ГБОУ Самарской области

«Самарский региональный центр для одаренных детей»

СОЗДАНИЕ ПРОТОТИПА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ ДЛЯ КОНТРОЛЯ ЗА ИЗМЕНЕНИЕМ ВЛАЖНОСТИ ВОЗДУХА И ТЕМПЕРАТУРЫ В ПОМЕЩЕНИИ

Автор:

Тудачков Илья,

обучающийся 10 Б класса СОЛ

Научный руководитель:

Ежов Данил Александрович,

учитель информатики и ИКТ

Самара, 2016

**ОГЛАВЛЕНИЕ**

1.Введение3

2.Введение в микроконтроллеры Arduino3

Что такое Arduino?4

Почему Arduino?5

3.Теоритические представления о прототипе7

Датчик влажности DHT117

Датчик атмосферного давления BMP1808

4.Практическая часть9

Устройство прототипа9

Блок-схема работы устройства10

Схема прототипа10

Полное описание работы прототипа11

Код программы16

5.Заключение21

6.Список Используемой литературы22

**1. ВВЕДЕНИЕ**

В настоящее время очень популярны всяческие системы для автоматизации рутинных процессов в быту человека. Например, система «Умный дом». Эта система предназначена для повышения комфортности проживания человека в доме. Она направлена на освобождение людей от выполнения разного рода каждодневной работы по дому. Автоматизированные действия, заменяющие такую работу, позволят сохранить больше свободного времени для досуга и отдыха человека. Обычно подобные системы изобилуют разнообразными функциями, которые разработчики посчитали необходимыми для комфортного проживания. Тем самым, стоимость подобных систем становится довольно высокой для рядового потребителя. Тем более, что часть функций системы «Умный дом» попросту даже не используются потребителями.

Поэтому мы решили предложить более доступную систему, которая строится на базе контроллера Arduino. Arduino – это платформа для проектирования разнообразных электронных устройств.

**2. ВВЕДЕНИЕ В МИКРОКОНТРОЛЛЕРЫ ARDUINO**

Существует такой открытый проект, который называется Arduino. Основа этого проекта – базовый аппаратный модуль и программа, в которой можно написать код для контроллера на специализированном языке, и которая позволяет этот модуль подключить и запрограммировать. Модули отличаются друг от друга своим видом, удобством использования. Но все позволяют подключить их к компьютеру, запустив программу Arduino, написать код, «прошить» полученную программу в модуль и проверить работу программы в живом виде. И все модули позволяют подключать к ним платы расширения, либо готовые, купленные в магазине, либо изготовленные самостоятельно. Кстати, проект открытый, то есть, можно не только бесплатно скачать программу и использовать её или переделать по своему вкусу, можно самостоятельно изготовить базовый модуль — есть его схема, есть прошивка, есть всё для самостоятельного изготовления.

С момента появления проекта Arduino у него появилось множество почитателей – достаточно ввести в поисковую строку слово «arduino», как вы обнаружите сотни сайтов, посвящённых этой теме, сотни проектов, основанных на Arduino.

Так для чего же я решил сделать свою работу на микроконтроллере?

Так как мы живём в 21 веке, веке информационных и технических инноваций, я решил сделать свой проект именно на новых компонентах компьютерной техники. Но для этого мне нужно разобраться с рядом программ, которые предназначены для работы с модулем Arduino, как сама программа для создания программ для микроконтроллера Arduino S4A, так и програама эмуляции VirtualBreadBoard. Эти программы – свободно распространяемые и устанавливаются как на компьютеры под управлением операционной системы Windows, так и под управлением операционной системы Linux.

Научившись программировать модуль Arduino, мы будем готовы создавать интересные и полезные электронные устройства, к которым относятся и роботы.

Я хочу познакомиться с увлекательнейшим проектом Arduino, и возможно, роботостроение станет моей будущей профессией. Но даже если нет, то, уверен, через много лет, вспоминая свои эксперименты в этой области, я буду рад, что не прошел мимо, не пожалел времени на освоение основ – мне будет, что вспомнить.

Что такое Arduino?

**Arduino** — торговая марка аппаратно-программных средств для построения простых систем автоматики и [робототехники](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE%D0%B1%D0%BE%D1%82%D0%BE%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0), ориентированная на непрофессиональных пользователей. Программная часть состоит из бесплатной программной оболочки ([IDE](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%80%D0%B5%D0%B4%D0%B0_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8)) для написания программ, их компиляции и программирования аппаратуры. Аппаратная часть представляет собой набор смонтированных [печатных плат](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%87%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D0%B0), продающихся как официальным производителем, так и сторонними производителями. Полностью [открытая архитектура](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%B0%D1%8F_%D0%B0%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0) системы позволяет свободно копировать или дополнять линейку продукции Ардуино.

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

В концепцию Ардуино не входит корпусный или монтажный конструктив. Разработчик выбирает метод установки и механической защиты плат самостоятельно. Сторонними производителями выпускаются наборы робототехнической электромеханики, ориентированной на работу совместно с платами Ардуино.

Проект Arduino был удостоен почётного упоминания при вручении призов [Prix Ars Electronica](https://ru.wikipedia.org/w/index.php?title=Prix_Ars_Electronica&action=edit&redlink=1) 2006 в категории Digital Communities.   
 Arduino представляет собой [линейку электронных блоков-плат](http://www.arduino.cc/en/Main/Boards), которые можно подключать к компьютеру по USB, а в качестве периферии — любые устройства от светодиодов до механизмов радиоуправляемых моделей и роботов. Программы для него пишутся на простом и интуитивно понятном си-подобном языке Wiring (c возможностью подключения сторонних библиотек на C/C++, например, для управления LCD-дисплеями или двигателями), компилируются и загружаются в устройство одной кнопкой, после чего вы тут же получаете работающий автономный гаджет. Никакого ассемблера, никаких лишних проводов и дорогущих деталей и программаторов — чистое творчество, включай и работай!

**Почему Arduino?**

Существует множество микроконтроллеров и платформ для осуществления «physical computing».  Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard и многие другие предлагают схожую функциональность. Все эти устройства объединяют разрозненную информацию о программировании и заключают ее в простую в использовании сборку.  Arduino, в свою очередь, тоже упрощает процесс работы с микроконтроллерами, однако имеет ряд преимуществ перед другими устройствами для преподавателей, студентов и любителей:

**Низкая стоимость** – платы Arduino относительно дешевы по сравнению с другими платформами. Самая недорогая версия модуля Arduino может быть собрана в ручную, а некоторые даже готовые модули стоят меньше 50 долларов.

**Кросс-платформенность** – программное обеспечение Arduino работает под ОС Windows, Macintosh OSX и Linux. Большинство микроконтроллеров ограничивается ОС Windows.

[**Простая и понятная среда программирования**](http://arduino.ru/Arduino_environment) – среда Arduino подходит как для начинающих пользователей, так и для опытных. Arduino основана на среде программирования Processing, что очень удобно для преподавателей , так как студенты работающие с данной средой будут знакомы и с Arduino.

Программное обеспечение с возможностью расширения и открытым исходным текстом – ПО Arduino выпускается как инструмент, который может быть дополнен опытными пользователями. Язык может дополняться библиотеками C++. Пользователи, желающие понять технические нюансы, имеют возможность перейти на язык AVR C на котором основан C++. Соответственно, имеется возможность добавить код из среды AVR-C в программу Arduino.

[***Аппаратные средства***](http://arduino.ru/Hardware) с возможностью расширения и открытыми принципиальными схемами – микроконтроллеры ATMEGA8 и ATMEGA168 являются основой Arduino.  Схемы модулей выпускаются с лицензией Creative Commons, а значит, опытные инженеры имеют возможность создания собственных версий модулей, расширяя и дополняя их. Даже обычные пользователи могут разработать опытные образцы с целью экономии средств и понимания работы.

**3. ТЕОРИТИЧЕСКИЕ ПРЕДСТАВЛЕНИЯ О ПРОТОТИПЕ**

Наша система предназначена, в первую очередь, для людей пожилого возраста и для людей с ограниченными возможностями, которым, порой, проблематично выполнять некоторые действия в доме.

Помимо автоматизированной системы регулирования влажности и температуры, в проекте, присутствует функция обнаружения утечки газа в помещении и моментальное информирование об этом в том случае, когда человек находится вне своего дома. Способом такого информирования может быть отправка SMS – сообщения или звонок со звуковым сигналом на мобильный телефон владельца дома. Одновременно с информированием хозяина дома происходит автоматическое перекрывание крана подачи газа.

Кроме того система предусматривает информирование о понижении температуры в доме ниже критического порога, который заранее запрограммирован в программе. Например, в том случае, когда по какой-либо причине произошло отключение системы отопления в квартире. Помимо информирования хозяина дома, система автоматически включает электрический обогреватель, если такой имеется в доме.

Также в макете присутствует инфракрасный датчик, который реагирует на присутствие человека в определённой комнате. Эта функция служит для автоматического проветривания помещения, когда в комнате никого нет.

Датчик влажности DHT11

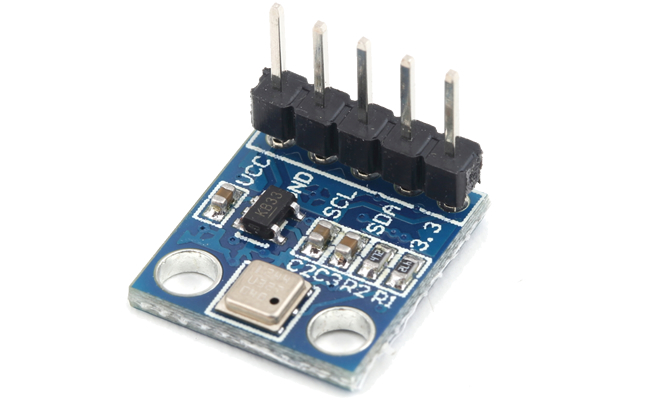
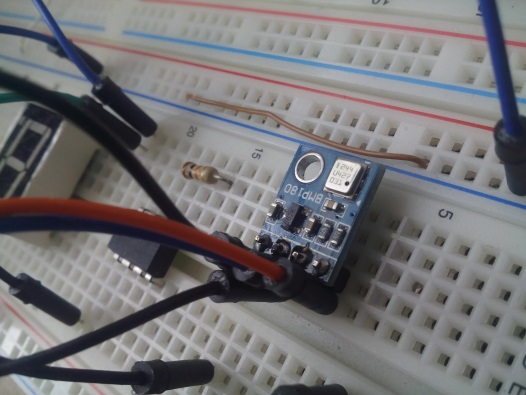
Для измерения влажности в помещениях отлично подойдет цифровой датчик влажности DHT11. подключать его мы будем как вы уже наверное догадались к плате [Arduino](http://www.radio-magic.ru/arduino-projects/41-chto-takoe-arduino). Для этих целей была создана специальная библиотека, все что нам остается это импортировать ее в проект. Измеренное значение влажности будет передаваться в com порт ноутбука. Полученное значение с arduino посылается в ноутбук где отображается в мониторинге порта среды Arduino IDE. Среда Arduino IDE работает абсолютно на любой операционной системе и на любом ноутбуке.

На основе данного датчика относительного давления можно спроектировать некое подобие климатической установки, регулирующей влажность в помещении либо в теплице с растениями. Для этого в программ у добавить условие: "если влажность менее 60% то подать напряжение на распылитель". Тут под "распылителем" понимается устройство распыляющее влагу.

Датчик атмосферного давления BMP180

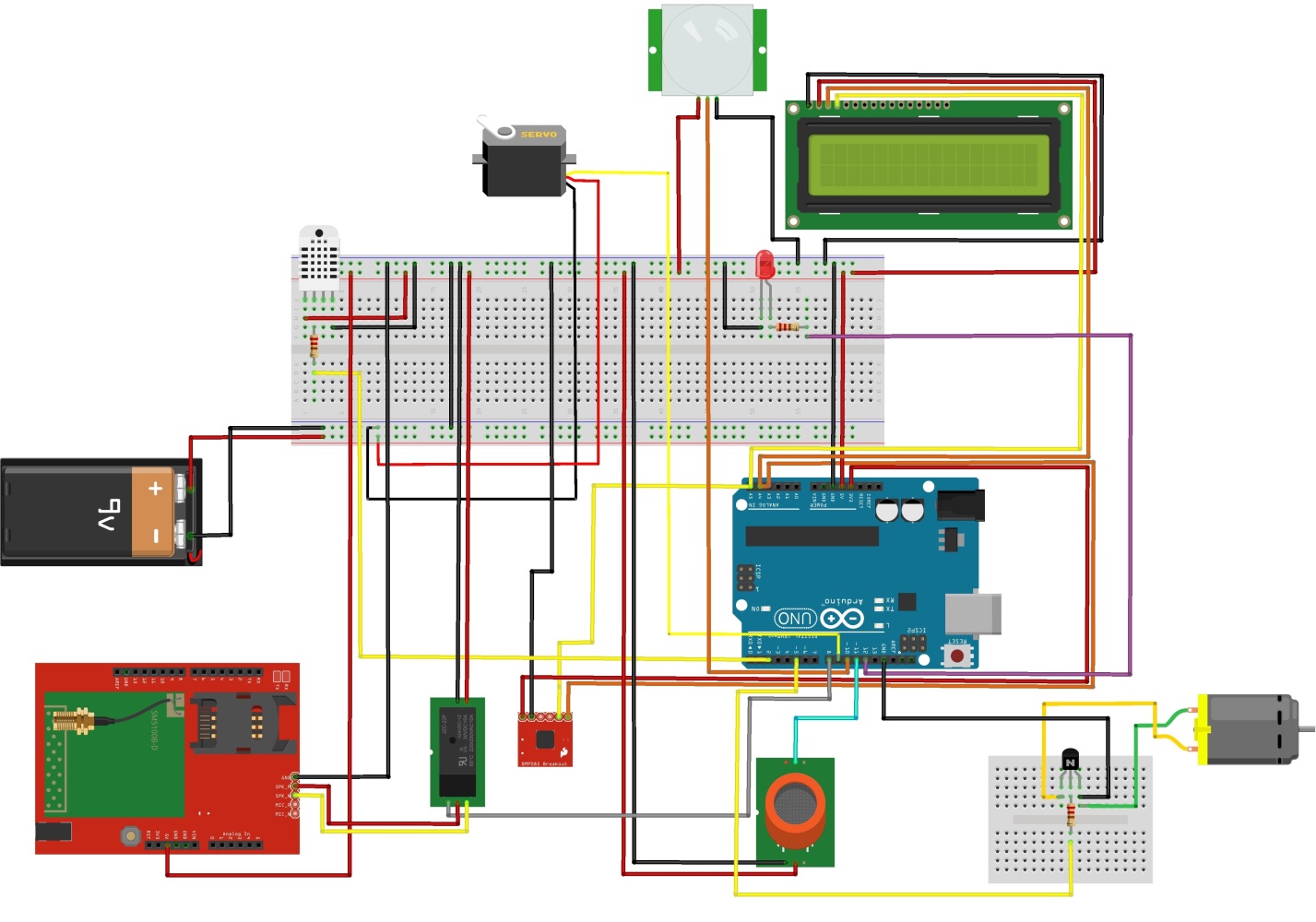
Модуль Барометра, построен на высокоточном чипе Bosch BMP085. Предназначен для определения барометрического давления, температуры и высоты. Он может широко измерять давление в диапазоне от 300hPa до 1100hPa, до 9000 м над уровнем моря, с супер высокой точностью 0.03hPa (0,25 м).

Этот датчик очень функционален, так как помимо измерения атмосферного давления и высоты он измеряет температуру.



**4. ПРАКТИЧЕСКАЯ ЧАСТЬ**

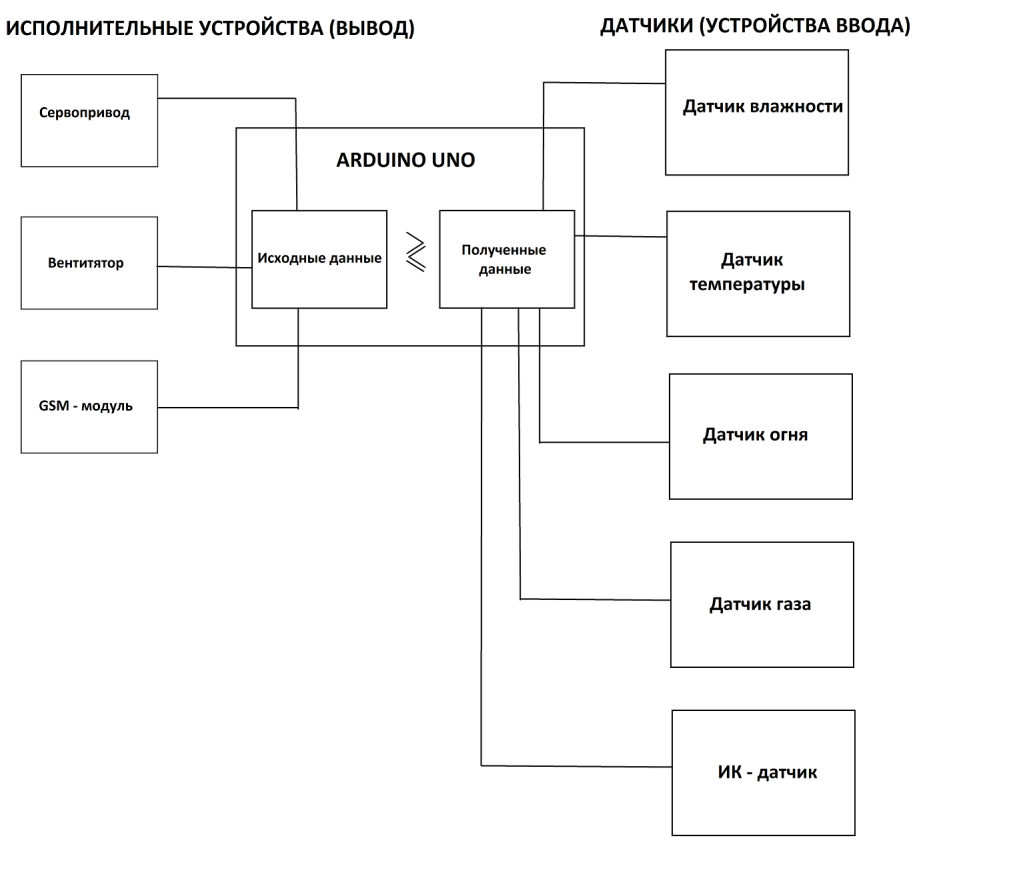
Устройство макета



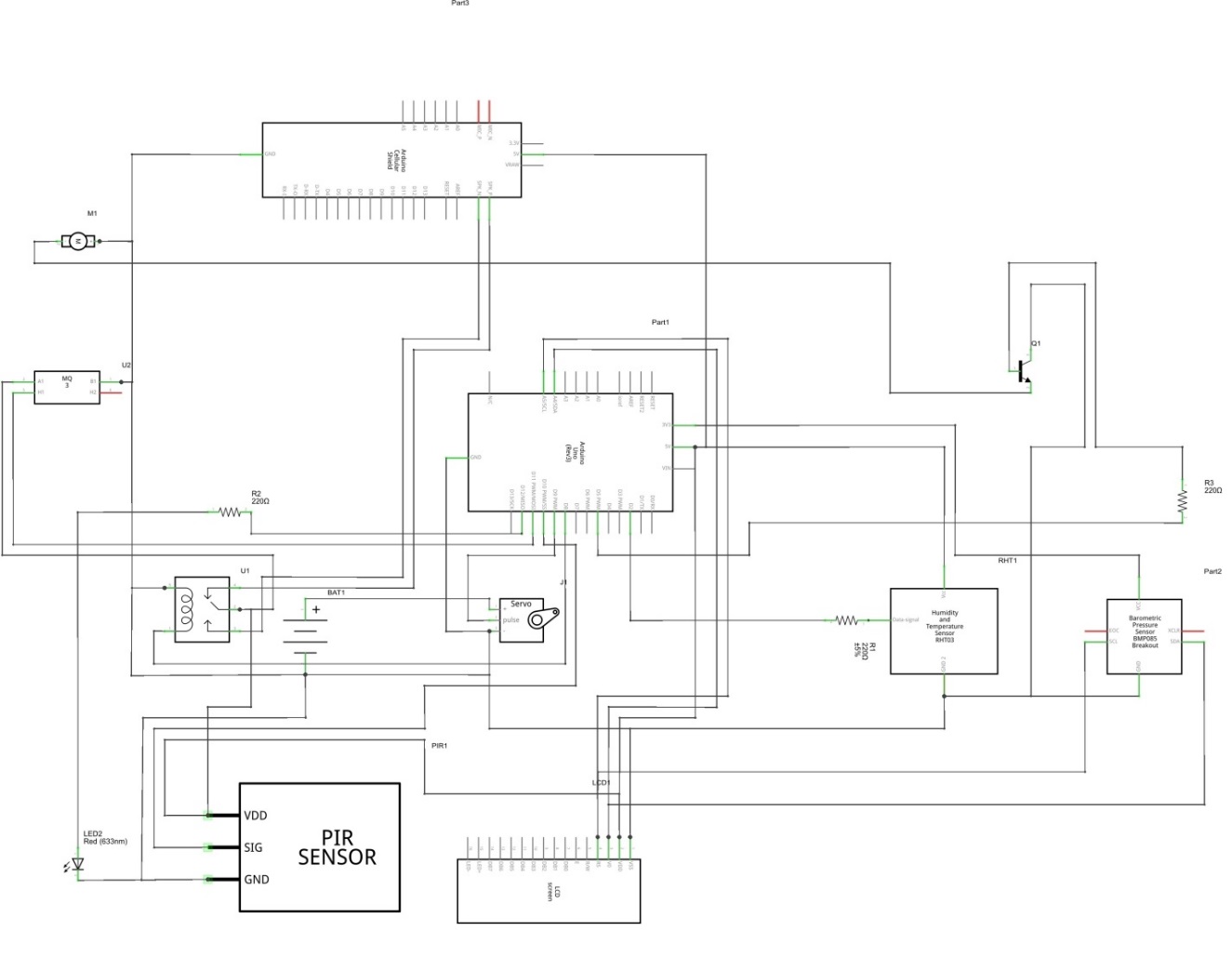
Состав системы:

* Arduino Uno – “сердце” всей системы
* Датчик влажности DHT11
* Датчик атмосферного давления и температуры BMP180
* GSM – модуль
* Серводвигатель
* Кулер
* Датчик газа MQ-2
* ИК датчик движения HC-SR501
* ЖК-дисплей
* Реле модуль

Блок-схема работы прототипа



Принципиальная схема макета



Полное описание работы прототипа

Наша автоматизированная система отслеживает параметры среды в помещении (влажность воздуха, температуру, пары газа и т.д.), пересылает полученные значения в логическое устройство (Arduino), сравнивает эти параметры с предельными (которые записаны в программу для выполнения условия) и отправляет команды на исполнительные устройства (мобильный телефон, вентилятор, сервопривод и т.д.), которые приводят полученные значения к заданным.

1. **Температура**

Ни для кого не секрет, что существуют определенные пределы комфортной температуры в помещении. Например, когда в помещении температура повышается до 25ºС и более, я чувствую себя не очень комфортно (слишком жарко), а если температура опускается ниже 18ºС, то мне становится холодно. Так вот в нашей автоматизированной системе при температуре 25ºС открывается окно для того, чтобы проветрить (охладить помещение). Вот так выглядит фрагмент программного кода, который отвечает за то, что будет происходить при повышении температуры до 25ºС:

Фрагмент 1.

if (t > 25)

{

myservo.write(45);

}

В первой строчке фрагмента указано условие, где температура принимает значение больше 25ºС. Далее команда myservo.write(45); включает сервопривод, который в свою очередь открывает окно для проветривания. При температуре меньше 23ºС окно закрывается:

Фрагмент 2.

if (t < 23)

{

myservo.write(0);

delay(1000);

myservo.detach();

}

Команда myservo.write(0); запускает сервопривод, который устанавливает окно в нулевое положение, т.е. закрывает. Команда delay(1000); устанавливает задержку в 1 с.(1000 мс.) в течение которой происходит закрывание окна. Команда myservo.detach(); выключает сервопривод для экономии энергии.

Но бывает так, что при жаркой погоде температура в помещении уже достигла 25ºС и продолжает расти. И даже открытое окно уже не может должным образом охлаждать температуру в помещении. Так вот для такого случая предусмотрен вентилятор. Он включается, когда температура превысит 30ºС. Но для того, чтобы вентилятор включался только в том случае, когда выполняется условие, заданное в программе, пришлось соорудить самодельный переключатель, который состоит из n-p-n транзистора, резистора на 10 кОм. Когда на базу транзистора через резистор поступает сигнал (1), то вентилятор включается.

Фрагмент 3.

if (t > 30)

{

digitalWrite(VEN, HIGH);

}

if (t < 26)

{

digitalWrite(VEN, LOW);

}

В коде произошло объявление о подключении вентилятора к 5 пину Arduino:

int VEN = 5;

pinMode(VEN, OUTPUT);

Строчка digitalWrite(VEN, HIGH); обозначает, что когда на резистор приходит значение HIGH, т.е. 1, вентилятор включается. Соответственно при значении LOW, т.е. 0, вентилятор выключается.

Таким образом, при температуре в помещении выше 30ºС включается вентилятор, работа которого вместе с открытым для проветривания окном будет способствовать понижению температуры в помещении. Когда же температура наконец-то начнёт падать и достигнет значения меньше 26ºС, то вентилятор выключится.

Также бывают такие случаи, (чаще всего в частных домах) когда во время вашего отсутствия в доме может произойти по каким либо причинам отключение отопительного газового котла. И до момента вашего возвращения домой помещение может значительно охладиться, тем более в зимнее время. Для того, чтобы избежать подобных неприятностей, мы предусмотрели автоматизированное информирование о произошедшем отключении системы отопления. Но обо всём поподробнее.

Итак, допустим, произошла остановка котла. Температура в доме начала стремительно падать и снизилась до значения меньше 15ºС.

В этот момент с Arduino на модуль реле приходит команда на включение. Модуль в свою очередь включает кнопку быстрого набора на сотовом телефоне, и происходит звонок со звуковым сообщением на телефон владельца дома, сообщающий, что дома выключилось отопление:

Фрагмент 4.

const int REL = 8;

pinMode(REL, OUTPUT);

if (t < 15.0)

{

digitalWrite(REL, HIGH);

}

Строчка digitalWrite(REL, HIGH); включает реле и включает звонок с телефона.

Также информирование происходит и при температуре выше 45ºС, потому что это может говорить о том, что в доме случился пожар:

Фрагмент 5.

if (t > 45)

{

digitalWrite(REL, HIGH);

}

1. **Влажность**

Помимо допустимых температурных параметров можно говорить об оптимальных значениях влажности воздуха в жилом помещении. Норма влажности воздуха в помещении находится в пределах от 30% до 60%, а 45% - самое оптимальное значение уровня влажности. Итак, при влажности воздуха больше 60% окно открывается:

Фрагмент 6.

if (h >= 60)

{

myservo.attach(SERVO);

myservo.write(45);

}

При влажности меньше 50% окно закрывается:

Фрагмент 7.

if (h < 50)

{

myservo.write(0);

delay(50);

myservo.detach();

}

Но мы столкнулись со следующей проблемой. Человек находится в комнате, уровень влажности воздуха в которой достиг 60%, соответственно, при заданном условии окно должно открыться. Но находящийся в комнате человек при открытом окне может пострадать от возникшего сквозняка и в дальнейшем простудиться (например, если это холодное время года).

Поэтому мы добавили в наш прототип инфракрасный датчик, который распознает присутствие человека в комнате. То есть, если в помещении находится человек, то окно не открывается. Как только человек уходит (т.е. датчик не наблюдает движения) окно сразу же открывается:

Фрагмент 8.

const int DVI = 10;

pinMode(DVI, INPUT);

val = digitalRead(DVI);

if (val == HIGH && h > 60)

{

digitalWrite(LED, LOW);

myservo.write(0);

delay(1000);

myservo.detach();

lcd.setCursor(14, 1);

lcd.print("WC");

Serial.println("Motion!");

}

else

{

digitalWrite(LED, HIGH);

myservo.attach(SERVO);

myservo.write(45);

lcd.setCursor(14, 1);

lcd.print("WO");

Serial.println("No motion!");

}

1. **Прочие датчики**

В нашем прототипе присутствует датчик огня, который при обнаружении возгорания информирует человека пронзительным звуковым сигналом и информирует звонком на сотовый телефон:

Фрагмент 9.

if (!digitalRead(FLAME))

{

digitalWrite(BUZ, LOW);

digitalWrite(REL, LOW);

}

else

{

digitalWrite(BUZ, HIGH);

}

Также в нашем прототипе присутствует датчик газа, который реагирует помимо паров газа на пары этилового спирта. Когда датчик улавливает пары этилового спирта или газа, он отправляет звуковой сигнал на телефон владельца:

Фрагмент 10.

if (digitalRead(GAS) == HIGH)

{

digitalWrite(REL, HIGH);

}

Код программы:

#include <Wire.h>

#include <Adafruit\_BMP085.h>

#include <LiquidCrystal\_I2C.h>

#include "DHT.h"

#define DHTPIN 2

#define DHTTYPE DHT11

#include "Servo.h"

const int SERVO = 9;

const int DVI = 10;

const int REL = 8;

const int FLAME = 3;

const int BUZ = 7;

int LED = 12;

int val = 0;

int angle = 1;

int VEN = 5;

const int GAS = 11;

DHT dht(DHTPIN, DHTTYPE);

Adafruit\_BMP085 bmp;

LiquidCrystal\_I2C lcd(0x27, 16, 2);

byte degree[8] =

{

B00110,

B01001,

B01001,

B00110,

B00000,

B00000,

B00000,

B00000,

};

Servo myservo;

void setup()

{

myservo.attach(SERVO);

Serial.begin(9600);

dht.begin();

lcd.begin();

lcd.backlight();

pinMode(VEN, OUTPUT);

pinMode(DVI, INPUT);

pinMode(LED, OUTPUT);

pinMode(REL, OUTPUT);

pinMode(FLAME, INPUT);

pinMode(BUZ, OUTPUT);

myservo.write(0);

if (!bmp.begin()) {

Serial.println("Could not find a valid BMP085 sensor, check wiring!");

while (1) {}

}

}

void loop()

{

val = digitalRead(DVI); // считывание состояния датчика

lcd.createChar(0, degree);

float h = dht.readHumidity();

float t = bmp.readTemperature();

int p = (bmp.readPressure()/133.3);

lcd.setCursor(0, 1);

lcd.print("Temp: ");

lcd.setCursor(5, 1);

lcd.print(t);

lcd.setCursor(10, 1);

lcd.write((byte) 0);

lcd.setCursor(11, 1);

lcd.print("C");

lcd.setCursor(11, 0);

lcd.print("H:");

lcd.setCursor(13, 0);

lcd.print(h);

lcd.setCursor(15, 0);

lcd.print("%");

lcd.setCursor(0, 0);

lcd.print("At.P: ");

lcd.setCursor(5, 0);

lcd.print(p);

lcd.setCursor(8, 0);

lcd.print("mm");

//Условия по открытию окна//////////////////////////////

if (h < 44)

{

myservo.write(0);

delay(50);

myservo.detach();

}

if (h == 23.0)

{

myservo.attach(SERVO);

}

if (h >= 23.1 && h < 55 && val == LOW)

{

myservo.detach();

}

if (h >= 45)

{

myservo.attach(SERVO);

myservo.write(45);

}

//Температура//////////////////////////////////////////

if (t < 15.0){

digitalWrite(REL, HIGH);

}

if (t > 45){

digitalWrite(REL, HIGH);

}

if (t > 25){

myservo.write(45);

}

if (t < 23){

myservo.write(0);

delay(1000);

myservo.detach();

}

if (t > 30){

digitalWrite(VEN, HIGH);

}

if (t < 26){

digitalWrite(VEN, LOW);

}

//Считывание с датчика движения////////////////////////////

if (val == HIGH){

digitalWrite(LED, LOW);

myservo.write(0);

delay(1000);

myservo.detach();

//delay(300000);

lcd.setCursor(14, 1);

lcd.print("WC");

Serial.println("Motion!");

}

else {

digitalWrite(LED, HIGH);

myservo.attach(SERVO);

myservo.write(45);

lcd.setCursor(14, 1);

lcd.print("WO");

Serial.println("No motion!");

}

if (t > 30){

myservo.attach(SERVO);

myservo.write(angle);

myservo.write(45);

}

//Датчик ГАЗА////////////////////////////

if (digitalRead(GAS) == HIGH){

digitalWrite(REL, HIGH);

}

//Датчик Огня////////////////////////////////

if (!digitalRead(FLAME)){

digitalWrite(BUZ, LOW);

//digitalWrite(REL, LOW);

}

else{

digitalWrite(BUZ, HIGH);

}

}

1. **ЗАКЛЮЧЕНИЕ**

Проведя опыты со сделанной моделью, мы убедились, что данный макет позволяет нам сократить количество каждодневных однообразных манипуляций в доме. Благодаря применению микроконтроллера Arduino макет получился полностью работоспособным (несмотря на огромное количество датчиков) и интересным в изготовлении и изучении, так как микроконтроллеры Arduino очень хороши для начального внедрения в мир микроэлектроники.

Практическая значимость. Результаты данной работы могут использоваться в дальнейшем для внедрения этой системы во многие жилые дома и квартиры для повышения комфортности проживания человека.

Личностная значимость. В ходе данной работы я приобрел навыки структурирования информации, работы с источниками, обработки больших объёмов исходных данных и программирования микроконтроллеров.

В дальнейшем работу в предложенном русле можно углубить (чем я сейчас и занимаюсь). А именно попробовать внедрить эту систему в свой дом и провести различные наблюдения, подсчёты и доработки устройства в случае необходимости.

**6. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

[1] Блум Дж. Изучаем Arduino: инструменты и методы технического волшебства. Пер. с англ. – СПб.:БХВ-Петербург, 2015 г. – 336 с.: ил.

[2] Кравченко А.В. “10 практических устройств на AVR -микроконтроллерах”. Книга 3. – “МК-Пресс”, СПб.: “КОРОНА - ВЕК”, 2011. – 416 с.

[3] Гололобов В. Н. «Умный дом» своими руками / В. Н. Гололобов. – М.: НТ Пресс, 2007. – 416с.:ил.-(В помощь радиолюбителю)

[4] Петин В. А. Проекты с использованием контроллера Arduino. – СПб.: БХВ-Петербург, 2014. – 400 с.: ил. – (Электроника)

[5] Соммер У. Программирование микроконтроллерных плат Arduino/Freeduino. – СПб.: БХВ-Петербург, 2012. – 256 с.: ил. – (Электроника)