ВВЕДЕНИЕ

Сфера использования устройств на базе микроконтроллеров и микропроцессоров постоянно расширяется. Микроконтроллеры применяются в различных механизмах и устройствах: автомобили, светофоры, промышленной оборудование, системы безопасности и многое другое. Средства автоматизации не обошли и сферу сельского хозяйства. Многие растения и культуры требуют особого ухода и условий выращивания, например, контроль окружающей температуры, контроль влажности почвы, контроль освещённости и проветривания.

Темой данного курсового проекта было выбрано микропроцессорное устройство мониторинга окружающей среды в теплице. Главной целью умной теплицы является контроль параметров среды, в которой находится растение: автоматическое доведение их до нормы или оповещение о выходе параметров за пределы нормы.

Основные функции подобных устройств – это контроль и доведение до нормы параметров освещённости растения, контроль и доведение до нормы параметров температуры окружающего воздуха, контроль и доведение до нормы параметров влажности почвы, контроль и доведение до нормы параметров влажности окружающего воздуха, а также контроль и доведение до нормы параметров химического состава воздуха (для особенно прихотливых культур).

Разрабатываемое устройство решает ряд поставленных задач: по контролю и доведению до состояния нормы освещённости и влажности почвы, а также контролю за температурой окружающей среды. С целью совместимости устройства с разными видами культур была создана возможность гибкой настройки параметров.

1 ОБЗОР ЛИТЕРАТУРЫ

1.1 Состав устройства

Как было упомянуто ранее, разрабатываемое должно решать поставленные задачи по мониторингу за параметрами окружающей среды растения и при необходимости доведения эти параметров до состояния нормы, а также иметь возможность настройки данных параметров с целью универсальности и совместимости оборудования с разными видами культур. Для решения вышеперечисленных задач проектируемое устройство должно обладать микроконтроллером, датчиком влажности почвы, датчиком температуры, датчиком уровня освещённости, световой индикацией. Кроме датчиков в состав устройства должны входить модули для манипуляции блоками доведения параметров окружающей среды до состояния нормы: модуль управления водяным насосом и модуль управления освещением.

1.2 Обзор микроконтроллеров

При выборе микроконтроллера для текущего проекты были рассмотрены микроконтроллеры Arduino и STM (см таблицу 1.1)

Таблица 1.1 – Сравнение микроконтроллеров

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | Arduino Nano | Arduino Mega | STM32f407-disc1 |
| Напряжение питания, В | 7-12 | 7-12 | 5 |
| Ток потребления, мА | 20 | 66 | 40 |
| Flash-память, Кб | 32 | 256 | 1024 |
| Частота, МГц | 32 | 32 | 168 |
| Разрядность, бит | 8 | 8 | 32 |
| Количество интерфейсов UART | 1 | 4 | 6 |
| Количество интерфейсов I2C | 1 | 2 | 3 |
| Количество интерфейсов SPI | 1 | 1 | 3 |

Для анализа параметров вышеупомянутых микроконтроллеров были изучены следующие источники [12,13].

Базовые навыки программирования на Arduino были получены вследствие изучения следующей литературы [14,15]. В вышеупомянутых источниках были изучены способы использования как аналоговых, так и цифровых выходов, способы подключения и обмена данными с внешне подключаемыми модулями. В качестве микроконтроллера в данном проекте выбрана плата Arduino Nano (рис. 1.2.1).



Рисунок 1.2.1 – Arduino Nano

Также для данного микроконтроллера создано большое количество пользовательских библиотек для упрощения работы с периферийными компонентами. Для курсового проекта было принято решение использовать некоторые библиотеки для раскрытия полного функционала некоторых модулей [16-18].

1.3 Литература о датчиках

Датчик – средство измерений, предназначенное для выработки сигнала измерительной информации в форме, удобной для передачи, дальнейшего преобразования, обработки и (или) хранения. Для подключения датчиков могут использовать различные проводные интерфейсы. Среди цифровых интерфейсов можно выделить наиболее используемые такие как I2C, SPI, 1-WARE. Некоторые определяют зависимость измеряемой величины от аналогового сигнала на выходе. Для снятия показаний с таких датчиков необходимо оцифровывать аналоговый сигнал с помощью АЦП. АЦП может как входить в состав самого датчика и быть оформленным отдельной микросхемой, так и входить в состав переферии микроконтроллера, в таком случае аналоговый сигнал датчика можно напрямую подавать на выводы микроконтроллера.

Датчик света – модуль, способный реагировать на изменение уровня освещённости. Зачастую в основе работы таких датчиков используются различные фотоэлементы. В этом проекте в качестве такого фотоэлемента используется фоторезистор (рис 1.3.1). Данные получены из источников [2,5] (см. таблицу 1.2).

Таблица 1.2 – Параметры датчика освещённости

|  |  |
| --- | --- |
| Параметр | MLG4416 |
| Световое сопротивление (10 лк), кОм | 5-10 |
| Диаметр, мм | 4 |
| Максимальное напряжение, В | 150 |
| Темновое сопротивление, МОм | 1 |
| Длина волны, нм | 560 |
| Рабочая температура, С | -30...+70 |
| Время отклика, мс | 30 |

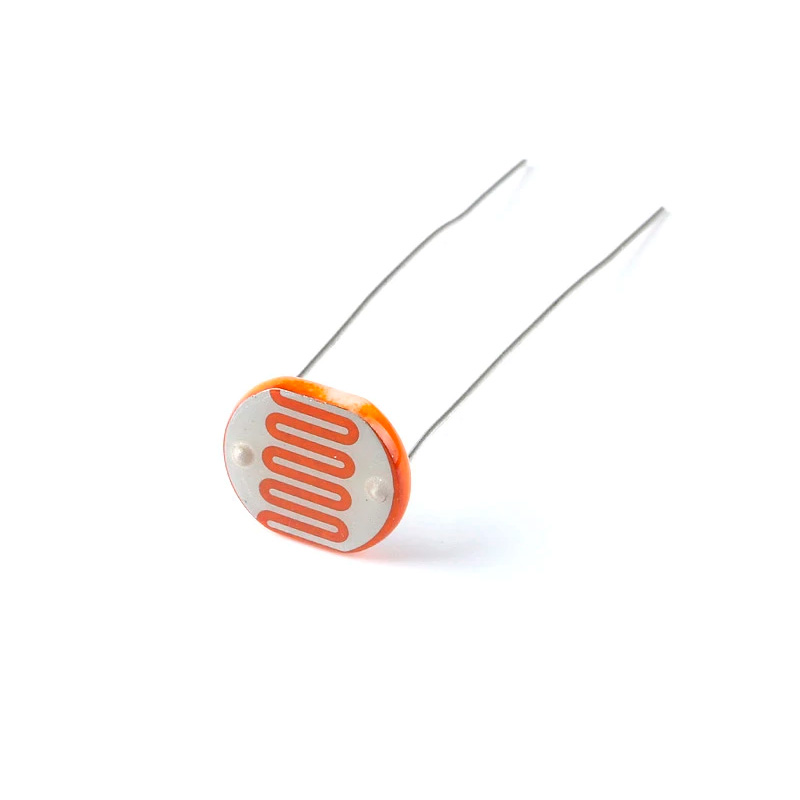


Рисунок 1.3.1 – Фоторезистор

Датчик температуры используется для определения температуры окружающей среды [4,7,8] (см. таблицу 1.3). В данном проекте был задействован датчик DHT22 (рис 1.3.2) передающий данные по цифровому интерфейсу 1-WARE.

Таблица 1.3 – Сравнение датчиков температуры

|  |  |  |  |
| --- | --- | --- | --- |
| Параметр | DS18B20 | DHT11 | DHT22 |
| Напряжение питания, В | 3–5,5 | 3-5 | 3-5 |
| Точность, % | 0,5 | 2 | 1 |
| Температурный диапазон, С | -55–+125 | 0-50 | -40–+80 |
| Потребляемый ток, мА | 9 | <2.5 | <2.5 |
| Измерение влажности воздуха | нет | да | да |

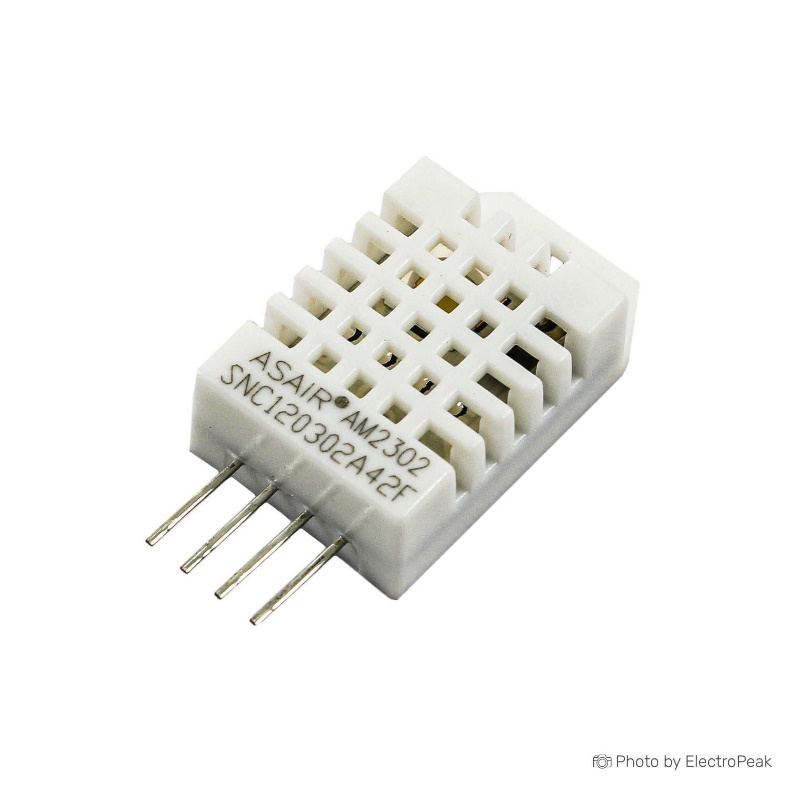


Рисунок 1.3.2 – DHT22

Датчик влажности почвы используется для определения содержания воды в почве для растения. Применительно к этому виду датчиков можно выделить два наиболее используемых принципа определения влажности почвы. Первый принцип заключается в измерении электрического сопротивления почвы. Датчики этого типа просты в исполнении и использовании, на выходе имеют аналоговый сигнал, но имеют существенный недостаток, заключающийся в относительно быстрой коррозии контактных площадок, которые имеют непосредственный контакт с почвой. Такая коррозия оказывает существенное влияние на показания датчика, что делает его недолговечным и не практичным. Второй принцип заключается в измерении ёмкостных параметров почвы. Датчики этого типа более сложные в исполнении, и требуют более тщательного подбора параметров схемы. Однако так же просты в использовании, поскольку на выходе формируют аналоговый сигнал и лишены главного недостатка резистивных датчиков – коррозии контактных площадок, поскольку контактные площадки могут быть покрыты защитным слоем. В этом проекте был задействован именно ёмкостный датчик влажности почвы CSMS-1.2 (рис 1.3.3). Параметры получены из литературы [9,10] (см. таблицу 1.4).

Таблица 1.4 – Сравнение датчиков пламени

|  |  |
| --- | --- |
| Параметр | CSMS-1.2 |
| Напряжение питания, В | 5 |
| Глубина погружения в почву, мм | 65-80 |
| Выходное напряжение, В | 3-1.75 |
| Потребляемый ток, мА | <5.3 |

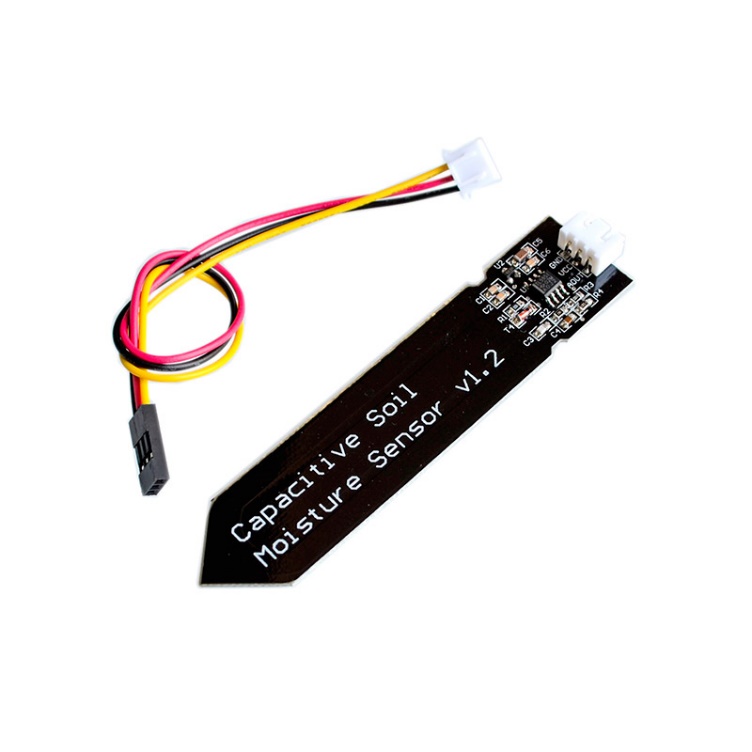


Рисунок 1.3.3 – CSMS-1.2

1.4 Литература о водяном насосе

В качестве водяного насоса в данном проекте была задействована небольшая погружная помпа. Параметры помпы представлены в таблице 1.5.

Таблица 1.5 – Параметры пьезодинамика

|  |  |
| --- | --- |
| Параметр | JT-DC3L |
| Напряжение питания, В | 3,5–6 |
| Потребляемый ток, мА | 180 |
| Производительность, л/мин | 1.7 |



Рисунок 1.4.1 – Водяная помпа

1.5 Литература о освещении для растений

Для обеспечения необходимого уровня освещённости растениям требуются дополнительные источники света. В качестве источников света целесообразно выбрать экономичные светодиодные лампы. Для обеспечения активного роста растения не подойдут обыкновенные лампы, используемые в быту. Растения наиболее активно поглощают световое излучение ультрафиолетового спектра, поэтому необходимо использовать источники света, длинна волны которых находится преимущественно в ультрафиолетовом спектре. В данном проекте задействована лампа, изображённая на рис 1.5.1.



Рисунок 1.5.1 – Фитолампа

2 РАЗРАБОТКА СТРУКТУРЫ УСТРОЙСТВА

2.1 Постановка задачи

Перед тем как составить структуру разрабатываемого устройства, необходимо выделить функции, которые будет выполнять устройство.

Необходимо разработать микропроцессорное устройство контроля жизненного цикла растений. Функции, выполняемые данным устройством:

- контроль влажности почвы;

- автоматическое включение и отключение системы полива;

- считывание значение температуры окружающей среды на выход за пределы допустимых значений;

- контроль уровня освещённости;

- автоматическое включения и отключение дополнительного освещения;

- световая индикация состояния устройства.

-программная конфигурация параметров устройства через интерфейс персонального компьютера.

Управление устройством включает возможность ручной установки пределов включения освещения, пределов включения и отключения системы автоматического полива, а также критических пределов температуры.

2.2 Определение компонентов структуры устройства

Компоненты структуры устройства выбираются исходя из функций, определенных в постановке задачи. Были определены следующие компоненты (см. приложение А).

1) Датчик температуры считывает температуру окружающей среды и передает значение на микроконтроллер.

2) Датчик влажности почвы считывает значение влажности почвы и передает значение на микроконтроллер.

3) Датчик света определяет уровень освещённости растения и передает значение на микроконтроллер.

4) Модуль световой индикации сообщает пользователю о соблюдении температурного режима растения.

5) Модуль коммутации помпы служит для управления помпой.

6) Помпа служит для переливания жидкости.

7) Микроконтроллер – ключевой компонент всей схемы. Выполняет функцию обработки поступаемой информации и выставления управляющих сигналов.

8) Модуль коммутации освещения служит для передачи сигналов микроконтроллера о включении и отключении освещения.

9) Модуль освещения обеспечивает растение светом в отсутствие источников естественного освещения.

2.3 Взаимодействие компонентов устройства

Микроконтроллер производит периодические запрос на считывание данных с датчиков, проверяет полученные данные на выход за пределы допустимых значений параметров и перезаписывает считанные параметры во флэш-память Arduino.

При обнаружении отклонений от нормы и/или сигнала с модуля управления микроконтроллер отдает сигнал модулю индикации для светового оповещения и модулям коммутации о включении систем полива и дополнительного освещения.

Дополнительно микроконтроллер производит проверку входного буфера UART. При наличии в нём данных, выполняется обработка и сохранение либо отправка необходимых данных, таких как порог включения системы автоматического полива, порог отключения системы автоматического полива, порог включения дополнительного освещения, а также значение критической температуры.

3 ОБОСНОВАНИЕ ВЫБОРА УЗЛОВ, ЭЛЕМЕНТОВ ФУНКЦИОНАЛЬНОЙ СХЕМЫ

* 1. Обоснование выбора микроконтроллера

Микроконтроллер является главным модулем разрабатываемого устройства, потому что берет на себя главные функции: управление всеми внешне подключёнными модулями, а также производить обработку поступаемых с датчиков данных.

В таблице 1.1 были продемонстрированы ключевые параметры устройств. Микроконтроллеры STM32F407 и Arduino Mega опережают Arduino Nano по многим параметрам, но было принято решение о достаточности параметров Arduino Nano для выполнения поставленных задач. Также данный микроконтроллер обладает большим количеством пользовательских библиотек для различных сценариев использования данного устройства.

* 1. Обоснование выбора подключаемых модулей

В качестве датчика освещённости был выбран обычный фоторезистор. На его основе можно реализовать обыкновенный делитель напряжения и подать выходной сигнал напрямую на микроконтроллер. Минусом такого решения является отсутствие аппаратного гистерезиса показаний датчика, однако данный функционал можно реализовать программно.

Датчиком температуры был выбран DHT22 потому, что данный датчик позволяет считывать показания температуры в достаточно широких диапазонах значений и делать это с достаточной точностью (см. таблицу 1.3). К минусам датчика можно отнести низкую скорость обновления показаний, однако, предполагая, что воздух – достаточно инерционная система, можно сделать вывод что этой скорости будет достаточно. Кроме того, датчик имеет возможность измерения влажности воздуха, что может оказаться полезным в дальнейшем развитии и улучшении устройства.

При выборе датчика влажности почвы было решено использовать CSMS-1.2. Это ёмкостный датчик, к преимуществам которого можно отнести простоту интерфейса подключения к микроконтроллеру: он выдаёт обыкновенный аналоговый сигнал, а также долговечность, поскольку ёмкостные контактные площадки ёмкостных датчиков не имеют прямого контакта с почвой.

В качестве водяного насоса используется обыкновенная помпа, параметры которой перечислены в подразделе 1.4 поскольку её характеристик вполне достаточно для реализации требуемого функционала.

Для коммутации питания дополнительного освещения был выбран модуль реле, поскольку в его составе есть специальный драйвер для коммутации катушки реле. Прямое подключение реле к микроконтроллеру может вывести последний из строя. Выбранный модуль можно увидеть ниже (рис 3.2.1).

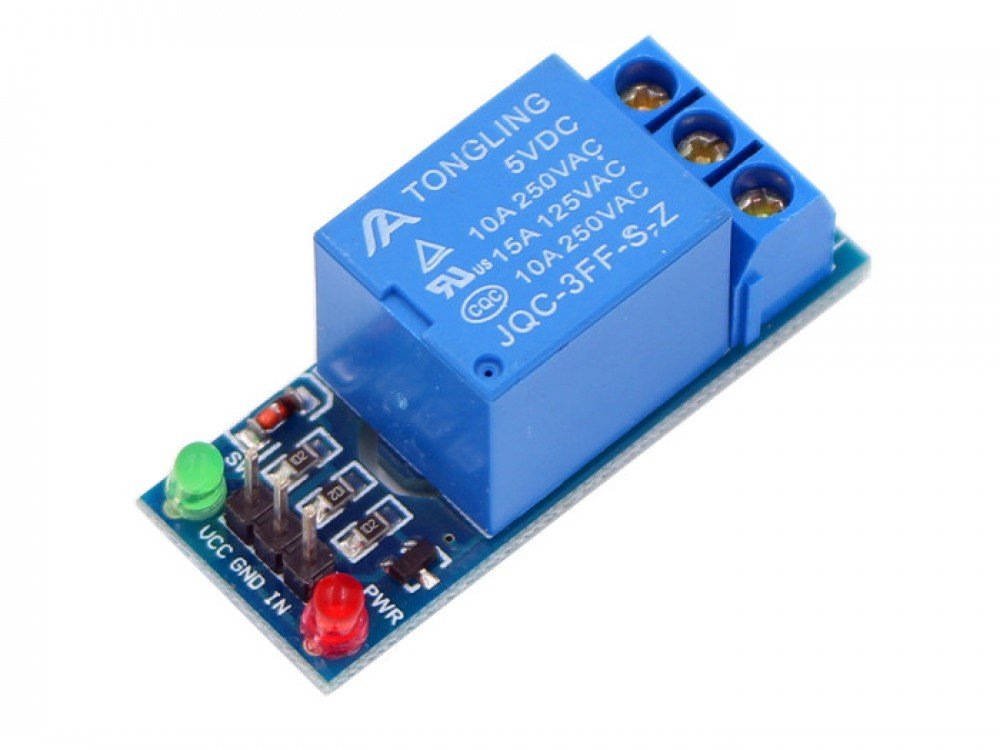


Рисунок 3.2.1 – Модуль реле

В модуль световой индикации входит два светодиода, красного и зелёного света.

* 1. Расчет питания

Перед выбором источника питания был проведен расчет потребляемой мощности каждого модуля (см. таблицу 1.6).

Таблица 3.1 – Расчет потребляемой мощности

|  |  |  |  |
| --- | --- | --- | --- |
| Имя устройства | Напряжение питания, В | Максимальный ток потребления, мА | Мощность, мВт |
| Arduino Nano | 5 | 20 | 100 |
| Водяная помпа | 5 | 180 | 900 |
| Модуль реле | 5 | 100 | 500 |
| CSMS-1.2 | 5 | 5.5 | 27.5 |
| DHT22 | 5 | 2.5 | 12.5 |
| Модуль световой индикации | 5 | 20 | 100 |
| Суммарная мощность, Вт | | | 1.6 |

Для правильного функционирования схемы надо обеспечить каждое из устройств стабильным напряжением номиналом 5В.

Питание устройства осуществляется от любого адаптера с выходным напряжением 5В или USB порта компьютера. Питание для всех датчиков расходится по плате с контактов Arduino.

* 1. Взаимодействие узлов

В приложении Б продемонстрирована функциональная схема устройства.

Датчик влажности почвы подключен к микроконтроллеру Arduino Nano и передает на микроконтроллер аналоговый сигнал обратно пропорциональный содержанию воды в почве растения.

Датчик освещения подключен к микроконтроллеру Arduino Nano и передает на микроконтроллер аналоговый сигнал пропорциональный уровню освещённости в помещении.

Датчик уровня температуры подключен к микроконтроллеру Arduino Nano и по запросу со стороны микроконтроллера передает данные о температуре окружающей среды по цифровому каналу использую протокол 1-Wire.

Модуль световой индикации подключен к микроконтроллеру Arduino Nano и по управляющему сигналу включает нужный светодиод.

Модуль коммутации помпы усиливает сигнал с микроконтроллера, делая его достаточно мощным для управления водяной помпой.

Модуль коммутации освещения подключен к микроконтроллеру и коммутирует сетевую линию фитоламп.

4 РАЗРАБОТКА ПРИНЦИПИАЛЬНОЙ ЭЛЕКТРИЧЕСКОЙ СХЕМЫ

В приложении В приведена функциональная схема программы.

4.1 Разработка схемы измерения освещённости

В роли датчика освещённости выступает фоторезистор, соответственно для измерения уровня освещения имеет смысл собрать на его основе делитель напряжения (рис 4.1.1).

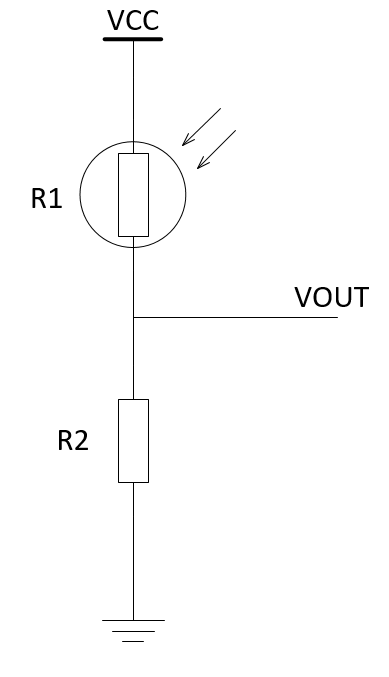


Рисунок 4.1.1 – Резистивный делитель на основе фоторезистора

Выходное напряжение делителя можно напрямую подать на вход микроконтроллера. Встроенный АЦП преобразует это напряжение в пригодное для цифровой обработки.

Особое внимание стоит уделить подбору резистора R2 поскольку от его номинала зависит диапазон выходного напряжения и, соответственно, точность измерений.

Поскольку номинал R1 соответствует 10 кОм, значение R2 было выбрано 4.7 кОм.

4.2 Расчет нагрузки светодиодов

В качестве световой индикации используются два светодиода разных цветов, подключенные к цифровым выходам платы по приведенной ниже схеме (см. рисунок 4.2.1).



Рисунок 4.2.1 – Схема подключения светодиода

Для ограничения тока, поступаемого на светодиод, к ножке анода подсоединяют последовательно резистор, номинал которого рассчитывается по след формуле:



где *U*Н.П – напряжение питания, *U*Д – напряжение падения на светодиоде, *I*Д – ток потребления светодиода.

В проекте используются светодиоды зеленого и красного цветов, со следующими параметрами: *I*Д = 20 мА. *U*Д = 2.3 В.

Однако из даташита на микроконтроллер Atmega328P, входящего в состав Arduino Nano, можно узнать, что максимальный выходной ток пинов GPIO соответствует 20 мА. Исходя из этого целесообразно уменьшить ток, протекающий через светодиод, тем самым предотвратить выход из строя порта микроконтроллера. Предполагается ограничить ток на уровне 10мА.

Получаем:



4.3 Разработка схемы коммутации помпы

Помпа, как и любая нагрузка, не может подключаться к микроконтроллеру напрямую, поскольку может вывести микроконтроллер из строя. Соответственно для включения и выключения помпы требуется схема коммутации. Среди наиболее популярных схем можно выделить схемы на основе полевых транзисторов. Полевые транзисторы управляются потенциалом на затворе, что снижает нагрузку на пины микроконтроллера, а также малое сопротивление открытого канала позволяет максимально снизить потери в схеме коммутации. Типовая схема коммутации нагрузки на основе полевого транзистора приведена ниже (рис 4.3.1).

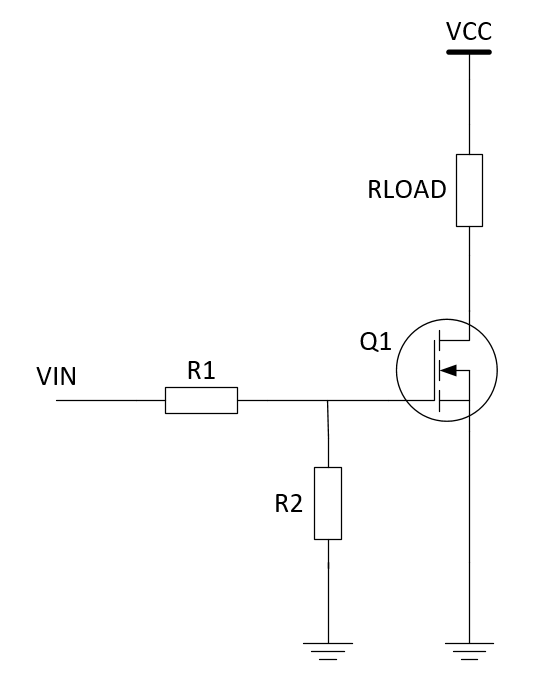


Рисунок 4.3.1 – Схема коммутации на полевом транзисторе

Резистор R1 служит для ограничения тока заряда затвора в момент открытия транзистора, поскольку полевой транзистор имеет паразитную ёмкость на затворе. R2 служит для более устойчивого управления транзистором: надёжно закрывает его в момент, когда на схему подано напряжение, однако выходные пины микроконтроллера ещё не сконфигурированы, а так же ускоряет закрытие транзистора, сливая заряд затвора на исток. RLOAD – коммутируемая нагрузка, в данном случае помпа.

В качестве ключа для этой схемы был выбран транзистор IRFZ24N в корпусе ТО-220 (рис 4.3.2).



Рисунок 4.3.2 – Транзистор IRFZ24N в корпусе ТО-220

Его электрический параметров хватает с запасом для выполнения возложенных задач.

4.4 Разработка печатной платы

Печатная плата для аппаратной части была разработана в специализированной программе Altium Designer (рис 4.4.1). Все дорожки имеют толщину 1 мм, кроме тех, через которые будет проходить питание помпы. Полигоны питания занимают максимальную площадь для уменьшения расхода раствора для травления и увеличения скорости травления.

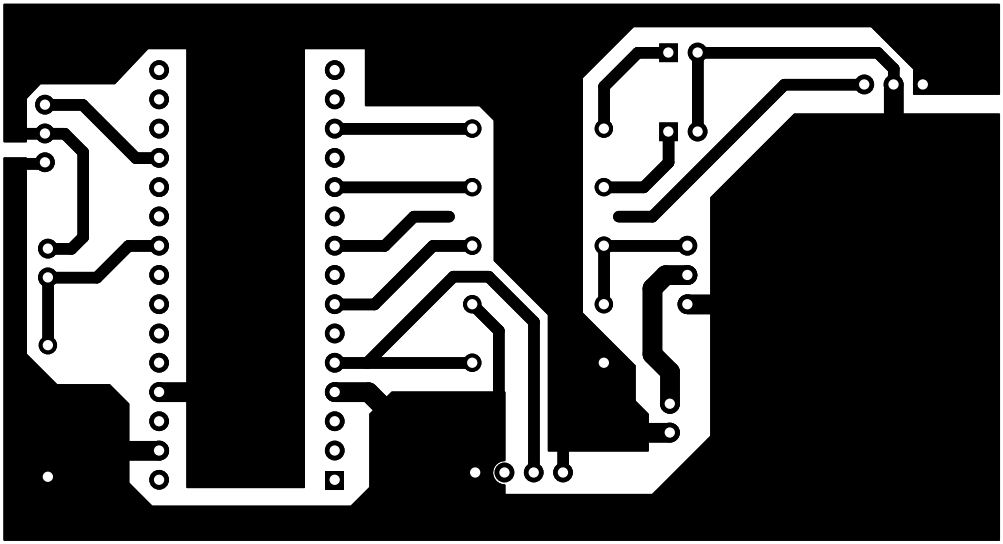


Рисунок 4.4.1 – Печатная плата устройства

5 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

5.1 Требования к программе

Разрабатываемая программа должна позволять выполнять все функции, указанные в п. 2.1. Кроме всего этого программная часть проекта делиться на две составляющие: прошивка микроконтроллера и графический интерфейс для персонального компьютера, который служит для настройки параметров устройства.

Прошивка микроконтроллера должна реализовывать следующий функционал:

1) Калибровка всех датчиков.

2) Считывание показание с датчика температуры.

3) Считывать показаний с датчика влажности почвы.

4) Считывание значения с датчика освещённости.

5) Автоматическое включение и выключение системы освещения.

6) Автоматическое включение и выключение системы полива.

7) Световое оповещение о статусе параметров датчиков.

8) Обработка данных, пришедших с персонального компьютера, по выбранному протоколу.

Графический интерфейс для ПК должен соответствовать следующим требованиям:

1) Автоматическое подключение к устройству и проверка дальнейшего подключения на отсутствие.

2) Отображение текущего состояния подключения.

3) Отображение текущего значения температуры.

4) Отображение текущего значения критической температуры.

5) Возможность ручной установки значения критической температуры.

6) Возможность ручной установки границ включения и отключения полива.

7) Возможность ручной установки границы включения освещения.

8) Обработка и отправка данных на устройство в соответствии с выбранным протоколом.

**5.2 Разработка протокола для передачи информации от ПК к устройству**

Разрабатываемое устройство должно работать в двух режимах: одиночный режим, в котором происходит устройство выполняет функции по контролю окружающей среды растения, и режим конфигурации, в котором устройство выполняет всё те же функции, но дополнительно появляется возможность конфигурации устройства, при этом в концепции подключения к ПК устройство является ведомым, а ПК ведущим. Протокол должен соответствовать простейшей структуре и организовывать считывание и запись данных в регистры устройства. В каждом пакете должен содержаться заголовок, адрес нужного регистра и данные соответственно. Разрядность АЦП микроконтроллера составляет 10 бит, а для значений с датчика температуры в данном проекте достаточно диапазона 0-50 градусов Цельсия. Точность измерений датчика температуры составляет 0.1 градуса. Таким образом данные со всех датчиков можно уместить в двухбайтное число, соответственно поле данных в пакетах протокола составляет два байта. Адресное пространство устройства условно поделено на регистры, в которых находятся данные. Структура пакета для записи данных на устройство показана ниже (см таблицу 5.1). Структура пакета для отправки данных с устройства на компьютер аналогична.

Таблица 5.1 – Структура пакета для записи данных на устройство

|  |  |  |  |
| --- | --- | --- | --- |
| Байт 1 | Байт 2 | Байт 3 | Байт 4 |
| Заголовок | Адрес | Младший байт данных | Старший байт данных |

Структура пакета для чтения данных с устройства показана ниже (таблица 5.2).

Таблица 5.1 – Структура пакета для записи данных на устройство

|  |  |
| --- | --- |
| Байт 1 | Байт 2 |
| Заголовок | Адрес |

Адреса регистров перечислены ниже (таблица 5.3).

Таблица 5.3 – Адреса регистров устройства

|  |  |
| --- | --- |
| Адрес (HEX) | Данные |
| 0x30 | Уникальный идентификатор устройства |
| 0x04 | Текущее значение температуры воздуха |
| 0x08 | Текущее значение критической температуры |
| 0x15 | Граница включения полива |
| 0x17 | Граница отключения полива |
| 0x42 | Граница включения освещения |

Стоит отметить что отправка данных в некоторые регистры не записывает данные из пакета в регистр, а вызывает считывание текущих показаний с датчиков в регистры микроконтроллера.

Заголовок пакета определят его назначение и суть содержащихся данных. Возможные заголовки пакетов и их назначение показаны в таблице ниже (таблица 5.4).

Таблица 5.4 – Адреса регистров устройства

|  |  |
| --- | --- |
| Байт заголовка (HEX) | Назначение пакета |
| 0x77 | Пакет для записи данных на устройство |
| 0x72 | Пакет для чтения данных с устройства |
| 0x61 | Ответ устройства на пакет чтения данных |

5.3 Разработка программы для микроконтроллера

В приложении Г приведена схема программы.

Программа состоит из трех основных частей:

- настройка;

- считывание параметров с датчиков;

- анализ считанных данных и соответствующее реагирование;

- обработка входного буфера UART.

Основной концепцией программного обеспечения микроконтроллера является бесконечный цикл, соответственно обработка всех данных происходит именно в нём.

Основным вопросом в этой концепции является считывание показаний с датчика температуры. Из его документации можно увидеть, что обновление показаний датчика происходит каждые две секунды. При подключении устройства к персональному компьютеру, анализ состояния подключения будет происходить переодическим опросом устройства. ПК будет запрашивать регистр уникального идентификатора устройства. Поскольку обработка буфера UART на устройстве происходит также в бесконечном цикле, цикл не должен иметь значительных задержек, или эти задержки должны иметь длительность меньшую чем период опроса устройства.

Из всех функций, находящихся в бесконечном цикле, блокировать бесконечный цикл может только функция считывания показаний с датчика температуры, соответственно для разработки программного обеспечения необходимо выяснить длительность задержки цикла на этой функции. С этой целью была разработана простая программа, позволяющая выяснить длительность этой задержки. Код программы приведён ниже.

#include "DHT.h"

#define DHT\_PIN 2

uint32\_t measurement\_delay;

float temp;

uint32\_t tmr;

DHT dht\_sensor(DHT\_PIN, DHT22);

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

pinMode(DHT\_PIN, INPUT);

}

void loop() {

// put your main code here, to run repeatedly:

if (millis() - tmr > 2000) {

tmr = millis();

measurement\_delay = millis();

temp = dht\_sensor.readTemperature();

measurement\_delay = millis() - measurement\_delay;

Serial.print("delay: ");

Serial.println(measurement\_delay);

Serial.print("temperature: ");

Serial.println(temp);

}

}

Результат работы программы показан ниже (рис 5.3.1).

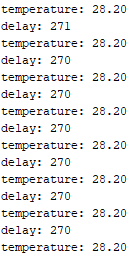


Рисунок 5.3.1 – Результат работы программы по определению задержки считывания показаний с датчика температуры

Задержка указана в миллисекундах. Соответственно период опроса устройства должен превышать 270 миллисекунд.

Для исключения эффекта дребезга в управлении автоматическими системами для показаний датчиков должен быть реализован программный гистерезис.

Стоит отметить что часть функционала программы вынесена в функцию обработки буфера UART. Код функции приведён ниже.

void check\_uart\_buff() {

uint8\_t package\_curr[WRITE\_PACKAGE\_SIZE];

uint16\_t package\_data;

if ((Serial.available() >= WRITE\_PACKAGE\_SIZE && Serial.peek() == WRITE\_HEADER) ||

(Serial.available() >= READ\_PACKAGE\_SIZE && Serial.peek() == READ\_HEADER)) {

switch (Serial.read()) {

case READ\_HEADER: {

package\_curr[HEADER\_OFFSET] = ANSWER\_HEADER;

switch (Serial.read()) {

case HW\_ID\_REG\_ADDR: {

package\_curr[ADDR\_OFFSET] = HW\_ID\_REG\_ADDR;

package\_curr[DATA\_OFFSET] = HW\_ID & 0xFF;

package\_curr[DATA\_OFFSET + 1] = (HW\_ID >> 8) & 0xFF;

Serial.write(package\_curr, 4);

standalone\_mode\_flag = false;

if (millis() - temperature\_tmr > 2000) {

temperature\_tmr = millis();

temperature = dht\_sensor.readTemperature();

}

break;

}

case TEMPERATURE\_REG\_ADDR: {

package\_data = temperature \* 10;

package\_curr[ADDR\_OFFSET] = TEMPERATURE\_REG\_ADDR;

package\_curr[DATA\_OFFSET] = package\_data & 0xFF;

package\_curr[DATA\_OFFSET + 1] = (package\_data >> 8) & 0xFF;

Serial.write(package\_curr, WRITE\_PACKAGE\_SIZE);

break;

}

case TEMPERATURE\_LIMIT\_REG\_ADDR: {

package\_data = temperature\_limit \* 10;

package\_curr[ADDR\_OFFSET] = TEMPERATURE\_LIMIT\_REG\_ADDR;

package\_curr[DATA\_OFFSET] = package\_data & 0xFF;

package\_curr[DATA\_OFFSET + 1] = (package\_data >> 8) & 0xFF;

Serial.write(package\_curr, WRITE\_PACKAGE\_SIZE);

break;

}

}

break;

}

case WRITE\_HEADER: {

switch (Serial.read()) {

case DRY\_SOIL\_REG\_ADDR: {

dry\_soil = analogRead(SOIL\_HUMIDITY\_PIN);

Serial.read();

Serial.read();

break;

}

case WET\_SOIL\_REG\_ADDR: {

wet\_soil = analogRead(SOIL\_HUMIDITY\_PIN);

Serial.read();

Serial.read();

break;

}

case LIGHT\_LEVEL\_REG\_ADDR: {

light\_level = analogRead(LIGHT\_PIN);

Serial.read();

Serial.read();

break;

}

case TEMPERATURE\_LIMIT\_REG\_ADDR: {

package\_data = Serial.read();

package\_data += Serial.read() << 8;

temperature\_limit = (float)package\_data / 10;

break;

}

}

break;

}

}

}

}

При подключении к персональному компьютеру и запуске программного обеспечения для конфигурации устройства, оно выходит из самостоятельного режима. Сброс самостоятельного режима происходит при считывании уникального идентификатора устройства. Кроме этого измерения температуры происходят после отправления ответа на запрос идентификатора устройства. Это необходимо для того чтобы исключить влияние задержки функции считывания температуры на передачу данных об идентификаторе устройства.

Данные с датчика температуры считываются в переменную с плавающей запятой. Поэтому перед отправкой данных о температуре необходимо перевести значение температуры в целочисленный формат и учесть это при приёме данных на компьютере. Поскольку точность измерений датчика составляет 0.1 градуса по Цельсию, значение температуры достаточно умножить на 10. Соответствуещее преобразование данных происходит и при приёме данных о критической температуре.

Отсутствие запросов на передачу идентификатора устройства в течение трёх секунд возвращает устройство в самостоятельный режим.

5.4 Разработка программного обеспечения для ПК

Требования к программному обеспечению для конфигурации устройства перечислены в подразделе 5.1.

Важным требованием является наличие функционала по автоматическому поиску подключённого устройства. Это требование значительно упрощает работу с программным обеспечение для конечного пользователя, поскольку избавляет его от необходимости ручного выбора порта подключённого устройства.

Использование программного обеспечения совместно с платформой Arduino Nano имеет одну особенность. На плате Arduino Nano установлен USB-UART конвертер, преобразующий сигналы, передаваемые по шине USB в сигналы, передаваемые по интерфейсу UART. Для работы с Arduino производителем были созданы драйвера, позволяющее разработчикам работать с платой как с COM-портом, то есть организовывать передачу данных по протоколу UART. Так же в микроконтроллеры, установленные на платах Arduino записывается загрузчик, позволяющий перезаписывать прошивку микроконтроллера через тот же UART. Данный загрузчик запускается при подаче напряжения на микроконтроллер либо при перезапуске микроконтроллера с помощью сигнала RESET.

Однако в отличие от шины USB, стандартный COM-порт имеет дополнительные сервисные сигнальные линии. Одна из таких линий на платах Arduino используется для перезапуска установленного на ней микроконтроллера, что можно увидеть в схемотехнике платы Arduino Nano (рис 5.4.1). Тем самым микроконтроллер временно входит в состояние записи новой прошивки.

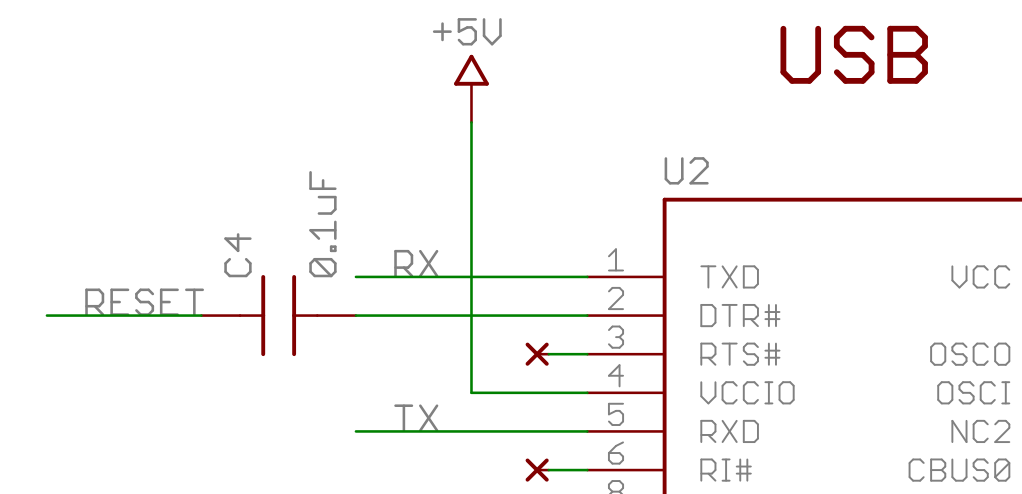


Рисунок 5.4.1 – Фрагмент схемы Arduino Nano

Для разработки программного обеспечения персонального компьютера важно учесть этот момент, поскольку драйвер для платформы разработан таким образом, что при подключении к порту происходит автоматический перезапуск микроконтроллера. Момент запуска загрузчика и ожидание новой прошивки нужно пропустить и только после этого начинать отсылать данные на устройство.

Кроме всего этого хорошим тоном считается асинхронная работа потоков, отвечающих за работу графического интерфейса, и потоков, выполняющих передачу данных. Соответственно буферы COM-порта должны обрабатываться в отдельном потоке.

Все вышеперечисленные моменты были учтены при разработке программного обеспечения для ПК.

5.5 Управление работой устройства

Разработанное программное обеспечение предоставляет пользователю удобный интерфейс для настройки параметров устройства (рис 5.5.1).

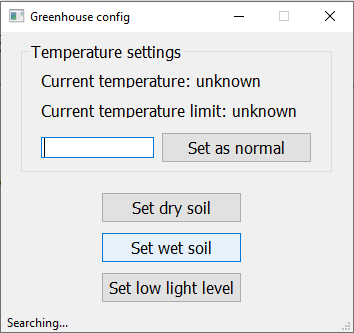


Рисунок 5.4.1 – Фрагмент схемы Arduino Nano

В отсутствие подключения к устройству все кнопки и поля являются неактивными, а в строке состояния отображается статус подключения “Searching…”, сообщающий о том, что ПО пытается найти устройство в системе.

При успешном подключении и идентификации устройства статус изменяется на “Device connected”. Данные о текущей температуре воздуха автоматически запрашиваются с устройства с периодом две секунды. Так же отображается значение нормальной температуры, записанное на устройстве.

Если пользователь хочет изменить значение нормальной температуры, ему достаточно ввести новой значения в поле ввода и нажать на кнопку “Set as normal”.

При нажатии на кнопку “Set dry soil” текущее значение датчика влажности почвы установится, как порог включения системы автоматического полива.

Кнопка “Set wet soil” предназначена для установки предела отключения системы автоматического полива. Устройство запомнит значение датчика влажности почвы при котором требуется отключить полив.

При нажатии на кнопку “Set low light level” текущее значение датчика освещения устройство установит, как порог включения системы автоматического освещения.

ЗАКЛЮЧЕНИЕ

В процессе выполнения курсовой работы было разработано устройство для контроля окружающей среды растений. Приобретены знания в области работы с микропроцессорными устройствами и различными датчиками. Получены знания в области проектирования электрических схем, разводки печатных плат и 3D моделирования.

Разработанное устройство является полностью работоспособным и выполняет возложенные на него обязанности. Однако в будущем планируется модернизация и расширения проекта, поскольку на текущий момент устройство подходит преимущественно для комнатного применения, но имеет возможности масштабирования. Модернизации может подлежать как аппаратная, так и программная часть проекта. Например, можно сделать динамическую индикацию температуры, добавить сохранение параметров конфигурации в энергонезависимую память, сделать закрытый корпус и внедрить в него систему обогрева и проветривания, нарастить количество независимых каналов полива и другое. Фото готового устройства приведено ниже.

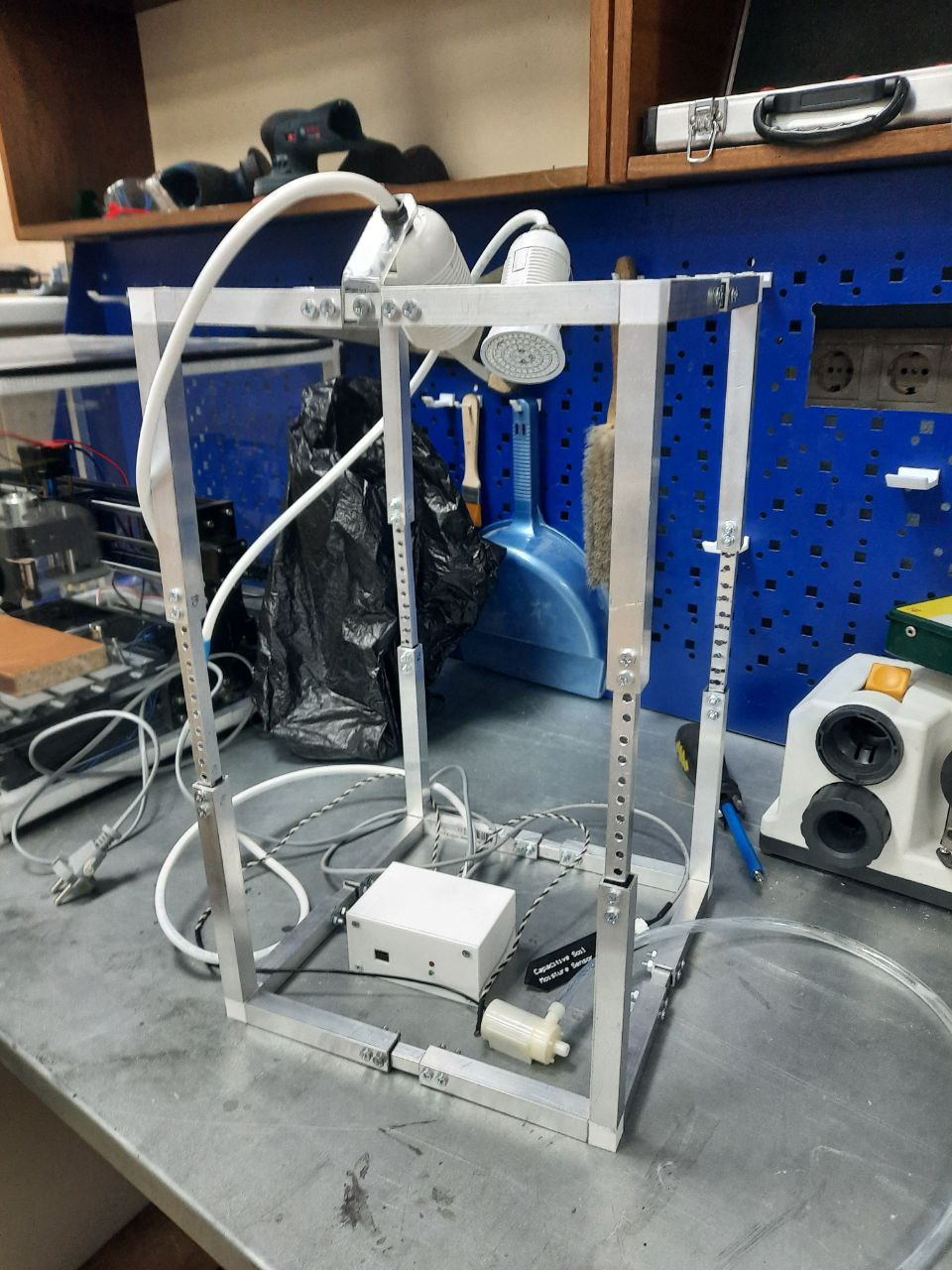


Рисунок 6.1 – Готовое устройство

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1]. Статья – Датчики [Электронный ресурс] – Электронные данные. – Режим доступа: <http://www.electrolibrary.info/subscribe/sub_16_datchiki.htm> – Дата доступа: 21.09.2021

[2]. MQ-2 new. [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.pololu.com/file/0J309/MQ2.pdf> – Дата доступа: 21.09.2021

[3]. MQ-5.doc [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://files.seeedstudio.com/wiki/Grove-Gas\_Sensor-MQ5/res/MQ-5.pdf](https://files.seeedstudio.com/wiki/Grove-Gas_Sensor-MQ5/res/MQ-5.pdf%20) – Дата доступа: 21.09.2021

[4]. DS18B20 - Programmable Resolution 1-Wire Digital Thermometer [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf> – Дата доступа: 21.09.2021

[5]. MH-Z19 CO2 Ver1.0.pdf [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.winsen-sensor.com/d/files/PDF/Infrared%20Gas%20Sensor/NDIR%20CO2%20SENSOR/MH-Z19%20CO2%20Ver1.0.pdf> – Дата доступа: 21.09.2021

[6]. RP2040 Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datasheets.raspberrypi.com/rp2040/rp2040-datasheet.pdf>– Дата доступа: 21.09.2021

[7]. DHT11 Humidity & Temperature Sensor [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>– Дата доступа: 21.09.2021

[8]. DHT22 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>– Дата доступа: 21.09.2021

[9]. Nivel\_de\_agua\_analogico [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://curtocircuito.com.br/datasheet/sensor/nivel_de_agua_analogico.pdf>– Дата доступа: 21.09.2021

[10]. BM4012\_\354\340\353.DOC [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://belchip.by/sitedocs/06273.pdf>– Дата доступа: 21.09.2021

[11]. Статья – Тепловые пожарные извещатели: виды и описание, принцип работы [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://fireman.club/statyi-polzovateley/teplovyie-pozharnyie-izveshhateli-vidyi-i-opisanie/> – Дата доступа: 22.09.2021

[12]. Статья – Arduino Nano [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://arduino.ru/Hardware/ArduinoBoardNano> – Дата доступа: 22.09.2021

[13]. Raspberry Pi Pico Datasheet [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf> – Дата доступа: 22.09.2021

[14]. Programming with Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: [https://www.halvorsen.blog/documents/technology/resources/resources/Arduino/Programming%20with%20Arduino.pdf](https://www.halvorsen.blog/documents/technology/resources/resources/Arduino/Programming%20with%20Arduino.pdf%20) – Дата доступа: 22.09.2021

[15]. Programming Arduino Getting Started [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://index-of.es/Varios-2/Programming%20Arduino.pdf> – Дата доступа: 22.09.2021

[16]. Статья – Dallas Temperature Control Library [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.milesburton.com/Dallas_Temperature_Control_Library>

– Дата доступа: 23.09.2021

[17]. Статья – OneWire Library [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.pjrc.com/teensy/td_libs_OneWire.html> – Дата доступа: 23.09.2021

[18]. Статья – Amperka/gas-sensor [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://wiki.amperka.ru/js:gas-sensor> – Дата доступа: 23.09.2021

[19]. Статья – starter\_kity\_i\_nabory\_dlya\_arduino\_info [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://purelogic.ru/docs/elektronika/starter_kity_i_nabory_dlya_arduino_info.pdf> – Дата доступа: 23.09.2021

[20]. PIEZO-PKM22EPPH4001-BO.pdf [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.arduino.cc/documents/datasheets/PIEZO-PKM22EPPH4001-BO.pdf> – Дата доступа: 23.09.2021

[21]. Ams1117.pdf [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://static.chipdip.ru/lib/029/DOC001029248.pdf> – Дата доступа: 23.09.2021

ПРИЛОЖЕНИЕ А

*(обязательное)*

Схема структурная

ПРИЛОЖЕНИЕ Б

*(обязательное)*

Схема функциональная

ПРИЛОЖЕНИЕ В

*(обязательное)*

Схема принципиальная

ПРИЛОЖЕНИЕ Г

*(обязательное)*

Схема программы

ПРИЛОЖЕНИЕ Д

*(обязательное)*

Ведомость документов