

Úvod do programování mikropočítačů

(Určeno pro vnitřní potřebu SOUE Plzeň, zveřejňování bez předchozího souhlasu je zakázáno)

Modul 1 Sekce 15.1 – detekce pohybu PIR senzor HC-SR501 a 7Seg displej

Nahradíme **DHT11** senzorem **HC-SR501 PIR**.

1.15.1.1 Příklad zapojení „1111“

Cíl: Při detekci pohybu zobrazíme na 7-segmentovém displeji čtyři jedničky ("1111").

Zapojení HC-SR501:

HC-SR501	Arduino
VCC	5V
GND	GND
OUT	Pin 5

OUT pin PIR senzoru připojíme na digitální pin 5, který byl dříve používán pro DHT11.

Zapojení 7-segmentového displeje zůstává stejné:

Displej	Arduino
Digit 1	12
Digit 2	9
Digit 3	8
Digit 4	6
Segment A	11
Segment B	7
Segment C	4
Segment D	3
Segment E	A1
Segment F	A2
Segment G	A3
Segment DP	A4

Upravený kód pro HC-SR501 a 7-segmentový displej:

Kód zobrazí "1111", když PIR detekuje pohyb, a "0000", když pohyb zmizí.

```
#include <SevSeg.h> // Knihovna pro 7-segmentový displej
```

```

#define PIR_PIN 5 // Pin, kam připojíme PIR senzor

SevSeg sevseg; // Objekt pro práci s displejem
volatile bool pohybDetekovan = false; // Proměnná pro uchování
stavu senzoru

void ISR_PIR() {
    pohybDetekovan = true; // Přerušení nastaveno při detekci pohybu
}

void setup() {
    Serial.begin(9600);

    // Nastavení pinu PIR senzoru jako vstup
    pinMode(PIR_PIN, INPUT);

    // Přerušení pro detekci pohybu na PIR senzoru
    attachInterrupt(digitalPinToInterrupt(PIR_PIN), ISR_PIR, RISING);

    // Nastavení 7-segmentového displeje
    byte numDigits = 4;
    byte digitPins[] = {12, 9, 8, 6}; // Piny pro číslice
    byte segmentPins[] = {11, 7, 4, 3, A1, A2, A3, A4}; // Piny pro
segmenty (A, B, C, D, E, F, G, DP)
    bool resistorsOnSegments = true; // Jsou rezistory na segmentech?
    byte hardwareConfig = COMMON_CATHODE; // Typ displeje
    bool updateWithDelays = false;
    bool leadingZeros = false;
    bool disableDecPoint = false;

    sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins,
resistorsOnSegments);
    sevseg.setBrightness(90);
}

void loop() {
    if (pohybDetekovan) {
        Serial.println("Pohyb detekován!");
        sevseg.setNumber(1111); // Zobrazení čísla "1111"
        delay(5000); // Zůstane zapnuté 5 sekund
        pohybDetekovan = false; // Resetujeme stav senzoru
    } else {
        sevseg.setNumber(0000); // Zobrazení "0000", když pohyb není
detekován
    }

    sevseg.refreshDisplay(); // Aktualizace displeje
}

```

Popis kódu:

- Když PIR detekuje pohyb, aktivuje přerušení `ISR_PIR()`, které nastaví `pohybDetekovan = true`.
- V `loop()` se podle proměnné `pohybDetekovan` zobrazí:
 - "1111" pokud byl detekován pohyb.
 - "0000" pokud pohyb zmizel.
- Po detekci displej zůstane 5 sekund na hodnotě "1111", poté se vrátí k „0000“.

1.15.1.2 Zobrazení času od posledního pohybu

Cíle:

- Při detekci pohybu se na displeji zobrazí "0000".
- Každou sekundu se číslo zvýší o 1, dokud nebude detekován další pohyb.
- Při novém pohybu se číslo resetuje zpět na 0000.

Upravený kód s počítáním času od posledního pohybu:

```
#include <SevSeg.h> // Knihovna pro 7-segmentový displej

#define PIR_PIN 5 // Pin pro PIR senzor

SevSeg sevseg; // Objekt pro práci s displejem
volatile bool pohybDetekovan = false; // Proměnná pro uchování
stavu senzoru
unsigned long posledniPohybCas = 0; // Ukládá čas posledního pohybu

// ISR pro PIR senzor - aktivuje přerušení při detekci pohybu
void ISR_PIR() {
    pohybDetekovan = true;
    posledniPohybCas = millis(); // Uložíme aktuální čas
}

void setup() {
    Serial.begin(9600);

    pinMode(PIR_PIN, INPUT);
    attachInterrupt(digitalPinToInterrupt(PIR_PIN), ISR_PIR, RISING);

    // Nastavení 7-segmentového displeje
    byte numDigits = 4;
    byte digitPins[] = {12, 9, 8, 6}; // Piny pro číslice
    byte segmentPins[] = {11, 7, 4, 3, A1, A2, A3, A4}; // Piny pro
segmenty (A, B, C, D, E, F, G, DP)
    bool resistorsOnSegments = true;
    byte hardwareConfig = COMMON_CATHODE;

    sevseg.begin(hardwareConfig, numDigits, digitPins, segmentPins,
resistorsOnSegments);
```

```

    sevseg.setBrightness(90);
}

void loop() {
    // Čas od poslední detekce pohybu (v sekundách)
    unsigned long casOdPoslednihoPohybu = (millis() -
posledniPohybCas) / 1000;

    if (pohybDetekovan) {
        Serial.println("Pohyb detekován!");
        posledniPohybCas = millis(); // Resetujeme čas při detekci
pohybu
        pohybDetekovan = false; // Reset proměnné
    }

    // Zobrazíme počet sekund od posledního pohybu
    sevseg.setNumber(casOdPoslednihoPohybu);
    sevseg.refreshDisplay(); // Aktualizace displeje

    delay(1000); // Každou sekundu se aktualizuje displej
}

```

Popis kódu:

- Při detekci pohybu se displej resetuje na "0000".
- Každou sekundu se číslo na displeji zvýší o 1, dokud nebude znovu detekován pohyb.
- Pokud PIR znovu detekuje pohyb, číslo se resetuje a počítání začne od nuly.

Odkazy:

Zbyšek Voda - Průvodce světem Arduina

<https://navody.drateg.cz/navody-k-produktum/pohybove-cidlo-hc-sr501.html>

office 365 copilot