

PRACOVNÍ LIST – SEMAFOR

V TÉTO LEKCI SI SESTAVÍME MODEL SVĚTELNÉ KŘIŽOVATKY (SEMAFORU) A NAUČÍME SE JÍ OVLÁDAT. BUDEME POSTUPOVAT OD JEDNODUCHÉ K SLOŽITĚJŠÍ. SOUČASNĚ SI NA TOMTO PŘÍPADĚ VYSVĚTLÍME PRINCIP A POUŽITÍ PŘERUŠENÍ.

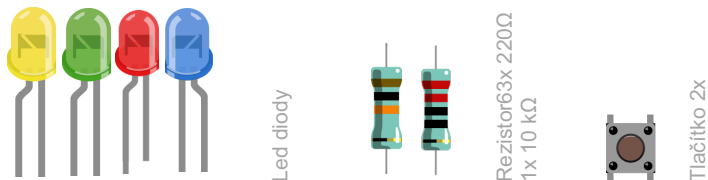
CO SE NAUČÍTE

- ① Princip semaforu.
- ② Jak fungují světelné křižovatky.
- ③ Co je to přerušení a jak jej použít.



CO BUDETE POTŘEBOVAT

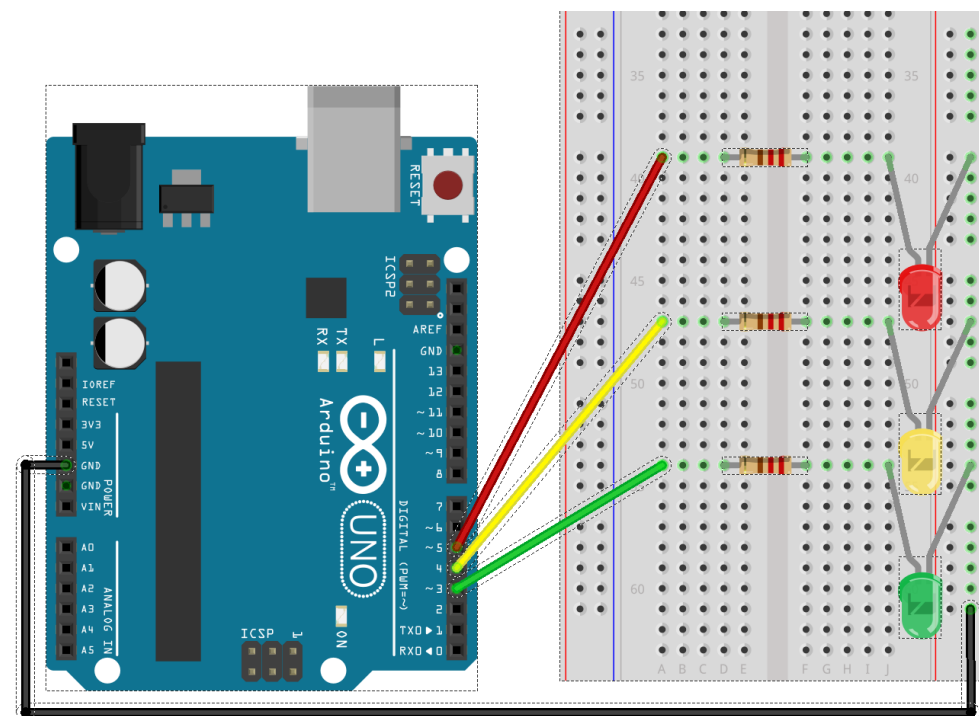
- ① LED diody (2 x červenou, 2 x zelenou, 1x žlutou, 1 x modrou).
- ② Tlačítko 2x.
- ③ Arduino.
- ④ Kontaktní pole.
- ⑤ Odpor 220 Ω (6x) a 10 k Ω (1x).
- ⑥ Vodiče typu zástrčka-zástrčka.



POUŽITÉ SOUČÁSTKY

A JDĚTE NA TO ...

- ① Podle schématu zapojte elektronický obvod.

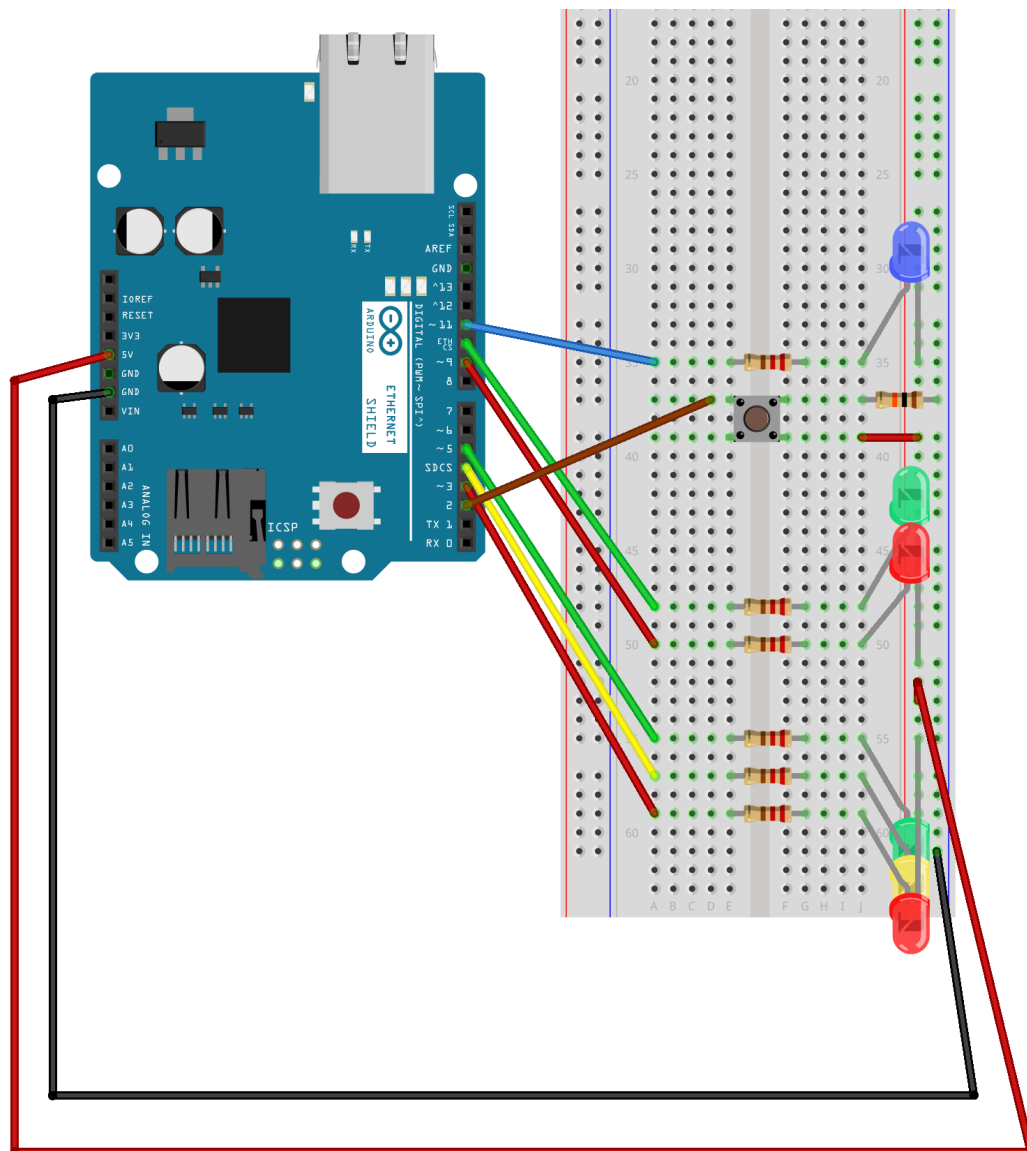


- ② Spustíte program Arduino IDE a napíšete následující programový kód.

```
1  int cervenal=3;
2  int oranzoal=4;
3  int zelenal=5;
4
5
6
7  void setup() {
8      pinMode(cervenal, OUTPUT);
9      pinMode(oranzoal, OUTPUT);
10     pinMode(zelenal, OUTPUT);
11
12 }
13
14
15 void loop() {
16     digitalWrite(cervenal,HIGH);
17     delay(1000);
18     digitalWrite(oranzoal,HIGH);
19     delay(1000);
20     digitalWrite(cervenal,LOW);
21     digitalWrite(oranzoal,LOW);
22     digitalWrite(zelenal,HIGH);
23     delay(2000);
24     digitalWrite(zelenal,LOW);
25     digitalWrite(oranzoal,HIGH);
26     delay(1000);
27     digitalWrite(oranzoal,LOW);
28     digitalWrite(cervenal,HIGH);
29     delay(1000);
30 }
31
32
33
34
```

Pokud vše funguje měli byste před sebou mít fungující semafor. Můžete experimentovat s dobou svícení jednotlivých světél. Až bude vše fungovat, jak si představujete, postupujte dál.

③ Nyní upravte a rozšiřte své zapojení dle následujícího schématu:



④ Do Arduina vložte následující kód:

```

1  int prepinac=2;
2  int tlacitko = 0;
3  int cervena1=3;
4  int oranzova1=4;
5  int zelena1=5;
6  int cervena3=9;
7  int zelena3=10;
8  int modra=11; //kontrolni dioda pro chodce
9
10 void setup() {
11     pinMode(prepinac, INPUT);
12     pinMode(cervena1, OUTPUT);
13     pinMode(oranzova1, OUTPUT);
14     pinMode(zelena1, OUTPUT);
15     pinMode(cervena3, OUTPUT);
16     pinMode(zelena3, OUTPUT);
17     pinMode(modra, OUTPUT);
18     digitalWrite(zelena1, HIGH);
19     digitalWrite(cervena3, HIGH);
20     attachInterrupt(digitalPinToInterrupt(prepinac),      zmena,
21     RISING);
22 }
23
24 void loop() {
25     delay(2000);
26     if (tlacitko)
27     {
28         digitalWrite(zelena1, LOW);
29         digitalWrite(oranzova1, HIGH);
30         delay(1000);
31         digitalWrite(oranzova1, LOW);
32         digitalWrite(cervena1, HIGH);
33         delay(500);
34         digitalWrite(zelena3, HIGH);
35         digitalWrite(cervena3, LOW);
36         digitalWrite(modra, LOW);
37         tlacitko=0;
38         delay(2000);
39         digitalWrite(zelena3, LOW);
40         digitalWrite(oranzova1, HIGH);
41         digitalWrite(cervena3, HIGH);
42         delay(1000);
43         digitalWrite(cervena1, LOW);
44         digitalWrite(oranzova1, LOW);
45         digitalWrite(zelena1, HIGH);
46     }
47 }
48

```

```

49 void zmena(){
50     tlacitko=1;
51     digitalWrite(modra, HIGH);
52 }

```

Úloha nyní simuluje přechod pro chodce vybavený tlačítkem pro rozsvícení zelené na přechodu.

VYSVĚTLENÍ

- Asi nejdůležitější (a nové) pro vás v tomto případě je přerušení a jeho obsluha.
- Přerušení se nastavuje pomocí funkce `attachInterrupt` v části `setup`.
- Samotná obsluha přerušení je ve funkci `zmena`. Všimněte si, že jediné, co tato funkce udělá, je že při stisku tlačítka změní hodnotu proměnné. Dle její hodnoty pak program pozná, zda tlačítko bylo od minulého průchodu stisknuté.



UPOZORNĚNÍ

- Pokud nemusíte, pak obvod na konci hodiny nerozpojujte a ponechejte si jej zapojený pro příští hodinu.



ÚKOLY PRO VÁS

- A) Přemýšlejte, jak by bylo možné naprogramovat tuto úlohu bez použití přerušení.
- Která možnost je jednodušší
- Zkuste vymyslet další případy, kde lze s úspěchem použít přerušení.

