

# PRACOVNÍ LIST – POLE

V TÉTO ČÁSTI BUDETE POKRAČOVAT V PROGRAMOVÁNÍ SOUSTAVY LED DIOD. NAUČÍTE SE OPTIMALIZOVAT PROGRAMOVÝ KÓD POMOCÍ PROGRAMOVÉ STRUKTURY – POLE.

## CO SE NAUČÍTE

- ① Definovat pole pro Arduino.
- ② Využívat pole při definici pinů.
- ③ Procvičíte si přístupy k prvkům pole.



## CO BUDETE POTŘEBOVAT

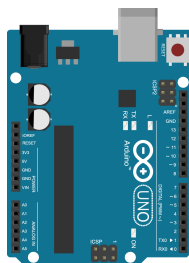
- ① LED diodu - 8x.
- ② Rezistor 220Ω – 8x.
- ③ Desku Arduino.
- ④ Kontaktní pole.
- ⑤ Vodiče typu samec-samec.



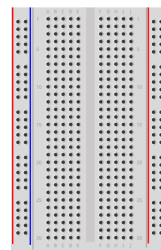
RGB led – 8 kusů



Rezistor 220Ω – 8 kusů



Deska Arduino



Kontaktní pole

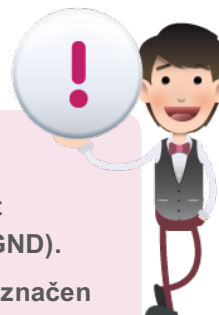
POUŽITÉ SOUČÁSTKY

## A JDĚTE NA TO ...

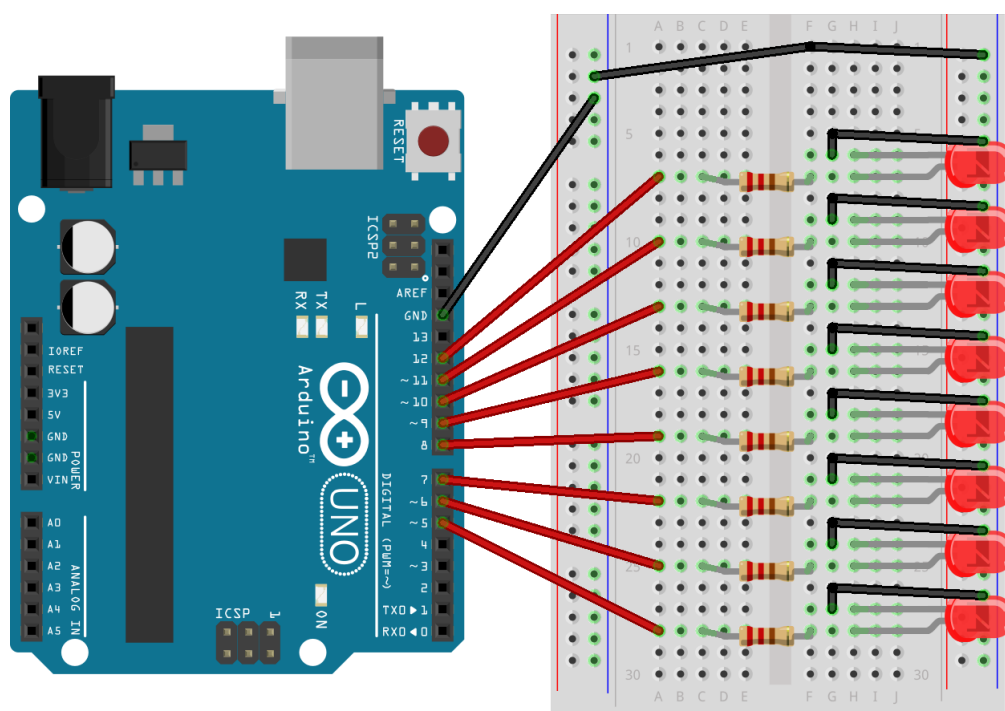
- 1 Pokud máte složený elektronický obvod z minulé hodiny, můžete se pustit rovnou do programování. V opačném případě obvod musíte opět složit podle přiloženého schématu.

### DEJTE SI POZOR

- ➔ Pozor si dejte na to, jak zapojujete LED diody. Delší vývod musí být připojen přes rezistor k pinu. Kratší vývod je připojen na zem (pin GND).
- ➔ Dejte si pozor na hodnotu rezistoru. Zkontrolujte si, že je barevně označen v pořadí červená, červená, modrá černá, zlatá.
- ➔ Všimněte si, jak je zapojen vodič zemnění. Pro přehlednost je veden na druhou stranu kontaktního pole. Následně je zemnění vedeno ke každé LED diodě (černý vodič).



- 2 Otevřete programový kód z minulé hodiny, kde je definována vlastní funkce pro jezdící světlo.



## DEKLARACE POLÍ

Deklarace polí lze provádět několika způsoby:

```
1  int myInts[6];
2  int myPins[] = {2, 4, 8, 3, 6};
3  int mySendVals[6] = {2, 4, -8, 3, 2};
4  char message[6] = "hello";
```

- ① **myInt[6]** – deklarace pole bez jeho inicializace.
- ② **myPins[]** – deklarace pole bez uvedení jeho konkrétní velikosti, kompilátor nejdříve spočítá prvky a pak pole vytvoří o odpovídající velikosti.
- ③ **mySendVals[6]** – deklarace pole s jeho přesným počtem prvků.
- ④ **message[6]** – u pole datového typu **char** musí být provedena jeho inicializace minimálně jedním prvkem.

## PŘÍSTUP K POLI

Prvky v poli jsou indexovány vždy od nuly. Tzn. první prvek v poli je na indexu 0. To znamená, že pole o deseti prvcích má poslední prvek s indexem 9.

```
1  // přístup k prvkům od nultého indexu
2  mySendVals[0] == 2; // index 0 tj. první prvek
3  mySendVals[1] == 4; // index 1 tj. druhý prvek
4  mySendVals[2] == -8; // atd.
5
6  // přidělení hodnoty pole
7  mySendVals[0] = 12; // na index 0 bude přidělena hodnota 12
8
9  // získání hodnoty z pole
10 x = mySendVals[4]; // do x se uloží hodnota z indexu 4 tj. 2
```

### OTÁZKY PRO VÁS

- Když se podíváte na otevřený programový kód, jak byste v něm využili pole?
- V čem byste spatřovali výhodu ve využití pole?



Pokud nedokážete odpovědět na uvedené otázky, vyzkoušejte funkčnost následujícího programu.

```
1  int pinArray[] = {5, 6, 7};
2
3  void setup() {
4      pinMode(pinArray[0], OUTPUT);    // dioda 1
5      pinMode(pinArray[1], OUTPUT);    // dioda 2
6      pinMode(pinArray[2], OUTPUT);    // dioda 3
7  }
8
9  void loop() {
10     changeLED (pinArray[0]);
11     changeLED (pinArray[1]);
12     changeLED (pinArray[2]);
13 }
14
15 void changeLED(int pin) {
16     digitalWrite(pin, HIGH);    // rozsvícení diody
17     delay(50);
18     digitalWrite(pin, LOW);    // zhasnutí diod
19 }
```

### ÚKOLY PRO VÁS

- A) Upravte otevřený program s vlastní funkcí tak, aby čísla pinů byla nahrazena odkazem na prvky pole.
- B) Změňte směr běžícího světla z opačné strany.
- C) Upravte pořadí prvků v poli tak, aby se diody rozsvěcovali od středu ke krajům.

