|  |
| --- |
| МУНИЦИПАЛЬНОЕ АВТОНОМНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «ГИМНАЗИЯ №1 Г. БЛАГОВЕЩЕНСКА» |
| ИНДИВИДУАЛЬНЫЙ ИТОГОВЫЙ ПРОЕКТ  Предметная область «Информатика и ИКТ» и «Робототехника» |
| **Тема: «**Модель «умный дом» »**»**  Выполнил:  Саяпин Никита Александрович,  обучающийся 9Е класса  МАОУ «Гимназия №1  г. Благовещенска»  Научный руководитель:  Федченко Галина Михайловна,  Преподаватель БГПУ;  Руководитель проекта:  Саяпина Маргарита Николаевна,  учитель информатики и ИКТ  МАОУ «Гимназия №1  г. Благовещенска»  г. Благовещенск - 2016 г  Содержание |

|  |  |
| --- | --- |
| Введение | 3 - 4 |
| Глава 1. Технологическая часть | 5 - 7 |
| Глава 2. Разработка устройства | 8 - 9 |
| Выводы | 10 |
| Список литературы | 11 |
| Приложение | 12 - 19 |

**Введение**

Само понятие "умный" дом (англ. smart house) не такое уж молодое. Оно возникло в США в начале 70-х годов прошлого века, в недрах "Института интеллектуальных зданий". На тот момент под умным домом подразумевалось "здание, обеспечивающее продуктивное и эффективное использование рабочего пространства…".

Однако годом рождения современного "умного" дома можно считать 1978-й год. В этом году в США компании Х10 USA и Leviton разработали и внедрили в производство технологию управления бытовыми приборами по проводам бытовой электросети.

Распространение эти разработки получили в то время лишь на территории Северной Америки, ибо были рассчитаны на работу при напряжении 110 В и частоту сети 60 Гц. Тем не менее, именно этим фирмам человечество обязано появлением "невероятных чудес прогресса" - автоматически открывающихся дверей, включающегося по хлопку света и прочих "фокусов", которыми состоятельные американцы удивляли своих гостей, а голливудские фильмы - весь остальной мир.

Для конца 70-х годов технология X10 (именно такое название закрепилось и сохранилось за ней и поныне) являлась, конечно, революционной. Однако она была рассчитана на поддержку всего шести управленческих команд и, в основном, использовалась для управления электроосвещением. Но людям хотелось большего. "Умный" дом должен был становиться все "умнее".

С начала нового тысячелетия человечество шагает в эпоху новых технологических открытий, одним из которых является бытовая автоматизация. Время современного человека имеет огромную ценность и такие системы автоматизации как «умный дом» существенно экономят этот жизненно-важный ресурс. Включить кондиционер, выключить свет в прихожей, активировать ночную сигнализацию - это лишь маленький перечень действий, которые можно возложить на систему «умного дома». Но такие устройства имеют один минус - большую рыночную стоимость. Поэтому разработка относительно дешевой системы, с аналогичными возможностями получает все большую **актуальность.**

**Целью** работы является разработка модуля системы «умный дом», а так же разработка и исследование алгоритмов системы, позволяющих увеличить экономию ресурсов.

**Задачи проекта**

Создание архитектуры аппаратных средств «умного дома»

Анализ программно-аппаратных систем управления «умным домом»

Отладка стенда

**Содержание работы**

Обзор систем управления зданиями

Обзор готовых программных решений

Выбор аппаратных средств

Создание архитектуры аппаратных средств «умного дома»

Разработка требований к многофункциональному программно-аппаратному стенду

Разработка структуры программно-аппаратного стенда

**Экспериментальная часть**

Отладка стенда

Глава 1. **Технологическая часть**

Arduino — это электронный конструктор и удобная платформа быстрой разработки электронных устройств для новичков и профессионалов. Платформа пользуется огромной популярностью во всем мире благодаря удобству и простоте языка программирования, а также открытой архитектуре и программному коду. Устройство программируется через USB без использования программаторов.

Arduino позволяет компьютеру выйти за рамки виртуального мира в физический и взаимодействовать с ним. Устройства на базе  Arduino могут получать информацию об окружающей среде посредством различных датчиков, а также могут управлять различными исполнительными устройствами.

Микроконтроллер на плате программируется при помощи языка С++ и среды разработки Arduino. Проекты устройств, основанные на Arduino, могут работать самостоятельно, либо же взаимодействовать с программным обеспечением на компьютере (напр.: Flash, Processing, MaxMSP). Платы могут быть собраны пользователем самостоятельно или куплены в сборе. Программное обеспечение доступно для бесплатного скачивания. Исходные чертежи схем (файлы CAD) являются общедоступными, пользователи могут применять их по своему усмотрению.

В 2006  Arduino получила признание в категории Digital Communities на фестивале Ars Electronica Prix .

**Аппаратная часть платформы Arduino**

Существует несколько версий платформ Arduino. Последняя версия Leonardo базируется на микроконтроллере ATmega32u4.

[**Uno**](http://arduino.ru/Hardware/ArduinoBoardUno), как и предыдущая версия **[Duemilanove](http://arduino.ru/Hardware/ArduinoBoardDuemilanove)** построены на микроконтроллере Atmel ATmega328.

Старые версии платформы **[Diecimila](http://arduino.ru/Hardware/ArduinoBoardDiecimila)** и первая рабочая [Duemilanoves](http://arduino.ru/Hardware/ArduinoBoardDuemilanove)были разработаны на основе Atmel ATmega168 ,более ранние версии использовали ATmega8.

[**Arduino Mega2560**](http://arduino.ru/Hardware/ArduinoBoardMega2560), в свою очередь, построена на микроконтроллере ATmega2560.

Язык Arduino можно разделить на три раздела:

|  |  |  |
| --- | --- | --- |
| **Операторы**   * [setup](http://arduino.ru/Reference/Setup)() * [loop](http://arduino.ru/Reference/Loop)()   **Управляющие операторы**   * [if](http://arduino.ru/Reference/If) * [if...else](http://arduino.ru/Reference/Else) * [for](http://arduino.ru/Reference/For) * [switch case](http://arduino.ru/Reference/SwitchCase) * [while](http://arduino.ru/Reference/While) * [do... while](http://arduino.ru/Reference/DoWhile) * [break](http://arduino.ru/Reference/Break) * [continue](http://arduino.ru/Reference/Continue) * [return](http://arduino.ru/Reference/Return) * [goto](http://arduino.ru/Reference/Goto)   **Синтаксис**   * [; (semicolon)](http://arduino.ru/Reference/SemiColon) * [{} (curly braces)](http://arduino.ru/Reference/Braces) * [// (single line comment)](http://arduino.ru/Reference/Comments) * [/\* \*/ (multi-line comment)](http://arduino.ru/Reference/Comments)   **Арифметические операторы**   * [= (assignment)](http://arduino.ru/Reference/Assignment) * [+ (addition)](http://arduino.ru/Reference/Arithmetic)   + [(subtraction)](http://arduino.ru/Reference/Arithmetic) * [(multiplication)](http://arduino.ru/Reference/Arithmetic) * [/ (division)](http://arduino.ru/Reference/Arithmetic) * [% (modulo)](http://arduino.ru/Reference/Modulo)   **Операторы сравнения**   * [==](http://arduino.ru/Reference/If) (equal to) * [!=](http://arduino.ru/Reference/If) (not equal to) * [<](http://arduino.ru/Reference/If) (less than) * [>](http://arduino.ru/Reference/If) (greater than) * [<=](http://arduino.ru/Reference/If) (less than or equal to) * [>=](http://arduino.ru/Reference/If) (greater than or equal to) * Логические операторы * [&& (И)](http://arduino.ru/Reference/Boolean) * [|| (ИЛИ)](http://arduino.ru/Reference/Boolean) * [! (Отрицание)](http://arduino.ru/Reference/Boolean)   **Унарные операторы**   * [++](http://arduino.ru/Reference/Increment) (increment) * [--](http://arduino.ru/Reference/Increment) (decrement) * [+=](http://arduino.ru/Reference/IncrementCompound) (compound addition) * [-=](http://arduino.ru/Reference/IncrementCompound) (compound subtraction) * [\*=](http://arduino.ru/Reference/IncrementCompound) (compound multiplication) * [/=](http://arduino.ru/Reference/IncrementCompound) (compound division) | **Данные**  **Константы**   * [HIGH](http://arduino.ru/Reference/Constants) | [LOW](http://arduino.ru/Reference/Constants) * [INPUT](http://arduino.ru/Reference/Constants) | [OUTPUT](http://arduino.ru/Reference/Constants) * [true](http://arduino.ru/Reference/Constants) | [false](http://arduino.ru/Reference/Constants) * [Целочисленные константы](http://arduino.ru/Reference/IntegerConstants) * [Константы с плавающей запятой](http://arduino.ru/Reference/Fpconstants)   **Типы данных**   * [boolean](http://arduino.ru/Reference/BooleanVariables) * [char](http://arduino.ru/Reference/Char) * [byte](http://arduino.ru/Reference/Byte) * [int](http://arduino.ru/Reference/Int) * [unsigned int](http://arduino.ru/Reference/UnsignedInt) * [word](http://arduino.ru/Reference/Word) * [long](http://arduino.ru/Reference/Long) * [unsigned long](http://arduino.ru/Reference/UnsignedLong) * [float](http://arduino.ru/Reference/Float) * [double](http://arduino.ru/Reference/Double) * [string](http://arduino.ru/Reference/String) - массив символов * [String](http://arduino.ru/Reference/StringObject) - объект класса * [массив (array)](http://arduino.ru/Reference/Array) * [void](http://arduino.ru/Reference/Void)   **Преобразование типов данных**   * [char()](http://arduino.ru/Reference/CharCast) * [byte()](http://arduino.ru/Reference/ByteCast) * [int()](http://arduino.ru/Reference/IntCast) * [long()](http://arduino.ru/Reference/LongCast) * [float()](http://arduino.ru/Reference/FloatCast)   **Область видимости переменных и квалификаторы**   * [Область видимости](http://arduino.ru/Reference/Scope) * [static](http://arduino.ru/Reference/Static) * [volatile](http://arduino.ru/Reference/Volatile) * [const](http://arduino.ru/Reference/Const) | **Функции**  **Цифровой ввод/вывод**   * [pinMode](http://arduino.ru/Reference/PinMode)() * [digitalWrite](http://arduino.ru/Reference/DigitalWrite)() * [digitalRead](http://arduino.ru/Reference/DigitalRead)() * **Аналоговый ввод/вывод** * [analogRead](http://arduino.ru/Reference/AnalogRead)() * [analogReference()](http://arduino.ru/Reference/AnalogReference) * [analogWrite](http://arduino.ru/Reference/AnalogWrite)()   **Дополнительные фунции ввода/вывода**   * [tone](http://arduino.ru/Reference/Tone)() * [noTone](http://arduino.ru/Reference/NoTone)() * [shiftOut](http://arduino.ru/Reference/ShiftOut)() * [pulseIn](http://arduino.ru/Reference/PulseIn)() * **Работа со временем** * [millis](http://arduino.ru/Reference/Millis)() * [micros](http://arduino.ru/Reference/Micros)() * [delay](http://arduino.ru/Reference/Delay)() * [delayMicroseconds](http://arduino.ru/Reference/DelayMicroseconds)()   **Математические функции**   * [min](http://arduino.ru/Reference/Min)() * [max](http://arduino.ru/Reference/Max)() * [abs](http://arduino.ru/Reference/Abs)() * [constrain](http://arduino.ru/Reference/Constrain)() * [map](http://arduino.ru/Reference/Map)() * [pow](http://arduino.ru/Reference/Pow)() * [sq](http://arduino.ru/Reference/Sq)() * [sqrt](http://arduino.ru/Reference/Sqrt)()   **Тригонометрические функции**   * [sin](http://arduino.ru/Reference/Sin)() * [cos](http://arduino.ru/Reference/Cos)() * [tan](http://arduino.ru/Reference/Tan)()   **Генераторы случайных значений**   * [randomSeed](http://arduino.ru/Reference/RandomSeed)() * [random](http://arduino.ru/Reference/Random)()   **Внешние прерывания**   * [attachInterrupt()](http://arduino.ru/Reference/AttachInterrupt) * [detachInterrupt()](http://arduino.ru/Reference/DetachInterrupt)   **Функции передачи данных**   * [Serial](http://arduino.ru/Reference/Serial) |

**Глава 2. Разработка устройства**

Проектирование системы можно разделить на 2 части: аппаратную и программную.

**Аппаратная часть** системы представлена платформой Arduino (рис.1) с расширениями и физическими модулями. Arduino – это инструмент для проектирования электронных устройств, более тесно взаимодействующих с окружающей физической средой, чем стандартные персональные компьютеры. В данной работе будет использована версия платформы Arduino UNO – самая популярная версия базовой платформы Arduino с USB-интерфейсом и возможностью подключения большого разнообразия плат расширения. Эта платформа предназначена для физических расчетов (physical computing) с открытым программным кодом, построенная на простой печатной плате с современной средой для разработки программного обеспечения .

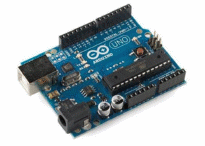


Рисунок 1 - Внешний вид платформы Arduino MEGA

Arduino использует микроконтроллер Atmega328, который имеет 32 Кб флеш памяти. Этого будет вполне достаточно для выполнения возложенной на платформу задачи. Остальная обработка будет распределена на веб-ресурс.

Микроконтроллеры Arduino отличаются наличием предварительно прошитого в них загрузчика. С помощью этого загрузчика пользователь загружает свою программу в микроконтроллер без использования традиционных отдельных аппаратных программаторов и соединяется с компьютером через USB-интерфейс.

**Программная часть** системы представлена в программном коде микроконтроллера (Arduino скетч) (см. Приложение 1)

Модуль управления кроватью

Модуль управления вентилятором

Модуль управления светом

**Выводы**

Разрабатываемая система управления исключит вероятность того, что неиспользуемые в данный момент приборы будут потреблять электроэнергию. Это позволит улучшить энергосбережение используемого помещения и получить экономическую выгоду.

В данный момент работа не окончена. Идет стадия аппаратного проектирования системы «умный дом».

Список литературы

1. В. Архипов «Системы для «интеллектуального» здания» - "СтройМаркет", № 45 1999 г.

2. Mike Riley «Programming Your Home Automate with Arduino, Android, and Your Computer» - « The Pragmatic Bookshelf Dallas, Texas • Raleigh, North Carolina », 2012 г.

3. Информация сайта <http://habrahabr.ru/post/118817/>

4. Информация сайта <http://ab-log.ru/>

5. Информация сайта <http://a-bolshakov.ru/index/0-163>

6. Информация сайта <https://xively.com/?from_cosm=true>

7. Информация сайта <http://www.tesli.com/ru/service/automation/smarthouse/>

8. Информация сайта <http://www.fibaro.com>

Приложение 1.

#include <Keypad.h>

#include <Servo.h>

const int

pin = 8,

DotDelay = 200;

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

//define the cymbols on the buttons of the keypads

char hexaKeys[ROWS][COLS] = {

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

};

byte rowPins[ROWS] = {31,33,35,37}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {39,41,43,45}; //connect to the column pinouts of the keypad

int a;

//initialize an instance of class NewKeypad

Keypad customKeypad = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

char text[]="sos";

Servo servo; //объявляем переменную servo типа Servo

void setup()

{

Serial.begin(9600);

pinMode(pin, OUTPUT);

servo.attach(9);

}

void dot()

{

digitalWrite(pin,HIGH);

delay(DotDelay);

digitalWrite(pin,LOW);

delay(DotDelay);

}

void dash()

{

digitalWrite(pin,HIGH);

delay(DotDelay \* 3);

digitalWrite(pin,LOW);

delay(DotDelay);

}

void A()

{

dot();

dash();

}

void B()

{

dash();

dot();

dot();

dot();

}

void C()

{

dash();

dot();

dash();

dot();

}

void D()

{

dash();

dot();

dot();

}

void E()

{

dot();

}

void f()

{

dot();

dot();

dash();

dot();

}

void G()

{

dash();

dash();

dot();

}

void H()

{

dot();

dot();

dot();

dot();

}

void I()

{

dot();

dot();

}

void J()

{

dot();

dash();

dash();

dash();

}

void K()

{

dash();

dot();

dash();

}

void L()

{

dot();

dash();

dot();

}

void M()

{

dash();

dash();

}

void N()

{

dash();

dot();

}

void O()

{

dash();

dash();

dash();

}

void P()

{

dot();

dash();

dot();

}

void Q()

{

dash();

dash();

dot();

dash();

}

void R()

{

dot();

dash();

dot();

}

void S()

{

dot();

dot();

dot();

}

void T()

{

dash();

}

void U()

{

dot();

dot();

dash();

}

void V()

{

dot();

dot();

dot();

dash();

}

void W()

{

dot();

dash();

dash();

}

void X()

{

dash();

dot();

dot();

dash();

}

void Y()

{

dash();

dot();

dash();

dash();

}

void Z()

{

dash();

dash();

dot();

dot();

}

void loop()

{

char customKey = customKeypad.getKey();

char customKey1 = customKeypad.getKey();

if (customKey)

{

Serial.println(customKey);

if (customKey == '5')

{

int a=0;

servo.write(180);

analogWrite(pin, 0);

delay(2000);

}

else

{

int a=1;

Serial.println(a);

servo.write(90);

analogWrite(pin, 400);

delay(1000);

morseText(text);

analogWrite(pin, 400);

delay(5000);

morseText(text);

analogWrite(pin, 400);

delay(10000);

morseText(text);

analogWrite(pin, 400);

delay(15000);

morseText(text);

analogWrite(pin, 400);

delay(20000);

morseText(text);

analogWrite(pin, 0);

delay(2000);

morseText(text);

}

}

}

void morseText(char text[])

{

int l=strlen(text);

for (int i=0; i<l; i++)

{

morseLetter(text[i]);

Serial.print(text[i]);

}

Serial.println();

}

void morseLetter(char letter)

{

if (letter==' ') delay(DotDelay \* 6);

if (letter=='a') A();

if (letter=='b') B();

if (letter=='c') C();

if (letter=='d') D();

if (letter=='e') E();

if (letter=='f') f();

if (letter=='g') G();

if (letter=='h') H();

if (letter=='i') I();

if (letter=='g') G();

if (letter=='k') K();

if (letter=='l') L();

if (letter=='m') M();

if (letter=='n') N();

if (letter=='o') O();

if (letter=='p') P();

if (letter=='q') Q();

if (letter=='r') R();

if (letter=='s') S();

if (letter=='t') T();

if (letter=='u') U();

if (letter=='v') V();

if (letter=='w') W();

if (letter=='x') X();

if (letter=='y') Y();

if (letter=='z') Z();

}

Свет

#define led\_pin1 13

#define led\_pin2 12

#define led\_pin3 11

#define led\_pin4 10

#define led\_pin5 9

#define ldr\_pin A0

#define pot\_pin A1

void setup()

{

pinMode(led\_pin1, OUTPUT);

pinMode(led\_pin2, OUTPUT);

pinMode(led\_pin3, OUTPUT);

pinMode(led\_pin4, OUTPUT);

pinMode(led\_pin5, OUTPUT);

}

void loop()

{

int lightness = analogRead(ldr\_pin);

int threshold = analogRead(pot\_pin);

boolean tooDark = (lightness < threshold);

if (tooDark)

{

digitalWrite(led\_pin1, HIGH);

digitalWrite(led\_pin2, HIGH);

digitalWrite(led\_pin3, HIGH);

digitalWrite(led\_pin4, HIGH);

digitalWrite(led\_pin5, HIGH);

}

else

{

digitalWrite(led\_pin1, LOW);

digitalWrite(led\_pin2, LOW);

digitalWrite(led\_pin3, LOW);

digitalWrite(led\_pin4, LOW);

digitalWrite(led\_pin5, LOW);

}

}