // изменение при быстром повороте
44
45 // ===== РАЗНОЕ =====
46 #define DRIVER_LEVEL 1 // 1 или 0 - уровень сигнала на драйвер/реле для привода
47 #define DRIVE_TOUT 10 // секунд. Время, которое система будет ждать после ручного управления дверью, прежде чем применить режим
48 #define PWM_RELAY_HZ 1 // частота ШИМ для каналов ШИМ-реле, Гц (раз в секунду) можно десятичные дроби (0.1 - период будет 10 секунд)
49 #define SETT_TIMEOUT 100 // таймаут неактивности (секунд) после которого автоматически откроется DEBUG и сохранятся настройки
50 #define WIRE_OVERCLOCK 1 // 1 - увеличить частоту шины i2c до 400 кГц
51 #define LCD_ADDR 0x3f // адрес дисплея - 0x27 или 0x3f. Смени если не работает!
52 #define BME_ADDR 0x76 // адрес BME280 - 0x76 или 0x77. Смени если не работает!
53
54 #define WDT_ENABLE 0 // 1 - включить, 0 - отключить watchdog (только для optiboot)
55 // Ссылка для optiboot: https://github.com/Optiboot/optiboot/releases/download/v8.0/package_optiboot_optiboot-additional_index.json
56
57 // ===== НАЗВАНИЯ КАНАЛОВ =====
58 // (только английские буквы)
59 const char *channelNames[] = {
60     "Channel 1",
61     "Channel 2",
62     "Channel 3",
63     "Channel 4",
64     "Channel 5",
65     "Channel 6",
66     "Channel 7",
67     "Servo 1",
68     "Servo 2",
69     "Drive",
70 };
71
72 // ===== ДАТЧИКИ =====
73 // ----- РН-Z19 -----
74 // входы SENS_1 или SENS_2 - датчик CO2
75 #define USE_CO2 0 // 1 - использовать РН-Z19, 0 - не использовать
76 #define CO2_PIN 1 // 1 или 2, соответственно подключать к SENS_1 или SENS_2 !!!
77 #define CO2_MAX 5000 // диапазон датчика (бывает 2000 и 5000)
78 #define CO2_CALIB 0 // 1 - оставить автокалибровку датчика (для жилых комнат), 0 - выключить (для теплиц)
79
80 // ----- bme280 -----
81 // шина i2c - цифровой датчик температуры и влажности
82 #define USE_BME 0 // 1 - использовать BME280, 0 - не использовать
83
84 // ----- ds18b20 -----
85 // вход SENS1 - цифровой датчик температуры
86 #define DALLAS_SENS1 0 // 1 - ко входу SENS1 подключен ds18b20, 0 - подключен обычный аналоговый датчик
87 #define DALLAS_AMOUNT 1 // количество далласов на ОДНОМ пине
88
89 // для нескольких далласов
90 #define DALLAS_MODE 0 // для нескольких датчиков, СЧИТАТЬ показания как:
91 // 0 - среднее из всех
92 // 1 - наибольшее из всех
93 // 2 - наименьшее из всех
94 #define DALLAS_DISP 0 // ОТОБРАЖЕНИЕ показаний:
95 // 0 - согласно режиму (показывает один результат)
96 // 1 - по очереди выводит показания в одно "окно" SENS1. Номер1, значение1, номер2, значение2...
97
98 // адреса датчиков. Количество должно соответствовать DALLAS_AMOUNT
99 #if (DALLAS_AMOUNT > 1)
100 const uint8_t dsAddress[][8] = {
101     {0x28, 0xff, 0x42, 0x5A, 0x51, 0x17, 0x04, 0xD2}, // адрес датчика 1
102     {0x28, 0xff, 0x53, 0xE5, 0x50, 0x17, 0x04, 0xC3}, // адрес датчика 2
103     {0x28, 0xff, 0x09, 0x80, 0x50, 0x17, 0x04, 0x4D}, // адрес датчика 3
104 }; // по аналогии можно добавить ещё
105 #endif
106
107 // ----- DHT11/DHT22 -----
108 // вход SENS2 - цифровой датчик температуры и влажности, отображается вместо BME280
109 #define DHT_SENS2 0 // 1 - ко входу SENS2 подключен DHT11/DHT22, 0 - подключен обычный аналоговый датчик
110 #define DHT_TYPE DHT22 // тип DHT датчика: DHT11 или DHT22
111
112 // ----- NTC термисторы -----
113 // входы SENS1 - SENS4
114 #define THERM1 0 // 1 - термистор на SENS1, 0 - нет
115 #define THERM2 0 // 1 - термистор на SENS2, 0 - нет
116 #define THERM3 0 // 1 - термистор на SENS3, 0 - нет
117 #define THERM4 0 // 1 - термистор на SENS4, 0 - нет
118
119 #define BETA_COEF1 3435 // температурный коэффициент термистора 1 (см. даташит)
120 #define BETA_COEF2 3435 // температурный коэффициент термистора 2 (см. даташит)
121 #define BETA_COEF3 3435 // температурный коэффициент термистора 3 (см. даташит)
122 #define BETA_COEF4 3435 // температурный коэффициент термистора 4 (см. даташит)
123
124 // =====
125
126 #include "a1_data.h"

```





<https://cloud.mail.ru/public/hYmj/9NQbHENGb> – видео работы с микроконтроллером