Введение

Актуальность темы –

Современный мир стал свидетелем интенсивного развития технологий и автоматизации процессов во всех сферах деятельности. Актуальность темы создания удаленной системы управления переносной мини-теплицей обусловлена растущим интересом к современным технологиям в области сельского хозяйства и увеличением потребности в свежих овощах, ягодах и зелени круглый год. Данная тема может быть полезна как для обычного человека, желающего выращивать свежие продукты в домашних условиях, так и для ресторанов и сельхоз предприятий.

Разработка удаленной системы управления переносной мини-теплицей позволит обычным людям выращивать овощи, ягоды и декоративные растения в любое время года в домашних условиях, что является важным шагом в направлении экологии, автономности и самообеспечения хозяйств, экономии средств на покупку продуктов. Для ресторанов и других предприятий общественного питания, выращивание микрозелени и ягод в мини-теплицах с помощью удаленной системы управления может стать выгодным и экономически эффективным решением, позволяющим сократить затраты на закупку свежих ингредиентов и обеспечить высокое качество блюд.

Одним из основных преимуществ создания удаленной системы управления переносной мини-теплицей является возможность контролировать все процессы выращивания растений удаленно с помощью мобильного устройства или компьютера. Это позволит пользователям получать информацию о температуре, влажности почвы и воздуха, уровне освещенности и других показателях, а также управлять системой полива, освещения и вентиляции из любой точки планеты. В результате, это существенно улучшит процесс выращивания овощей и растений в условиях городской среды и сельского хозяйства, способствуя повышению качества продукции и улучшению экологической ситуации в стране.

Создание удаленной системы управления теплицами также может быть полезна для сельскохозяйственной промышленности в целом, обеспечивая более эффективное и точное управление производственными процессами, а также улучшение качества и урожайности продукции.

Цель работы –

Целью данной работы является создание удаленной системы управления переносной мини-теплицей, которая позволит обеспечить круглогодичное выращивание овощей, ягод и декоративных растений в домашних условиях. Данная система будет представлять собой комплекс из аппаратного и программного обеспечения, которое позволит управлять теплицей из любой точки мира с помощью интернета. Основная задача заключается в создании удобного и эффективного способа выращивания растений в условиях ограниченного пространства и времени.

В рамках работы будут рассмотрены существующие подходы к управлению теплицами и разработана новая система, которая будет основана на использовании современных технологий интернета вещей и удаленного управления. Результатом работы будет готовый прототип системы, который можно будет использовать для реализации проектов в области сельского хозяйства и создания удобных условий для выращивания растений в домашних условиях.

Основные задачи –

1. Создание и разработка базы данных для хранения информации о состоянии теплицы, ее настройках, пользователях системы и производимых измерениях.
2. Создание и разработка архитектуры веб-сервисов для реализации всего необходимого функционала системы, способного выдерживать необходимую нагрузку и обеспечивать заданную задержку.
3. Создание и разработка прототипа теплицы и программирование микроконтроллера, который будет обеспечивать связь между теплицей и веб-сервисами, а также обрабатывать данные с датчиков и поддерживать автономный режим работы.
4. Создание удобного и интуитивно понятного пользовательского интерфейса для использования системы.

Архитектура системы и ее сервисы

1. Разработка архитектуры

В данном разделе будут подробно описаны основные архитектурные решения, принятые при разработке системы, а также факторы, повлиявшие на их выбор.

* 1. «Тонкий клиент»

Учтенные факторы:

1. Ограничения микроконтроллера – микроконтроллеры имеют ограниченный объем памяти и вычислительных мощностей, что сильно ограничивает возможность реализации логики на стороне пользователя.
2. Обновляемость - изменения в функционале системы могут потребовать изменения кода микроконтроллера, для чего каждому пользователю необходимо будет лично подключить теплицу к ПК и перепрошить микроконтроллер.

В связи с данными факторами для системы было принято решение выбрать архитектуру «Тонкий клиент», роль клиента в которой будет исполнять микроконтроллер, а роль сервера – веб-сервисы на виртуальных машинах.

Таким образом, большая часть обработки и хранения данных будет происходить на серверной стороне, функционал микроконтроллера будет ограничен только поддержкой системных функций, отправлением данных с датчиков и исполнением максимально простых команд.

* 1. RESTful API.

Учтенные факторы:

1. Расширяемость – система должна обеспечивать возможность быстрого внесения новой функциональности и изменения уже существующей без необходимости серьезных изменений в архитектуре. Это связанно с широким спектром целевой аудитории – от сельскохозяйственной промышленности до физических лиц.
2. Вариативность ui-клиентов – дальнейшее развитие системы может потребовать создания множества различных ui клиентов для пользователей – сайт, мобильное приложение, desktop приложение, боты для социальных сетей и т.д., в связи с чем серверное приложение должно предоставлять единообразное API для всех клиентов.
3. Сложность тестирования – в процессе работы приложения, входящие данные будут проходить через большое количество операций – приведение к системе СИ, анализ для работы автоматических систем управление, запись в БД и др. Данный фактор серьезно усложняет тестирование системы и создает потребность в возможности тестирования работы на различных этапах обработки данных.

В связи с данными факторами для серверного приложения было принято решение выбрать паттерн архитектуры RESTful API c трехслойной архитектурой.

Таким образом разделение приложения на слои позволит реализовать принципы SOLID, что обеспечит легкую расширяемость функционала и возможность тестирования корректности обработки данных на различных слоях, а слой представления данных организованный в стиле REST – единообразный унифицированный интерфейс для различных ui-клиентов.

* 1. «Диспетчер»

Учтенные факторы:

1. Большой объем данных – для возможности реализации автоматического управления теплицей и фиксации статистических выбросов во входящих данных каждая теплица на постоянной основе и с достаточной частотой должна отправлять результаты измерений параметров среды. Однако, полная обработка и запись данных в БД –трудоемкий процесс, более того, теплица не является объектом критически важной инфраструктуры и для пользователя нет необходимости хранить результаты с той же точностью и частотой, которая нужна для данного функционала.
2. Масштабирование – с увеличением числа пользователей система должна легко масштабироваться для обеспечения заданного уровня задержки управления. Однако создание нескольких экземпляров API может спровоцировать «гонку за ресурсы» за ресурсы БД

В связи с данных факторами было принято решение создание отдельного сервиса между клиентом-микроконтроллером и API – диспетчера. Данный сервис будет получать от API текущую конфигурацию для подключаемого микроконтроллера, генерировать для него команды на основе входящих данных и отправлять API усредненные результаты измерений за определенный период.

* 1. Сайт и «конфигуратор»

В качестве основного ui-клиента для данной системы было принято решение разработать отдельный сайт. Он наиболее универсален, так как изначально доступен для большинства устройств и предоставляет больше возможностей для реализации функционала чем приложения и боты для социальных сетей.

Также существует потребность в создании «конфигуратора» - desktop приложения для первоначальной настройки теплицы. Приложение должно взаимодействовать с подключенной к ПК по USB порту теплицей, давая возможность пользователю настроить используемую ею Wi-Fi сеть и зарегистрировать ее в системе под своими данными.

2. Общая архитектура

В результате проектирования была принята следующая версия архитектуры:



Серверные приложения включают в себя: БД, API, сайт и один или несколько экземпляров диспетчера. Экземпляры сервисов распространяются в виде docker образов и могут располагаться как на нескольких виртуальных машинах, так и на одной, используя разные порты.

Развертывание и тестирование экземпляров приложений автоматизировано по методологии CI/CD.

В процессе разработки диплома для экономии средств все серверные приложения будут располагаться на одной ВМ. В случае дальнейшего развития проекта и его коммерческого использования целесообразно расположить все приложения на разных ВМ для повышения отказоустойчивости и масштабируемости системы.

Приложение «конфигуратор» распространяется через сайт и устанавливается на ПК пользователя вместе с необходимыми для сообщения с МК драйверами.

Общая схема взаимодействия приложений:



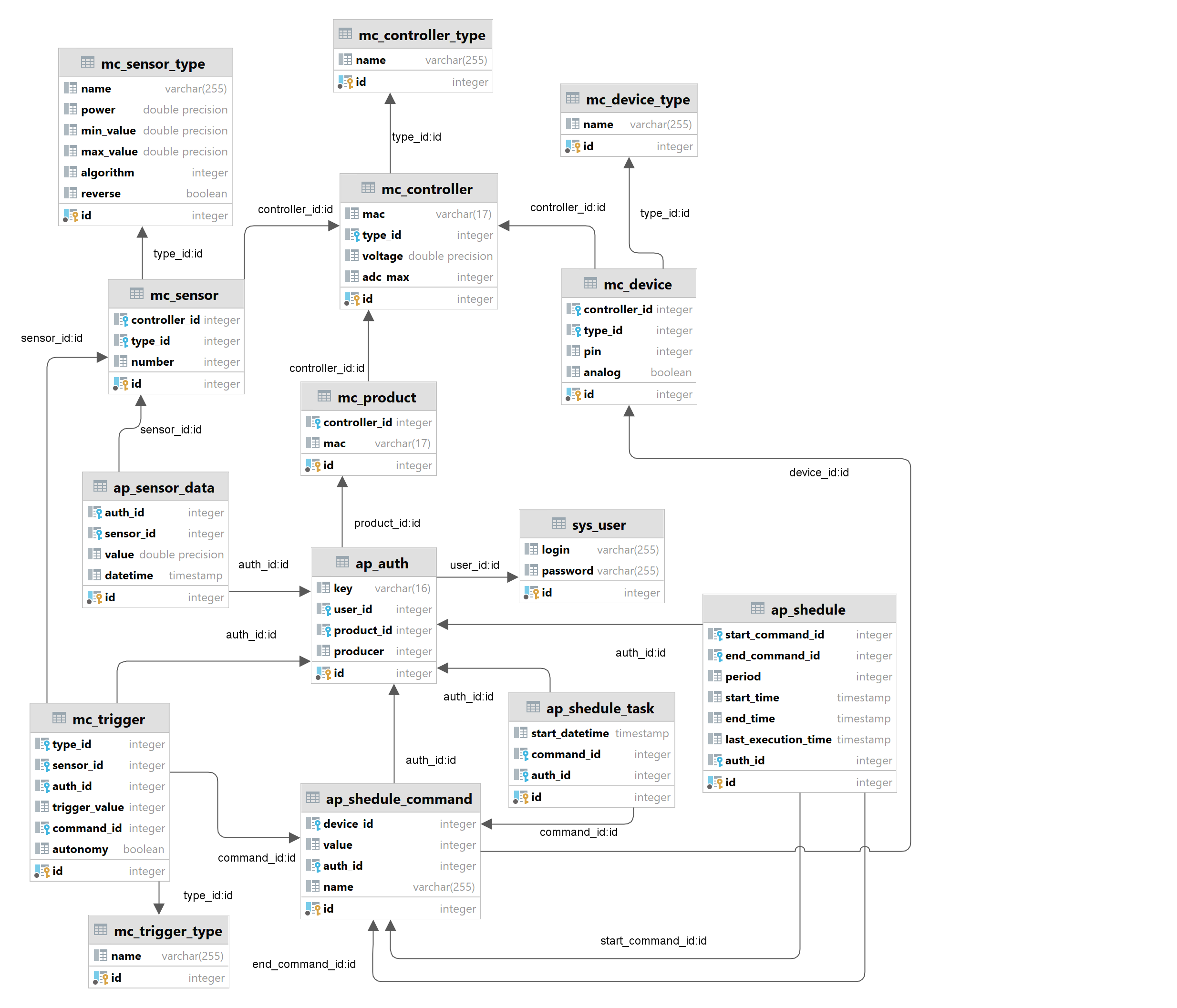
2.1. База данных

Для разработки системы выдвинуты следующие требования:

1. Высокая производительность чтения и записи данных – диспетчер периодически производит как запись усредненных результатов измерений, так и чтение текущей конфигурации контроллера, в связи с чем требуется БД с оптимальной скоростью записи/чтения данных.
2. Поддержка масштабирования и гибкости в добавлении новых таблиц и полей – см 1.2 «Расширяемость»
3. Открытый исходный код – на сегодняшний день в Российской федерации происходит массовый отказ от иностранного лицензионного ПО и переход на использование либо отечественных систем, либо систем с открытым исходным кодом.

Таким образом, с учетом всех вышеперечисленных требований, в качестве основной для реализации системы была выбрана реляционная БД PostgreSQL, однако, планируется сохранить поддержку подключения других похожих БД, таких как MySQL и MS SQL.

Схема БД:



|  |  |  |
| --- | --- | --- |
| MC\_CONTROLLER\_TYPE – модель МК | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Name | VARCHAR | Наименование типа контроллера |

|  |  |  |
| --- | --- | --- |
| MC\_SENSOR\_TYPE – модель датчика | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Name | VARCHAR | Наименование типа датчика |
| Power | DOUBLE | Рабочее напряжение датчика |
| Min\_value | DOUBLE | Минимальное значение датчика в СИ |
| Max\_value | DOUBLE | Максимальное значение датчика в СИ |
| Algorithm | INT | Дискриминатор алгоритма для приведения в СИ |
| Reverse | BOOL | Обратная зависимость возрастания напряжения к результату измерения |

|  |  |  |
| --- | --- | --- |
| MC\_DEVICE\_TYPE – модель устройства | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи, генерируемый системой, начиная с 1 |
| Name | VARCHAR | Наименование типа девайса |

|  |  |  |
| --- | --- | --- |
| MC\_CONTROLLER – версия прошивки МК | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Type\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_CONTROLLER\_TYPE].[ID] |
| Voltage | DOUBLE | Опорное напряжение микроконтроллера |
| Adc\_max | INT | Разрядность АЦП |

|  |  |  |
| --- | --- | --- |
| MC\_SENSOR – датчик | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Controller\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_CONTROLLER].[ID] |
| Type\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_SENSOR\_TYPE].[ID] |
| Number | INT | Порядковый номер датчика |

|  |  |  |
| --- | --- | --- |
| MC\_DEVICE - устройство | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Controller\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_CONTROLLER].[ID] |
| Type\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_SENSOR\_TYPE].[ID] |
| Pin | INT | Номер пина, к которому подключено устройство |
| Analog | BOOL | ШИМ- подключение |

|  |  |  |
| --- | --- | --- |
| MC\_RPODUCT – конкретный прошитый МК | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Controller\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_CONTROLLER].[ID] |
| Mac | VARCHAR(17) | MAC-адрес микроконтроллера |

|  |  |  |
| --- | --- | --- |
| AP\_AUTH – зарегистрированный МК | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Key | VARCHAR(16) | Ключ регистрации |
| User\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [SYS\_USER].[ID] |
| Product\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_RPODUCT].[ID] |
| Producer | INT | Дискриминатор режима работы теплицы |

|  |  |  |
| --- | --- | --- |
| AP\_SENSOR\_DATA – данные с датчиков | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Sensor\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_SENSOR].[ID] |
| Auth\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_AUTH].[ID] |
| Value | DOUBLE | Значение датчика |
| Datetime | DATETIME | Дата и время измерения |

|  |  |  |
| --- | --- | --- |
| SYS\_USER – пользователь системы | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Login | VARCHAR | Логин пользователя |
| Password | VARCHAR | Пароль пользователя |

|  |  |  |
| --- | --- | --- |
| AP\_COMMAND – базовые команды МК | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Name | VARCHAR | Наименование команды |
| Device\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_DEVICE].[ID] |
| Auth\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_AUTH].[ID] |
| Value | INT | Значение устройства |

|  |  |  |
| --- | --- | --- |
| AP\_TRIGGER – триггер для режима работы «по правилам» | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Type\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_TRIGGER\_TYPE].[ID] |
| Sensor\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_SENSOR].[ID] |
| Auth\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_AUTH].[ID] |
| Trigger\_value | INT | Значение срабатывания триггера |
| Command\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [AP\_COMMAND].[ID] |
| Autonomy | BOOL | Включается ли триггер в автономный конфиг |

|  |  |  |
| --- | --- | --- |
| MC\_TRIGGER\_TYPE – тип триггера | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Name | VARCHAR | Наименование типа триггера |

|  |  |  |
| --- | --- | --- |
| AP\_SHEDULE – расписание команд для режима работы «ручной» | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Start\_command\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [AP\_COMMAND].[ID] |
| End\_command\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [AP\_COMMAND].[ID] |
| Start\_time | DATETIME | Время запуска стартовой команды |
| End\_time | DATETIME | Время запуска финальной команды |
| Last\_execution\_time | DATETIME | Дата и время последнего исполнения |
| Auth\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_AUTH].[ID] |
| Period | INT | Кол-во дней между запусками |

|  |  |  |
| --- | --- | --- |
| AP\_SHEDULE\_TASK – запланированные на текущий день команды | | |
| ID | INT | (PK) Первичный ключ — автоинкрементный номер записи |
| Start\_datetime | DATETIME | Время запуска команды |
| Command\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [AP\_COMMAND].[ID] |
| Auth\_id | INT | (FK) Внешний ключ – ссылка на первичный ключ [MC\_AUTH].[ID] |

2.1 Датчики

Для прототипа теплицы было решено использовать следующие датчики:

1. Датчик температуры (терморезистор) – для контроля температуры воздуха внутри теплицы.

Параметры:

Тип данных: аналоговый

Рабочее напряжение: 3.3-6 В

Рабочий ток: 10 мА

Рабочая температура: 0-70 °C

1. Датчик влаги (YL-69) – для контроля влажности почвы.

Параметры:

Тип данных: аналоговый

Рабочее напряжение: 3,3 – 5 В

Рабочий ток: 35 мА

1. Датчик света (фоторезистор) – для контроля уровня освещенности внутри теплицы.

Параметры:

Тип данных: аналоговый

Рабочее напряжение: 3.3-6 В

Рабочий ток: 10 мА

1. Датчик расхода воды (YF-S201) – для контроля полива.

Параметры:

Тип данных: аналоговый, ШИМ

Рабочее напряжение: 3,3 – 18 В

Рабочий ток: 15 Ма

Диапазон работы: 1-30 л/мин

Импульсы на литр: 450

Могут использоваться как цифровые, так и аналоговые датчики. Цифровые подключаются напрямую к пинам МК. Аналоговые подключаются либо через внешний АЦП, либо, при наличии, к пинам МК подключенным к внутреннему АЦП.

2.2 Устройства

Для прототипа теплицы было решено использовать следующие устройства:

1. Фито-лента – для освещения внутри теплицы.

Параметры:

Рабочее напряжение: 5 В DC

Максимальная мощность: 11 Вт

Вид ламп: светодиод

1. Греющий кабель (SRL 16-2) – для обогрева внутри теплицы.

Параметры:

Максимальная мощность: 16 Вт/метр

Рабочее напряжение: 220 В AC

Максимальная температура: 65 °C

1. Кулер – для вентиляции воздуха внутри теплицы.

Параметры:

Количество: 2 шт

Рабочее напряжение: 12 В DC

Максимальная мощность: 0,96 Вт

Скорость вращения: 6000 об/мин

1. Насос (КЗАЭ ЭНЦ 2,5-12) – для полива.

Рабочее напряжение: 12 В

Максимальный ток: 4 А

Давление: 2,5 кг/см2

Расход воды: 25 мл/c

Могут использоваться как цифровые, так и аналоговые устройства. Аналоговые устройства подключаются с помощью реле или транзисторов, устройства, работающие от переменного напряжения, подключаются через MOSFET-транзисторы.

2.3 МК

При разработке системы управления переносной мини-теплицей одним из ключевых компонентов является микроконтроллер, который отвечает за управление устройствами, сбор и обработку данных. При рассмотрении вариантов микроконтроллеров, которые могут быть использованы в данной системе, обязательным требованием будет наличие Wi-Fi модуля стандарта IEEE 802.11, так как разработка на таком МК упрощается и убирает необходимость дополнительно покупать и присоединять сетевой модуль, также поддержка UART для работы конфигуратора и наличие flash памяти для хранения конфигурации.

Для выбора оптимального варианта будем использовать следующие характеристики:

1. Цена – стоимость одного микроконтроллера.
2. Память – количество внутренней памяти.
3. Доступность – возможность приобретения МК в Санкт-Петербурге в течении 7 дней.
4. Количество аналоговых пинов
5. Количество цифровых пинов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| МК | Цена  (usd) | Память  (кб) | Доступность | Аналоговые пины | Цифровые пины |
| ESP8266 | 2-3 | 80 | + | 1 | 11 |
| ESP32 | 6-8 | 320 | + | 18 | 36 |
| Raspberry PI | 35-55 | 1024 | + | 0 | 40 |
| STM32WB | 4-8 | 256 | - | 10 | 36 |

ESP8266 – 5 баллов

ESP32 – 9 баллов

Raspberry PI – 7 баллов

STM32WB – 7 баллов

Таким образом, был выбран микроконтроллер ESP32 - он соответствует требованиям проекта, имеет встроенный Wi-Fi модуль и поддерживает нужный функционал для создания удаленной системы управления.

ESP32 - это микроконтроллер с двумя ядрами, основанный на архитектуре Xtensa LX6, с тактовой частотой до 240 МГц, имеющий встроенный Wi-Fi стандарта 802.11 и Bluetooth v4.2, 520 Кб ОЗУ, 4 Мб flash-памяти, 34 цифровых входа/выхода (GPIO) и 18 аналоговых входов, а также поддержку многих периферийных устройств, таких как SPI, I2C, UART и других. ESP32 использует стандарт языка C++ 11.

Функциональные требования

|  |  |
| --- | --- |
| FR-1 | Подключение к сети |