

Programi za projekte v članku

Načrtovanje projektov iz elektronike (3. del)

Umetna inteligenca pri kodiranju programov za programirljive krmilnike

Program za utripajočo LED-diodo (01_ArduinoLED.ino):

```
void setup() {  
  pinMode(2, OUTPUT); // pin 2 kot izhod  
}
```

```
void loop() {  
  digitalWrite(2, HIGH); // LED sveti  
  delay(500); // počakaj 500 ms  
  digitalWrite(2, LOW); // LED ne sveti  
  delay(500); // počakaj 500 ms  
} //nazaj na Loop
```

Program za letečo luč s petimi LED-diodami (02_ArduinoLetecaLuc.ino):

```
void setup() {  
  // Nastavimo pine od 2 do 6 kot izhode  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
}
```

```
void loop() {  
  // 1. Lučka (Pin 2)  
  digitalWrite(2, HIGH); // Prižgi
```

```
delay(200); // Počakaj 0.2 sekunde
digitalWrite(2, LOW); // Ugasni

// 2. Lučka (Pin 3)
digitalWrite(3, HIGH);
delay(200);
digitalWrite(3, LOW);

// 3. Lučka (Pin 4)
digitalWrite(4, HIGH);
delay(200);
digitalWrite(4, LOW);

// 4. Lučka (Pin 5)
digitalWrite(5, HIGH);
delay(200);
digitalWrite(5, LOW);

// 5. Lučka (Pin 6)
digitalWrite(6, HIGH);
delay(200);
digitalWrite(6, LOW);

// Program se bo tukaj samodejno vrnil na začetek loop-a
}
```

Program za časovno določeno trajanje vklopa LED-diode:

```
void setup() {
// Pin 2 nastavimo kot izhod za LED
pinMode(2, OUTPUT);

// Pin 3 nastavimo kot vhod s PULLUP uporom
```

```
// To pomeni, da je pin normalno "visoko" (HIGH), ko pritisneš tipko, pa "nizko" (LOW)
```

```
pinMode(3, INPUT_PULLUP);
```

```
// Na začetku ugasnemo LED
```

```
digitalWrite(2, LOW);
```

```
}
```

```
void loop() {
```

```
// Preverimo, če je tipka pritisnjena (stanje LOW)
```

```
if (digitalRead(3) == LOW) {
```

```
digitalWrite(2, HIGH); // Prižgi LED
```

```
delay(5000); // Počakaj 5 sekund (5000 milisekund)
```

```
digitalWrite(2, LOW); // Ugasni LED
```

```
}
```

```
}
```

Program za krmiljenje LED-diode prek tipke in glavnega stikala:

```
// Konstanta za čas (v milisekundah)
```

```
const unsigned long TRAJANJE = 5000;
```

```
unsigned long casZacetka = 0;
```

```
bool alarmAktiven = false;
```

```
void setup() {
```

```
pinMode(2, OUTPUT);
```

```
pinMode(3, INPUT_PULLUP); // Tipka
```

```
pinMode(4, INPUT_PULLUP); // Stikalo
```

```
}
```

```
void loop() {
```

```

// 1. Preveri, če je glavno stikalo IZKLOPLJENO
// Ker uporabljamo INPUT_PULLUP, je stikalo vklopljeno, ko je LOW
if (digitalRead(4) == HIGH) {
digitalWrite(2, LOW);
alarmAktiven = false;
return; // Program se vrne na začetek, ne dela ničesar več
}

// 2. Preveri pritisk tipke (samo če alarm še ne teče)
if (digitalRead(3) == LOW && alarmAktiven == false) {
alarmAktiven = true;
casZacetka = millis();
digitalWrite(2, HIGH);
}

// 3. Preveri, če je čas potekel
if (alarmAktiven == true) {
if (millis() - casZacetka >= TRAJANJE) {
digitalWrite(2, LOW);
alarmAktiven = false;
}
}
}

```

Tu je še nekaj idej, ki jih lahko izvedemo s pomočjo usvojenega znanja in UI-orodij:

- Od krmilnika zahtevamo, da ugotovi, kateri od treh predalov je bil odprt.
- Dodamo še tri LED-diode, po eno za vsak predal. Njihova naloga naj bo, da potem, ko piskač utihne, pokažejo, kateri predal je bil v času naše odsotnosti odprt, tudi če je nepovabljeni gost zaradi sproženega alarma zaprl vse predale.
- Projekt lahko razširimo na več predalov.

- Alarm lahko izklopimo le ob hkratnem držanju katere od tipk oziroma ob odprtem točno določenem predalu, s čimer preprečimo, da bi nepovabljeni oseba alarm preprosto izključila.
- Alarm razširimo še na okna, za kar lahko uporabimo isti ali dodatni piskač.