

1 Arduino – sériová linka

Velmi užitečnou vlastností Arduina je pohodlná práce se sériovou linkou. Pomocí té se dá snadno vypisovat aktuální stav programu a tím odhalovat chyby. Přes sériovou linku lze také snadno zasílat do mikrokontroléru příkazy a tím měnit chod programu. Prostřednictvím sériové linky je také do mikrokontroléru nahráván program.

Po připojení arduino desky k počítači přes USB a po nainstalování ovladačů se vytvoří virtuální sériový port. Data zaslaná po tomto portu putují přes USB do arduino desky. Pro zajištění převodu dat z USB na sériovou linku (RS232 nebo také UART) se používal čip FT232RL (FTDI), pro který jsou již Windows schopné sami nalézt ovladač. Avšak dnes Arduino nahradilo tento specializovaný čip levnějším obecným mikrokontrolérem Atmega8U2 obsahujícím podporu USB. V tomto mikrokontroléru je od výrobce Arduino desky nahrán program pro emulaci sériové linky přes USB. Pro tuto variantu je však potřeba nahrát do Windows ovladač dodávaný od Arduina.

1.1 Metoda Serial

1.1.1 Syntax

`Serial.begin(speed)`

`Serial.begin(speed, config)`

Komunikace přes UART probíhá po dvou vodičích, první z nich je pro příjem dat (Rx) a druhý pro vysílání (Tx). Nejsou zasílány hodinové impulsy pro synchronizaci, je to tedy asynchronní přenos. Z toho důvodu musí být vysílací i přijímací zařízení nastaveno na stejnou přenosovou rychlost, tzv. „baud rate“. Typicky používané rychlosti jsou 9600bd, 19200bd a 115200bd definováno je však i mnoho dalších. Rychlost na vysílacím a přijímacím zařízení by se neměla lišit o více než 5%, pak by již mohlo začít docházet k chybně interpretovaným datům na přijímací straně. Komunikace je vždy zahájena a synchronizována „Start bitem“. To je první přechod z logické jedničky do nuly. Následuje 5 až 9 datových bitů, případný paritní bit a stop bit. Stop bit je vždy v logické jedničce a jeho délka může být 1bd, 1,5bd nebo 2bd. Tyto parametry musí být shodně nastaveny na obou komunikujících zařízeních.


Typicky se nastavuje rychlost 9600bd, 8 datových bitů, bez parity a jeden stop bit.

	Start bit	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	parita	Stop bit
--	--------------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	--------	-------------

Na desce Arduino UNO jsou piny pro sériovou linku umístěny na digitálních pinech 0 a 1. Z toho důvodů má deska pro připojení přípravků propojky umožňující připojit piny 0 a 1 konektoru D buď na piny 0 (Rx) a 1 (Tx) nebo na piny 16 a 17. Pokud by se k portu D připojil přípravek impedančně zatěžující piny Rx a Tx nemuselo by být možné mikrokontrolér naprogramovat.

Pro inicializaci sériové linky se v proceduře `void setup()` volá metoda `begin(int bd)` z instance třídy `HardwareSerial Serial` inicializuje sériovou linku pro přenosovou rychlost 9600 baudů `Serial.begin(9600)`. Pro odeslání dat po sériové lince disponuje Arduino řadou metod, které bývají přetíženy pro různé datové typy vstupních parametrů. Pro zobrazení na konzoli je vhodná metoda `Serial.println()`, pro odeslání binárních dat `Serial.write()`. Pro přijetí textového řetězce se hodí

metoda `Serial.readString()` a pro kontrolu, zda jsou nějaká data pro příjem, slouží `Serial.available()`.

Jednoduchou ukázkou použití sériové linky v Arduino najdete v „Soubor“ -> „Příklady“ -> „01.Basics“ -> „DigitalReadSerial“. Pokud na konektor D připojíte přípravek M8DIPSW bude po nahrání programu odesílat mikrokontrolér každou milisekundu stav třetího přepínače (pin 2). Okno s výstupem sériové linky si můžete otevřít v prostředí Arduino tlačítkem .

1.1.2 Odeslání dat – `Serial.print()` a `Serial.println()`

Sériová komunikace využívá piny TX / RX používá logické úrovně TTL (5V nebo 3.3V v závislosti na desce). Nespojíte tyto piny přímo k sériovému portu RS232 ten pracuje na +/- 12V a může poškodit vaši Arduino desku. Pro přímou komunikaci přes sériovou linku je třeba použít převodník z TTL na RS232. Tyto funkce slouží k odeslání hodnoty z Arduino do PC, nebo jiného zařízení. Liší se od sebe tím, že funkce `Serial.print()` posílá data stále na jednom řádku. Funkce `Serial.println()` vždy na konci odeslaných dat přidá znak pro zalomení řádku. Rozdíl mezi těmito dvěma funkcemi můžete vidět na následujících příkladech.

1.2 Funkce `random()`

Tato funkce je určena pro generování pseudonáhodného čísla.

1.2.1 Syntax

`random(max)`

`random(min, max)`

Parameters

min - dolní limit náhodné hodnoty, včetně (volitelně)

max - horní mez náhodné hodnoty, nevčetně

2 Vzorové příklady

2.1 Přečtení hodnoty z M8DIPSW a odeslání na terminál sériové linky

Připojte přípravek M8DIPSW na port D a pomocí přepínačů nastavujte různé hodnoty. Zkontrolujte, zda jsou na konektorové desce propojky pinů 0 a 1 nastaveny na piny 16 a 17 mikrokontroléru. Nastavená hodnota je odeslána na terminál sériové linky každou 1 sekundu. Soubor je uložen na LMS MOODLE jako M8DIPSW_SERIAL.ino

```
//přípravek M8DIPSW na portu D, piny 0,1 připojeny na piny 16,17 mikrokontroleru

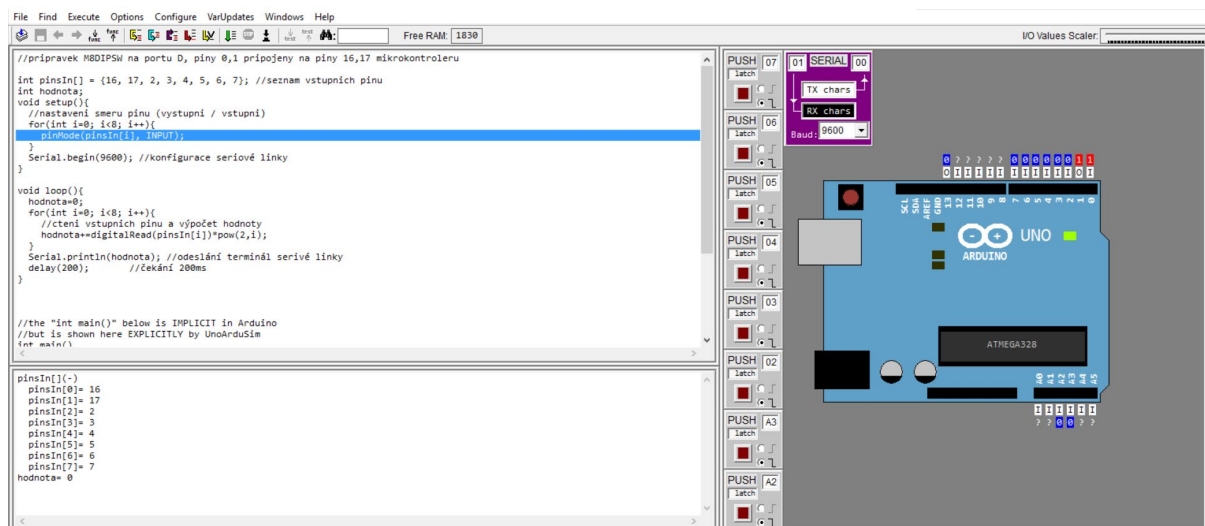
int pinsIn[] = {16, 17, 2, 3, 4, 5, 6, 7}; //seznam vstupních pinů
int hodnota;
void setup(){
    //nastavení smeru pinu (vstupní)
    for(int i=0; i<8; i++){
        pinMode(pinsIn[i], INPUT);
    }
    Serial.begin(9600); //konfigurace sériové linky
}

void loop(){
    hodnota=0;
    for(int i=0; i<8; i++){
        //čtení vstupních pinů a výpočet hodnoty
        hodnota+=digitalRead(pinsIn[i])*pow(2,i);
    }
    Serial.println(hodnota); //odeslání terminál sériové linky
    delay(1000);             //čekání 1s
}
```

Popis vzorového programu, nejprve si definujeme pořadí vstupních pinů. Definice pinů byla již probrána v prvním cvičení a je dána HW převodníkem pro připojení periférií. Následuje definice proměnné hodnota. Metoda setup pro nastavení vstupních pinů arduina a konfigurace sériové linky. Hlavní blok programu, metoda loop, jak již bylo popsáno v prvním cvičení, se automaticky opakuje. Nejprve dojde k vynulování proměnné hodnota a pak k vypočítání hodnoty, jež je určena postupným přečtením jednotlivých vstupních pinů vynásobených vahou daného pinu (poslední pin má váhu 128) tím dostaneme celkovou hodnotu vyjádřenou jednotlivými piny na vstupu (číslo od 0 do 255). Následně je hodnota odeslána na sériovou linku a odesláno odřádkování. Než se děj začne opakovat, počkáme 1s. Po nahrání programu do mikroprocesoru otevřete okno terminálu sériové linky a na modulu přepínačů M8DIPSW připojeného k portu D měňte hodnoty a sledujte je na terminálu.

2.1.1 Přečtení hodnoty z M8DIPSW a odeslání na terminál sériové linky pomocí UnoArduSim

Po spuštění aplikace UnoArduSim je třeba načíst zdrojový kód aplikace, to uděláte pomocí menu „File“ a volby „Load INO or PDE Prog.“ a načtete soubor M8DIPSW_SERIAL.ino. Po načtení programu je ještě třeba nahrát HW konfiguraci. To můžete udělat buď ručně pomocí menu „Configure“ a volby „I/O devices“ nebo následným načtením přes tlačítko „Load“ je uložena v souboru M8DIPSW_SERIAL_HW.txt. Okno terminálu otevřete dvojklikem na fialovém čtverečku SERIAL nebo pomocí menu Windows a volbou seriál monitor. Aby tlačítka držela příslušnou hodnotu, je třeba aktivovat na každém tlačítku závoru „latch“ tím je tlačítko změněno na přepínač. Program spustíte zelenou šipkou vedle značky stop, která naopak program zastavuje.



Obr. 1 Přečtení hodnoty z M8DIPSW a odeslání na terminál sériové linky pomocí UnoArduSim

2.2 Přijetí hodnoty z terminálu sériové linky a vygenerování matice čísel

Pro tuto aplikaci není potřeba periférie. Pomocí terminálu sériové linky odešlete číselnou hodnotu maximálně do 10 a na základě přijaté hodnoty bude vygenerována čtvercová matice čísel daného rozměru zpět na terminál.

```
/*Program přijme vstupní číslo z terminálu sériové linky
a na základě tohoto čísla vygeneruje matici náhodných čísel
*/

unsigned int vstup; //vstupní číslo
unsigned int x,y;
void setup()
{
    Serial.begin(9600);
}

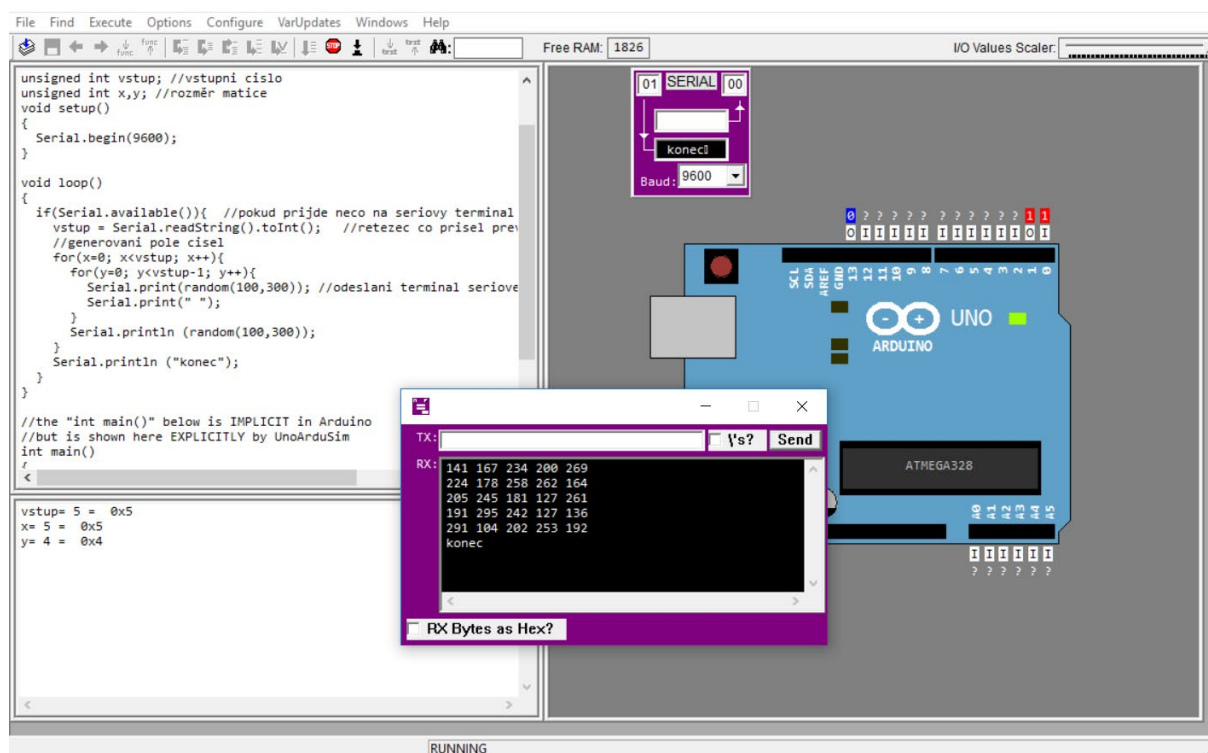
void loop()
{
    if(Serial.available()){ //pokud přijde něco na sériový terminál
        vstup = Serial.readString().toInt(); //retezec co přišel převed na Int
        //generování pole čísel
        for(x=0; x<vstup; x++){
            for(y=0; y<vstup-1; y++){
                Serial.print(random(100,300)); //odeslání terminálu sériové linky
                Serial.print(" ");
            }
            Serial.println (random(100,300));
        }
        Serial.println ("konec");
    }
}
```

Popis vzorového programu, nejprve si nadefinujeme proměnné vstup pro hodnotu, která přijde z terminálu sériové linky. Proměnné x a y jsou proměnné určující rozměr matice a pro budoucí programy budou sloužit jako index do pole. Metoda „setup ()“ obsahuje pouze definici sériové linky a její rychlost. Hlavní blok programu je zjišťuje, zda došlo k příjmu nějakého řetězce, a pokud ano pak jej načte a převede na číslo (pokud by nebylo přijato číslo, tak jako by byla poslána 0). Následuje využití for cyklu ve for cyklu a vypsaní náhodných hodnot zpět na terminál sériové linky po každém řádku dojde k odřádkování. Toho se udělá tak, že poslední hodnota se již nevypisuje uvnitř, ale po ukončení

cyklu, jež je zkrácen o 1, aby bylo možné odeslat ukončení řádku. Po ukončení vnějšího cyklu for je ještě odeslán text konec.

2.2.1 Přijetí hodnoty z terminálu sériové linky a vygenerování matice čísel pomocí UnoArduSim

Po spuštění aplikace UnoArduSim je třeba načíst zdrojový kód aplikace, to uděláte pomocí menu „File“ a volby „Load INO or PDE Prog..“ a načtete soubor SERIAL.ino. Po načtení programu je ještě třeba nahrát HW konfiguraci. To můžete udělat buď ručně pomocí menu „Configure“ a volby „I/O devices“ nebo následným načtením přes tlačítko „Load“ je uložena v souboru SERIAL_HW.txt. Okno terminálu otevřete dvojklikem na fialovém čtverečku SERIAL nebo pomocí menu Windows a volbou seriál monitor.



Obr. 2 Přijetí hodnoty z terminálu sériové linky a vygenerování matice čísel pomocí UnoArduSim

Program spustíte zelenou šipkou vedle značky stop, která naopak program zastavuje. V okně terminálu sériové linky (okno otevřete např. dvojklikem) zapište hodnotu a odešlete tlačítkem „Send“ a následuje výpis programu zpět na sériovou linku.

3 Cvičení 2 pro Arduino UNO

Zadané úkoly jsou navrženy tak, aby bylo možné je otestovat v programu UnoArduSim.

Odevzdávejte jednotlivé úkoly do samostatného adresáře. A do každého z adresářů nahrajte 4 soubory. Zdrojový kód (*.ino) konfiguraci MCU (myArduPrefs.txt) a konfiguraci periférií (myIODevs.txt) a výstřížek terminálu aby bylo jasné, že aplikace pracuje dle zadání.

3.1 Úkol 1 – vygenerujte čtvercovou matici čísel

Pomocí terminálu pošlete do Arduino Uno číslo, které bude určovat rozměr čtvercové matice minimálně 3 a maximálně 10 (aplikaci udělejte tak aby byla provedena kontrola rozsahu vstupního čísla. Vaším úkolem je vygenerovat matici maximálně 10 x 10 a naplnit ji čísly od 0 do 199 a vypsát ji tak, aby čísla byla zarovnaná. A pod výpis matice ještě určíte vlastnosti vygenerovaných čísel. Jaké číslo je nejmenší a jaké je největší.

Výpis na terminálu sériové linky by mohl vypadat například takto:

```
Zadejte cislo (max 10):3
 3 11 65
35  5  9
88 35 12

Min:  3
Max: 88

Zadejte cislo (max 10):
```

3.2 Úkol 2 – rozklad na prvočinitele

Pomocí terminálu pošlete do Arduino Uno číslo a toto číslo rozložte na prvočinitele. Vstupní číslo bude maximálně do 9999.

Příklad zobrazení v terminálu:

```
Zadejte cislo (max 9999): 280
280=2*2*2*5*7

Zadejte cislo (max 9999): 47
47 je prvocislo

Zadejte cislo (max 9999):
```

Například zadáte číslo 280 a program již provede příslušný rozklad. Pak zadáte například číslo 47 a program vám odpoví, že se jedná o prvočíslo.