Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет Телекоммуникаций

Кафедра Защита информации

Дисциплина Цифровые и микропроцессорные устройства

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

СИГНАЛИЗАЦИЯ НА PIR-СЕНСОРЕ

БГУИР КП 1-45 01 02 021 ПЗ

|  |  |
| --- | --- |
| Студент гр. 562101  Руководитель | Семеник В. В.  Петров С.Н. |
|  |  |

Минск 2017

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение…………………………………………………………………. | 3 |
| 1. Описание используемого микроконтроллера………………………. | 4 |
| 2. Структурная схема устройства………………………...……………. | 11 |
| 3. Электрическая принципиальная схема……………………………... | 12 |
| 4. Программа для микроконтроллера………………………………….  5. Программа отправки сообщения на электронную почту (на языке Python)…………………………………………………………………… | 13  14 |
| Заключение……………………………………………………………… | 15 |
| Список литературы……………………………………………………... | 16 |
| Приложение А…………………………………………………………...  Приложение Б…………………………………………………………… | 17  23 |
|  |  |
|  |  |

**ВВЕДЕНИЕ**

В современном мире вместе с технологиями в инфокоммуникационной среде развивается вся микроэлектроника в целом, а также микропроцессорные системы. В современной технике все системы телекоммуникаций строятся на основе программируемых логических схем и микропроцессоров. Современные специалисты по инфокоммуникациям должны обладать знаниями в области микропроцессорных устройств.

Выполнение курсовой работы по теме «Сигнализация на PIR-сенсоре» по дисциплине «Цифровые и микропроцессорные устройства» является углубленным методом изучения микропроцессорных систем и дисциплины в целом.

**Сигнализация на PIR-сенсоре –** система, которая реагирует на сигнал поступивший от датчика движения (PIR-сенсор) и сообщает о проникновении или подает предупреждающий сигнал на пункт приема сигнала. В данном проекте сигнализация устанавливается и снимается с режима работы при помощи приемо-передающих модулей, а также, устройство отправляет предупредительное сообщение на указанный адрес электронной почты.

Преимущества сигнализации на PIR-сенсоре:

– автономность в работе;

– мобильность (легко переустановить в другое место);

– возможность вариации режимов работы устройства.

Недостатки:

– дальность работы приемопередатчиков составляет около 1 км;

– необходимо устанавливать в месте что бы нельзя было обойти PIR-сенсор;

– при желании отправлять письмо о проникновении на электронную почту необходим доступ к сети интернет.

**1 ОПИСАНИЕ ИСПОЛЬЗУЕМОГО МИКРОКОНТРОЛЛЕРА И ПЕРИФЕРИЙНЫХ УСТРОЙСТВ**

**1.1 Микроконтроллер**

ATmega328/P - маломощный CMOS 8-разрядный микроконтроллера на основе расширенной архитектуры RISC с AVR®. Для выполнение мощных команд за один такт в ATmega328/P пропускная способность достигает 1MIPS на МГц. Это позволяет создать система оптимизации энергопотребления устройства в зависимости от обработки скорости.

– Периферийные Характеристики

– два 8-разрядных таймера/счетчика с отдельным предделителем и режим сравнения

– один 16-разрядный Таймер/счетчик с отдельным предделителем, режим сравнения, режим захвата

– счетчик реального времени с отдельным генератором

– шесть каналов PWM

– 8-канальный 10-разрядный АЦП в TQFP и пакет QNF/MLA package

– измерение Температуры

– 6-канальный 10-разрядный АЦП в корпусе PDIP package

– измерение Температуры

– два мастер/подчиненный последовательный SPI интерфейс

– один программируемый последовательный USART

– один байт-ориентированный 2-проводной последовательный интерфейс (Филипс I2C совместимый)

– программируемый Сторожевой таймер с отдельным генератором

– один на чип аналогового компаратора

– прерывание и пробуждение изменить PIN-код

– специальные Функции Микроконтроллера

– схема сброса при подаче питания и Программируемый коричневый обнаружения

– внутренний Калиброванный Генератор

– внешние и внутренние источники прерывания

– шесть режимов сна: холостой ход, снижение шума АЦП, экономии энергии, питания, ожидания и расширенный режим ожидания

– ввода/вывода и packages

– 23 программируемые линии Ввода/Вывода

– 28-выводных корпусах PDIP, 32-выводном корпусе TQFP, 28-колодка корпус QFN/MLF и 32-колодка корпус QFN/MLF

– рабочее Напряжение:

– 1.8 - 5.5 в

– диапазон Температур:

– от -40°C до 105°C

– скорость Класс:

– 0 - 4мгц @ 1.8 - 5.5 в

– 0 - 10мгц @ 2.7 - 5.5 в

– 0 - 4.5 20мгц @ - 5.5 в

– расход энергии на 1 МГц, 1.8 в, 25°C

– активный режим: 0,2 мА

– режим пониженного энергопотребления: 0,1 мкА

– режим экономии энергии: 0.75 мкА (в том числе 32 кГц RTC)

Блок-схема выбранного микроконтроллера приведена на рисунке 1.1

Распиновка для двух видов корпусов на рисунке 1.2

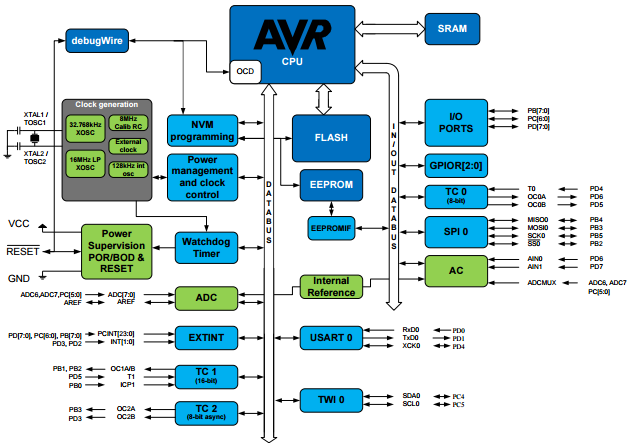


Рисунок 1.1 – Блок-схема микроконтроллера ATmega328/P

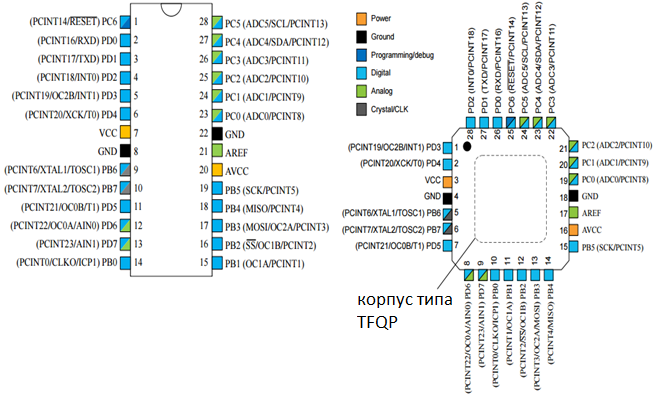


Рисунок 1.2 – Распиновка в двух видах корпусов для ATmega328/P

**1.2 Устройство индикации и ввода**

Индикация построена на LCD-индикаторе (дисплее). Жидкокристаллический дисплей (англ. liquid crystal display, LCD) – плоский

дисплей на основе жидких кристаллов, а также устройство (монитор, телевизор) на основе такого дисплея.

Простые приборы с дисплеем (электронные часы, телефоны, плееры, термометры и пр.) могут иметь монохромный или 2—5-цветный дисплей. Многоцветное изображение формируется с помощью RGB-триад.

Дисплей на жидких кристаллах используется для отображения графической или текстовой информации в компьютерных мониторах (также и в ноутбуках), телевизорах, телефонах, цифровых фотоаппаратах, электронных, навигаторах, планшетах, электронных переводчиках, калькуляторах, часах и т. п., а также во многих других электронных устройствах.

На 2008 год в большинстве настольных мониторов на основе TN- (и некоторых \*VA) матриц, а также во всех дисплеях ноутбуков используются матрицы с 18-битным цветом (6 бит на каждый RGB-канал), 24-битность эмулируется мерцанием с дизерингом.

Настройка режима работы осуществляется при помощи 5 кнопок (UP, LEFT, DOWN, RIGHT, SELECT). При нажатии на одну из клавиш уровень сигнала поступающего на аналоговый вход 0 меняется.

Устройство индикации и ввода изображено на рисунке 1.3

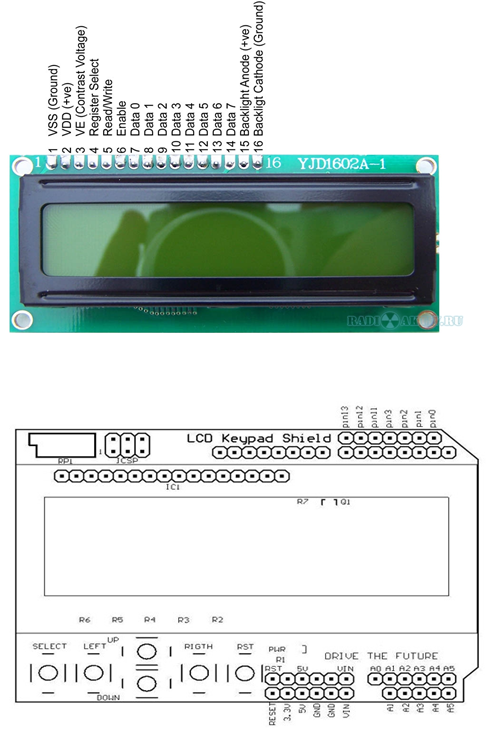


Рисунок 1.3 – Устройство индикации и ввода.

**1.3 PIR модуль, его назначение и принцип работы**

PIR датчики движения по сути состоят из пироэлектрического чувствительного элемента (цилиндрическая деталь с прямоугольным кристаллом в центре), который улавливает уровень инфракрасного излучения. Все вокруг излучает небольшой уровень радиации. Чем больше температура, тем выше уровень излучения. Датчик фактически разделен на две части. Это обусловлено тем, что нам важен не уровень излучения, а непосредственно наличие движение в пределах его зоны чувствительности. Две части датчика установлены таким образом, что если одна половина улавливает больший уровень излучения, чем другая, выходной сигнал будет генерировать значение high или low.

Пироэлектрический датчик движения состоит из двух основных частей. Каждая из частей включает в себя специальный материал, чувствительный к инфракрасному излучению. В данном случае линзы особо не влияют на работу датчика, так что мы видим два участка чувствительности всего модуля. Когда датчик находится в состоянии покоя, оба сенсора определяют одинаковое количество излучения. Например, это может быть излучение помещения или окружающей среды на улице. Когда теплокровный объект (человек или животное), проходит мимо, он пересекает зону чувствительности первого сенсора, в результате чего на модуле ПИР датчика генерируются два различных значения излучения. Когда человек покидает зону чувствительности первого сенсора, значения выравниваются. Именно изменения в показаниях двух датчиков регистрируются и генерируют импульсы HIGH или LOW на выходе.

На рисунке 1.4 изображен PIR модуль.

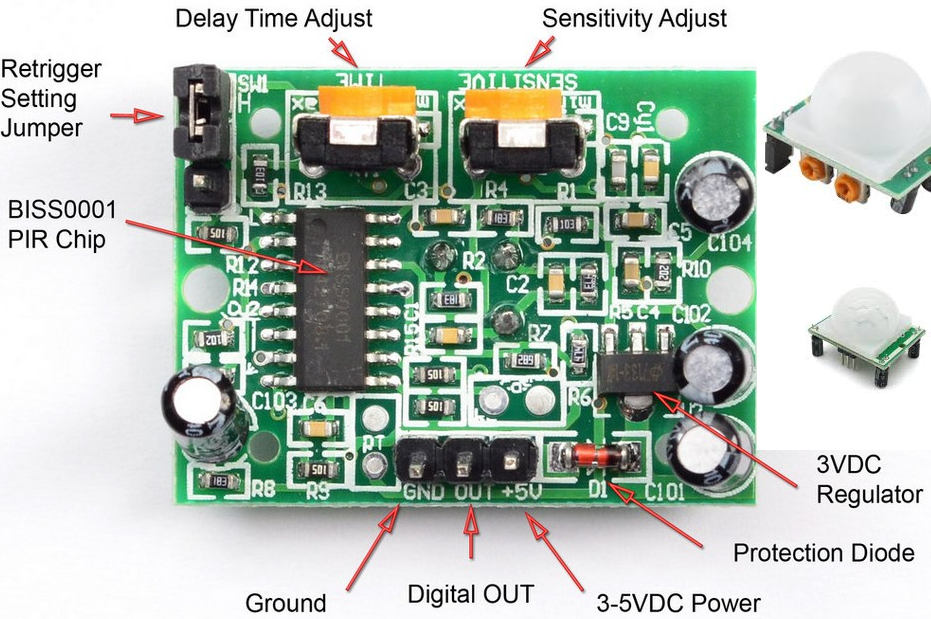


Рисунок 1.4 – PIR модуль.

**1.4 Приемо-передатчик NRF24L01**

Существует целая серия радио модулей 2.4 ГГц, которые созданы на базе чипа Nordic Semiconductor nRF24L01+ Nordic nRF24L01+ включает в себя 2.4 ГГц RF трансивер, логику, которая поддерживает высокоскоростной SPI интерфейс для подключения и обмена данными.

Технические характеристики:

-Частота передачи/приема 2.4ггц.  
- Дальность до 100 м.В помещении - до 30 м.   
- Скорость до 2 Мб.Возможные варианты: 250kbps, 1Mbps и 2Mbps.  
- Интерфейс взаимодействия с микроконтроллером- SPI. Выводы толеранты к 5 вольтам.  
- Напряжение: 3-3.6В (рекомендуется 3,3) В. При попытке подключения 5 В вывод из строя маловероятен,но работать устройство отказывается..  
- Максимальная выходная мощность: 0 dBm  
- Коэффициент усиления антенны (пиковая): 2dBm  
-Количество каналов: 126. Нулевой канал начинается с 2400мгц и далее с шагом 1 мгц,например 70 канал находится соответственно на 2470мгц.

На рисунке 1.5 изображен модуль nRF24L01 и его печатная плата.

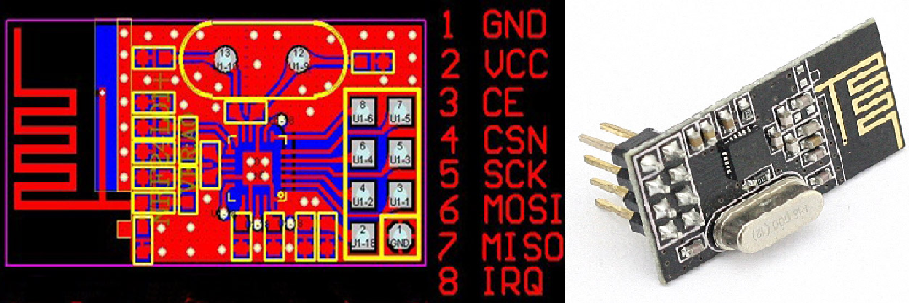
****

Рисунок 1.5 – модуль nRF24L01 и его печатная плата.

**2 СТРУКТУРНАЯ СХЕМА УСТРОЙСТВА**

Структурная схема сигнализации состоит из 2-х основных блоков:

1. Управляющую сторону (приёмо-передатчик)

– Устройство ввода – ввод нужной информации;

– Микроконтроллер – обработка введенной информации;

– Блок индикации – демонстрация обработанной информации;

– Приёмо-передатчик – передает данные о режиме работы и принимает данные от информационной стороны;

1. Информационную сторону (приёмо-передатчик)

– PIR-сенсор – получает данные о движении;

– Приёмо-передатчик – принимает данные о режиме работы и пересылает данные с датчика движения на управляющую сторону;

– Микроконтроллер – обработка введенной информации;

Структурная схема данного проекта представлена в приложении Б

.

**3 ЭЛЕКТРИЧЕСКАЯ ПРИНЦИПИАЛЬНАЯ СХЕМА**

Принципиальная схема данного проекта достаточно проста. В ней присутствует: две платы Arduino UNO (LCD keypad shield, PIR sensor и два приемо-передатчика nRF24L01), 2 резистора (в том числе подстроечный U1) и один светодиод.

Принципиальная схема данного устройства представлена в приложении Б.

Перечень используемых элементов приведен в приложении Б.

**4 ПРОГРАММА ДЛЯ МИКРОКОНТРОЛЛЕРА**

**4.1 Алгоритм функционирования управляющего устройства**

При подаче питания на управляющей стороне включается меню выбора режима работы сигнализации, при выборе количества часов работы начиная с данного момента времени, модуль nRF24L01 устанавливается в режим передачи значения режима на информационную сторону. После приема значения от управляющей стороны информационная сторона начинает пересылать, в течении установленного времени, информацию (управляющая сторона становится в режим приема, информационная в режим передачи) с датчика движения (PIR сенсора). Обрабатывая эту информацию, управляющая сторона сигнализирует пользователю о проникновении. Так же информационная сторона при наличии доступа к сети интернет отправляет на электронную почту владельца сообщение о проникновении.

Блок-схема программы управляющей стороны приведена в приложении Б.

**4.2 Алгоритм функционирования информационного устройства**

При подаче питания на информационной стороне начинает работать основной цикл программы, который устанавливает приемо-передающий модуль в режим приема (слушает канал в надежде приема времени работы). После приема информации о времени работы nRF-модуль устанавливается в режим передачи и на протяжении установленного времени передает информацию с Pir sensor на управляющую сторону. Так же информационный модуль отправляет электронное письмо на почту владельца (при проникновении).

Блок-схема программы информационной стороны приведена в приложении Б.

**5 ПРОГРАММА ОТПРАВКИ СООБЩЕНИЯ НА ЭЛЕКТРОННУЮ ПОЧТУ (НА ЯЗЫКЕ PYTHON)**

Программа на языке Python осуществляет отправку сообщения на электронную почту. Принцип работы основан на считывании данных поступающих по последовательному интерфейсу USART через USB порт компьютера. Программа проверяет входящие сообщения и если находит сообщение о движение в пределах датчика, то происходит отправка сообщения на почту.

Блок-схема программы отправки сообщения на электронную почту представлена в приложении Б.

**ЗАКЛЮЧЕНИЕ**

Основной целью проекта была разработка сигнализации на базе микроконтроллера Atmega328/P интегрированного в плату Arduino UNO.

В ходе создания проекта были изучены такие программы, как Atmel Studio, Arduino IDE, консольный редактор NANO и VIM, Proteus, Fritzing и Paint, при помощи которых он и был реализован.

Код прошивки был создан в Arduino IDE, проверка работоспособности программы и разработанного устройства производилась с помощью системы автоматизированного проектирования PROTEUS, позволяющая моделировать работу электронных схем, а также непосредственно на самих платах.

Подводя итоги проделанной работы можно сделать вывод, что микроконтроллеры AVR – это подходящая платформа для глубокого изучения микропроцессоров и программируемых логических схем. Так же можно сказать, что платформа разработки программируемых устройств такая как Arduino является очень перспективной и удобной не только в обучении, но и в реализации серьезных проектов.

**СПИСОК ЛИТЕРАТУРЫ**

1. Токхайм Р. Микропроцессоры: Курс и упражнения / под ред. А. Г. Грасевича. – М.: Энергоатомиздат, 1987. – 338 с.

2. Микропроцессорные системы [Электронный ресурс] : электрон. учеб. пособие / О. В. Непомнящий, Е. А. Вейсов, Г. А. Скотников, М. В. Савицкая. – Электрон. дан. (4 Мб). – Красноярск : ИПК СФУ, 2009. – (Микропроцессорные исследования) режим доступа : [http://www.tverhtk.ru/library/predmets/pc\_systems/Mikroprocessornye\_sistemy\_200 9.pdf](http://www.tverhtk.ru/library/predmets/pc_systems/Mikroprocessornye_sistemy_200%209.pdf)

3. Евстифеев, А.В. Микроконтроллеры AVR семейства Mega : руководство пользователя [Текст] / А.В. Евстифеев. – М.: Изд. дом «Додэка-XXI», 2007. – 592 с.

4. Белов, А.В. Разработка устройств на микроконтроллерах AVR [Текст] / А.В. Белов. – М. : Наука и техника, 2012. – 528 с.

5. Хартов, В.Я. Микроконтроллеры AVR. Практикум для начинающих [Текст] / В.Я. Хартов. – М. : Изд-во МГТУ им. Н.Э. Баумана, 2012. – 240 с.

6. Трамперт В. Измерение, управление и регулирование с помощью AVR микроконтроллеров / В. Трамперт. – К.: МК-Пресс, 2006. – 200 с.

7. Шпак, Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров [Текст] / Ю.А. Шпак. – К. : МК-Пресс, 2006 – 400 с.

8. «Цифровые и микропроцессорные устройства. Микроконтроллеры AVR. Лабораторный практикум», Петров С.Н., 2016 г.

9. [Электронный ресурс] – «Техническое описание микроконтроллера ATmega8515», Atmel Corporation 2004.

10. [Электронный ресурс] – http://wiki.amperka.ru/.

11. [Электронный ресурс] – https://ru.wikipedia.org/wiki/Микроконтроллер

12. [Электронный ресурс] – http://www.atmel.com/ru/ru/products/microcon trollers/default.aspx?src=parent.

13. [Электронный ресурс] – «ЭРУД ЦиМПУ – Электронная библиотека БГУИР», Прищепа С.Л., 2013 г.

**ПРИЛОЖЕНИЕ А**

**Листинг программы**

Код для управляющей стороны:

#include <LiquidCrystal.h>

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

// индитификатор передачи, "труба"

const uint64\_t pipe = 0xF0F1F2F3F4LL;

RF24 radio(3, 10); // CE, CSN

void setup() {

//предворительная настройка lcd

lcd.begin(16, 2);

lcd.clear();

lcd.setCursor(0, 0);

Serial.begin(9600);

//настройка nrf

radio.begin();

delay(2);

radio.setChannel(9); // канал (0-127)

radio.setAutoAck(1);

radio.enableAckPayload();

// скорость, RF24\_250KBPS, RF24\_1MBPS или RF24\_2MBPS

// RF24\_250KBPS на nRF24L01 (без +) неработает.

// меньше скорость, выше чувствительность приемника.

radio.setDataRate(RF24\_1MBPS);

// мощьность передатчика, RF24\_PA\_MIN=-18dBm, RF24\_PA\_LOW=-12dBm, RF24\_PA\_MED=-6dBM,

radio.setPALevel(RF24\_PA\_HIGH);

radio.powerUp();

}

//-------------------global values--------------------

byte col = 0;

boolean menu = true;

boolean select;

//----------переменные передоваемые по каналу--------------

float hours = 0;

boolean siga = false;

//---------------------functions-----------------------------

//

//----------функция--индикации--нажатой--кнопки----------------

byte key(){

int dig = analogRead(0);

if(dig < 50) return 5;

else if(dig < 200) return 4;

else if(dig < 400) return 3;

else if(dig < 550) return 2;

else if(dig < 800) return 1;

else if(dig < 1100) return 0;

}

//---------------------стартовое---меню-------------------------

void menu\_1(){

if(menu){

lcd.print("turn device?");

lcd.cursor();

switch(key()){

case 5: col+=2;

break;

case 4: ;

break;

case 3: ;

break;

case 2: col-=2;

break;

case 1: if(col == 0)menu = false; else if(col == 2) {if(!select) select = true; else {select = false;} hours = 0;}

break;

case 0: ;

break;

}

if(col < 0) col = 2;

if(col > 2) col = 0;

lcd.setCursor(0,1);

lcd.print("+|-");

lcd.setCursor(13,1);

if(col == 0)

lcd.print("yes");

else if(col == 2) lcd.print(" no");

lcd.setCursor(15,0);

lcd.print(select); // cdecb toje ne nado

lcd.setCursor(col,1); //идет последним что бы был курсор где нужно

}

}

//----------------------меню--настройки--режима---------------------------

void menu\_2(){

if(!menu){

lcd.noCursor();

lcd.setCursor(0,0);

lcd.print("-30 <-mode-> +30");

switch(key()){

case 1: delay(200); if(key() == 1 ){ if(!select) select = true; else select = false;}

break;

case 2: hours-=0.5;

break;

case 5: hours+=0.5;

break;

case 3: menu = true;

break;

case 4: if(hours == 1e+38) hours = 0.5; else hours = 1e+38;

break;

}

if(hours < 0.5) {hours = 24;}

if(hours > 24 && hours < 100) hours = 0.5;

lcd.setCursor(0,1);

lcd.print(hours);

lcd.print("\*60 minuts");

lcd.print(select); //----------------------

lcd.setCursor(0,1);

}

}

//-------------------функция---передачи---------------------------

void transmit(){

radio.openWritingPipe(pipe); // открываем трубу на передачу.

//radio.setRetries(1, 1); // Кол-во попыток <-> время между попытками

radio.write(&hours, sizeof(hours)); // отправляем данные и указываем сколько байт пакет

Serial.print("hours: ");

Serial.println(hours);

}

//-----------------------------прием-------------------------------

void receiv(){

radio.openReadingPipe(1,pipe);

radio.startListening();

if (radio.available()){

radio.read(&siga, sizeof(siga));

Serial.print("siga: ");

Serial.println(siga);

}

}

//------------------------------диод--------------------------------

void light(){

if(siga){

analogWrite(1,HIGH);

delay(300);

}else{

analogWrite(1, HIGH);

delay(150);

analogWrite(1,HIGH);

delay(150);

}

}

//------------------------------------main\_loop--------------------------------------------------

void loop() {

lcd.clear();

menu\_1();

menu\_2();

if(select){

transmit();

select = false;

}

else{

receiv();

}

light();

//select = false;

//lcd.print(millis()/1000);

//delay(300);

}

Код для информационной стороны:

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

const uint64\_t pipe = 0xF0F1F2F3F4LL; // индитификатор передачи, "труба"

//--------------global---------values---------------------------

float hours = 0;

boolean siga = false;

int minSecsBetweenEmails = 3600\*hours; // Время- со второй ардуины

long lastSend = -minSecsBetweenEmails \* 1000l;

RF24 radio(9, 10); // CE, CSN

void setup() {

Serial.begin(9600);

//настройка nrf

radio.begin();

delay(2);

radio.setChannel(9); // канал (0-127)

radio.setAutoAck(1);

radio.enableAckPayload();

// скорость, RF24\_250KBPS, RF24\_1MBPS или RF24\_2MBPS

// RF24\_250KBPS на nRF24L01 (без +) неработает.

// меньше скорость, выше чувствительность приемника.

radio.setDataRate(RF24\_1MBPS);

// мощьность передатчика, RF24\_PA\_MIN=-18dBm, RF24\_PA\_LOW=-12dBm, RF24\_PA\_MED=-6dBM,

radio.setPALevel(RF24\_PA\_HIGH);

radio.powerUp();

}

//-------------------------listning----------------chanalll-----------

void receiv(long now){

radio.openReadingPipe(1,pipe);

radio.startListening();

if (radio.available()){

radio.read(&hours, sizeof(hours));

lastSend = now;

Serial.print("hours: ");

Serial.println(hours);

}

}

//---------------------------tramsmit--------------------------------------

void transmit(){

radio.openWritingPipe(pipe); // открываем трубу на передачу.

radio.write(&siga, sizeof(siga)); // отправляем данные и указываем сколько байт пакет

Serial.print("siga: ");

Serial.println(siga);

}

//----------------------------------main\_loop---------------------------------------------------

void loop() {

long now = millis();

//проверка аналогового входа

if(analogRead(0) > 300) {Serial.println("MOVEMENT"); siga = true;}

else {siga = false; Serial.println("Too soon");}

//алгоритм выбора отправки, приема

if(now >= (lastSend + minSecsBetweenEmails \* 1000l)){

receiv(now);

} else {

if(hours > 0) transmit();

}

delay(500);

}

**Script для отправки email:**

import time  
  
import serial  
  
import smtplib  
  
TO = '[vlad.semenik@gmail.com](https://vk.com/write?email=vlad.semenik@gmail.com)'  
  
GMAIL\_USER = '[vlad.semenik@gmail.com](https://vk.com/write?email=vlad.semenik@gmail.com)'  
  
GMAIL\_PASS = 'тут мой пароль'  
  
SUBJECT = 'Intrusion!!'  
  
TEXT = 'Your PIR sensor detected movement'  
  
ser = serial.Serial("/dev/ttyUSB0", 9600)  
  
while True:  
  
if (ser.inWaiting()>0):

message = ser.readline()

print(message)

if message[0] == 'M' :

def send\_email():

print ('Sending email')

smtpserver = smtplib.SMTP('smtp.gmail.com',587)

smtpserver.ehlo()

smtpserver.starttls()

smtpserver.ehlo

smtpserver.login(GMAIL\_USER, GMAIL\_PASS)

header = 'To:' + TO + '\n' + 'From: ' + GMAIL\_USER

header = header + '\n' + 'Subject:' + SUBJECT + '\n'

print (header)

msg = header + '\n' + TEXT + ' \n\n'

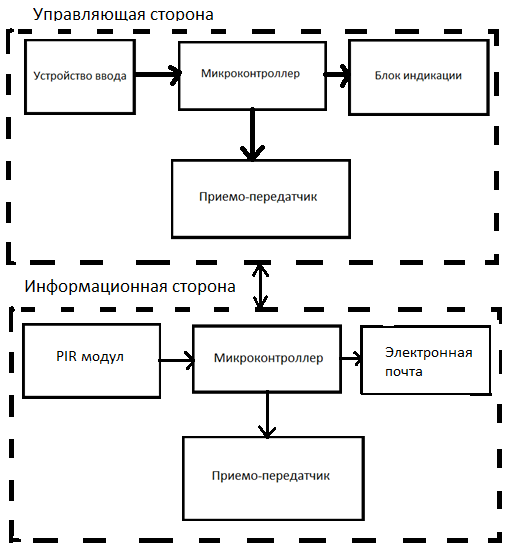
smtpserver.sendmail(GMAIL\_USER, TO, msg)

smtpserver.close()

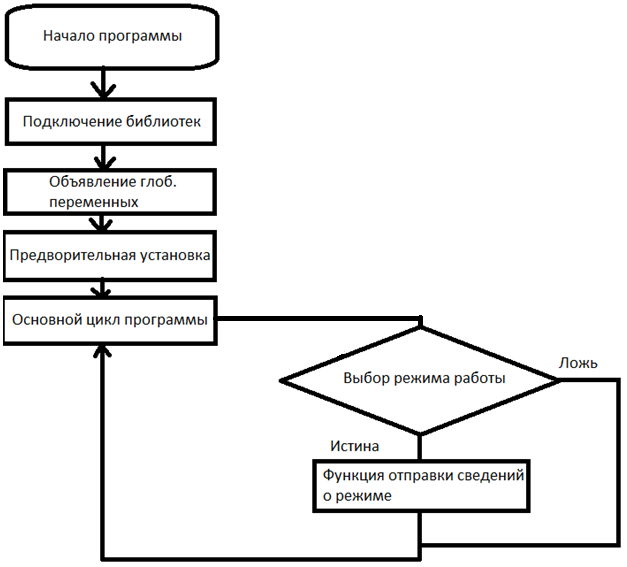
time.sleep(0.5)

**ПРИЛОЖЕНИЕ Б**

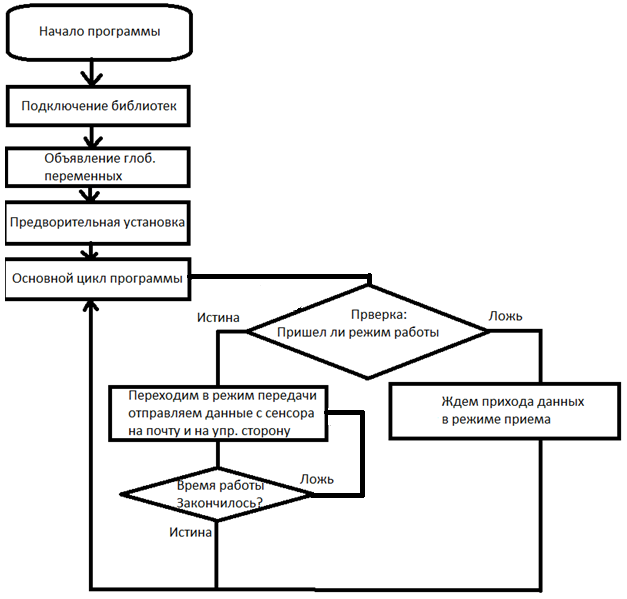
**Структурная схема проекта**



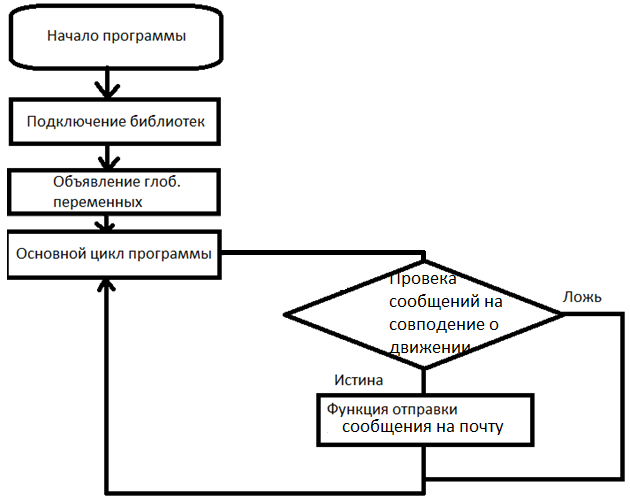
Блок схема программы управляющей стороны

****

Блок схема программы информационной стороны

****

Блок-схема программы отправки сообщения на электронную почту

****