

**Нижегородский Государственный Технический
Университет им. Р.Е.Алексеева**

Всероссийская Робототехническая Олимпиада

Проект «СОВ.А»

**Система Обнаружения Возгорания.
Автоматическая.**

Авторы проекта:

Команда РУР-ИНЭЛ

В составе:

Краюшкин А.В.

Журова С.А.

Бокарев К.А.

Руководитель проекта:

Краюшкин В.А.

Н.Новгород

2014

I. Введение

Каждый год происходят катастрофы природного и техногенного характера. Борьбсь с природными катастрофами невозможно, их можно прогнозировать, предупреждать, предугадывать. Но существуют катастрофы техногенного характера, связанные непосредственно с деятельностью человека: утечки химического производства, ядерные утечки, пожары на больших территориях.

Анализу пожаров и способов их предотвращения посвящается наш проект.

Леса России — ценные породы дерева, пятая часть мирового запаса древесины и один из главных источников кислорода нашей планеты. Они занимают около 45% площади современной России и являются средой обитания для тысячи видов животных и растений. Лесные пожары - самое разрушительное стихийное бедствие, уничтожающее гектары лесных ресурсов ежегодно.

Самым показательным по пожарной обстановке страны можно считать 2010 год. По официальным данным МЧС России, всего за время аномалии 2010 года возникло около 31 тыс. пожаров на общей площади примерно в 1,5 млн га (В два раза большую цифру — 3 млн га — приводит начальник авиалесоохраны Федерального агентства лесного хозяйства России Николай Ковалёв)..Есть и ещё более радикальные оценки. Так, по данным Всемирного центра мониторинга пожаров (Global Fire Monitoring Center, Германия), данные МЧС занижены в десятки раз, а общая площадь пожаров на территории России составила более 16 млн га.

Экономический ущерб, причиненный только лесу, составил более 300 миллиардов долларов, заявил на пресс-конференции представителей природоохранных организаций директор Центра охраны дикой природы Алексей Зименко.

"По очень осторожным подсчетам, экономический ущерб от пожаров составляет не менее 25 тысяч долларов на один гектар леса. Таким образом, в масштабах страны отсутствие действенной системы лесоохраны в текущем году обошлось минимум в 375 миллиардов долларов", - заявил Зименко.

Свои подсчеты экологи основывают на данных Всемирного центра мониторинга пожаров, они основаны на двух показателях: стоимость деловой древесины и стоимость стандартных средств восстановительных работ. При этом в них не учитываются расходы по уходу за лесными культурами первые 5-10 лет после их посадки, а также гибель и восстановление растений и животных, в том числе, занесенных в Красную книгу и ценных с хозяйственной точки зрения.

Кроме ущерба, причиненного лесным угодьям страны, не обошлось без человеческих жертв. Не удалось спасти около 60 человек и 2,5 тыс. домов. На восстановление одного дома было выделено 2 мл. руб.

Из сказанного выше следует, что борьба с пожарами является трудоемкой и опасной, упростить ее поможет своевременное обнаружение очага возгорания. Наш проект был задуман именно с этой целью.

II. Анализ систем обнаружения очагов возгорания.

На сегодняшний день существует несколько способов обнаружения очагов возгорания

1. Человеческий - обнаружение возгораний лесных массивов силами сотрудников лесоохранных служб.

К минусам такого способа можно отнести человеческий фактор и низкую скорость патрулей.

2. Аэроразведка

Минусы-сравнительно дорогостоящий вид (стоимость одного часа полета вертолета Ми-8- около 124 тыс. руб. В настоящее время большинство вертолетов находятся в частной собственности, а не на гос. обеспечении)

3. Космическая разведка

Наиболее существенный минус-спутник не сможет засечь мелкие пожары

4. Видеонаблюдение

Минусы – нет возможности обнаружить низовые пожары, которые горят практически без дыма и вовремя приступить к мерам по его предотвращению. Система видеонаблюдения засекает уже разгоревшийся пожар, когда горят живые ветки и огонь охватывает все ярусы леса.

На основе данного анализа мы решили создать такую систему, которая была бы лишена недостатков вышеперечисленных систем обнаружения очагов возгорания.

III. Описание проекта

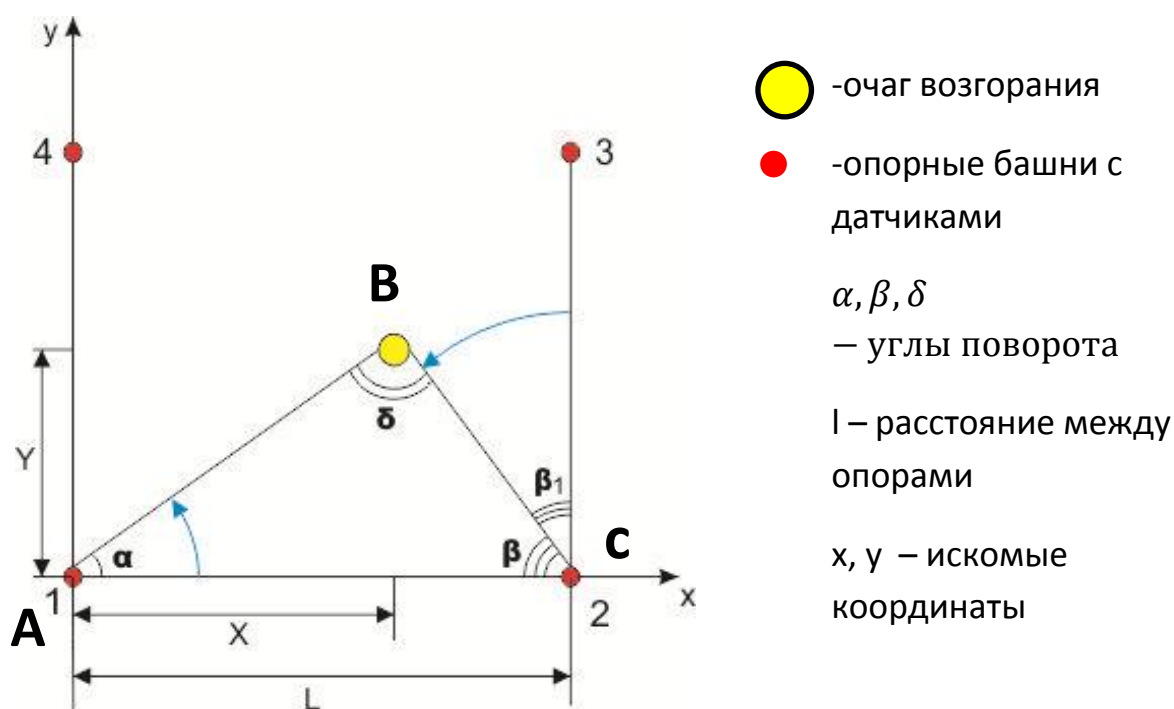
«СОВА» - Система Обнаружения Возгораний Автоматическая.

предназначена для контроля как открытых участков местности, так и малоэтажных населенных поселков, лесных массивов на предмет воспламенения, фиксации координат воспламенения и передачи сигнала опасности в диспетчерскую службу

Общий принцип работы

Принцип работы системы основан на обнаружении очага возгорания посредством улавливания волн инфракрасного излучения (пламя излучает волны от 0,75 до 100 Мкм в инфракрасном диапазоне) сборкой диодов, установленных на четырех опорах (1,2,3,4), расположенных в углах квадратного участка охраняемой территории. Датчики, установленные на подвижной площадке, управляемой системой «ARDUINO MEGA», сканируют местность, и на основе полученных данных, система определяет угол, посредством тригонометрических функции, находит X и Y координаты очага возгорания. Подает звуковой

сигнал оповещения населения и отправляет данные диспетчеру поста пожарной части.



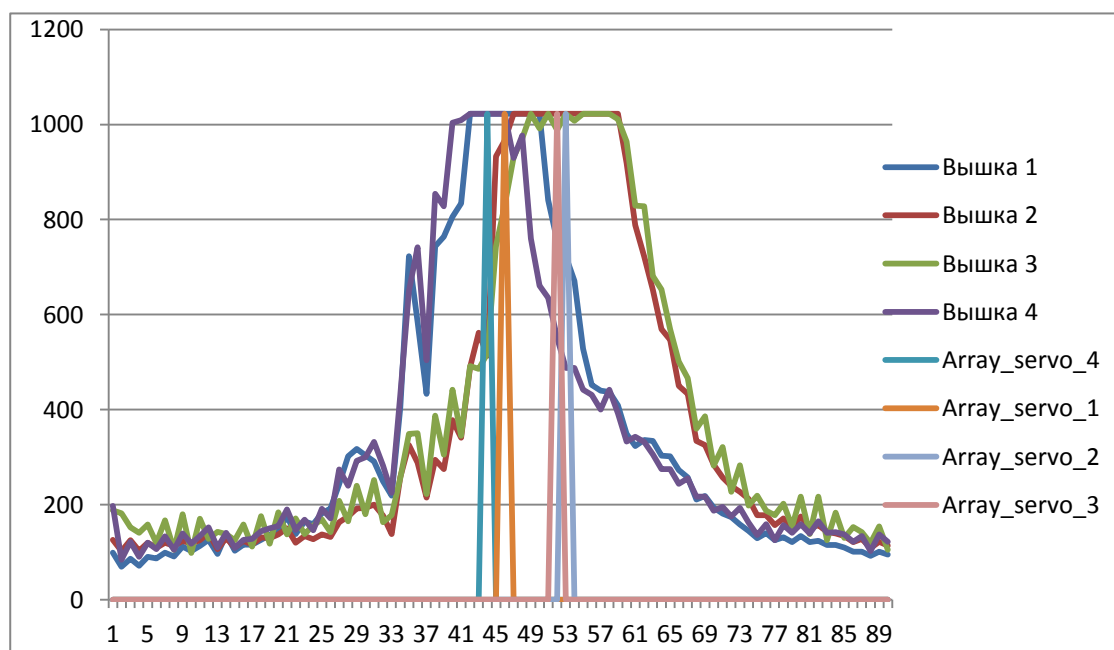
$$\delta = (180 - (180 - \beta_1 + \alpha))$$

$$AB = \frac{l \cdot \sin \beta}{\sin \delta} \text{ — по теореме синусов}$$

$$y = AB \cdot \sin \alpha$$

$$x = AB \cdot \cos \alpha$$

График зависимости значений ИК излучения пламени от угла поворота датчиков



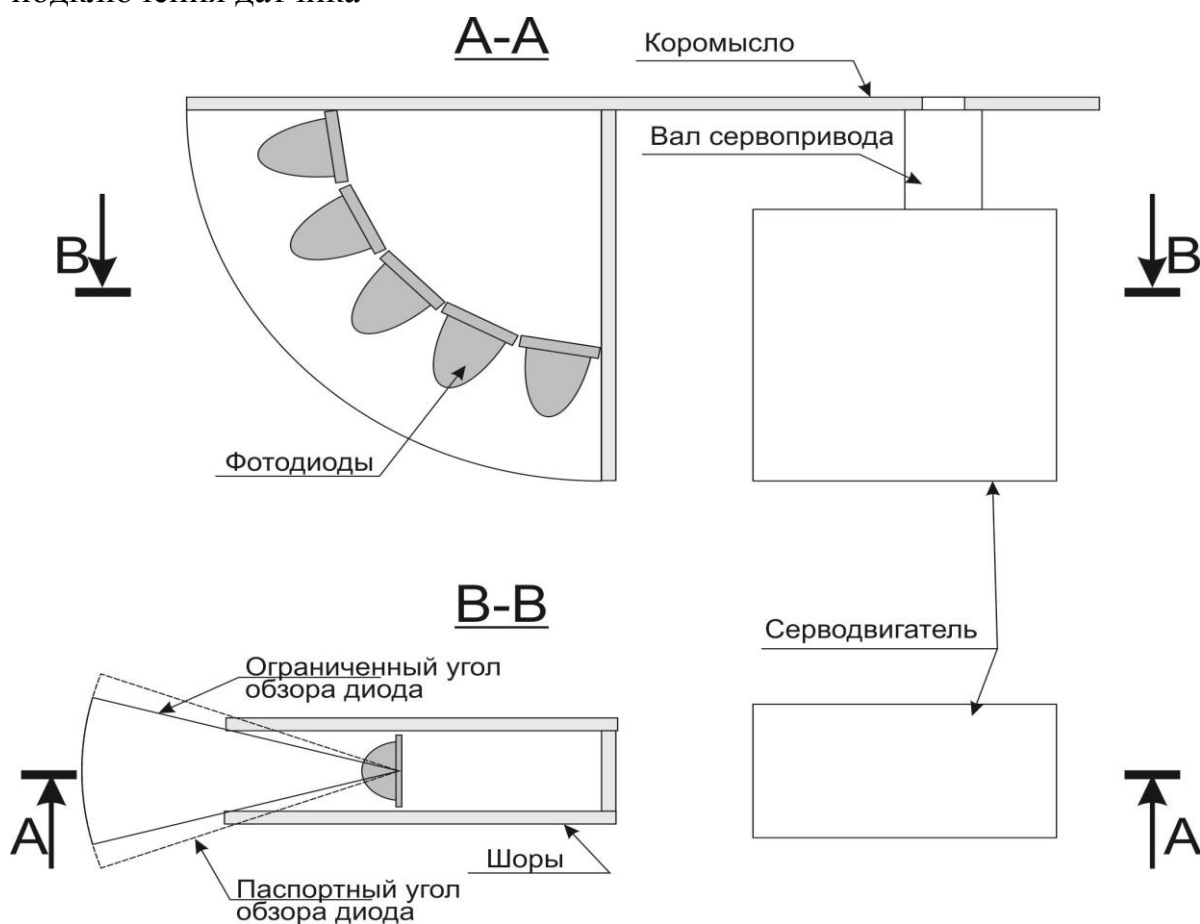
Принцип работы каждого узла системы

1. Датчики.

В данной системе использованы фотодиоды инфракрасного диапазона с чувствительностью для волн с излучением 940 Нм. И углом обзора 20°.

Для большей точности, устранения боковых засветов и сужения угла обзора были установлены «шоры»

Ниже на рисунке представлена конструкция и электрическая схема подключения датчика



2. Привод датчика

Для управления поворотом датчика использован сервопривод FS90 (диапазон 180° с усилием 3кг/см , скоростью вращения 60°/12сек)

Для снижения скорости поворота были использованы программные задержки.

3. Система управления и вычисления

Система управления и вычисления основана на плате «ARDUINO MEGA». В задачи системы входит управление сервоприводами, сбор данных с датчиков, обработка данных, нахождение пиковых значений, принятие решений и передача обработанных данных.

- Для управления сервоприводами в программном коде предусматриваются задержки для снижения угловой скорости вращения в пределах $90^\circ/2$ сек.
 - Данные датчиков приходят на аналоговый вход со значениями от 0 до 1023
 - Принятые данные программа анализирует с целью вычисления
 - a) Среднего (естественного) фонового уровня инфракрасного излучения
 - b) Возможных пиковых значений и сравнения их с полученными фоновыми значениями для вычисления угла очага возгорания
 - c) Координат очагов возгорания по данным с двух датчиков и с применением тригонометрических функций
 - d) Возможных погрешностей и проверки данных с использованием данных, полученных с третьего и четвертого датчиков.
4. Узел индикации

Узел индикации выполнен на основе ЖКИ. На него выводятся данные о наличии очага возгорания в относительных координатах.

Ниже приведен скетч программы.

```
//-----Подключаем библиотеки-----
#include <Servo.h>
#include <LiquidCrystal.h>
#include <math.h>
//-----

//-----Назначаем Выводы-----
const int Pin_Servo_1 = 42;
const int Pin_Servo_2 = 43;
const int Pin_Servo_3 = 44;
const int Pin_Servo_4 = 45;
const int Pin_Da4ik_1 = A3;
const int Pin_Da4ik_2 = A2;
const int Pin_Da4ik_3 = A1;
const int Pin_Da4ik_4 = A0;
const int Pin_Dinamik = 5;
const int Pin_stop_key = 4;
const int Pin_start_key = 3;
const int Pin_col_kof_da4ik_1 = A15;
const int Pin_col_kof_da4ik_2 = A14;
const int Pin_col_kof_da4ik_3 = A13;
const int Pin_col_kof_da4ik_4 = A12;
//-----

//-----Создаем экземпляры класса-----
Servo Servo_1;
Servo Servo_2;
Servo Servo_3;
Servo Servo_4;
LiquidCrystal LCD_2(26, 27, 25, 24, 23, 22);
LiquidCrystal LCD_1(28, 29, 30, 31, 32, 33);
//-----

//-----Объявляем переменные-----
const int Servo_1_1 = 34;
const int Servo_2_1 = 50;
const int Servo_3_1 = 25;
const int Servo_4_1 = 60;
const int Servo_1_90 = 138;
const int Servo_2_90 = 157;
const int Servo_3_90 = 125;
const int Servo_4_90 = 158;
int array_servo_znach[90][4];
boolean array_servo[90][4];
int AC = 42; //Размер площадки
int pog = 15;
int pause_1 = 50;
int pause_2 = 200;
```



```
float dob_kof_1 = 0.99;
float dob_kof_2 = 0.99;
float dob_kof_3 = 0.99;
float dob_kof_4 = 0.99;
//-----

void setup(){
  start_LCD();
  bip();
  attachInterrupt(1, kalibrovka, RISING);
  Servo_1.attach(Pin_Servo_1);
  Servo_2.attach(Pin_Servo_2);
  Servo_3.attach(Pin_Servo_3);
  Servo_4.attach(Pin_Servo_4);
  pinMode(Pin_start_key, INPUT);
  pinMode(Pin_stop_key, INPUT);
  pinMode(Pin_Dinamik, INPUT);
  //-----Инициализация Серво-привода-----
  for (int i = Servo_1.read(); i >= Servo_1_1; i--){
    Servo_1.write(i);
    delay(pause_1/3);
  }
  for (int i = Servo_2.read(); i >= Servo_2_1 ; i--){
    Servo_2.write(i);
    delay(pause_1/3);
  }
  for (int i = Servo_3.read(); i >= Servo_3_1 ; i--){
    Servo_3.write(i);
    delay(pause_1/3);
  }
  for (int i = Servo_4.read(); i >= Servo_4_1 ; i--){
    Servo_4.write(i);
    delay(pause_1/3);
  }
  randomSeed(0); //инициализируем генератор псевдослучайных чисел

  for (int i = 0; i < 90; i++){
    for (int j = 0; j < 4; j++){
      array_servo[i][j] = false;
      array_servo_znach [i][j] = 0;
    }
  }

  Serial.begin(9600);
}

void loop(){
  scan();
}
```

```
    ras4et();
    return_servo();
}
//-----Крутим Серво и считываем данные-----
void scan(){
    int max_znach_1 = 0, max_znach_2 = 0, max_znach_3 = 0,
max_znach_4 = 0;
    long sred_znach_1 = 0, sred_znach_2 = 0, sred_znach_3 = 0,
sred_znach_4 = 0;
    boolean flag = false;
    for (int i = 0; i < 90; i++){
        array_servo_znach[i][0] = analogRead(Pin_Da4ik_1);
        array_servo_znach[i][1] = analogRead(Pin_Da4ik_2);
        array_servo_znach[i][2] = analogRead(Pin_Da4ik_3);
        array_servo_znach[i][3] = analogRead(Pin_Da4ik_4);
        sred_znach_1 = sred_znach_1 + array_servo_znach[i][0];
        sred_znach_2 = sred_znach_2 + array_servo_znach[i][1];
        sred_znach_3 = sred_znach_3 + array_servo_znach[i][2];
        sred_znach_4 = sred_znach_4 + array_servo_znach[i][3];
        max_znach_1 = max(max_znach_1, array_servo_znach[i][0]);
        max_znach_2 = max(max_znach_2, array_servo_znach[i][1]);
        max_znach_3 = max(max_znach_3, array_servo_znach[i][2]);
        max_znach_4 = max(max_znach_4, array_servo_znach[i][3]);
        Servo_1.write(map(i, 0, 90, Servo_1_1, Servo_1_90));
        Servo_2.write(map(i, 0, 90, Servo_2_1, Servo_2_90));
        Servo_3.write(map(i, 0, 90, Servo_3_1, Servo_3_90));
        Servo_4.write(map(i, 0, 90, Servo_4_1, Servo_4_90));
        delay(pause 1);
    }
    sred_znach_1 = sred_znach_1/90;
    int granica_1 = max_znach_1 * dob_kof_1;
    sred_znach_2 = sred_znach_2/90;
    int granica_2 = max_znach_2 * dob_kof_2;
    sred_znach_3 = sred_znach_3/90;
    int granica_3 = max_znach_3 * dob_kof_3;
    sred_znach_4 = sred_znach_4/90;
    int granica_4 = max_znach_4 * dob_kof_4;

    for (int i = 0; i < 90; i++){
        if (array_servo_znach[i][0]>granica_1) array_servo[i][0] =
true;
        if (array_servo_znach[i][1]>granica_2) array_servo[i][1] =
true;
        if (array_servo_znach[i][2]>granica_3) array_servo[i][2] =
true;
        if (array_servo_znach[i][3]>granica_4) array_servo[i][3] =
true;
    }
}
```

```

    for (int j = 0; j < 4; j++){

        int i_n = 0, i_k = 89;
        for (int i = 0; i < 90; i++){
            if ((array_servo[i][j]==true)&&(array_servo[i-1][j]==false)&&(array_servo[i-2][j]==false))    i_n = i;
            if
            ((array_servo[i][j]==true)&&(array_servo[i+1][j]==false)&&(array_servo[i+2][j]==false))    i_k = i;
        }
        if(i_n<i_k)
        {
            for (int i = i_n; i <= i_k; i++)
            {
                if(i!=i_n + ceil((i_k-i_n)/2)) array_servo[i][j] = false;
            }
        }
        for (int i = 0; i < 90; i++) Serial.print(array_servo[i][j]);
    }

//-----Возвращаем серво-----
void return_servo(){
    delay(pause_2);
    for (int i = 90; i > 0; i--){
        Servo 1.write(map(i,0,90,Servo 1 1,Servo 1 90));
        Servo 2.write(map(i,0,90,Servo 2 1,Servo 2 90));
        Servo 3.write(map(i,0,90,Servo 3 1,Servo 3 90));
        Servo 4.write(map(i,0,90,Servo 4 1,Servo 4 90));
        delay(pause 1/3);
    }
    delay(pause 2);
    for (int i = 0; i < 90; i++){
        for (int j = 0; j < 4; j++){
            array_servo[i][j] = false;
        }
    }
}

//-----Здесь математика-----
void ras4et()
{
    int m = 0;
    for (int i = 0; i < 90; i++)    if(array_servo[i][0]==true)    for
(int j = 0; j < 90; j++)    if (array_servo[j][1]==true)    m++;

```

```
double mas[m][3];
int n=0;
for (int i = 0; i < 90; i++)
{
    if(array_servo[i][0]==true)
    {
        for (int j = 0; j < 90; j++)
        {
            if (array_servo[j][1]==true)
            {
                mas[n][0] = x_kor(i,j);
                mas[n][1] = y_kor(i,j);
                mas[n][2] = 1;
                n++;
            }
        }
    }
}

//-----

//----Проверка мнимых точек относительно 3 вышки
for(int i = 0; i < m; i++)
{
    int b = 0;
    double BC = sqrt(sq(AC - mas[i][0])+sq(AC - mas[i][1]));
    double C = RadToGrad(asin((AC - mas[i][1])/BC));
    for(int j = C - pog; j<= C + pog; j++)
    {
        b = b + array_servo[j][2];
    }
    if (b==0) mas[i][2] = 0;
}
//-----

//----Проверка мнимых точек относительно 4 вышки
for (int i = 0; i < m; i++)
{
    int b = 0;
    double BC = sqrt(sq(mas[i][0])+sq(AC - mas[i][1]));
    double C = RadToGrad(asin((mas[i][0])/BC));
    for(int j = C - pog; j<= C + pog; j++)
    {
        b = b + array_servo[j][3];
    }
    if (b==0) mas[i][2] = 0;
}
//----Проверка мнимых точек выходящих за размер полотна
```

```
    if ((AC<=mas[i][0])||(AC<=mas[i][1])) mas[i][2] = 0;
    //-----

}

//-----

LCD_1.clear();
LCD_2.clear();
LCD_1.setCursor(0,0);
LCD_1.print("X=");
LCD_1.setCursor(0,1);
LCD_1.print("Y=");

int index_cursor = 2;
for (int i = 0; i < m; i++){
    if (mas[i][2] > 0) {
        LCD_1.setCursor(index_cursor,0);
        LCD_1.print(int(mas[i][0]));
        LCD_1.setCursor(index_cursor,1);
        LCD_1.print(int(mas[i][1]));
        yes();
        if((mas[i][0]<9)&&(mas[i][1]<9)) index_cursor+=2;
        else index_cursor+=3;
    }
}

void kalibrovka(){

    if (digitalRead(Pin_start_key)==true) {
        LCD_2.clear();
        LCD_2.clear();
        LCD_1.setCursor(0,0);
        LCD_1.print("Kp 1 = ");
        LCD_1.setCursor(0,1);
        LCD_1.print("Kp 3 = ");
        LCD_2.setCursor(0,0);
        LCD_2.print("Kp_2 = ");
        LCD_2.setCursor(0,1);
        LCD_2.print("Kp_4 = ");
        while(1){

dob_kof_1=float(map(analogRead(Pin_col_kof_da4ik_1),1,1023,1,100))/
100;

        LCD_1.setCursor(7,0);
        LCD_1.print(dob_kof_1);
```

```
dob_kof_3=float(map(analogRead(Pin_col_kof_da4ik_3),1,1023,1,100))/
100;
    LCD_1.setCursor(7,1);
    LCD_1.print(dob_kof_3);

dob_kof_2=float(map(analogRead(Pin_col_kof_da4ik_2),1,1023,1,100))/
100;
    LCD_2.setCursor(7,0);
    LCD_2.print(dob_kof_2);

dob_kof_4=float(map(analogRead(Pin_col_kof_da4ik_4),1,1023,1,100))/
100;
    LCD_2.setCursor(7,1);
    LCD_2.print(dob_kof_4);

    if (digitalRead(Pin_stop_key)==true) break;
    delay(10);
}
start_LCD();
}

//-----Перевод из Градусов в Радианы-----
double GradToRad(int grad){
    double rad = grad * PI / 180;
    return rad;
}
//-----

//-----Перевод из Радиан в Градусы-----
double RadToGrad(double rad){
    double grad = rad * 180 / PI;
    return grad;
}
//-----

//-----Нахождение Y координаты-----
double y_kor(int A, int C)
{
    int B = 180 - A - 90 + C;
    double AB = AC * sin(GradToRad(90-C)) / sin(GradToRad(B));
    double h = AB * sin(GradToRad(A));
    return h;
}
//-----
```

```

//-----Нахождение X координаты-----
double x_kor(int A, int C)
{
    int B = 180 - A - 90 + C;
    double AB = AC * sin(GradToRad(90-C)) / sin(GradToRad(B));
    double h = AB * cos(GradToRad(A));
    return h;
}
//-----

//-----Функция определения длины отрезка
float dlina_otrezka(int x1, int y1, int x2, int y2)
{
    int d = sqrt(sq(x2-x1)+sq(y2-y1));
    return d;
}
//-----

void bip()
{
    tone(Pin_Dinamik, 450);
    delay(1000);
    noTone(Pin_Dinamik);
}

void start_LCD(){
    //-----Инициализация LCD - дисплея-----
    LCD_1.begin(16, 2);
    LCD_1.print("      \xA8p\xB8\xB3""e\xBF");
    LCD_1.setCursor(0, 1);
    LCD_1.print("      WRO 2014");
    LCD_2.begin(16, 2);
    LCD_2.print("      \xA8p\xB8\xB3""e\xBF");
    LCD_2.setCursor(0, 1);
    LCD_2.print("      WRO 2014 ");
    //-----
}

void yes(){
    for(int i =0; i<3; i++){
        tone(Pin_Dinamik, 1000);
        delay(300);
        noTone(Pin_Dinamik);
        delay(300);
    }
}

```

IV. Анализ финансовых затрат при создании проекта.

Основные составляющие проекта, элементы и узлы сведены в таблицу.

Наименование	Количество	Цена (руб.)	Стоимость (руб.)
ARDUINO MEGA	1	2290	2290
Диоды	20	6	120
Сервоприводы	4	290	1160
Провода	8 м.	20	160
Макетная плата	2	340	680
ЖКИ	2	490	980
Переходник ПВХ	4	30	120
Хомуты	1	50	50
Краска	1	90	90
Работа	384 чел/часа.	94	36000
Итог			41520

Стоимость работы рассчитана из средней заработной платы по Нижнему Новгороду в размере 15 тыс. руб. В таблице не учтены стоимость «бросовых» материалов: трубы ПВХ, обрезки пластика и пленки, так как они не покупались и закончили бы свои дни на свалке.

V. Анализ финансовых вложений при массовом применении.

Рассмотрим возможные случаи применения данной системы.

1) Деревня, поселок, дачный массив.

При использовании данной системы необходимо сократить количество проводных соединений. Для этого в реальную систему вводится модуль беспроводной связи. Для обеспечения независимого питания используется солнечная батарея и аккумуляторы. Для увеличения площади покрытия и чувствительности датчика устанавливается большее количество диодов и

добавляется собирающая линза. Усилены сервоприводы, так же добавлен сервопривод для раскачивания рынды (или добавляется автомобильная сирена). Корпус выполнен из ударопрочного материала (поликарбоната) методом термоформовки либо пресслитьем по форме. Один из модулей оснащен GPRS-модулем для передачи сообщений в пожарную диспетчерскую службу.

Расчет стоимости опытного мелкосерийного образца.

Наименование	Кол-во	Цена	Стоимость
ARDUINO UNO	4	1150	4600
WiFi Shilg	4	3990	15960
Сервопривод FS510613	5	490	2450
Солнечные батареи Exmork 10 Вт 12 В	4	900	3600
Аккумулятор 12(В.)	4	250	1000
Диоды (ИК + УФ)	40	40	1600
линзы	4	300	1200
GSM модуль 2990 12 В x1,3 А; 12 Вт.	1	2990	2990
П-карб. NOVATTRO 3050x2050x2мм	1/4		3530
Работа			36000
Элементы крепления	4	1000	4000
Итог			76930

В таблице приведены розничные цены и стоимость работа рассчитана из 15 тыс. руб. и использовании ручного труда. При запуске данной системы в серийное производство прогнозируется падение стоимости до 40-50%.

Общая стоимость системы составляет 77 тыс. руб., что не сопоставимо даже со стоимостью восстановления от пожара одного жилого дома.

2) Зерновые угодья, лес.

Использование системы в данных условиях незначительно отличается от использования в дачных массивах. Из системы «СОВА» удалена система звукового оповещения и добавлен еще один комплект датчиков.

Исходя из проведенной работы и основываясь на экономических расчетах, мы предполагаем, что данная система может быть запущена в производство и позволит обеспечить безопасность не только лесных угодий, но и человеческих жизней.

VI. Основные направления развития проекта «СОВА»

На основе анализа проекта, были обнаружены недостатки, которые наметили пути дальнейшей модернизации системы.

Основные направления модернизации включают в себя:

А. Улучшение работы ИК-датчика на аппаратном уровне – повышение помехоустойчивости, увеличение дальности обнаружения. Проведение натурных испытаний.

Б. Разработка модуля (SLAVE), с поддержкой возможности передачи закодированного сигнала по не лицензируемому радиоканалу, проведение натурных испытаний, для определения максимального расстояния связи.

В. Разработка модуля (MASTER) для приема и обработки информации

Г. Разработка программного обеспечения, позволяющего на основании географических координат (Долгота, Широта) положения модуля (SLAVE) определять местоположение очага возгорания по Долготе и Широте.

Д. Создание на базе модулей типа SLAVE и модулей типа MASTER иерархической системы обнаружения возгорания, с выводом информации на монитор и наложением проекции очага возгорания на карту местности в центральном диспетчерском пункте (Представлена на схеме)

