2. Komunikace s PC a ladění pomocí sériové linky

Obsah

Sériová komunikace

Sériová komunikace se často využívá pro přenos dat mezi mikropočítačem a počítačem, ale i pro přenos dat mezi dvěma nebo více mikropočítači. Sériová komunikace se (na rozdíl od komunikace paralelní) vyznačuje sekvenčním přenosem jediného bitu za jednotku času.

Arduino a sériová komunikace

Aby mohlo Arduino komunikovat s počítačem přes USB port, je nutné aby se mezi PC a čipem nacházel **Serial-USB převodník**. Mezi různými typy Arduina nalezneme desky, které mají tento převodník napevno připájený k základní desce (př. Arduino Uno), dále ty které potřebují převodník externí (př. Arduino Mini) a nakonec ty, jejichž čip v sobě již má převodník zabudovaný, jako například u Esplory mikroprocesor **Atmega32u4**.

Každý přípravek Arduino obsahuje alespoň jeden sériový port (u Arduino Mega nalezneme dokonce rovnou čtyři), také známý jako **UART** (Universal Asynchronous Receiver/Transmitter). Tento Univerzální Asynchronní Přijímač/Vysílač pomocí dvou digitálních pinů RX a TX odesílá a přijímá data. Povšimněte si rozsvícení RX a TX LED na Arduinu Esplora vlevo nahoře po spuštění následujících příkladů.

Obrázek 1. Příklad externího převodníku

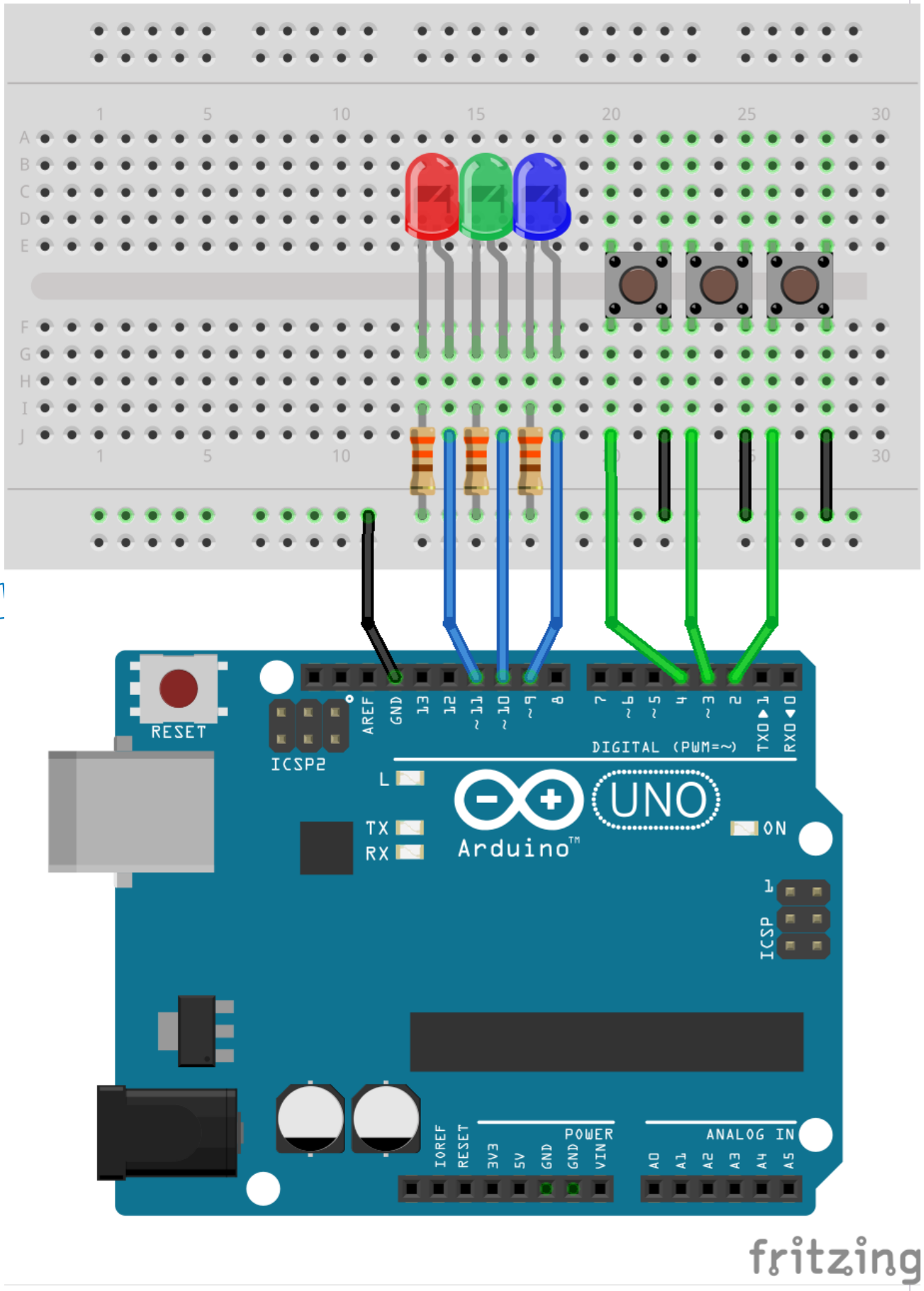
Projděte si funkce knihovny Serial na https://www.arduino.cc/en/Reference/Serial. Některé využijete v následujících příkladech.

Zapojení

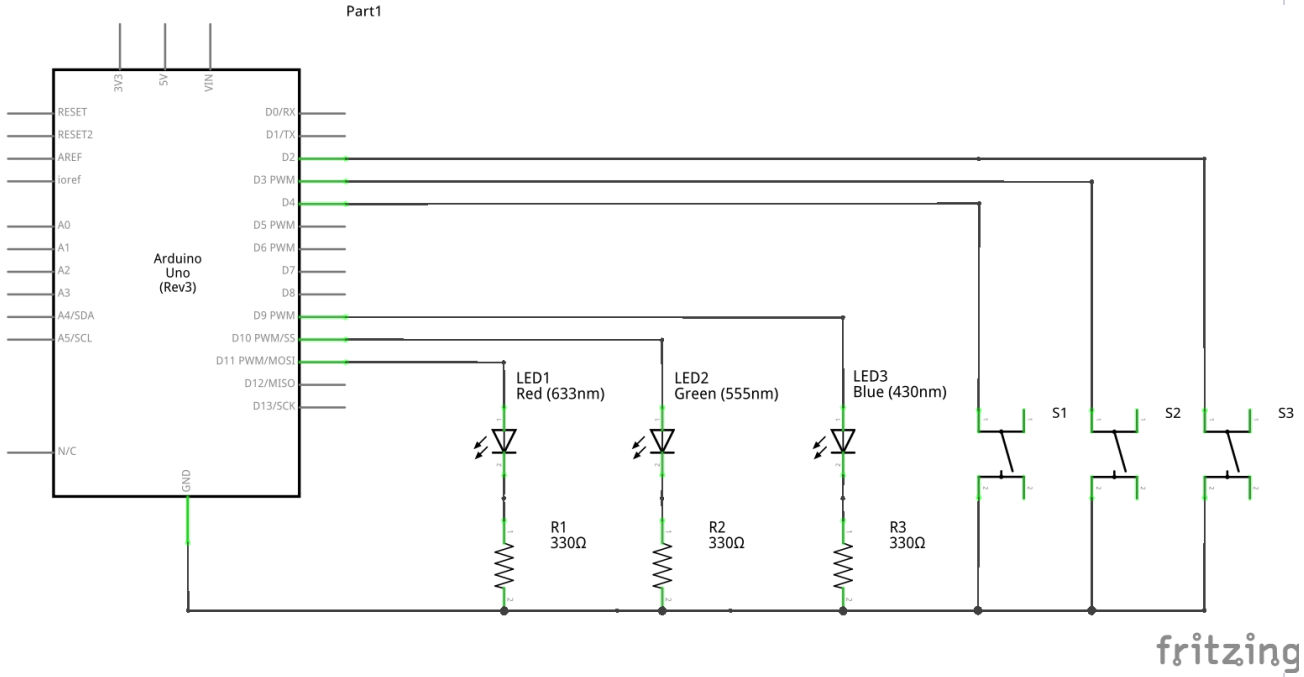
Pro dnešní cvičení budete potřebovat stejné zapojení jako na cvičení 1, s jedním rozdílem: použijete 3 tlačítka (`S1` až `S3`).

Tedy: 3 LED, 3 tlačítka, 3 rezistory

▼ Příklad zapojení



Obrázek 2. Zapojení součástek na breadboardu



Obrázek 3. Schéma zapojení

1. úloha - Výpis na sériový monitor (1b)

V první úloze si vyzkoušíte výpis na sériový monitor. Nejdříve na začátku programu zapněte sériovou komunikaci pomocí funkce `Serial.begin()` a její parametr nastavte na 9600 baudů. Baud rate je jednotka modulační rychlosti, která udává počet změn stavu média za jednu sekundu – zde tedy určujete rychlost přenosu 9600 bitů za sekundu.

Detekujte zmáčknutí tří tlačítek a v případě stisknutí rozsviťte LED (každé tlačítko rozsvítí jinou barvu) a název dané barvy vypište *jednou* na sériový monitor. V případě, že není žádné tlačítko zmáčknuté, ponechte LED vypnuté.

Sériový monitor spustíte jednoduše kliknutím na ikonu lupy v pravém horním rohu Arduino IDE.

2. úloha - Morseova abeceda (2b)

V prvním cvičení jste si vyzkoušeli pomocí LED zablikat písmena ARD v Morseově abecedě. Dnes na tuto úlohu navážeme a vaším cílem bude zablikat písmena, slova a věty poslaná z PC přes sériovou linku.

Nejdříve si stáhněte soubor morse_letters.h (../../media/tutorials/02/morse-letters.zip) , ve kterém najdete pole řetězců letters obsahující všechna písmena v Morseově kódu – použijete toto pole ve svém programu při překladu písmen v latince zaslaných z PC.

Doporučujeme použít funkci `flash(int duration)` pro zablikání jednoho znaku (tečky nebo čárky) z minulého cvičení a vytvořit novou funkci `flashSequence(char* sequence)` , pro zablikání jednoho celého písmene.

Nezapomeňte odlišovat délku mezer mezi jednotlivými písmeny a slovy.

Povšimněte si rozdílu funkcí `Serial.print()` a `Serial.println()`.

Pokud vám nefunguje include, můžete si tabulku písmen prostě okopírovat.

Ukázka 1. Tabulka písmen v Morseově abecedě

```
char* letters[]={
  ".-", "-...", "-.-.", "-..", ".", "..-.", "--.",  // A-G
  "...." , "..", ".---", "-.-", ".-..", "--", "-.", // H-N
  "---", ".--.", "--.-", ".-.", "...", "-", "..-",  // O-U
  "...-", ".--", "-..-", "-.--", "--.."             // V-Z
};
```

Stavový automat

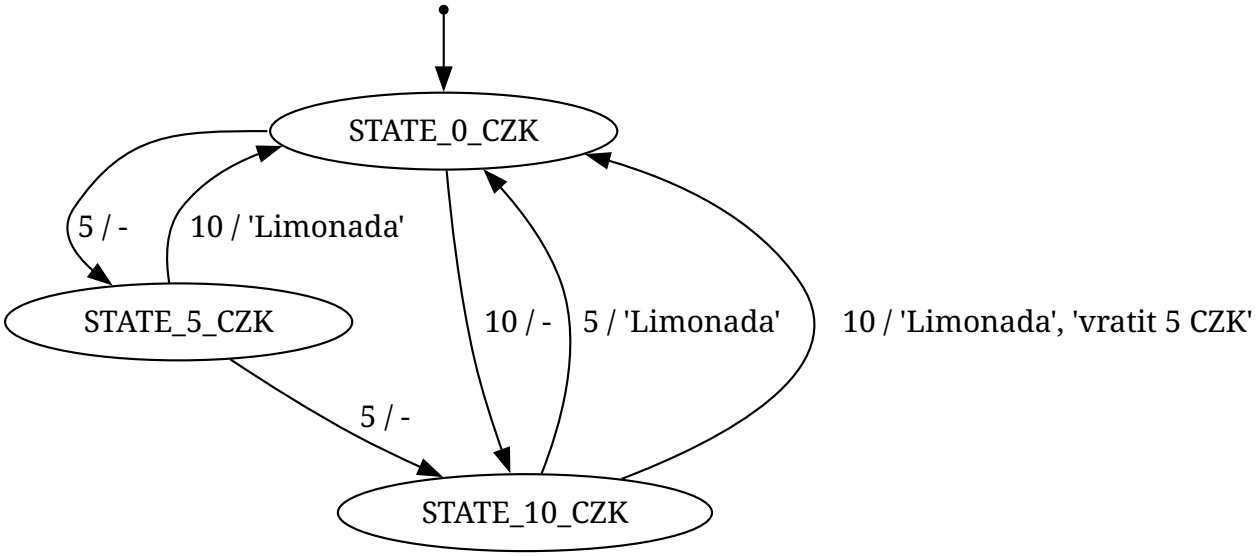
Jedná se o výpočetní model jednoduchého počítače, který se skládá z několika stavů mezi kterými přechází na základě symbolů, které čte ze vstupu. Přesnější definici se dozvíte v předmětu BI-SAP (3. přednáška – Sekvenční obvody (https://courses.fit.cvut.cz/BI-SAP/lectures/03/index.html)). Stavový automat popisujeme pomocí tzv. *grafu přechodů* (State Transition Graph).

Příklad návrhu a implementace stavového automatu

Nyní uvedeme krátký příklad návrhu a implementace jednoduchého **automatu na limonádu**. Do automatu můžete vhazovat pouze mince o hodnotě 5 a 10 Kč. Limonáda stojí 15 Kč, pokud vhodíte mince navíc, automat je vrátí.

- Nejprve navrhneme a graficky znázorníme řešení pomocí grafu přechodů. V tomto případě jednotlivé stavy znázorňují celkový počet korun, který jsme již do automatu vložili.
- Hrany neboli přechody automatu jsou označeny vstupem a výstupem. Vstup jsou zde vhozené peníze (př. 5 Kč) a výstup vydaná limonáda, případně i vrácené peníze navíc.
- **Vstup** je v grafu označen znakem vedle hrany **před** lomítkem, **výstup** naopak **za** lomítkem.

Pozn.: Jedná se o automat typu Mealy (podrobněji v předmětu BI-SAP).



Obrázek 4. Příklad stavového automatu „automat na limonádu“

Implementace automatu na limonádu v C:

```
enum states {
  STATE_0_CZK, STATE_5_CZK, STATE_10_CZK
};

enum states STATE, NEXT_STATE;

void setup() {

  // ...  Pocatecni inicializace
  STATE = STATE_0_CZK;
}

void loop() {

  // ...  Nacteni vstupu automatu (5 nebo 10 Kc)

  switch (STATE) {
    case STATE_0_CZK:
      if (vstup == 5) {
        NEXT_STATE = STATE_5_CZK;
      }
      else if (vstup == 10) {
        NEXT_STATE = STATE_10_CZK;
      }
      break;
```

```
    case STATE_5_CZK:

     if (vstup == 5) {
       NEXT_STATE = STATE_10_CZK;
     }
     else if (vstup == 10) {
       Vydej_limonadu();
       NEXT_STATE = STATE_0_CZK;
     }
     break;


   case STATE_10_CZK:

     if (vstup == 5) {
       Vydej_limonadu();
       NEXT_STATE = STATE_0_CZK;
     }
     else if (vstup == 10) {
       Vydej_limonadu();
       Vrat_penize();
       NEXT_STATE = STATE_0_CZK;
     }
   }

   STATE = NEXT_STATE;

 }
```

3. úloha - Výpis tlačítek na monitor (2b)

Naprogramujte stavový automat, který bude reagovat na stisknutí (resp. uvolnění) tlačítek `S1` a `S2` výpisem na sériový monitor a bliknutím LED. Doporučujeme si před začátkem implementace nakreslit graf přechodů.

Akce A

- **Akci A** proveďte po stisknutí *a uvolnění* tlačítka `S1`:
  - výpis „AKCE A“ na sériový monitor (pouze jednou)
  - 100 ms bliknutí červené LED

Akce B

- **Akci B** proveďte po stisknutí *a uvolnění* tlačítka `S2`:
  - výpis „AKCE B“ na sériový monitor (pouze jednou)
  - 100 ms bliknutí zelené LED

Akce C

Pokud došlo ke stisknutí (a uvolnění) obou tlačítek **S1 i S2**, proveďte **Akci C:**

- **Akci C** proveďte po stisknutí *a uvolnění **obou*** tlačítek `S1` a `S2`:
  - výpis „AKCE C“ na sériový monitor (pouze jednou)
  - 100 ms bliknutí modré LED

- Každou akci proveďte důsledně až *po uvolnění* tlačítka (ne již během stisknutí).
- Pokud zmáčknete `S1` i `S2` a poté jeden pustíte, měli byste zůstat ve stavu `STATE_SW1SW2`.

Šablona programu

Použijte tuto šablonu a do ní pouze doplňte chybějící kód.

```
enum states {
  STATE_START, STATE_SW1, STATE_SW2, STATE_SW1SW2
};

enum states STATE, NEXT_STATE;

void setup() {
  // ... todo
}


void loop() {

    switch (STATE)
    {
      case STATE_START:
          if (sw1_pressed == true) {
            NEXT_STATE = STATE_SW1;

          }
          else if (sw2_pressed == true) {
            NEXT_STATE = STATE_SW2;
          }

          // ... todo
          break;

      case STATE_SW1:
         if (sw1_pressed == false) {
           // ... todo
         }
         // ... todo
         break;
```

```
      case STATE_SW2:
         if (sw2_pressed == false) {
           // ... todo
         }
         // ... todo
         break;

      case STATE_SW1SW2:
         // ... todo

   }

   STATE = NEXT_STATE;
}
```