**Materials for Connected Home**

*from Dana Agybayeva, work was done together with Alua Zholgazy*

What kind of circuits were done:

1. LED on/off with IR remote controller (for RM-link)
2. IR switch
3. Контроль света на основе датчика движения (микроволновой, инфракрасный /PIR/)
4. ESP flash over USB
5. ESP flash over air (WiFi)
6. ВКЛ/ВЫКЛ лампочек с веб-страницы с помощью микроконтроллера ESP32

All codes are saved on GitHub (private):

<https://github.com/danadanon/Materials-for-Connected-Home>

|  |  |
| --- | --- |
| **#1. LED on/off with IR remote controller (for RM-link)** | |
| #include "IRremote.h"  IRrecv irrecv(3); // указываем вывод, к которому подключен приемник  int ledBlue = 4;  int ledRed = 7;  int ledYellow = 2;  decode\_results results;  int valBlue = 0;  int valRed = 0;  int valYellow = 0;  void setup() {  irrecv.enableIRIn(); // запускаем прием  pinMode(ledBlue, OUTPUT);  pinMode(ledRed, OUTPUT);  pinMode(ledYellow, OUTPUT);  }  void loop() {  valBlue = digitalRead(ledBlue);  valRed = digitalRead(ledRed);  valYellow = digitalRead(ledYellow);    if ( irrecv.decode( &results )) { // если данные пришли  switch ( results.value ) {  case 0xFF6897: // if press 0 all LEDs are OFF  if (valBlue==HIGH&&valRed==HIGH&&valYellow==HIGH){  digitalWrite( ledBlue, LOW );  digitalWrite( ledYellow, LOW );  digitalWrite( ledRed, LOW );  }  else if (valBlue==LOW&&valRed==LOW&&valYellow==LOW){  digitalWrite( ledBlue, HIGH );  digitalWrite( ledYellow, HIGH );  digitalWrite( ledRed, HIGH );  }  break;    case 0xFF30CF: // press 1 -> LED blue - ON  if (valBlue==LOW){  digitalWrite( ledBlue, HIGH );  }  else {  digitalWrite( ledBlue, LOW );  }  break;  case 0xFF18E7: // press 2 -> LED red - ON  if (valRed==0){  digitalWrite( ledRed, 1 );  }  else if (valRed==HIGH){  digitalWrite( ledRed, LOW );  }  break;  case 0xFF7A85: // press 3 -> LED yellow - ON  if (valYellow==LOW){  digitalWrite( ledYellow, HIGH );  }  else if (valYellow==HIGH){  digitalWrite( ledYellow, LOW );  }  break;  }  Serial.println( results.value, HEX );  irrecv.resume(); // принимаем следующую команду  }  } | * This circuit is necessary for self-learned RM-link. * Necessary materials:   Arduino UNO/NANO  3 LEDs, 3 resistors  IR receiver  Arduino IR remote controller (4 buttons are used) |
| https://schematicdiagrams.net/diagrams/21/2014/01/arduino-led-ir-remote-schematic.jpg |
| **#2. IR switch** | |
| int Sensor = 2; // RCWL-0516 Input Pin  int LED = 13; // Use the onboard Uno LED  int isObstaclePin = 7; // This is our input pin  int isObstacle = HIGH; // HIGH MEANS NO OBSTACLE  byte led=0; //Состояние светодиода  byte oldled=1; //Последнее состояние //светодиода, для исключения ложных //переключений  void setup() {  pinMode (Sensor, INPUT); // RCWL-0516 as input  pinMode(isObstaclePin, INPUT);  pinMode (LED, OUTPUT); // LED as OUTPUT  digitalWrite(LED, LOW); // Turn LED Off  Serial.begin(9600);  }  void loop() {  isObstacle = digitalRead(isObstaclePin);    if (isObstacle == LOW) //sensor is activated  { Serial.println("OBSTACLE!!, OBSTACLE!!");  if (led==oldled) { //Проверка, что //состояние кнопки изменилось  led=!led;  Serial.println("LED was changed");  }  } else { //Когда не нажата  oldled=led;  }  digitalWrite(LED,led);  } | * This circuit is necessary for turn ON/OFF any light by no touching the switch itself * Necessary materials:   Arduino NANO/UNO  1 LED, 1 resistor  Line Tracking Sensor Module – KY-033 |
|  |
| **#3. Контроль света на основе датчика движения (микроволновой, инфракрасный /PIR/)** | |
| 1. RCWL-0516 microwave proximity sensor   RCWL-0516 Pinout   1. PIR Motion Sensor   pir | Цель: построить цепь, чувствующую присутствие человека – но работает на основе чувствительности к движению человека  Results of comparison: RCWL are similar to PIR (when PIR has sensitivity MAX and delay MIN)  !!! Delay of the adjusted creates delay not only for input HIGH, but also LOW |
|  |  |
| **#4. ESP flash over USB** | |
| 1. Удаляем существующую прощивку:   esptool.py --port COM9 --baud 460800 erase\_flash   1. Загружаем нужную нам прошивку:   esptool.py --port COM9 --baud 460800 write\_flash --flash\_size=detect -fm dio 0x1000 bootloader.bin 0x10000 espruino\_esp32.bin 0x8000 partitions\_espruino.bin | Work with ESP32 on ESPRuino WEB IDE   * Board: ESP32 * Language: JavaScript * Software: ESPRuino WEB IDE * Source from the WEB: <http://tech.sparkfabrik.com/2017/03/01/espruino-nodemcu-step-by-step/> |
|  |  |
| **#5. ESP flash over air (WiFi)** | |
| //boot Flash OTA on ESP32 \*\*\*  #include <WiFi.h>  #include <WiFiClient.h>  #include <WebServer.h>  #include <ESPmDNS.h>  #include <Update.h>  #include <ArduinoOTA.h>  const char\* host = "esp32-webupdate";  const char\* ssid = "BeInTech-Ventures";  const char\* password = "BI15711588";  WebServer server(80);  const char\* serverIndex = "<form method='POST' action='/update' enctype='multipart/form-data'><input type='file' name='update'><input type='submit' value='Update'></form>";  void setup() {  Serial.begin(115200);  pinMode(16,OUTPUT);  Serial.println();  Serial.println("Booting Sketch...");  WiFi.mode(WIFI\_AP\_STA);  WiFi.begin(ssid, password);  if (WiFi.waitForConnectResult() == WL\_CONNECTED) {  MDNS.begin(host);  server.on("/", HTTP\_GET, []() {  server.sendHeader("Connection", "close");  server.send(200, "text/html", serverIndex);  });  server.on("/update", HTTP\_POST, []() {  server.sendHeader("Connection", "close");  server.send(200, "text/plain", (Update.hasError()) ? "FAIL" : "OK");  ESP.restart();  }, []() {  HTTPUpload& upload = server.upload();  if (upload.status == UPLOAD\_FILE\_START) {  Serial.setDebugOutput(true);  Serial.printf("Update: %s\n", upload.filename.c\_str());  if (!Update.begin()) { //start with max available size  Update.printError(Serial);  }  } else if (upload.status == UPLOAD\_FILE\_WRITE) {  if (Update.write(upload.buf, upload.currentSize) != upload.currentSize) {  Update.printError(Serial);  }  } else if (upload.status == UPLOAD\_FILE\_END) {  if (Update.end(true)) { //true to set the size to the current progress  Serial.printf("Update Success: %u\nRebooting...\n", upload.totalSize);  } else {  Update.printError(Serial);  }  Serial.setDebugOutput(false);  } else {  Serial.printf("Update Failed Unexpectedly (likely broken connection): status=%d\n", upload.status);  }  });  server.begin();  MDNS.addService("http", "tcp", 80);  Serial.printf("Ready! Open http://%s.local in your browser\n", host);    Serial.printf("Was flashed on: ", host);  Serial.println(WiFi.localIP());  } else {  Serial.println("WiFi Failed");  }  }  void loop() {  server.handleClient();  delay(1);    digitalWrite(16,1);  delay(500);  digitalWrite(16,0);  delay(500);  } | * Board: ESP32 * Language: C++ * Software: Arduino IDE * Source***:***   <https://www.youtube.com/watch?v=1aE9CnBfH3I>  Settings:    ***Result:***  ***Firstly***, code is downloaded from **Arduino IDE** onto **DOIT ESP32 DEVKIT** through **COM9:**    ***Secondly***, WEB page with IP address **10.3.178.96** was created:    ***Now***, we can send FLASH on ESP32 **OTA** with the help of **WiFi** (**BeInTech-Ventures**) by downloading the sketch on the above mentioned web site.  As a result, sketch is booted on ESP32 and **OK** message as a SUCCESS message is shown. |
| **#6. ВКЛ/ВЫКЛ лампочки с веб-страницы с помощью микроконтроллера ESP32** | |
| var wifi = require("Wifi");  wifi.startAP('EspruinoAP', {  password: '0123456789',  authMode: 'wpa2' },function() {  console.log(`AP started`);  });  function onPageRequest(req, res) {  var b = url.parse(req.url, true);    if (b.pathname=="/") {  res.writeHead(200, {'Content-Type': 'text/html'});  res.write("<H1><a href=\"/on\"><button><center>ON</center></button></a></H1>");  res.end("<H1><a href=\"/off\"><button><center>OFF</center></button></a></H1>");  } else if (b.pathname=="/on") {  res.writeHead(200, {'Content-Type': 'text/html'});  res.write("<a href=\"/on\"><button><svg></svg><center>ON</center></button></a>");  res.end("<H1><a href=\"/off\"><button><center>OFF</center></button></a></H1>");  //res.end("Enable");  digitalWrite(D16, false);  } else if (b.pathname=="/off") {  res.writeHead(200, {'Content-Type': 'text/html'});  res.write("<H1><a href=\"/on\"><button><center>ON</center></button></a></H1>");  res.end("<H1><a href=\"/off\"><button><center>OFF</center></button></a></H1>");  //res.end("Disable");  D16.write(true);  } else {  res.writeHead(404, {'Content-Type': 'text/plain'});  res.end("404: Page "+b.pathname+" not found");  }  }  require("http").createServer(onPageRequest).listen(80); | * Board: ESP32 * Language: JavaScript * Software: ESPRuino WEB IDE * Source: <https://www.espruino.com/Internet>  1. Из платы ESP32 делаем точку доступа с названием 'EspruinoAP', паролем '0123456789'. В браузере заходим на IP address '192.168.4.1', где должна открыться следующая вкладка:      1. При нажатии на ON, включается синяя LED на пине D16        1. При нажатии на OFF, отключается синяя LED на пине D16 |
|  |  |