

PRACOVNÍ LIST – SEMAFOR

V TÉTO LEKCI SI SESTAVÍME MODEL SVĚTELNÉ KŘÍŽOVATKY (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

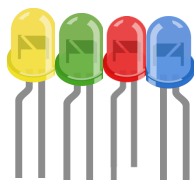
- a** Princip semaforu.
- b** Jak fungují světelné křižovatky.
- c** Co je to přerušení a jak jej použít.



CO BUDETE POTŘEBOVAT

- a** LED diody (2 x červenou, 2 x zelenou, 1x žlutou, 1 x modrou).
- b** Tlačítko
- c** Arduino.
- d** Kontaktní pole.
- e** Odpor 220 Ω (6x) a 10 k Ω (1x)
- f** Vodiče typu samec-samec.

Odpor 6x 220 Ω a 1x 10 k Ω).



LED diody
(4x červená, 4 x
zelená, 2x žlutá, 1x
modrá)



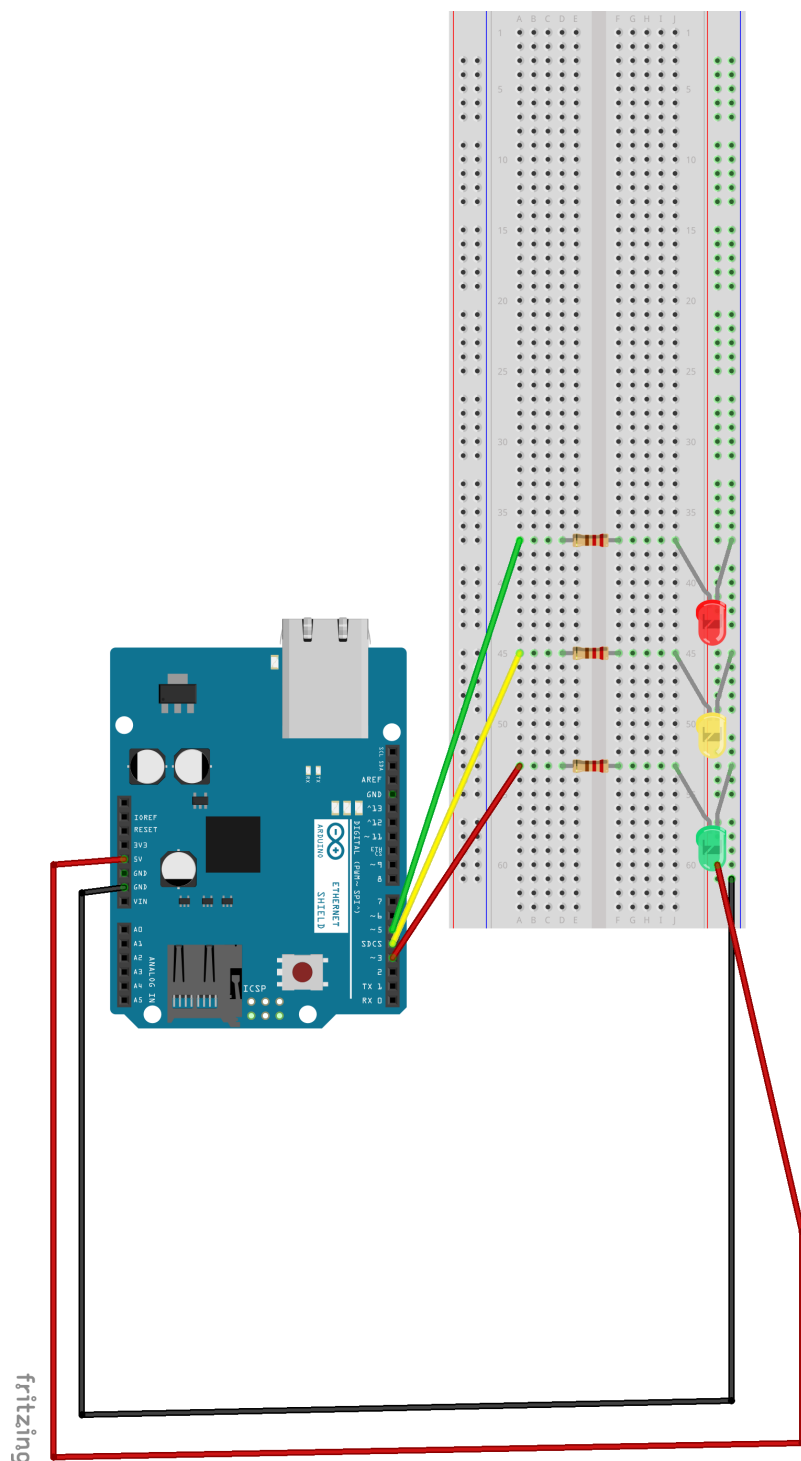
Rezistor 220 Ω 8x, 10
k Ω 1x



Tlačítko 2x

A JDĚTE NA TO ...

Podle schématu zapojte elektronický obvod.



Spustíte program Arduino IDE a napište následující programový kód.

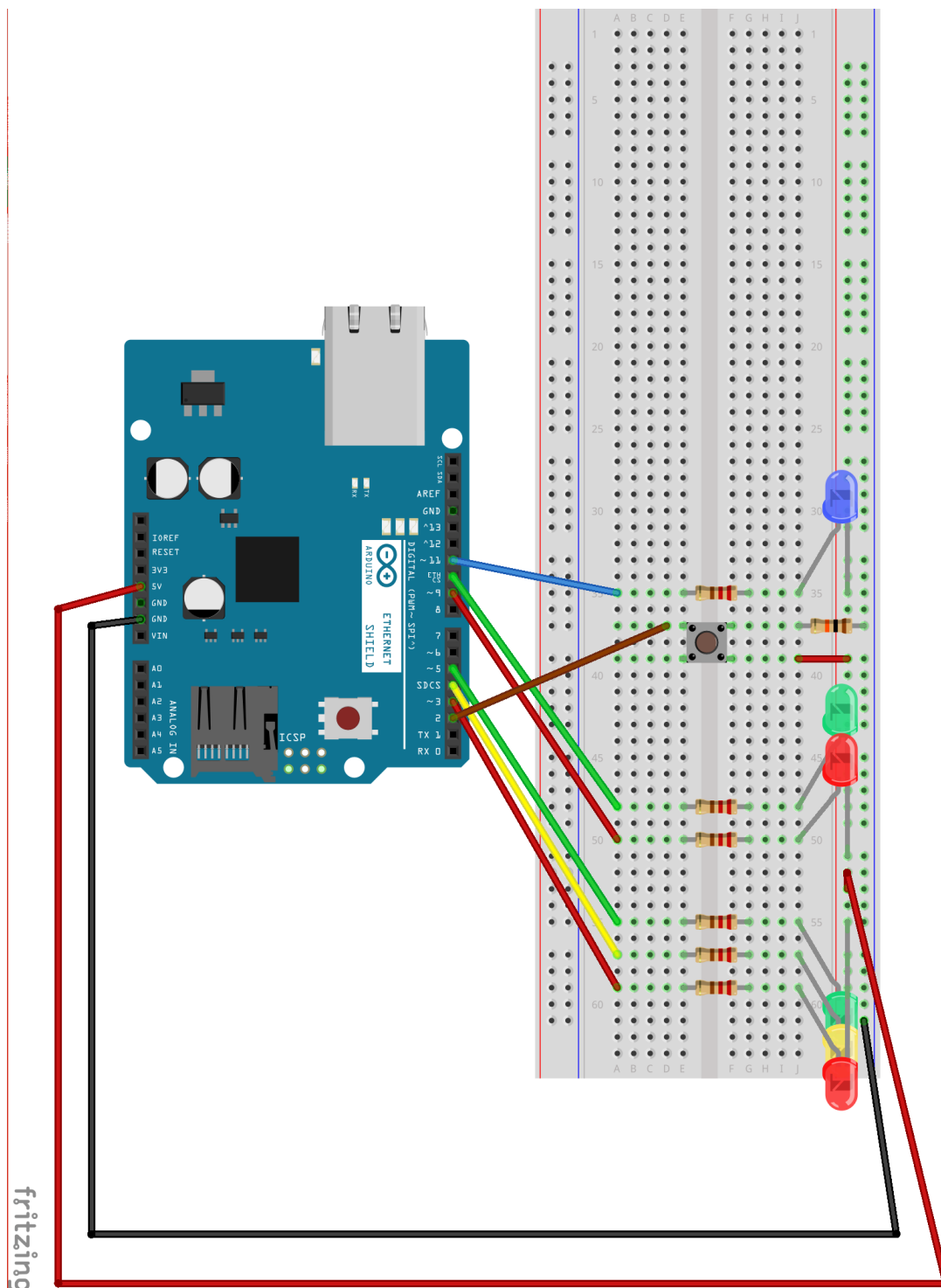
```
int cervenal=3;
int oranzoval=4;
int zelenal=5;

void setup() {
  pinMode(cervenal, OUTPUT);
  pinMode(oranzoval, OUTPUT);
  pinMode(zelenal, OUTPUT);
}

void loop() {
  digitalWrite(cervenal,HIGH);
  delay(1000);
  digitalWrite(oranzoval,HIGH);
  delay(1000);
  digitalWrite(cervenal,LOW);
  digitalWrite(oranzoval,LOW);
  digitalWrite(zelenal,HIGH);
  delay(2000);
  digitalWrite(zelenal,LOW);
  digitalWrite(oranzoval,HIGH);
  delay(1000);
  digitalWrite(oranzoval,LOW);
  digitalWrite(cervenal,HIGH);
  delay(1000);
}
```

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

Nyní upravte a rozšiřte své zapojení dle následujícího schématu:



Do Arduina vložte následující kód:

```
int prepinac=2;
int tlacitko = 0;
int cervena1=3;
int oranzova1=4;
int zelena1=5;
int cervena3=9;
int zelena3=10;
int modra=11; //kontrolni dioda pro chodce

void setup() {
  pinMode(prepinac, INPUT);
  pinMode(cervena1, OUTPUT);
  pinMode(oranzova1, OUTPUT);
  pinMode(zelena1, OUTPUT);
  pinMode(cervena3, OUTPUT);
  pinMode(zelena3, OUTPUT);
  pinMode(modra, OUTPUT);
  digitalWrite(zelena1, HIGH);
  digitalWrite(cervena3, HIGH);
  attachInterrupt(digitalPinToInterrupt(prepinac),      zmena,
  RISING);
}

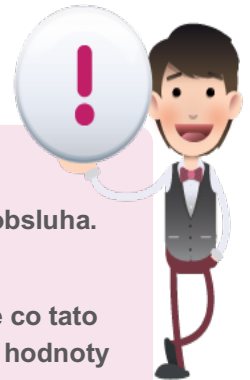
void loop() {
  delay(2000);
  if (tlacitko)
  {
    digitalWrite(zelena1, LOW);
    digitalWrite(oranzova1, HIGH);
    delay(1000);
    digitalWrite(oranzova1, LOW);
    digitalWrite(cervena1, HIGH);
    delay(500);
    digitalWrite(zelena3, HIGH);
    digitalWrite(cervena3, LOW);
    digitalWrite(modra,LOW);
    tlacitko=0;
    delay(2000);
    digitalWrite(zelena3, LOW);
    digitalWrite(oranzova1, HIGH);
    digitalWrite(cervena3, HIGH);
    delay(1000);
    digitalWrite(cervena1, LOW);
    digitalWrite(oranzova1, LOW);
    digitalWrite(zelena1, HIGH);
  }
}
```

```
void zmena(){
    tlacitko=1;
    digitalWrite(modra, HIGH);
}
```

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 jedinné co tato funkce udělá, je že při stisku tlačítka změní hodnotu proměnné. Dle její hodnoty pak program pozná, zda tlačítko bylo od minulého průchodu stisklé.



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.



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

