

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ПЕНЗЕНСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ НЕТИПОВОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ПЕНЗЕНСКОЙ ОБЛАСТИ «ГУБЕРНСКИЙ ЛИЦЕЙ»

СИСТЕМА АВТОПОЛИВА НА ПЛАТФОРМЕ
«ARDUINO»

Выполнил:

учащийся 8 физико-математического класса

Хряпов Кирилл

Научный руководитель: Танасов Иван Дмитриевич

Пенза, 2019

Содержание

Введение	3
Основная часть	5
Функциональные особенности платформы «Arduino».....	5
Определение тематики исследования.....	7
Моделирование и конструирование технической части проекта.....	7
Разработка и внедрение программной части проекта.....	9
Заключение	13
Список используемой литературы.....	14
Приложение.....	14

Введение

Образование должно происходить с использованием современных технологий. Человеку нужно постоянно развиваться и следить за появлением новейших технологий в электронике, которые играют очень важную роль в нашем обществе. Одной из таких технологий является семейство контроллеров Arduino.

«Arduino – это открытая платформа для прототипирования электроники, основанная на гибком, простом в использовании оборудовании и программном обеспечении. Она предназначена для новичков, профессионалов и всех тех, кто заинтересован в создании интерактивных объектов или сред» - так сказано об Arduino разработчиками. Проще говоря это инструмент для создания электронных устройств и воплощения в жизнь различных идей, связанных с объединением программирования и робототехники.

Arduino может принимать цифровые и аналоговые сигналы с различных устройств и имеет возможность управления различными исполнительными модулями.

В этом учебном году мне самому стала интересна тема создания роботизированной системы на «Arduino», и я, вдохновившись несколькими видео на YouTube, решил заказать себе базовый комплект по работе с Arduino. На протяжении нескольких недель мы изучали различные проекты, изготовленные на платформе. Далее мы приступили к созданию кода и подключению различных модулей. В ходе экспериментов с различными модулями и датчиками, в совокупности с

изученной информацией, мы выбрали тему исследования и приступили к этапу прототипирования.

Цель проекта: Создать программируемую систему с функцией автоматического полива комнатных растений.

Задачи:

1) Научиться языку программирования Arduino и уметь использовать полученные знания при создании кода

2) Научиться грамотно использовать все возможности платформы

3) Создать систему автоматического полива, соответствующую современным тенденциям: актуальный дизайн, эргономичность, технологичность.

4) В дальнейшем улучшать свой проект основываясь на обратной связи от пользователей.

Проблематика: Довольно часто в домашних условиях погибают комнатные растения из-за невнимательности людей к регулярности полива, а специальные горшки с системой полива могут подойти не ко всем растениям, и таймеры для полива стоят довольно больших денег.

Актуальность: У системы автоматического полива нет аналогов в массовом производстве и программный код проекта универсален, как для комнатных растений, так и для грядки на огороде. Система имеет актуальный эстетичный дизайн для комнатной среды, не требует высокого потребления энергии и постоянного человеческого вмешательства.

Объект исследования: Платформа для программирования Arduino.

Предмет исследования: Возможность интеграции платформы «Arduino» в социальную среду, в форме проекта, с целью повседневного использования.

Основная часть

Функциональные особенности платформы «Arduino»

Arduino — торговая марка аппаратно-программных средств для построения простых систем автоматике и робототехники, ориентированная на непрофессиональных пользователей.

Программная часть состоит из бесплатной программы «Arduino IDE» для написания кодов, их компиляции. Аппаратная часть представляет собой набор смонтированных печатных плат. Полностью открытая архитектура системы позволяет свободно копировать или дополнять линейку продукции Arduino.

Arduino может использоваться как для создания автономных объектов автоматике, так и подключаться к программному обеспечению на компьютере через стандартные проводные и беспроводные интерфейсы.

"Мозг" Arduino - это микроконтроллер семейства «Atmega». Микроконтроллер представляет из себя процессор с оперативной памятью и периферийными устройствами, реализованный на одной микросхеме. Фактически Arduino является микрокомпьютером,

который способен выполнять некоторые простые задачи. Разные модели из семейства Arduino оснащены разными микроконтроллерами. «Руками» в данном случае служат электрические выводы, или так называемые «порты» размещённые по периметру платы.

В линейку печатных плат входят такие модели как «Arduino UNO», «ArduinoNano», «ArduinoMega», «ArduinoMini», «ArduinoLeonardo». Платы различаются количеством оперативной памяти и портов, размером.

На плате есть выводы цифровые, а есть аналоговые. Принципиальная разница между ними в том, что на цифровых выводах может быть только два значения: либо логическая "1", выдающая сигнал в 5В, либо логический "0", выдающий 0В, а на аналоговых выводах - диапазон от логической 1 до 0 может быть разбит на множество участков.

В итоге такое количество "рук" у Arduino позволяет подключать к нему огромное количество различных периферийных устройств. Среди них, например: кнопки, светодиоды, сервоприводы, ЖК дисплеи, двигатели, Bluetooth, WiFi или GPS модули, сенсоры отпечатков пальцев, самые различные датчики и многое другое.

Общение с платой происходит с помощью языка программирования «Wiring» (упрощенного и адаптированного, специально для Arduino, языка программирования C++.) Само написание программы происходит в среде «Arduino IDE». Сам язык программирования достаточно простой и при желании любой человек может освоиться в нем.

Я выбрал именно эту платформу для программирования так как она довольно проста и удобна в использовании, относительно недорогая и совместима со многими модулями, что значительно упрощает их поиск в нашем городе.

Итак, данная платформа для программирования помогает нам реализовать свои идеи в реальную жизнь, не затрачивая при этом большие силы и время.

Определение тематики исследования.

После того как «Arduino» попал мне в руки я усердно начал искать актуальную информацию в IT сфере для выбора тематики своего проекта. У меня были различные варианты. Я хотел сделать домашнюю метеостанцию или же светомузыку, но такие проекты и так имеются в активной продаже. Мне хотелось сделать что-то уникальное и не имеющее аналога на рынке.

Моя мама и я любим комнатные растения, но они довольно часто погибали из-за того, что мы много времени проводим вне дома и забываем поливать вовремя. Мы смотрели на вариант купить горшок с автоматическим поливом, но он не подходил под растение. У меня появилась идея создать программируемую систему полива, не требующую постоянного ухода.

Моделирование и конструирование технической части проекта

Для начала нужно было определиться с технической частью проекта, а именно наличие каких компонентов необходимо для оптимальной работы

системы. Помимо платы Arduino понадобилась различная периферия для исполнения работы полива.

Для того чтобы поставлять воду из емкости к растению понадобилась погружная помпа на 12В (100 л/час). Для питания этой помпы я взял 3 батареи на ~3в, которые соединены последовательно, соответственно дающие необходимое питание для помпы, но в будущем планирую заменить источник питания на кабельный (тоже 12В).

Для удобства настройки периодичности и времени полива мной было принято решение поставить ЖК-дисплей с кнопками, с помощью которого можно в любой момент поменять настройки.

Работа системы осуществляется таким образом: с помощью дисплея мы задаем период между поливами растения. После данного периода сигнал с платы идет на полевой транзистор, к которому подключено питания батареи. Ток через транзистор идет на насос такое количество времени, которое мы также задаем с помощью дисплея. После того как работа закончена, сигнал с платы опять идет к транзистору и снова закрывает ток от источника.

Приближенная схема подключения компонентов приведена в приложении (Приложение 1).

Следующим этапом по созданию системы автоматического полива является создание корпуса. В качестве материала я выбрал оргстекло. Корпус представляет собой нескольких скрепленных между собой деталей. Чертеж был сделан с помощью программы «AutoCAD 2018» (Приложение 2) Из 3-х

миллиметровых пластин оргстекла, с помощью лазерного станка, я вырезал детали, собрал их и загерметизировал емкость для воды с помощью аквариумного герметика. Объем емкости для воды составляет 2,4 литра.

В качестве горшка для растений я решил сделать додекаэдр из тех же пластин оргстекла, которые я спаял между собой.

Комнатным растением я выбрал пару цветков из вида суккулентов (Приложение 3). Они хорошо накапливают влагу в своих толстых листьях и долгое время могут обойтись без воды. Я определил, как грамотно можно будет посадить растения этого вида. Сначала я заложил дренажный слой (при это протягивая шланг для полива) из крупных, мелких камней и песка, который обеспечивает оптимальный отток воды и воздухопроницаемость грунта. И только потом я положил грунт и посадил цветки.

Разработка и внедрение программной части проекта

Написание программы для системы автоматического полива происходила в программе «Arduino IDE». У программы есть 3 основные задачи. 1-ая задача это – полив с заданной регулярностью на заданное время. 2–ая задача настраивать значения периодов между поливом и время полива с помощью ЖК дисплея. 3-я задача – включить "экстренный полив" если мы нажимаем на соответствующую кнопку на дисплее. Для того чтобы выполнялись первые 2 цели я задал 2 переменные `Period` и `Volume`. С помощью

первой части программы мы можем взаимодействовать с дисплеем, тем самым регулируя значение этих двух переменных.

Вторая часть кода (рис 1), которая отвечает за подачу сигнала на транзистор, принимает новые значения переменных и меняет время между поливами и время поливов.

```
void WaterFillingTimer()
{
    if (millis() - timer > period*1000*60*60*24)//можно поменять
    {
        timer = millis();
        digitalWrite(12,HIGH);
        delay(volume*1000);
        digitalWrite(12,LOW);
    }
}
```

рис. 1

Рассмотрим конкретно эту часть кода.

Функция `millis` отсчитывает количество миллисекунд с начала работы микроконтроллера. Из `millis` мы вычитаем переменную `timer` которая с самого начала равна 0. `Period` мы настраиваем с помощью дисплея. Допустим, что `period = 2`, то `period*1000*60*60*24` – это количество миллисекунд в двух сутках. И когда с начала включения микроконтроллера пройдет более чем двое суток, то условие будет выполнено.

После того как условие выполнено идет выполнение дальнейшего кода. Сначала мы присваиваем переменной `timer` количество миллисекунд в двух сутках, чтобы команда могла повториться

повторно. Далее мы подаем сигнал с 12 выхода на транзистор.

Функция `delay` означает «ждать» (в миллисекундах). После того как пройдет заданное время код выполняется дальше. Но почему же тогда нельзя изначально использовать функцию `delay`, если она такая простая? Потому что во время того как микроконтроллер выполняет эту функцию, он не может больше ничего ждать, и именно для этого существует функция `millis`.

Итак, допустим, что переменная `volume = 3.3*1000` – это количество миллисекунд в трех секундах. То есть, функция `delay` будет выполняться 3 секунды. После того как проходит 3 секунды сигнал больше не подается на транзистор.

Соответственно, на практике мы увидим, как после 2 суток со времени включения системы насос включится на 3 секунды, а потом выключится еще на двое суток.

Также рассмотрим пару нюансов по работе с ЖК-дисплеем. Для того чтобы плата определяла какая кнопка нажата нужно прописать это в программе, но для начала нужно разобраться как именно работает дисплей.

На плате присутствуют пять управляющих кнопок, работа с которыми ведется через один аналоговый вход A0. В дисплее использован достаточно распространенный способ простого кодирования сигнала, при котором каждая кнопка формирует определенное значение напряжения, которое

преобразуется в соответствующее значение от 0 до 1023. Таким образом, мы можем передавать информацию о нажатии разных кнопок через один вход, считывая его при помощи функции `analogRead()`.

Итак, рассмотрим часть кода, отвечающую за распознавание нажатой кнопки:

```
const int BUTTON_NONE = 0; //ничего не нажато
const int BUTTON_RIGHT = 1; //нажата кнопка вправо
const int BUTTON_UP = 2; //нажата кнопка вверх
const int BUTTON_DOWN = 3; //нажата кнопка вниз
const int BUTTON_LEFT = 4; //нажата кнопка влево
const int BUTTON_SELECT = 5; //нажата кнопка выбор

void getPressedButton()
{
  int buttonValue = analogRead(0);
  if (buttonValue < 100) {
    return BUTTON_RIGHT;
  }
  else if (buttonValue < 200) {
    return BUTTON_UP;
  }
  else if (buttonValue < 400) {
    return BUTTON_DOWN;
  }
  else if (buttonValue < 600) {
    return BUTTON_LEFT;
  }
  else if (buttonValue < 800) {
    return BUTTON_SELECT;
  }
  return BUTTON_NONE;
}
```

Сначала мы задаем константы для каждой кнопки. Далее, переменную, которой присваиваем функцию `AnalogRead()` для выхода A0. Так как каждой нажатой кнопке соответствует некоторое значение напряжения на выходе A0 (значение которых можно найти в интернете), то мы для каждого этого числа возвращаем значение нажатой кнопки.

И теперь рассмотрим, как использовано распознавание нажатой кнопки. В качестве примера

возьмем часть программы, отвечающую за настройку периодичности полива:

```
case BUTTON_UP:
  button = 0;
  if (period < 50){
    period++;
  }
  message = ("Period = " + String (period) );
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print(message);
  break;
case BUTTON_DOWN:
  button = 0;
  if (period > 1) period--;
  message = ("Period = " + String (period) );
  lcd.clear();
  lcd.setCursor(1, 0);
  lcd.print(message);
  break;
```

Функция case заменяет функцию if. То есть – case является условием при котором будет выполняться команда. Когда нажата кнопка «вверх» и когда значение переменной меньше 50 то присваивается новое значение больше предыдущего на 1. Далее мы задаем сообщение, выбираем строку для его написания и печатаем само сообщение. Аналогично для нажатой кнопки «вниз».

Заключение

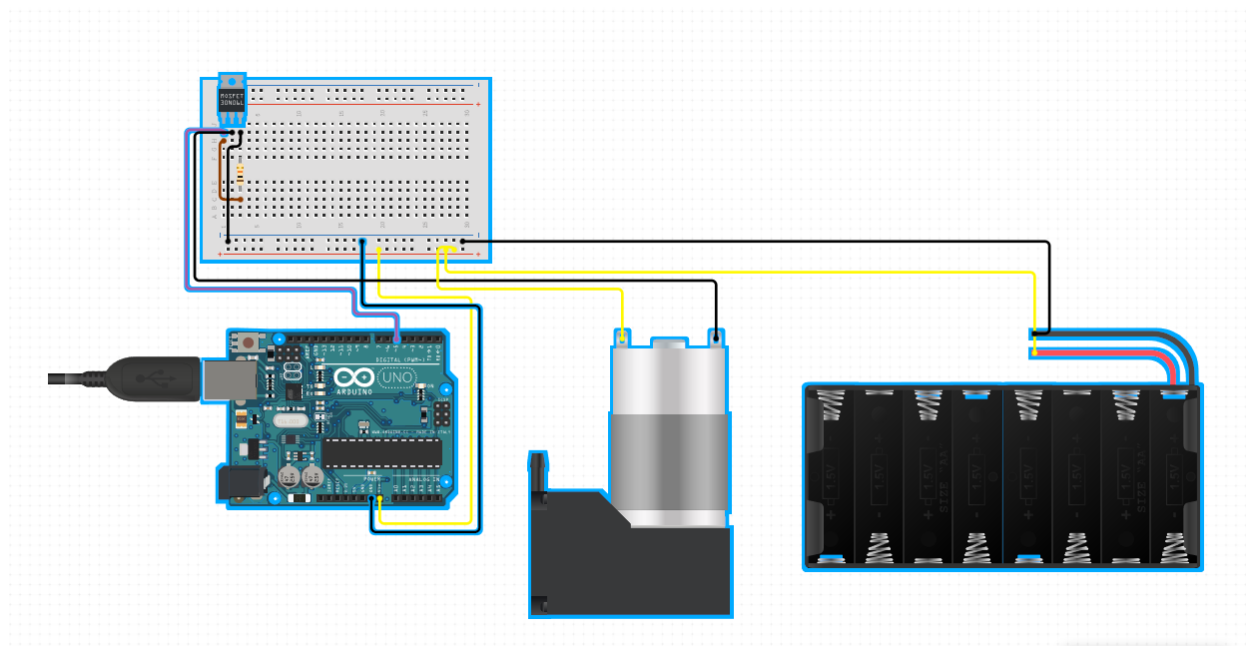
В ходе подготовки к защите проекта, с помощью видео-уроков и занятий с наставниками, я овладел языком программирования Wiring на базовом уровне и смог применить полученные знания на практике при создании программы. Результатом проекта является рабочий образец системы автономного полива, который соответствует современным тенденциям в сфере бытовых систем для полива растений. Определив недостатки системы, в будущем, мы планируем заняться совершенствованием проекта. В планах: доработать дизайн и особенности конструкции корпуса, сделать систему полива более компактной и удобной для бытового использования.

Список используемой литературы

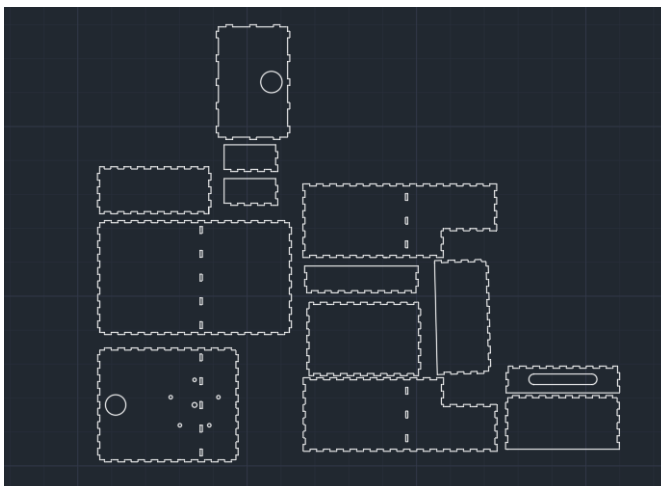
- 1) [LXF100-101:Arduino](#). — Цикл статей по Arduino
- 2) [Arduino Russian](#). — Неполная русская документация по языку и библиотекам
- 3) [arduino.cc](#) — официальный сайт ветви arduino.cc
- 4) [ArduinoPlus.ru](#) – любительский сайт с обсуждениями и библиотеками.

Приложение

Приложение 1



Приложение 2



Приложение 3

