

Arduino TFT LCD 128 x 160 px

1. POPIS

Jedná se o kompaktní LCD displej, který má rozlišení úměrné ke své úhlopříčce. Pracuje na rozhraní SPI, tudíž komunikace s Arduinem je relativně rychlá. Modul disponuje také slotem pro micro SD kartu. Vhodné pro rozšíření aplikací o grafické rozhraní.

Základní charakteristika:

- Sběrnice SPI
- Rozlišení 160 x 128
- Úhlopříčka 1,77"
- Slot pro micro SD kartu
- Kompatibilní s vývojovými kity Arduino



2. SPECIFIKACE

Driver	ST7735	Sběrnice	SPI
Napájecí napětí	5 V	Úhlopříčka disp.	1,77"
Pracovní proud	cca 50 mA	Typ SD karty	micro
Barevná hloubka	18 bitů	Rozměry (mm)	60 x 42
Rozlišení	160 x 128	Hmotnost	16 g

3. ZAPOJENÍ

Názorná ukázka připojení k Arduino UNO R3. Pro správnou funkci displeje doporučujeme používat tuto knihovnu https://github.com/Bodmer/TFT_ST7735. Pokud uživatel zapojí displej podle uvedeného obrázku, je nutné přepsat definici pinů přímo v knihovně viz. níže.

Místo uložení knihovny: Dokumenty >> Arduino >> libraries

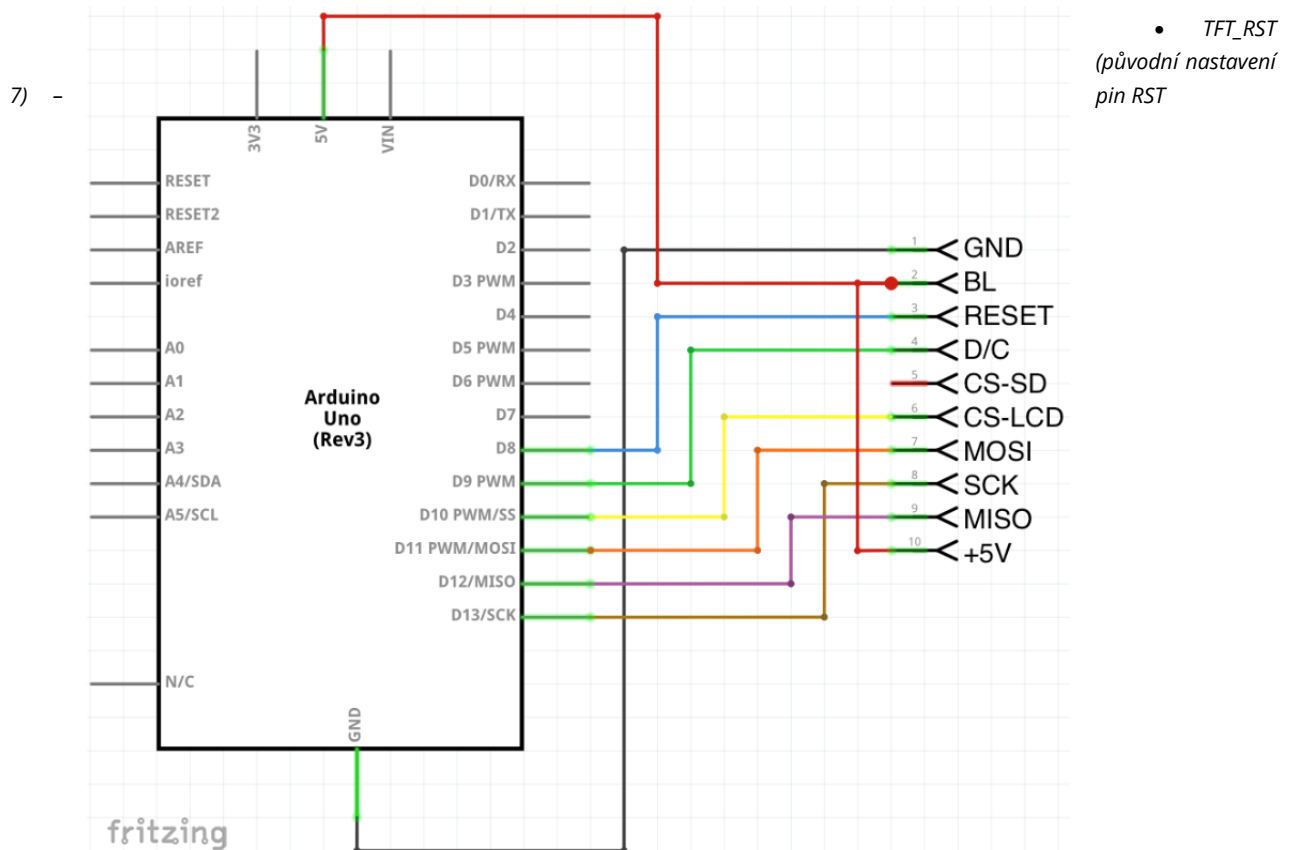
Definice pinů v User_Setup.h

Cesta k souboru: Dokumenty >> Arduino >> libraries >> TFT_ST7735 >> User_Setup.h

```
#define TFT_CS 10
#define TFT_DC 9
#define TFT_RST 8
```

poznámka:

- *TFT_CS (původní nastavení 9) – pin CS-LCD*
- *TFT_DC (původní nastavení 8) – pin D/C*



```
00101
01001
00001
```

4. UKÁZKA PROGRAMU

Jedná se o příklad z knihovny TFT_ST7735. Pro správnou funkčnost, je žádoucí dodržet kroky v předchozí kapitole.

```
#include <TFT_ST7735.h> // Graphics and font library for ST7735 driver chip
#include <SPI.h>

TFT_ST7735 tft = TFT_ST7735(); // Invoke library, pins defined in User_Setup.h

uint32_t targetTime = 0; // for next 1 second timeout
byte omm = 99;
boolean initial = 1;
byte xcolon = 0;
unsigned int colour = 0;

static uint8_t conv2d(const char* p) {
    uint8_t v = 0;
    if ('0' <= *p && *p <= '9')
        v = *p - '0';
    return 10 * v + *++p - '0';
}

uint8_t hh=conv2d(__TIME__), mm=conv2d(__TIME__+3), ss=conv2d(__TIME__+6); // Get H, M, S from compile time

void setup(void) {
    tft.init();
    tft.setRotation(1);
    tft.fillScreen(ST7735_BLACK);

    tft.setTextColor(ST7735_YELLOW, ST7735_BLACK); // Note: the new fonts do not draw the background colour

    targetTime = millis() + 1000;
}

void loop() {
    if (targetTime < millis()) {
        targetTime = millis()+1000;
        ss++; // Advance second
        if (ss==60) {
            ss=0;
            omm = mm;
            mm++; // Advance minute
            if(mm>59) {
                mm=0;
                hh++; // Advance hour
                if (hh>23) {
                    hh=0;
                }
            }
        }
    }
}
```

```

if (ss==0 || initial) {
  initial = 0;
  tft.setTextColor(ST7735_GREEN, ST7735_BLACK);
  tft.setCursor (8, 52);
  tft.print(__DATE__); // This uses the standard ADAfruit small font

  tft.setTextColor(ST7735_BLUE, ST7735_BLACK);
  tft.drawCentreString("It is windy",120,48,2); // Next size up font 2

  //tft.setTextColor(0xF81F, ST7735_BLACK); // Pink
  //tft.drawCentreString("12.34",80,100,6); // Large font 6 only contains characters [space] 0 1 2 3 4 5 6 7 8 9 . : a p m
}

byte xpos = 6;
byte ypos = 0;
if (omm != mm) { // Only redraw every minute to minimise flicker
  // Uncomment ONE of the next 2 lines, using the ghost image demonstrates text overlay as time is drawn over it
  tft.setTextColor(0x39C4, ST7735_BLACK); // Leave a 7 segment ghost image, comment out next line!
  //tft.setTextColor(ST7735_BLACK, ST7735_BLACK); // Set font colour to back to wipe image
  // Font 7 is to show a pseudo 7 segment display.
  // Font 7 only contains characters [space] 0 1 2 3 4 5 6 7 8 9 0 : .
  tft.drawString("88:88",xpos,ypos,7); // Overwrite the text to clear it
  tft.setTextColor(0xFBEO, ST7735_BLACK); // Orange
  omm = mm;

  if (hh<10) xpos+= tft.drawChar('0',xpos,ypos,7);
  xpos+= tft.drawNumber(hh,xpos,ypos,7);
  xcolon=xpos;
  xpos+= tft.drawChar(':',xpos,ypos,7);
  if (mm<10) xpos+= tft.drawChar('0',xpos,ypos,7);
  tft.drawNumber(mm,xpos,ypos,7);
}

if (ss%2) { // Flash the colon
  tft.setTextColor(0x39C4, ST7735_BLACK);
  xpos+= tft.drawChar(':',xcolon,ypos,7);
  tft.setTextColor(0xFBEO, ST7735_BLACK);
}
else {
  tft.drawChar(':',xcolon,ypos,7);
  colour = random(0xFFFF);
  // Erase the text with a rectangle
  tft.fillRect (0, 64, 160, 20, ST7735_BLACK);
  tft.setTextColor(colour);
  tft.drawRightString("Colour",75,64,4);
  String scolour = String(colour,HEX);
  scolour.toUpperCase();
  char buffer[20];
  scolour.toCharArray(buffer,20);
  tft.drawString(buffer,82,64,4);
}
}
}
}

```