Alarm pohybu v miestnosti

**Vypracoval:** Mgr. Dalibor Gajda

**Rok :**2024/2025

**Predmet:** Internet vecí

**Ročník:**2. RŠi

**Cieľ projektu**

Cieľom projektu je vytvoriť detektor pohybu využívajúci mikrokontrolér ESP32, PIR senzor HC-SR501, OLED displej SSD1106 a LED diódu s rezistorom. Pri detekcii pohybu sa LED dióda rozsvieti a na displeji sa zobrazí správa „Pohyb zaznamenaný!“. Ak pohyb prestane, displej zobrazí „Žiaden pohyb“ a LED dióda zhasne.

**Použitie v praxi**

Projekt môže byť využitý v rôznych oblastiach automatizácie:

* **Zabezpečenie domácností**: Aktivácia osvetlenia alebo alarmu pri pohybe.
* **Monitoring priestoru**: Zisťovanie prítomnosti osôb v miestnostiach.

**Použité komponenty a ich funkcia**

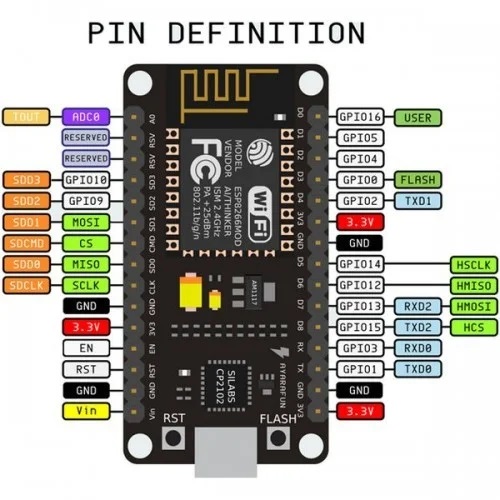
**1. Mikrokontrolér ESP32 (NodeMCU Lolin V3)**

Hlavný ovládací prvok projektu, ktorý spracováva údaje zo senzora, ovláda LED diódu a komunikuje s OLED displejom.

Obrázok, na ktorom je elektrinžinierstvo, elektronika, elektronická súčiastka, súčiastka obvodu

Automaticky generovaný popis

Rozloženie pinov na našom ESP32



**2. PIR senzor HC-SR501**

Senzor detekujúci pohyb na základe zmien teploty v prostredí. Generuje digitálny signál pre ESP32 pri zaznamenaní pohybu.

Obrázok, na ktorom je hračka, vnútri

Automaticky generovaný popis

**Nastavenie PIR senzora HC-SR501**

Na senzore sú dva nastaviteľné potenciometre:

1. **Časové oneskorenie**: Nastavuje, ako dlho zostane výstup HIGH po detekcii pohybu.
2. **Citlivosť detekcie**: Nastavuje vzdialenosť, na ktorú senzor reaguje.

**3. OLED displej SSD1106**

Displej zobrazuje textové správy o prítomnosti alebo absencii pohybu.

Obrázok, na ktorom je elektronika, elektronické zariadenie, gadget, podlaha

Automaticky generovaný popis

**4. LED dióda**

Indikátor pohybu, ktorý sa rozsvieti pri detekcii a zhasne pri absencii pohybu.

Obrázok, na ktorom je elektronika, elektroinštalácia, elektrinžinierstvo, kábel

Automaticky generovaný popis

**5. Rezistor (220 Ω)**

Obmedzuje prúd cez LED diódu, čím zabraňuje jej poškodeniu.

**6. Prepojovacie vodiče a breadboard**

Používajú sa na flexibilné a dočasné spojenie komponentov.

**Zapojenie komponentov**

**ESP32 a PIR senzor HC-SR501**

* VCC (PIR senzor) → 3.3V (ESP32)
* GND (PIR senzor) → GND (ESP32)
* OUT (PIR senzor) → GPIO16 (D0 na ESP32)

**ESP32 a OLED displej SSD1106 (I2C)**

* VDD (OLED) → 3.3V (ESP32)
* GND (OLED) → GND (ESP32)
* SCK (OLED) → GPIO5 (ESP32)
* SDA (OLED) → GPIO4 (ESP32)

**ESP32 a LED dióda cez rezistor**

* Katóda LED diódy → GND (ESP32)
* Anóda LED diódy → GPIO2 (D4 na ESP32) cez rezistor (220 Ω)

Zapojenie vytvorené vo wokwi.com

Obrázok, na ktorom je snímka obrazovky, elektronika, elektrinžinierstvo, okruh

Automaticky generovaný popis

**Postup práce**

1. **Inštalácia vývojového prostredia**: Stiahnite a nainštalujte Arduino IDE z [arduino.cc](https://www.arduino.cc/).
2. **Inštalácia drivera pre ESP32**: Postupujte podľa návodu na [Instructables](https://www.instructables.com/Get-Started-With-ESP8266-NodeMCU-Lolin-V3/).
3. **Pripojenie ESP32**: Použite Micro USB kábel na pripojenie ESP32 k počítaču.
4. **Testovanie drivera**: Nahrajte jednoduchý program pre výstup do sériového monitora na overenie správneho fungovania drivera. (blikanie integrovanej diódy na ESP32)

#define LEDpin D4 //set led pin

// the setup function runs once when you press reset or power the board

void setup() {

// initialize digital pin LED\_BUILTIN as an output.

pinMode(LEDpin, OUTPUT);

}

// the loop function runs over and over again forever

void loop() {

digitalWrite(LEDpin, HIGH); // turn the LED on (HIGH is the voltage level)

delay(1000); // wait for a second

digitalWrite(LEDpin, LOW); // turn the LED off by making the voltage LOW

delay(1000); // wait for a second

}

1. **Zapojenie komponentov**: Spojte ESP32, PIR senzor, LED diódu a OLED displej podľa vyššie uvedeného schémy.
2. **Inštalácia knižníc**: V Arduino IDE nainštalujte knižnice Adafruit BusIO, Adafruit GFX Library a OLED SSD1306 - SH1106, ktoré umožňujú ovládanie displeja a senzora.
3. **Nahratie programu**: Nahrajte kód do ESP32 a sledujte výstupy na OLED displeji a LED dióde.
4. **Testovanie**: Skúste detegovať pohyb a pozorujte zmeny na LED dióde a displeji.

Fotografia zariadenia keď nezaznamenáva pohyb

Obrázok, na ktorom je kábel, elektronika, elektroinštalácia, elektrinžinierstvo

Automaticky generovaný popis

Fotografia zariadenia keď zaznamenáva pohyb

Obrázok, na ktorom je kábel, elektronika, elektroinštalácia, elektrinžinierstvo

Automaticky generovaný popis

**Kód pre Arduino IDE**

#include <SPI.h> // SPI - serial peripheral interface - komunikacia s perifernymi zariadeniami

#include <Wire.h> // Wire - umoznuje komunikaciu s I2C zariadeniami//

#include <Adafruit\_GFX.h> // zahrnutie grafickej kniznice

#include <oled.h> // zahrnieme konkretny typ nasho OLED displeja

#define SCREEN\_WIDTH 128 // sirka OLED, v pixeloch

#define SCREEN\_HEIGHT 64 // vyska OLED, v pixeloch

OLED display(D2,D1,NO\_RESET\_PIN,OLED::W\_128,OLED::H\_64,OLED::CTRL\_SH1106,0x3C); // vytvorenie objektu OLED displeja pripojeneho k I2C

#define MOTION\_SENSOR\_PIN D0 // ESP32 pin GPI016 pripojeny k OUTPUT-pinu pohyboveho senzora

#define LED\_PIN D4 // ESP32 pin GPI02 pripojeny k LED pin

int motionStateCurrent = LOW; // sucasny stav pinu pohyboveho senzora

int motionStatePrevious = LOW; // predchadzajuci stav pinu pohyboveho senzora

void setup() {

  Serial.begin(9600); // zaciatok, nastavenie prenosovej rychlosti

  pinMode(MOTION\_SENSOR\_PIN, INPUT); // nastavenie ESP32 pinu do vstupneho modu

  pinMode(LED\_PIN, OUTPUT); // nastavenie ESP32 pinu do vystupneho modu

  // inicializacia OLED displeja s I2C adresou 0x3C

display.begin();

}

void loop() { // opakujuci sa cyklus

  motionStatePrevious = motionStateCurrent; // ulozenie povodneho stavu

  motionStateCurrent = digitalRead(MOTION\_SENSOR\_PIN); // detekcia noveho stavu

  if (motionStatePrevious == LOW && motionStateCurrent == HIGH) { // zmena stavu na pine: LOW -> HIGH (senzor nieco registruje)

    Serial.println("Pohyb zaznamenany! zapnutie LED"); // zobrazenie hlasenia v monitore serioveho portu

    digitalWrite(LED\_PIN, HIGH); // zapnutie pinu LED diody

display.clear(); // zmazanie displeja

 display.draw\_string(1,1,"Pohyb zaznamenany!"); // zobrazenie hlasenia v monitore serioveho portu

  display.display(); // nastavenie obsahu zobrazeneho textu

  delay(500);

  } else if (motionStatePrevious == HIGH && motionStateCurrent == LOW) { // zmena stavu na pine: HIGH -> LOW (senzor uz nic neregistruje)

    Serial.println("Pohyb ukonceny!, vypnutie LED");

   display.clear(); // zmazanie displeja

   display.draw\_string(1,1,"Ziaden pohyb!"); // zobrazenie hlasenia v monitore serioveho portu

   display.display(); // nastavenie obsahu zobrazeneho textu

   delay(500); // oneskorenie (0,5 sekunda)

   digitalWrite(LED\_PIN, LOW); // vypnutie pinu LED diody

  }

  delay(1000);

  } // oneskorenie (1 sekunda)

**Zdroje:**

[**https://esp32io.com/tutorials/esp32-motion-sensor  
https://esp32io.com/tutorials/esp32-motion-sensor-led**](https://esp32io.com/tutorials/esp32-motion-sensorhttps:/esp32io.com/tutorials/esp32-motion-sensor-led) [**https://esp32io.com/tutorials/esp32-oled  
https://lastminuteengineers.com/oled-display-arduino-tutorial/**](https://esp32io.com/tutorials/esp32-oledhttps://lastminuteengineers.com/oled-display-arduino-tutorial/)