

1 Arduino - displej

Pro výpis textu na textovém LCD displeji poskytuje Arduino třídu LiquidCrystal dovolující komunikovat s každým displejem kompatibilním s Hitachi HD44780, což podporuje značné množství textových LCD displejů. Po zapnutí je potřeba provést inicializaci displeje, nastavit přenosový formát, typ kurzoru, a další, o to se postará třída LiquidCrystal. Komunikace může být buď čtyřbitová, nebo osmibitová. Z důvodu úspory pinů je často volena čtyřbitová varianta. Dále se používá pin RS pro rozlišení, zda je odesílán příkaz nebo znak, pin Enable pro synchronizaci přenosu a případně RW pro rozlišení zda se má číst či zapisovat. Pin RW se často nepoužívá a je tažen k nule, což udává směr zápisu do displeje.

Po používání třídy LiquidCrystal je potřeba do kódu vložit její hlavičkový soubor a vytvořit instanci této třídy. Při vytváření instance třídy LiquidCrystal se přes parametry konstruktoru předávají čísla pinů, na kterých jsou připojeny jednotlivé signály displeje. Konstruktore je přetížen pro několik variant konfigurace. Minimálně je potřeba zadat číslo pinu RW, Enable a horních čtyř datových bitů.

Přípravek s LCD displejem se jmenuje MLCDGEN a je upraven tak, aby jej bylo možno připojit ke konektoru D. Arduino v sobě obsahuje několik ukázek použití displeje v sekci „File“ -> „Examples“ -> „LiquidCrystal“. Pro jejich správnou funkčnost s přípravkem MLCDGEN na portu D je potřeba upravit čísla pinů v konstrukturu následujícím způsobem:

```
//          RS, E, DB4, DB5, DB6, DB7
LiquidCrystal lcd(7, 6, 2, 3, 4, 5);
```

2 Vzorové příklady

2.1 Zobrazení textu ze sériové linky na displeji (reálný displej)

Připojte přípravek MLCDGEN ke konektoru D. Do mikrokontroléru nahrajte následující kód.

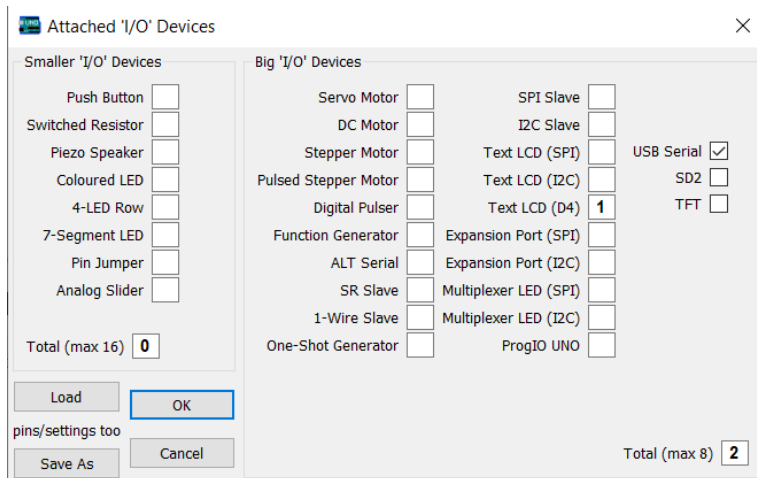
```
// include the library code:
#include <LiquidCrystal.h>
// initialize the library with the numbers of the interface pins
//          RS, E, DB4, DB5, DB6, DB7
LiquidCrystal lcd(7, 6, 2, 3, 4, 5);

void setup() {
  Serial.begin(9600); //inicializace sériové linky
  lcd.begin(8, 2);    //inicializace displeje (8 sloupců, 2 řádky)
  lcd.print("Hi there"); //výpis textu na první řádek
  lcd.setCursor(0, 1);   //přechod na druhý řádek
  lcd.print("Cheers!");  //výpis textu na druhý řádek
  lcd.setCursor(0, 0);
}

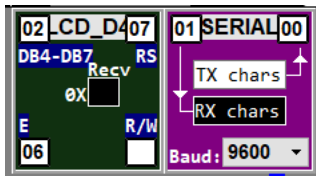
void loop() {
  if(Serial.available()){ //kontrola zda je něco odesláno
    const int MAX_MSG_LENGTH = 50; //maximální délka textu
    char message[MAX_MSG_LENGTH]; //řetězec pro uložení textu
    Serial.readBytesUntil('\n', message, MAX_MSG_LENGTH); //načtení textu
    message[MAX_MSG_LENGTH-1] = 0; //zaručení ukončení řetězce
    lcd.setCursor(0, 0); //přechod na první řádek
    lcd.print(message);  //výpis textu
  }
}
```

2.2 Zobrazení textu ze sériové linky pomocí UnoArduSim - simulace

Nejprve si nakonfigurujte HW konfiguraci dle potřeby:



Dle použitých pinů je třeba konfiguraci doplnit

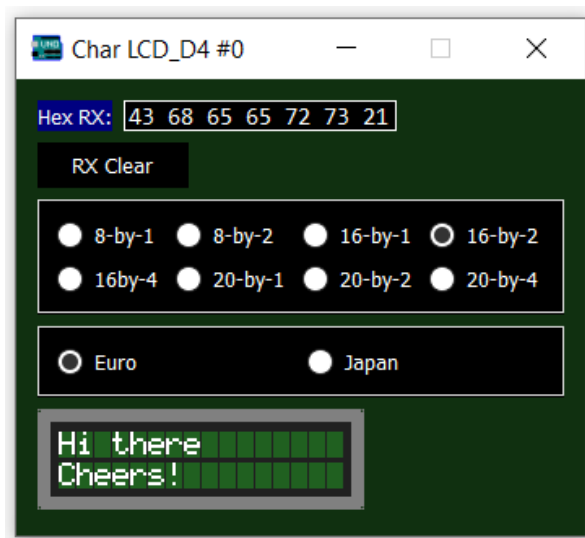


Výpis programu:

```
#include "Native_LiquidCrystal.h"
/* Definice komunikacnich pinu. Zde je potreba dat pozor na datove piny, protoze
UnoArduSim
ma pouze jedno okenko pro nejnizsi dat. pin (DB4) a potom bere, ze dalsi tri
nasledujici piny jsou DB5, DB6 a DB7.
Take je mozne vynechat pin R/W - coz delam, protoze muj realny displej ho ma
natvrdo na GND.
*/
#define RS_PIN 7
#define EN_PIN 6
#define D4_PIN 2
#define D5_PIN 3
#define D6_PIN 4
#define D7_PIN 5
/* Instance lcd displeje, podle toho jakou si vyberu knihovnu */
Native_LiquidCrystal lcd(RS_PIN, EN_PIN, D4_PIN, D5_PIN, D6_PIN, D7_PIN);

void setup() {
  /* Init displeje */
  lcd.begin(16, 2);
  /* Tohle tu musi byt */
  lcd.setCursor(0, 0);
  /* Staticky vypis */
  lcd.print("Hi there"); //vypis textu na první řádek
  lcd.setCursor(0, 1);
  lcd.print("Cheers!"); //vypis textu na druhý řádek
}

void loop() {
}
```



Vzorový program pro statické zobrazení na LCD displeji. Pokud chcete, je možné jej doplnit o načítání textu, který bude zobrazen na displeji LCD. Je však třeba doplnit inicializaci sériové linky do metody `setup()`

```
{
    if(Serial.available()){ //kontrola zda je neco poslano
        String msg = Serial.readString(); //nacteni textu
        msg.trim(); //oriznuti delicich znaku (\r \n)
        lcd.clear(); //vymazani displeje
        lcd.setCursor(0, 0);
        lcd.print(msg); //vypis textu
    }
}
```

3 Úkoly

3.1 Úkol 1 – Autoscroll

Upravte výše uvedený kód tak, aby při zaslání textu delšího než 8 znaků došlo k automatickému odrolování. Využijte přitom metodu `autoscroll()`, viz příklad „File“ -> „Examples“ -> „LiquidCrystal“ -> „Autoscroll“. Před výpisem nového textu display smažte metodou `clear()`. Kratší text než 8 znaků zobrazte přímo, u delšího textu začněte výpisem čtyř mezer, poté prvních čtyř znaků textu a následně začněte vypisované znaky scrollovat metodou `autoscroll()`. Na konci výpisu scrollování vypněte metodou `noAutoscroll()`.

3.2 Úkol 2 – Scrollování po řádcích

Upravte ukázkový kód tak, aby po vložení nového textu ze sériové linky se tento vložil na druhý řádek. Text zobrazený dříve na druhém řádku vypíše na řádek první. S tím souvisí to, že je potřeba si text z předchozího řádku pamatovat v proměnné.