

Игра через инфракрасный порт на платформе Arduino

Кизянов Антон Олегович

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс создания игры на базе Arduino и управления через инфракрасный порт. Для создания используется плата Arduino, инфракрасный датчик, пульт управления для него и LCD панель. Созданный макет позволяет управлять платой Arduino через инфракрасный порт и отображать результат на LCD панель.

Ключевые слова: Arduino, LCD панель, инфракрасный сенсор

Infrared play on Arduino

Kizyanov Anton Olegovich

Sholom-Aleichem Priamursky State University

student

Abstract

This article will guide you through the process of creating an Arduino game and an infrared board. For creation, an Arduino board, an infrared sensor, a control panel for it and an LCD panel are used. The created breadboard allows you to control the Arduino board via infrared port and display the result on the LCD panel.

Keywords: Arduino, LCD panel, infrared sensor

Все пульты дистанционного управления основаны на инфракрасных передатчиках. У Arduino также есть инфракрасный датчик позволяющий принимать и отправлять сигнал, а затем производить какое-либо действие в зависимости от программы. В этой статье сигнал будет служить сигналом к движению на LCD панели через инфракрасный датчик.

Цель исследования – создать игру с дистанционным управлением Arduino через инфракрасный датчик.

Ранее этим вопросом интересовались А.И. Халиуллин, М.В. Медведев развивали тему «Система охраны помещения на платформе arduino с использованием инфракрасного датчика объема и gsm-модуля для оповещения» [1] в которой обсуждается один из способов построения простой системы охраны помещения с использованием инфракрасного датчика объема и GSM-модуля для оповещения о нештатной ситуации. А.А. Курусканова с темой «Учебно-исследовательская программа для управления линейным и псевдослучайным движениями механической тележки» [2], а подробнее про режим дистанционного управления,

реализующийся путем дешифрации сигнала инфракрасного пульта управления. Режим псевдослучайного перемещения реализуется путем получения псевдослучайного числа при помощи генератора случайных чисел. В процессе псевдослучайного перемещения тележка отслеживает наличие сигналов при помощи ультразвукового дальномера и движется объезжая преграды. М.А. Охрименко, Ю.Е. Зинченко опубликовали статью «Архитектура подсистемы дистанционного управления бытовыми приборами для системы "умный дом"» [3] рассказал про архитектуру подсистемы дистанционного управления бытовыми приборами для системы «Умный дом» и выбранных моделях Arduino. Рассмотрен принцип функционирования подсистемы. В разработке задействован инфракрасный передатчик, технология Ethernet. Относительно существующих устройств, предлагаемое авторами устройство отличается высокой надежностью и универсальностью.

Для этого потребуется:

- Плата Arduino;
- Соединительные провода;
- Инфракрасный датчик;
- LCD панель.

Схема подключения представлена на рисунке 1.

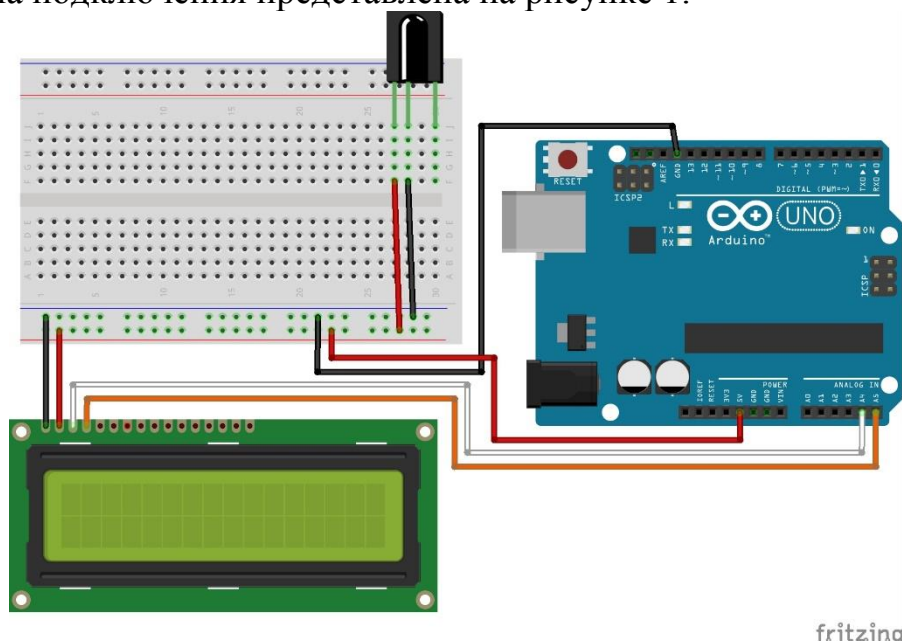


Рис. 1 Схема подключения к плате Arduino

```
#include <IRremote.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
char up = 3;
char down = 1;
char ok = 2;
```

```
byte up1[8] = {
    B00100, B01110, B10101, B00100,
    B00100, B00100, B00100, B00100
};

byte down1[8] = {
    B00100, B00100, B00100, B00100,
    B00100, B10101, B01110, B00100
};

byte ok1[8] = {
    B01110, B01010, B01110, B10000,
    B10100, B11000, B10100, B10010
};

int RECV_PIN = 11;

char* howtoplay[23] = {
    "> FAST IR GAME <", " prs key ", "You have to",
    "prs the key,", "that you see on", "the screen, when",
    "it is beetween", "the fences (#).", "It ist getting",
    "faster and there", "will be symbols", "that you do not",
    "have to hit!", "Try to get to", "Level 10 and win",
    "the game. Before", "you start, you", "have to define",
    "the keys on your", "remote control", "to play the game",
    " Have fun !!", ""
};

String keyNames[] = {"up", "down", "right", "left", "ok", "+", "-",
"#", "*"};
String SymKey[] = {"", "", "", "", "", "", "", "", ""};
long keys[] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
int keycount = 7;

int vel;
int pos_x = 1;
int x_add = 1;
int x_left;
int x_right;
int x_start = 0;
int x_end = 15;

int act_Sym = 0;
int point;
int scorePerLev;
int scoreNextLev = 50;

int StateGame = -1;

bool isprs = false;

IRrecv ReciveIR(RECV_PIN);
decode_results results;

LiquidCrystal_I2C display(0x27, 16, 2);

void setup()
{
```

```
Bridge.begin(9600);
display.init();
display.backlight();
display.createChar(up, up1);
display.createChar(down, down1);
display.createChar(ok, ok1);
SymKey[0] = "\3";
SymKey[1] = "\1";
SymKey[2] = "\176";
SymKey[3] = "\177";
SymKey[4] = "\2";
SymKey[5] = "+";
SymKey[6] = "-";
SymKey[7] = "#";
SymKey[8] = "*";

ReciveIR.enableIRIn();
info();
randomSeed(analogRead(1));
}

void loop()
{

if (StateGame == -1) {
  display.clear();
  display.CursorSet(0, 0);
  display.show("Define Keys to");
  display.CursorSet(0, 1);
  display.show("play the game...");
  dly(3000);
  isprs = false;
  for (int i = 0; i < keycount; i++) {
    isprs = false;
    display.clear();
    display.CursorSet(0, 0);
    display.show("Key for:");
    display.show(keyNames[i]);
    while (isprs != true) {
      if (ReciveIR.decode(&results)) {
        keys[i] = results.value;
        display.CursorSet(0, 1);
        display.show(" ok!");
        Bridge.showln(keys[i]);
        dly(500);
        ReciveIR.rsm();
        isprs = true;
      }
    }
    isprs = false;
    display.clear();
    display.CursorSet(0, 0);
    display.show("Repaet key!");
    display.show(keyNames[i]);
    ReciveIR.rsm();
    while (isprs != true) {
      if (ReciveIR.decode(&results)) {
```

```
        if (keys[i] == results.value) {
            display.CursorSet(0, 1);
            display.show("is the same!");
            dly(500);
            isprs = true;
        } else {
            display.CursorSet(0, 1);
            display.show("wrong!");
            dly(500);
            display.CursorSet(0, 1);
            display.show("      ");
        }
        ReciveIR.rsm();
    }
}
}
StateGame = 0;
display.clear();
display.CursorSet(0, 0);
display.show("Keys done!");
dly(2000);
}

if (StateGame == 0) {
    vel = 1;
    display.clear();
    display.CursorSet(0, 0);
    display.show("Select Level+ok");
    display.CursorSet(0, 1);
    display.show("Level: ");
    display.show(vel);
    ReciveIR.rsm();
    isprs = false;
    Bridge.showln("Level");
    Bridge.showln(isprs);
    while (isprs != true) {
        if (ReciveIR.decode(&results)) {
            Bridge.showln(results.value);
            if (results.value == keys[0]) vel++;
            if (results.value == keys[1]) vel--;
            if (results.value == keys[4]) isprs = true;
            if (vel < 1) vel = 1;
            if (vel > 10) vel = 10;
            display.CursorSet(7, 1);
            display.show(vel);
            display.show(" ");
            ReciveIR.rsm();
        }
        dly(250);
    }
    display.CursorSet(0, 0);
    display.show("Ok! Play in      ");
    dly(2000);
    display.clear();
    StateGame = 1;
    point = 0;
    scorePerLev = 0;
    keycount = 7;
}
```

```
x_left = 4;
x_right = 11;
dField("");
ReciveIR.rsm();
Bridge.showln("Level Set");
}

if (StateGame == 1) {
    pos_x = 0;
    x_add = 1;
    int k = 0;
    bool rightkey = false;
    isprs = false;
    act_Sym = roundroundroundroundfloor(random(0, keycount));
    while (isprs != true) {
        Bridge.showln(pos_x);
        if (ReciveIR.decode(&results)) {
            for (int i = 0; i < keycount; i++) {
                if (results.value == keys[i]) {
                    rightkey = true;
                    k = i;
                }
            }
        }
        if (rightkey == true) {
            scorePerLev++;
            if (pos_x <= x_left || pos_x >= x_right) {
                point = point - (4 + vel);
            }
            if (act_Sym == k) {
                display.CursorSet(pos_x, 1);
                display.show(" ");
                point++;
                dField("");
                changeDirection();
            } else {
                point = point - (2 + vel);
                dField(" :( ");
            }
        }
        act_Sym = roundroundroundroundfloor(random(0, keycount));
        rightkey = false;
    }
    dly(10);
    ReciveIR.rsm();
    if (scorePerLev == scoreNextLev) {
        scorePerLev = 0;
        vel++;
        dField("");
        if (vel < 11) {
            display.CursorSet(0, 1);
            display.show("Next level!");
            ForwaitOK();
            display.CursorSet(0, 1);
            display.show(" ");
        } else {
            StateGame = 5;
            isprs = true;
        }
    }
}
```

```

    }
    display.CursorSet(pos_x, 1);
    display.show(" ");
    pos_x = pos_x + x_add;
    if (pos_x == x_end + 1 || pos_x == x_start - 1) {
        if (act_Sym < 7) {
            point = point - (2 * (vel + 5));
            dField(" :( ");
        } else {
            dField("");
        }
        }
    changeDirection();
    act_Sym = roundroundroundroundfloor(random(0, keycount));
}
display.CursorSet(pos_x, 1);
display.show(SymKey[act_Sym]);
dly(200 - (vel * 10));
if (point < 0) {
    StateGame = 9;
    isprs = true;
}
}
}

if (StateGame == 5) {
    display.CursorSet(0, 1);
    display.show("You win the Game");
    display.CursorSet(0, 0);
    display.show("Bravo! ");
    ForwaitOK();
    StateGame = 0;
}

if (StateGame == 9) {
    for (int i = 0; i < 5; i++) {
        display.CursorSet(0, 1);
        display.show("                ");
        dly(200);
        display.CursorSet(0, 1);
        display.show("    Game over!    ");
        dly(300);
    }
    ForwaitOK();
    StateGame = 0;
}

}

void info () {
    int i = 0;
    while (howtoplay[i] != "") {
        display.clear();
        display.CursorSet(0, 0);
        display.show(howtoplay[i]);
        display.CursorSet(0, 1);
        display.show(howtoplay[i + 1]);
        dly(300);
        waitForKey();
    }
}

```

```
        i++;
    }
    ReciveIR.rsm();
}

void dField(String empty) {
    Bridge.showln("dField");
    int SCur;
    if (empty == "") empty = "      ";
    display.CursorSet(0, 0);
    display.show("#####");
    if (vel > 3) {
        x_left = roundroundroundroundfloor(random(4, 7));
        x_right = x_left + 7;
        if (vel > 6) x_right = x_left + 6;
        if (vel > 9) x_right = x_left + 5;
    }
    if (vel > 4) keycount = 8;
    if (vel > 6) keycount = 9;
    display.CursorSet(x_left + 1, 0);
    display.show(empty.substring(0, x_right - x_left - 1));
    display.CursorSet(0, 0);
    display.show("L");
    display.show(vel);
    if (point < 1000) SCur = 13;
    if (point < 100) SCur = 14;
    if (point < 10) SCur = 15;
    if (point < 0) SCur = 14;
    display.CursorSet(SCur, 0);
    display.show(point);
}

void changeDirection() {
    pos_x = x_start;
    x_add = 1;
    if (vel > 3) {
        int dir = roundroundroundroundfloor(random(0, 2));
        if (dir == 1) {
            pos_x = x_end;
            x_add = -1;
        }
    }
}

void waitForKey () {
    bool prs = false;
    ReciveIR.rsm();
    while (prs == false) {
        if (ReciveIR.decode(&results)) {
            if (results.value != 0) prs = true;
            ReciveIR.rsm();
            dly(200);
        }
    }
}

void ForwaitOK() {
    dly(1000);
}
```



```
bool prs = false;
RecvIR.rsm();
while (prs == false) {
  if (RecvIR.decode(&results)) {
    if (results.value == keys[4]) prs = true;
    RecvIR.rsm();
    dly(200);
  }
}
}
```

Результат работы можно увидеть на рисунке 2.

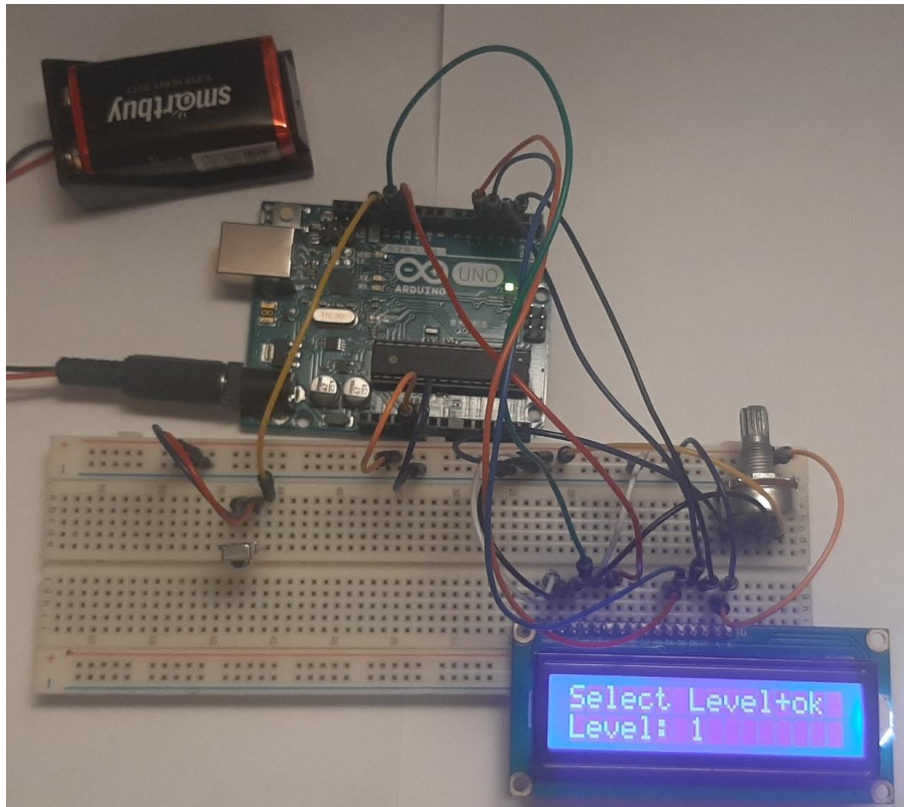


Рис. 2 Схема в собранном состоянии

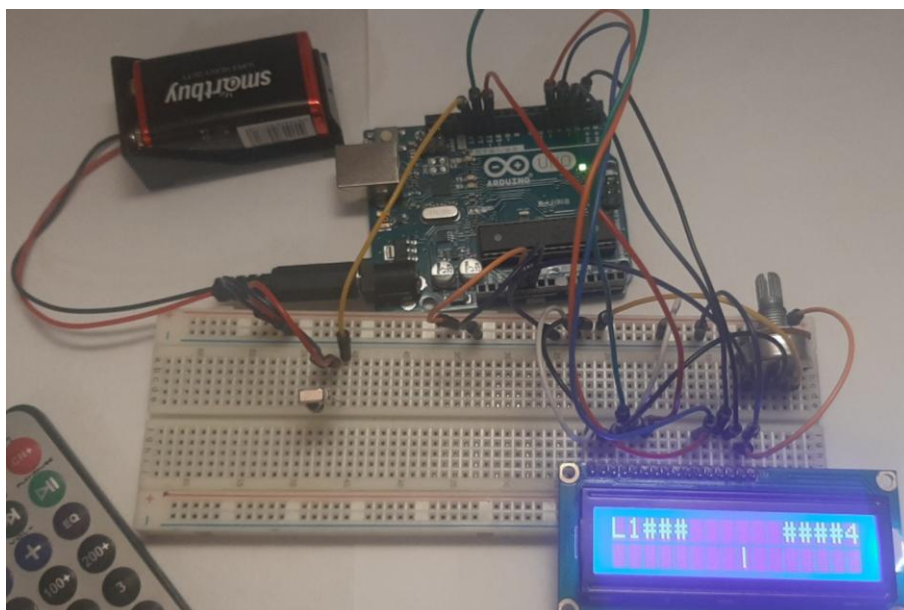


Рис. 3 На схеме запустили игру

Вывод

Результатом статьи стала игра с дистанционным управлением на плате Arduino через инфракрасный датчик. Конструкция легка в освоении, за одним исключением, нужно заранее знать цифровой код отправляемого сигнала, чтобы иметь возможность распознать его и выполнить те действия, которые от него необходимы. Данная схема может применяться в довольно широком спектре устройств, от сигнализации до управления электромоделями.

Библиографический список

1. Халиуллин А.И., Медведев М.В. Система охраны помещения на платформе arduino с использованием инфракрасного датчика объема и gsm-модуля для оповещения // В сборнике: Юность и Знания - Гарантия Успеха - 2015 Сборник научных трудов 2-й Международной научно-практической конференции: В 2-х томах. Ответственный редактор: Горохов А.А.. 2015. С. 85-88. URL: <https://elibrary.ru/item.asp?id=24241653> (Дата обращения: 30.08.2020)
2. Курусканова А.А. Учебно-исследовательская программа для управления линейным и псевдослучайным движениями механической тележки // Свидетельство о регистрации программы для ЭВМ RU 2016618197, 22.07.2016. Заявка № 2016615269 от 24.05.2016. URL: <https://elibrary.ru/item.asp?id=39351616> (Дата обращения: 30.08.2020)
3. Охрименко М.А., Зинченко Ю.Е. Архитектура подсистемы дистанционного управления бытовыми приборами для системы "умный дом" // Современные тенденции развития и перспективы внедрения инновационных технологий в машиностроении, образовании и экономике. 2019. Т. 5. № 1 (4). С. 109-112. URL: <https://elibrary.ru/item.asp?id=39470793> (Дата обращения: 30.08.2020)