

## Arduino: displej 16x2 s I2C

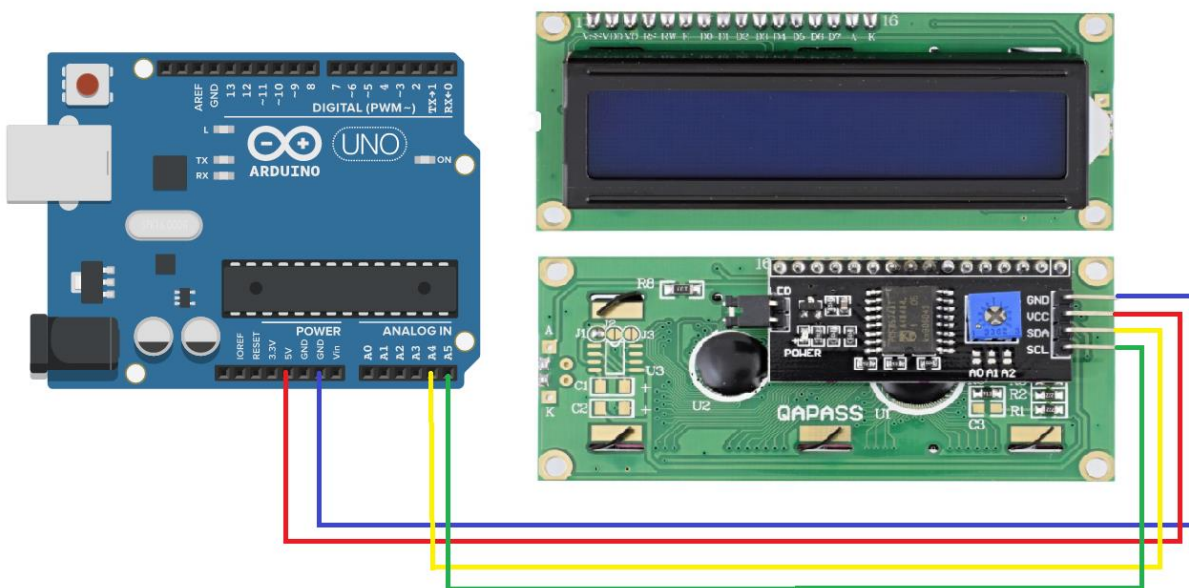
**LCD displej 16x2 s I2C** je veľmi vhodný pre jednoduché projekty s malými požiadavkami na zobrazenie. Veľkou výhodou je ovládanie zbernice I2C, pretože na komunikáciu s displejom sú potrebné iba 2 dátové kolíky SDA (pin A4) a SCL (pin A5). Okrem toho sa samozrejme pripája GND a +5V.

Podsvietenie LED môže byť vypnuté vytiahnutím prepajky, adresa môže byť nastavená pomocou jumperu adresy odpojením alebo skrátením spájkovacích plôch. Z výroby je adresa nastavená na hodnotu 0x27 alebo 0x3F. Kontrast je nastaviteľný pomocou potenciometra.

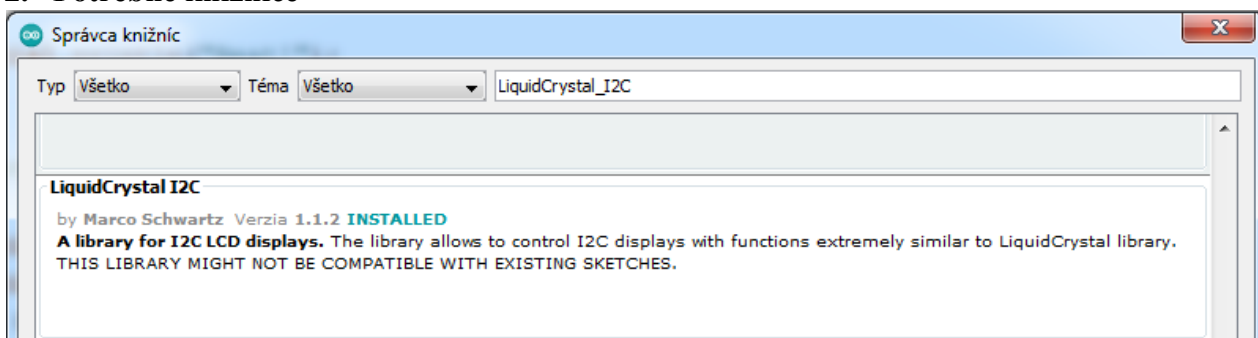


### 1. Schéma zapojenia

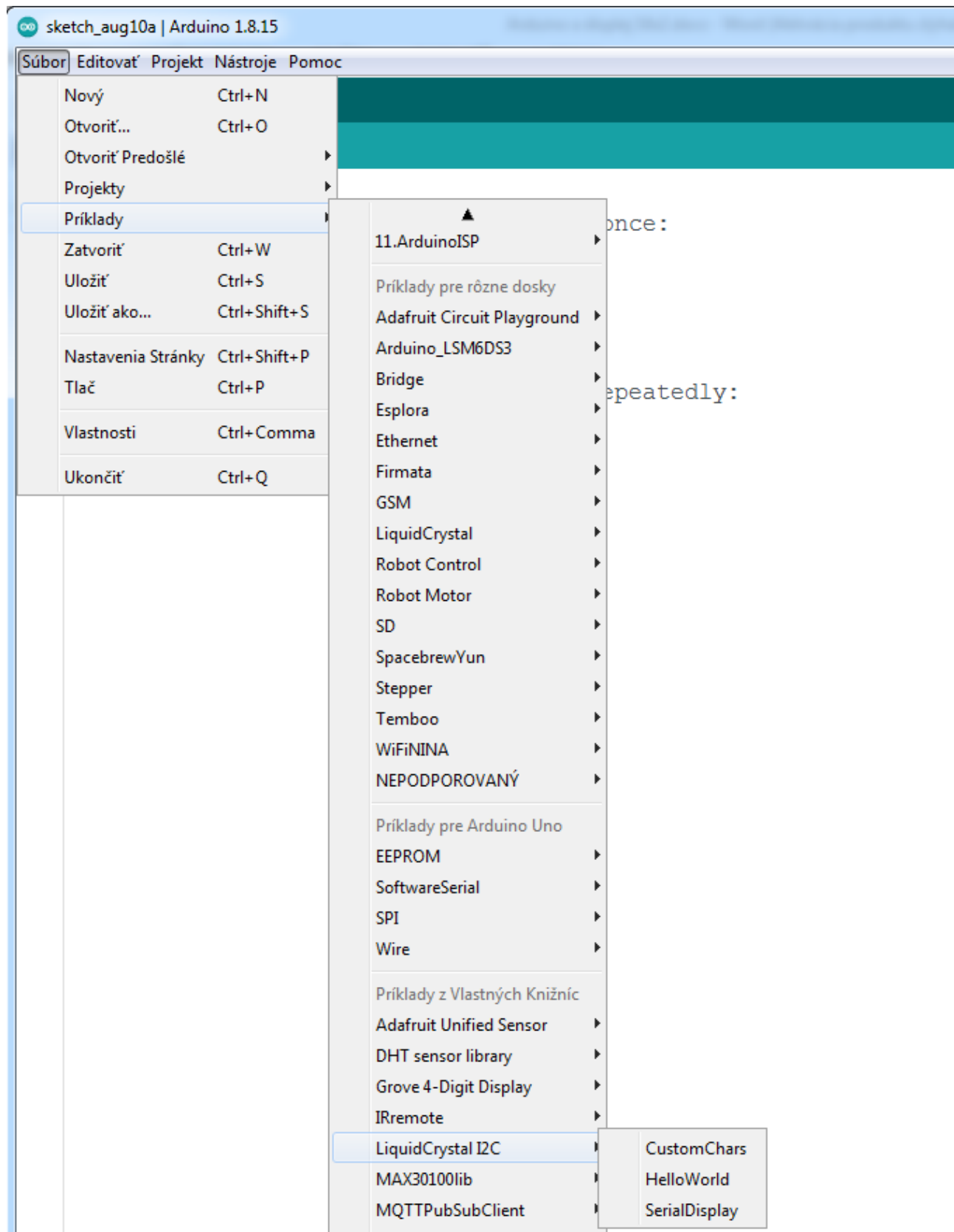
UNO R3	LCD1602_IIC
GND	GND
+5V	VCC
A4	SDA
A5	SCL



### 2. Potrebne knižnice



Po nainštalovaní knižnice LiquidCrystal\_I2C.h sú v sekcii Príklady k dispozícii 3 vzorové programy:

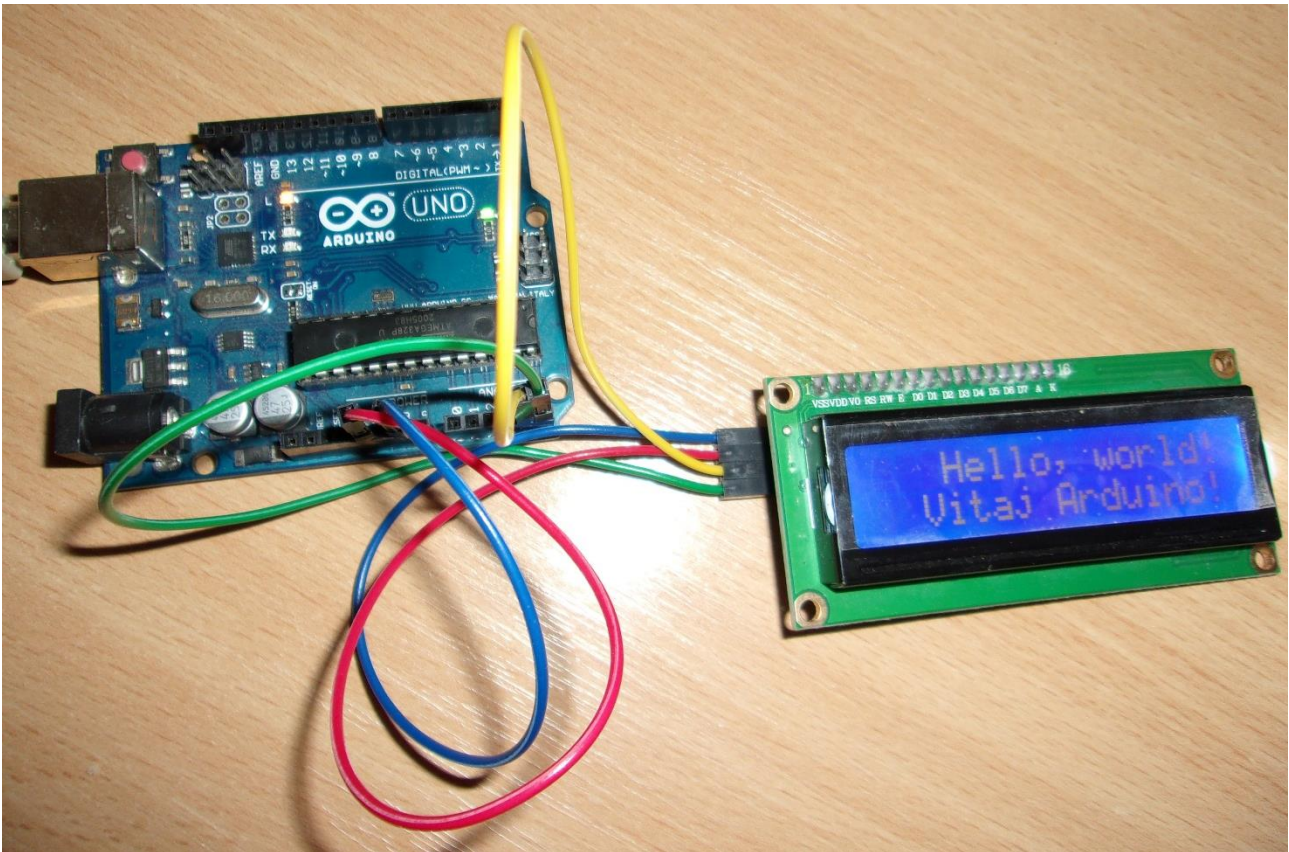


### 3. Upravený program Hello world

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// set the LCD address to 0x27 for a 16 chars and 2 line display
LiquidCrystal_I2C lcd(0x27,16,2);

void setup()
{
  lcd.init();           // initialize the lcd
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Hello, world!");
  lcd.setCursor(2,1);
  lcd.print("Vitaj Arduino!");
}

void loop()
{
}
```



### 4. Upravený program Vlastné znaky

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#if defined(ARDUINO) && ARDUINO >= 100
#define printByte(args) write(args);
#else
#define printByte(args) print(args,BYTE);
#endif

uint8_t bell[8] = {0x4,0xe,0xe,0xe,0x1f,0x0,0x4};
uint8_t note[8] = {0x2,0x3,0x2,0xe,0x1e,0xc,0x0};
uint8_t hodiny[8] = {0x0,0xe,0x15,0x17,0x11,0xe,0x0};
uint8_t heart[8] = {0x0,0xa,0x1f,0x1f,0xe,0x4,0x0};
```

```

uint8_t duck[8] = {0x0,0xc,0x1d,0xf,0xf,0x6,0x0};
uint8_t check[8] = {0x0,0x1,0x3,0x16,0x1c,0x8,0x0};
uint8_t cross[8] = {0x0,0x1b,0xe,0x4,0xe,0x1b,0x0};
uint8_t retarrow[8] = { 0x1,0x1,0x5,0x9,0x1f,0x8,0x4};
LiquidCrystal_I2C lcd(0x27,20,4); // set the LCD address to 0x27 for a 16 chars and 2 line
display

void setup()
{
  lcd.init(); // initialize the lcd
  lcd.backlight();
  lcd.createChar(0, bell);
  lcd.createChar(1, note);
  lcd.createChar(2, hodiny);
  lcd.createChar(3, heart);
  lcd.createChar(4, duck);
  lcd.createChar(5, check);
  lcd.createChar(6, cross);
  lcd.createChar(7, retarrow);
  lcd.home();
}

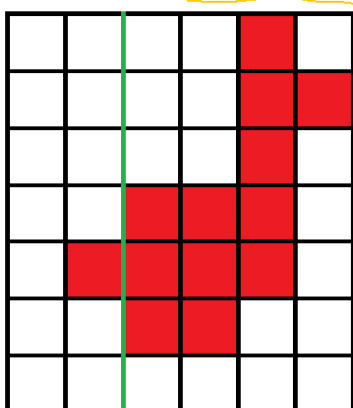
void loop()
{
  lcd.setCursor(0, 0);
  lcd.print("Vlastne znaky:");
  lcd.setCursor(0, 1); lcd.print("0=");lcd.printByte(0);
  lcd.setCursor(4, 1); lcd.print("1=");lcd.printByte(1);
  lcd.setCursor(8, 1); lcd.print("2=");lcd.printByte(2);
  lcd.setCursor(12, 1); lcd.print("3=");lcd.printByte(3);
  delay(5000);
}

```

## 5. Ako sa kóduje vlastný znak

Každý znak displeja je matica 7x5 pixelov. Vysvietený pixel kódujeme 1, zhasnutý kódujeme 0. Štvoricu, resp. dvojicu binárnych cifier prevedieme do hexadecimálnej sústavy. Čiže napr. 011110 rozdelíme na 01 a 1110, po prevedení do hexadecimálnej sústavy je to 1 a E, do programu tento riadok matice zapíšeme ako 0x1e.

note[8]={0x2, 0x3, 0xe, 0x1e, 0xc, 0x0};



$00_2 = 0_{10}$     $0010_2 = 2_{10}$

$01_2 = 1_{10}$     $1110_2 = E_{10}$

