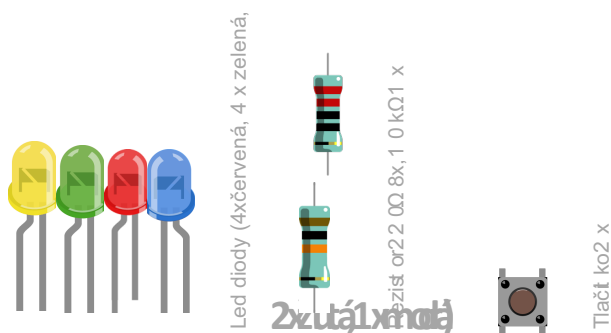


OVLÁDÁNÍ SVĚTELNÉ KŘIŽOVATKY POMOCÍ ARDUINA – SEMAFOR

VĚTŠINA Z VÁS DENNĚ PŘI CESTĚ DO ŠKOLY, NA NÁKUPY ATD. POTKÁVÁ SVĚTELNÉ KŘIŽOVATKY. JISTĚ ČASTO MÁTE POCIT, ŽE INTERVALY SVITU ČERVENÉ JSOU DLOUHÉ A NAOPAK INTERVALY SVITU ZELENÉ ABNORMÁLNĚ KRÁTKÉ. V TÉTO KAPITOLE SI SESTROJÍTE MODEL KŘIŽOVATKY, KTEROU BUDETE ŘÍDIT POMOCÍ ARDUINA.

CÍLE

- a** Sestavit složitější konstrukci z LED diod
- b** Pochopit princip přerušení a způsob jakým se k němu v Arduino přistupuje.
- c** Od jednoduché křižovatky postupovat ke složitější



POUŽITÉ SOUČÁSTKY

Čas: **90 min**

Úroveň: **□□□□□**

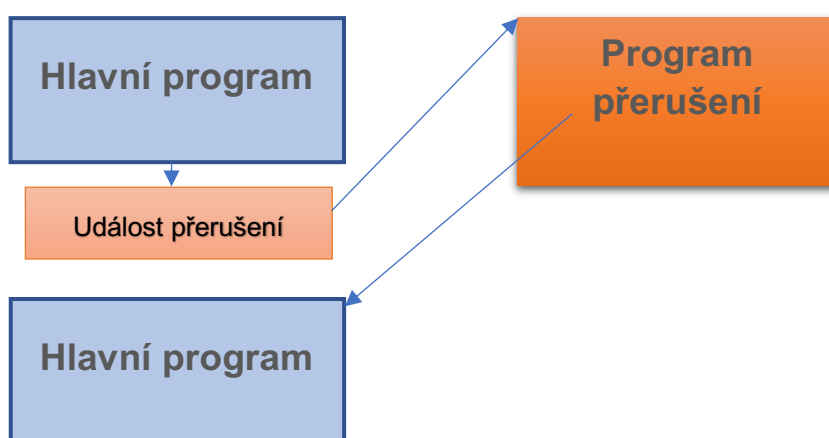
Vychází z: **1, 2**

PŘERUŠENÍ

Představme si, že je naše křižovatka umístěna na místě, kde se nepohybuje příliš chodců, ale nelze je zcela vyloučit. Je tedy zbytečné rozsvěcet zelenou pro chodce, pokud žádný chodec nechce křižovatku přejít. V praxi se tato situace řeší umístěním tlačítka, které si musí chodci stisknout, pokud chtějí přecházet. Program cyklicky ošetřuje jednotlivé stavy a v okamžiku, kdy dojde řada na chodce, tak si pouze ověří, zda od poslední bylo stisknuto tlačítko. Pokud ne, jedou auta z dalšího směru, pokud ano, naskočí nejprve zelená pro chodce.

Bylo by sice možné pravidelně kontrolovat stisk tlačítka – např. každou vteřinu, ale je to nepraktické a náročné na výpočetní prostředky. Navíc velmi krátký stisk (menší než 1 vteřina) nemusí být zaznamenán. Proto se tato situace řeší pomocí tzv. přerušení.

Přerušení si můžeme představit nezávisle běžící program, čekající na nějakou událost. Zde je to stisk tlačítka. Pokud tato situace nastane, pak hlavní program, přenechá výpočetní prostředky pro nezbytnou dobu programu pro obsluhu přerušení, ten vykoná, co potřebuje a opět vše vrátí hlavnímu programu.



V našem případě, se přerušení vyvolá stiskem tlačítka. Program přerušení, pak vykoná pouze to, že do **logické proměnné** uloží informaci o tom, že tlačítko bylo stisknuto (změní její hodnotu z nuly na jedničku). Pak se opět pokračuje vykonáváním hlavního programu.

V okamžiku, kdy se program rozhoduje, zda má nastavit zelenou pro chodce, se dotáže na hodnotu proměnné. Pokud je nula semafor pro chodce se nenastavuje a pokračuje se vykonáváním programu. Pokud je jedna, zapíná se zelená na semaforu pro chodce (a všude jinde je samozřejmě červená). Současně se hodnota proměnné nastaví na nulu.

Pro obsluhu přerušení je zde použit jako vstupní pin 2. Kromě něj je na Arduinou ještě možné pro přerušení použít pin 3. Na jiných pinech nelze přerušení nastavit. Je zde použit typ přerušení **RISING**. To znamená, že se sleduje stav tohoto pinu, jakmile na něj přijde signál, zavolá se přerušení. Arduino zná tyto typy přerušení:

Typ	Význam
LOW	Přerušení je voláno, pokud na daném pinu není signál (logická 0).
CHANGE	Přerušení je voláno, pokud na daném pinu dojde ke změně signálu (změna logické 0 na 1 nebo naopak).
RISING	Přerušení je voláno, pokud na daný pin přijde signál (změna z logické 0 na 1).
FALLING	Přerušení je voláno, pokud na daném pinu je signál ukončen (změna z logické 1 na 0).

Kód pro informaci o nastavení přerušení píšeme do části setup a vypadá takto:

```
attachInterrupt(digitalPinToInterrupt(prepinac), zmena, RISING);
```

Proměnná prepinac obsahuje číslo pinu, na kterém sledujeme přerušení. Funkce zmena je volána pro obsluhu přerušení, je umístěna na koci programu a vypadá takto:

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

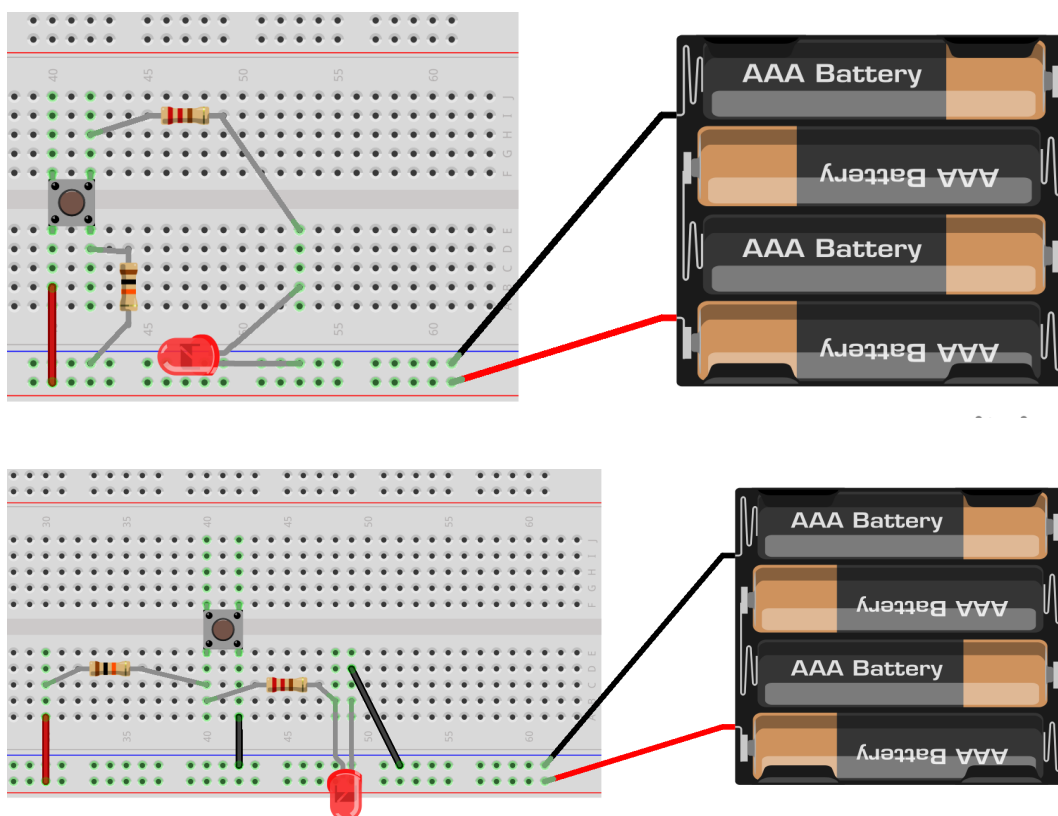
v našem příkladu, jako vedlejší efekt ještě rozsvítíme modrou diodu, která chodce upozorňuje, že jeho požadavek na umožnění přechodu byl přijat.

TLAČÍTKO

Pasivní (pro svou činnost nemusí být přímo napájené) zařízení, které nám umožňuje naprogramovat nějakou událost, řízenou uživatelem – sepnutí nebo uvolnění tlačítka.

Vaše stavebnice obsahuje dva druhy tlačítek – malé a velké (které ještě lze ozdobit barevnou „čepičkou“). Jejich zapojení je totožné, takže si můžete vybrat dle nálady.

Tlačítka můžeme zapojit tak, že jejich sepnutím buď proud začne nebo naopak přestane procházet. Pro pochopení principu si zkuste sestavit dva následující obvody (bez Arduina) a vyzkoušejte si je. Potřebujete dva odpory (220 a 10k), tlačítko a libovolnou diodu. Pro napájení můžete s úspěchem použít součástku YwRobot z vaší stavebnice. Pozor na správné zapojení diody (delší nožička na + a kratší na -). U prvního proud prochází po sepnutí tlačítka, u druhého to je naopak proud prochází a po sepnutí tlačítka přestane.



V našem příkladu použijeme první možnost a namísto diody vyvedeme vodič, který připojíme do Arduina na port 2 a budeme pomocí přerušení sledovat, zda je pod napětím (stisknuté tlačítko) či nikoliv (normální stav).

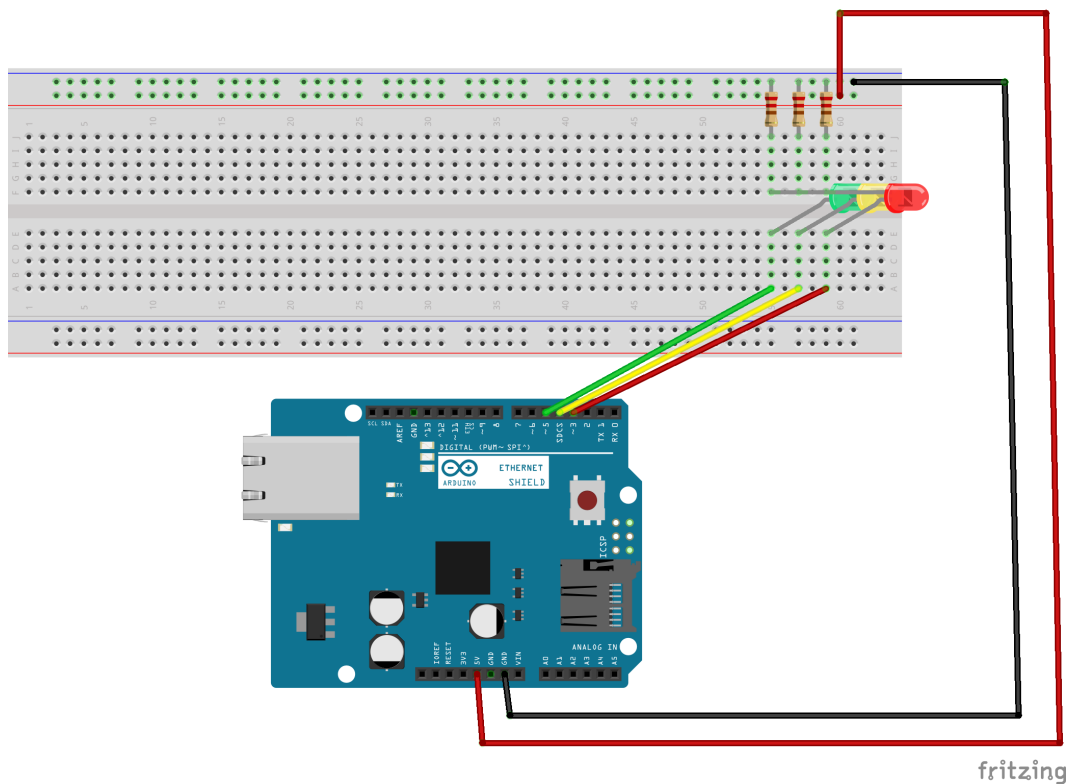
POZNÁMKA

Následující úkoly na sebe navazují, a proto po vyřešení prvního úkolu není nutné obvod rozpojovat, ale naopak postupně k němu budete přidávat další součástky. Stejně tak i program je koncipován od jednoduššího ke složitějšímu.

ÚKOL 1

Sestavte a naprogramujte model semaforu. V prvním úkolu se obejdeme bez přerušení.

ZAPOJENÍ OBVODU



Sestavte obvod dle následujícího obrázku:

PROGRAMOVÝ KÓD

```
1  int cervena1=3;
2  int oranzova1=4;
3  int zelena1=5;
4
5
6  void setup() {
7      pinMode(cervena1, OUTPUT);
8      pinMode(oranzova1, OUTPUT);
9      pinMode(zelena1, OUTPUT);
10 }
11
12 void loop() {
13     digitalWrite(cervena1,HIGH);
14     delay(1000);
15     digitalWrite(oranzova1,HIGH);
16     delay(1000);
17     digitalWrite(cervena1,LOW);
18     digitalWrite(oranzova1,LOW);
19     digitalWrite(zelena1,HIGH);
20     delay(2000);
21     digitalWrite(zelena1,LOW);
22     digitalWrite(oranzova1,HIGH);
23     delay(1000);
24     digitalWrite(oranzova1,LOW);
25     digitalWrite(cervena1,HIGH);
26     delay(1000);
27
28 }
```

a

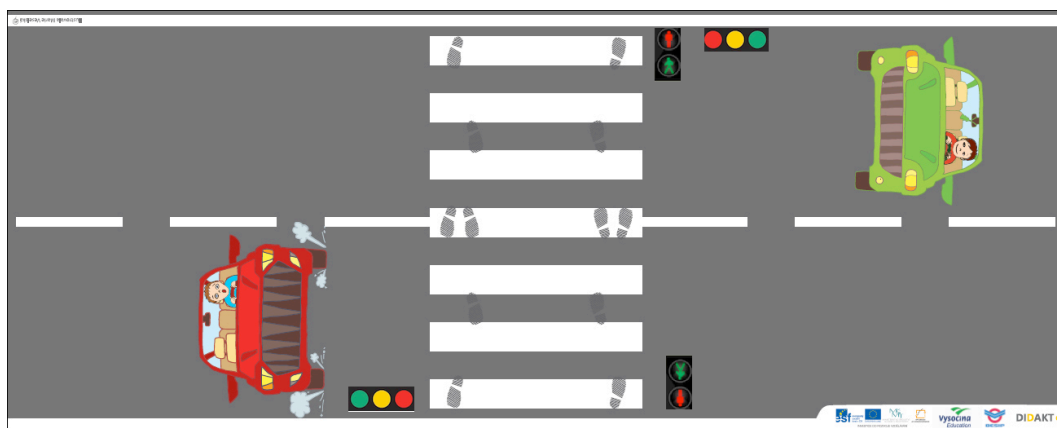
b

c

- a** Nastavení proměnných. Jsou použité piny 3 až 5, pin 2 bude později použit pro přerušení.
Proměnné jsou nazvány cervena1, oranzova1 a zelena1, protože v dalších zadáních přibudou další semaforey.
- b** Všechny piny budou výstupní.
- c** Rozsvěcení světel dle obvyklého cyklu pro semafor pro řízení automobilového provozu.

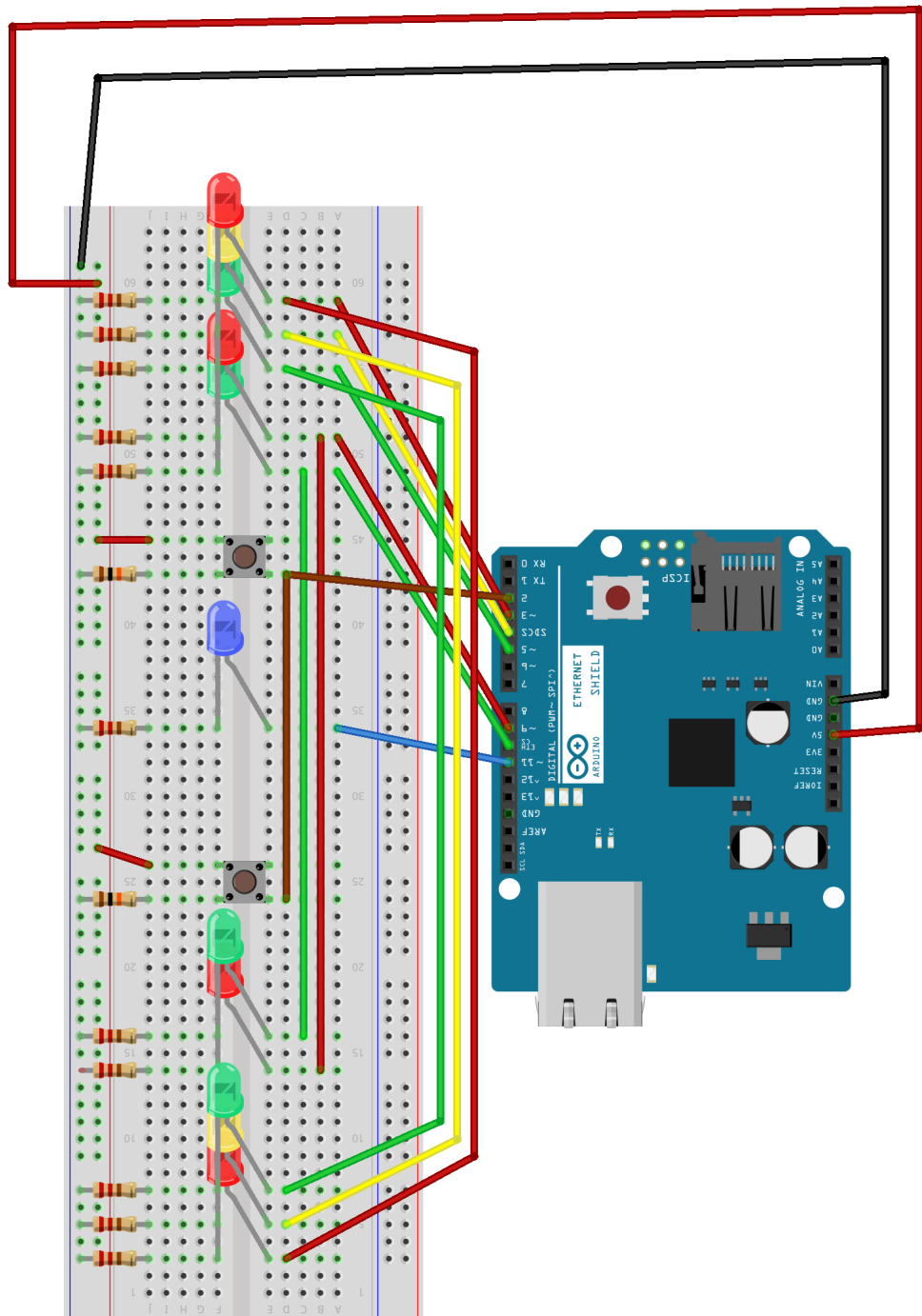
ÚKOL2

Sestrojte skupinu semaforů řídících přechod pro chodce. Z obou stran je semafor pro automobily a rovněž tak z obou stran ulice je semafor pro chodce a tlačítko, kterým chodci dávají na vědomí svůj požadavek na přecházení. V případě tohoto požadavku rozsvítíte kontrolní diodu, aby chodci věděli, že jejich požadavek byl přijat. Kontrolujte jedenkrát za dvě sekundy, zda tlačítko bylo stisknuto. Pro obsluhu tlačítka použijte přerušení. Viz obrázek.



ZAPOJENÍ OBVODU

Všimněte si, že jsou semafony propojené, neboť pro automobily z obou stran se musí nastejno rozsvěcet jednotlivé barvy a stejně tak i pro chodce. Rovněž si všimněte, že jsou vynechány tři piny na Arduinu (6-8), které budou použity v dalším úkolu.



PROGRAMOVÝ 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),
21 zmena, 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);
```

a

b

c

d

```

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 }
53

```

d

e

- a Nastavení proměnných. Tlačítko (tlačítka) bude připojeno na pin 2 (viz kapitola o **přerušení**). Proměnná tlacitko udává, zda byl stisknut přepínač (hodnota 1) nebo ne (hodnota 0). Defaultní je nula. Následuje nastavení proměnných určujících, na které piny budou připojeny jednotlivé LED semaforů. Piny 3 až 5 jsou první semafor, 9 a 10 semafor pro chodce. Na pinu 11 je pak připojena modrá dioda pro informaci pro chodce, že jejich požadavek byl přijat.
- b Určení, který z pinů bude vstupní (zde pouze pin 2) a které budou výstupní.
- c Informace, že pro pin 2 je nastaveno **přerušení**, o jaký typ přerušení se jedná a jaká funkce se bude zpracovávat při jeho vyvolání.
- d Základní část kódu. Program každé dvě vteřiny kontroluje, zda bylo stisknuto tlačítko. Pokud ano umožní chodcům přecházení. Stav se zjišťuje dle hodnoty tlačítka (1 = bylo stisknuté, 0 = nebylo stisknuté). Na konci této části se zhasne dioda a nastaví se hodnota tlačítka na nulu.
- e Obsluha přerušení. Nastavuje se hodnota proměnné tlacitko na jedna a současně se rozsvěcí modrá dioda.

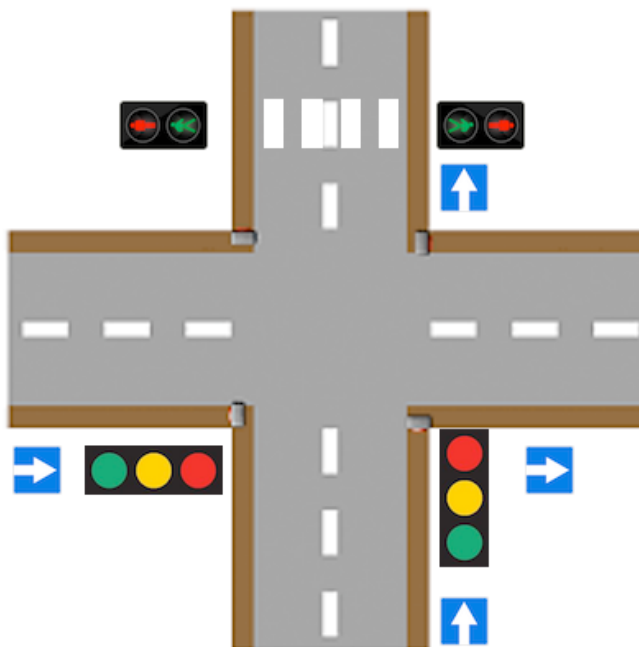


Bylo by možné tuto úlohu řešit bez použití přerušení?

Ano bylo by to poměrně jednoduše možné. Nejlepší řešení by bylo tak, že bychom zkrátili interval cyklu na cca. 0.2 sekundy a rovnou bychom kontrolovali, zda není stisknuté tlačítko. Museli bychom pravděpodobně přidat hlídání délky zelené pro automobily, aby nebyla příliš krátká. Také bychom obtížněji implementovali stavovou diodu.

ÚKOL 3

Sestrojte model křižovatky dvou jednosměrných cest s jedním přechodem pro chodce řízený pomocí semaforu. Viz obrázek.



Jedná se vlastně o zobecnění předchozího případu. Jediný rozdíl bude v tom, že oba směry budou řešit dva nezávislé semafore. Při složitější křižovatce, bychom potřebovali více výstupů, než má Arduino k dispozici. (14 digitálních) Na takovou křižovatku nám postačí deset pinů:

- Po třech na každý směr vozidel.
- Dva pro přechod pro chodce. Ty jsou pak rozvedeny do dvou semaforů.
- Jeden pro tlačítko pro přechod chodců. Správně bychom měli opět umístit dvě tlačítka na každou stranu jedno a výstup z nich spojit.
- Jeden pro diodu, která chodcům signalizuje, že byl přijat požadavek pro jejich přecházení.

Jedná se vlastně o systém se třemi následujícími stavy:

- **a** Jedou auta ve směru „zdola“.

- b** Jedou auta ve směru „zleva“.
- c** Přecházejí chodci, pokud bylo stisknuto tlačítko.

V rámci bezpečnosti při každém přechodu mezi stavy na určitou dobu (zde modelově 1s) musí svítit na všech semaforech červená.

Stavy u směrů automobilů a jejich časy jsou definovány dle následující tabulky:

červená	1s
Červená a oranžová	1s
Zelená	2s
Oranžová	1s

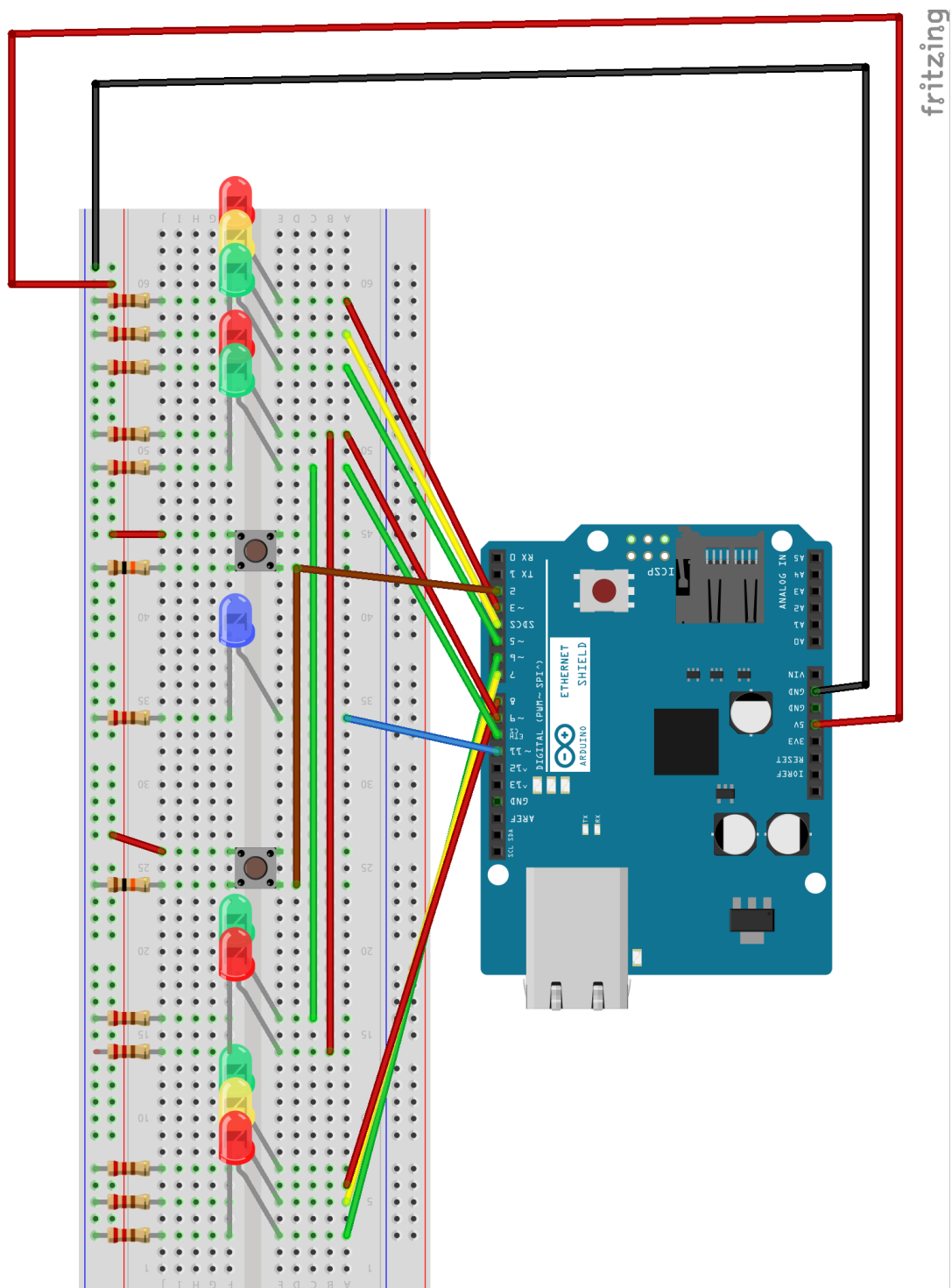
Stavy u přechodu pro chodce:

Červená	1s
Zelená	2s

Tento model křižovatky lze snadno převést na „klasickou“ křižovatku, kde auta jezdí z obou směrů a mají zelenou vždy oba směry proti sobě a chodci mohou přecházet po všech stranách. Postačí sestavit další semaforey z diod a rozvést k nim vodiče. Budete rovněž potřebovat celkem osm tlačítek a pokud chceme dát ke všem tlačítkům informační diodu, pak i osm diod. To je však již množství, na které vám nepostačí součástky z jedné stavebnice. Můžete však pracovat ve větší skupině a sestavit takovouto křižovatku.

Pro začátek ale zůstaneme pro jednoduchost u modelu se čtyřmi semaforey (dva pro chodce) a dvěma tlačítky.

ZAPOJENÍ OBVODU



3PROGRAMOVÝ KÓD

```
1  int prepinac=2;
2  int tlacitko=0;
3  int cervena1=3;
4  int oranzova1=4;
5  int zelena1=5;
6  int cervena2=6;
7  int oranzova2=7;
8  int zelena2=8;
9  int cervena3=9;
10 int zelena3=10;
11 int modra=11;
12
13 void setup() {
14     pinMode(prepinac, INPUT);
15     pinMode(cervena1, OUTPUT);
16     pinMode(oranzova1, OUTPUT);
17     pinMode(zelena1, OUTPUT);
18     pinMode(cervena2, OUTPUT);
19     pinMode(oranzova2, OUTPUT);
20     pinMode(zelena2, OUTPUT);
21     pinMode(cervena3, OUTPUT);
22     pinMode(zelena3, OUTPUT);
23     pinMode(modra, OUTPUT);
24     attachInterrupt(digitalPinToInterrupt(prepinac),
25                     zmena, RISING);
26 }
27
28 void loop() {
29     digitalWrite(cervena1,HIGH);
30     digitalWrite(cervena2,HIGH);
31     digitalWrite(cervena3,HIGH);
32     delay(1000);
33     digitalWrite(oranzova1,HIGH);
34     delay(1000);
35     digitalWrite(cervena1,LOW);
36     digitalWrite(oranzova1,LOW);
37     digitalWrite(zelena1,HIGH);
38     delay(2000);
39     digitalWrite(zelena1,LOW);
40     digitalWrite(oranzova1,HIGH);
41     delay(1000);
```

a

b

c

d

<pre> 42 digitalWrite(oranžova1,LOW); 43 digitalWrite(cervena1,HIGH); 44 delay(1000); 45 digitalWrite(oranžova2,HIGH); 46 delay(1000); 47 digitalWrite(cervena2,LOW); 48 digitalWrite(oranžova2,LOW); 49 digitalWrite(zelena2,HIGH); 50 delay(2000); 51 digitalWrite(zelena2,LOW); 52 digitalWrite(oranžova2,HIGH); 53 delay(1000); 54 digitalWrite(oranžova2,LOW); 55 digitalWrite(cervena2,HIGH); 56 delay(1000); 57 if (tlacitko) 58 { 59 tlacitko=0; 60 digitalWrite(zelena3,HIGH); 61 digitalWrite(cervena3,LOW); 62 digitalWrite(modra,LOW); 63 delay(2000); 64 digitalWrite(zelena3,LOW); 65 } 66 } 67 68 void zmena(){ 69 tlacitko=1; 70 digitalWrite(modra,HIGH); 71 } 72 </pre>	<div style="border-left: 1px solid black; height: 100px; position: relative;"> <div style="position: absolute; top: 0; right: 0; width: 20px; height: 20px; border: 1px solid black; background-color: white;"></div> <div style="position: absolute; top: 40%; right: 0; width: 20px; height: 20px; border: 1px solid black; background-color: white;"></div> <div style="position: absolute; top: 80%; right: 0; width: 20px; height: 20px; border: 1px solid black; background-color: white;"></div> </div> <div style="display: flex; justify-content: flex-end; align-items: center; margin-top: 10px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: white; margin-right: 5px;"></div> d </div> <div style="display: flex; justify-content: flex-end; align-items: center; margin-top: 10px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: white; margin-right: 5px;"></div> e </div> <div style="display: flex; justify-content: flex-end; align-items: center; margin-top: 10px;"> <div style="width: 20px; height: 20px; border: 1px solid black; background-color: white; margin-right: 5px;"></div> f </div>
---	---

- d** Nastavení proměnných. Tlačítko (tlačítka) bude připojeno na pin 2 (viz kapitola o **přerušení**). Proměnná tlacitko udává, zda byl stisknut přepínač (hodnota 1) nebo ne (hodnota 0). Defaultní je nula. Následuje nastavení proměnných určujících, na které piny budou připojeny jednotlivé LED semaforů. Piny 3 až 5 jsou první semafor, 5 až 8 druhý semafor, 9 a 10 semafor pro chodce. Na pinu 11 je pak připojena modrá dioda pro informaci pro chodce, že jejich požadavek byl přijat.
- e** Určení, který z pinů bude vstupní (zde pouze pin 2) a které budou výstupní.
- f** Informace, že pro pin 2 je nastaveno **přerušení**, o jaký typ přerušení se jedná a jaká funkce se bude zpracovávat při jeho vyvolání.

- g** Procházení té části kódu, která se vykoná vždy. Jedná se o oba směry silničního provozu. Časy jsou nastaveny dle tabulky X.
- h** Část kódu, která obsluhuje semafor pro chodce a je aktivována na základě hodnoty proměnné tlačítko. Pokud je její hodnota rovna 1, pak se tato část kódu provede, jinak nikoliv. Během provádění této části kódu se hodnota proměnné opět nuluje. Rovněž se zhasíná modrá dioda.
- i** Obsluha přerušení. Nastavuje se hodnota proměnné tlačítko na jedna a současně se rozsvěcí modrá dioda.



NESVÍTÍ DIODA

Zapojení diody – zkontrolujte jejich zapojení v kontaktní poli. Zkuste vyměnit diodu.

Zapojení v desce – zkontrolujte, zda jsou vývody zapojeny do odpovídajících pinů desky Arduino.

Rezistory – zkontrolujte, zda jste použili správný rezistory.

NEJDE NAHRÁT KÓD DO DESKY

USB kabel – ujistěte se, že máte desku Arduino připojenou k počítači.

Správný port – ujistěte se, že máte vybraný správný port pro připojení k desce Arduino pomocí USB kabelu.

NEFUNGUJE TLAČÍTKO

Zapojení tlačítka – ověřte, zda máte správně připojené piny dle schématu. Tlačítko má zespod čísla. Na číslo 1 se připojuje napájení, číslo 2 propojíte přes odpor 10k se zemí a z čísla 3 jde výstup.

Propojení tlačítek – zkuste odebrat propojení se sekundárním tlačítkem (to od kterého nejde vodič k Arduino) a zkuste nyní primární tlačítko.

JINÉ

Nesprávné pořadí rozsvěcení diod – ověřte pečlivě zapojení diod (dle schématu i zdrojového kódu). Ověřte, zda nemáte překlep ve zdrojovém kódu.



Bylo by možné tuto úlohu řešit bez použití přerušení?

Zde již by to bylo významně složitější. Museli bychom neustále kontrolovat stisk tlačítka a v případě stisku volat funkci pro obsluhu přerušení.

Zkuste si popsat nějakou vám známou křížovátku z vašeho okolí. Přemýšlejte, v jakých cyklech a jak se tam mění stavy a zkuste si ji nasimulovat.

ZÁVĚR

Na tomto příkladu jste si ukázali sestavení a naprogramování složitějšího obvodu z tlačítek a LED diod. Dále jste pochopili princip přerušení a naučili se jej používat.