

www.abico.cz/Arduino



Video-návod k tomu-
to dílu
najdete
ZDE!

11. DÍL

ARDUINO



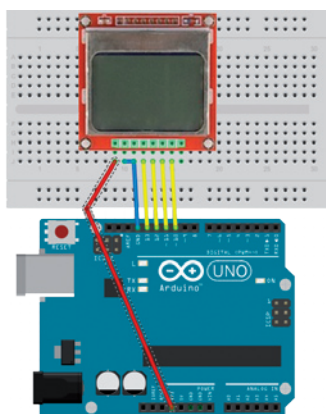
DISPLEJ Z TELEFONU

Jednou z možností, jak nám může Arduino něco sdělit, je displej. Využijeme starý displej z kdysi velmi rozšířeného a osvědčeného mobilního telefonu Nokia 5110.

1 Displej lze vymontovat ze starého telefonu nebo si koupit velice levně modul s tímto displejem.

2 Rozlišení je 84 x 48 bodů a je monochromatický. Ani zdaleka se nemůže měřit s dnešními tablety a mobilními telefony, pro naše účely je ovšem ideální. Je jednoduché ho ovládat, zapojení je také jednoduché a spotřeba minimální.

3 Pro práci s displejem si musíme přidat do prostředí knihovnu obsahující některé přístupové funkce, které nám usnadní programování. Tuto knihovnu najdete v odkazech ke stažení a nakupíte si ji do adresáře libraries, který najdete v adresáři s nainstalovaným vývojovým prostředím.



4 Dnes zde máme programy dva. První umí jen smazat obrazovku a malovat čáru, respektive zobrazuje posouvající se bod, který se „odráží“ od hran obrazovky.

```
#include <Arduino.h>

// vlnodni LCD na 4 piny SCLK, DW, D/C, RST
PCD8544 nokia = PCD8544( 10, 11, 12, 0, 13);

void setup(void)
{
  nokia.init(50); // nastave kontrast
  nokia.clear(); // maže obrazovku
  nokia.display(); // zobraz smazanou
}

byte k = 7; //startovací pozice bodu
byte y = 0;
byte zmena = 2; //velikost zmeny
byte yzmena = 1;

void loop(void)
{
  nokia.setPixel(k,y,1); // rozsvit bod na
  x += zmena;
  y += yzmena;
  if(k <= 0 || x >= LCDWIDTH){
    zmena = -zmena;
  }
  if(y <= 0 || y >= LCDHEIGHT){
    yzmena = -yzmena;
  }
  nokia.display(); // aktivuj zmenu
}
```

5 Druhý program obsahuje jednoduchou animaci. Po obrazovce se pohybují dvoje přesýpací hodiny. Všimněte si, že nejdříve se obrázek namaluje pomocí funkce setPixel() v paměti a poté se pomocí funkce display() přenesou na obrazovku. Touto technikou je dosaženo rychlejšího zobrazování, ale i toho, že obrázek neblíká.

```
#include <PCD8544.h>

// vlnodni LCD na 4 piny SCLK, DW, D/C, RST
PCD8544 nokia = PCD8544( 10, 11, 12, 0, 13);

int g_uhlevak = LCDWIDTH / 2;
int g_dhlevak = LCDHEIGHT / 2;

int STEPA_OBRAZKY = 6;
int VYSDA_OBRAZKY = 8;

byte obrazek[6][6] = {
  {1,1,1,1,1,1},
  {1,0,0,0,0,1},
  {0,1,0,0,0,0},
  {0,0,1,0,0,0},
  {0,0,1,0,0,0},
  {1,1,1,1,1,1}
};

byte obrazek[6][12] = {
  {0,0,0,0,1,1,0,0,0,0,0,1,1},
  {0,0,0,0,1,1,0,0,0,0,1,1},
  {0,0,0,0,1,1,0,0,0,0,1,1},
  {0,0,0,0,1,1,0,0,0,0,1,1},
  {0,0,0,0,1,1,0,0,0,0,1,1},
  {0,1,1,0,0,0,0,0,0,1,1,1}
};

void namalujObrazek(int iPosiceK, int iPosiceY)
{
  for (int k = 0; k < STEPA_OBRAZKY; k++)
  {
    for (int y = 0; y < VYSDA_OBRAZKY; y++){
      nokia.setPixel(iPosiceK+iPosiceK/2,
        iPosiceY+y)-(VYSDA_OBRAZKY/2),obrazek[k][y]);
    }
  }
}

void setup(void)
{
  nokia.init(60); // set contrast
  nokia.clear();
  nokia.display();
}

int g_uhlevak = 3;
int g_dhlevak = 2;

void loop(void)
{
  nokia.clear();
  namalujObrazek(LCDWIDTH - g_uhlevak, LCDHEIGHT - g_dhlevak);
  namalujObrazek(g_uhlevak, g_dhlevak);
  nokia.display();
  delay(1000);
  g_uhlevak += g_uhlevak;
  g_dhlevak += g_dhlevak;
  if(g_uhlevak <= (STEPA_OBRAZKY/2)+1 || g_uhlevak > (LCDWIDTH - (STEPA_OBRAZKY/2))-1){
    g_uhlevak = -g_uhlevak;
  }
  if(g_dhlevak <= (VYSDA_OBRAZKY/2)+1 || g_dhlevak > (LCDHEIGHT - (VYSDA_OBRAZKY/2))-1){
    g_dhlevak = -g_dhlevak;
  }
}
```

6 Všimněte si také použití konstant LCDWIDTH a LCDHEIGHT. Je výhodné používat pojmenované konstanty, protože pak nemusíte přepisovat celý program, pokud vyměníte display za nějaký s větším rozlišením. Pouze změňte hodnoty těchto předdefinovaných konstant.