

TFT LCD shield MCFRIEND

1. POPIS

Tento TFT LCD displej s úhlopříčkou 2,4 palce je určen především pro vývojové kity Arduino/Genuino UNO, kde je možno jej použít jako shield. Je však kompatibilní i s dalšími vývojovými kity Arduino/Genuino a spoustou dalších (např. Raspberry PI).

Základní charakteristika modulu:

- Přímá kompatibilita s Arduino/Genuino UNO
- Zobrazuje až 262 tisíc barev
- Používá digitální piny 5 až 13 a analogové 0 až 3
- Rezistivní dotyková vrstva
- Slot pro microSD kartu ze spodní strany

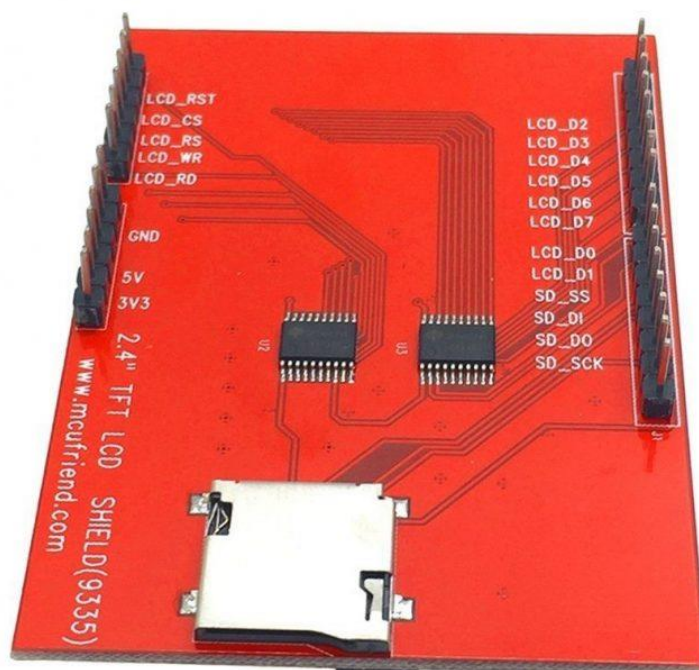


2. SPECIFIKACE

Grafický driver	IL9341	Napájecí napětí	5 V
Kontrolér obsahující VRAM	SPFD5408	Logické napětí	3,3 nebo 5 V
Rozlišení	320 x 240	Typ dotykové plochy	rezistivní
Úhlopříčka	2,4"	Typ paměťové karty	micro SD
Podsívění	4 LED diody	Rozměry (mm)	71 x 52 x 7
Barevné hloubka	18 bitů	Hmotnost	33 g

3. Zapojení

Shield stačí připojit do vývojového kitu Arduino UNO nebo klonů Arduino Mega (originální kit Arduino Mega je možné použít pouze po úpravě pinové lišty). Žádné další zapojování není třeba.





4. UKÁZKA PROGRAMU

Pro chod tohoto programu je zapotřebí nainstalovat knihovny Adafruit_GFX.h a MCUFRIEND_kbv (https://github.com/prenticedavid/MCUFRIEND_kbv) jež obsahuje všechny ostatní potřebné knihovny využitelné pro tento shield.

```
#include <Adafruit_GFX.h>
#include <UTFTGLUE.h>
UTFTGLUE myGLCD(0x0154,A2,A1,A3,A4,A0);

// Declare which fonts we will be using
#ifndef SmallFont
extern uint8_t SmallFont[];
#endif

void setup()
{
  randomSeed(analogRead(5));
  pinMode(A0, OUTPUT);
  digitalWrite(A0, HIGH);

  // Setup the LCD
  myGLCD.InitLCD();
  myGLCD.setFont(SmallFont);
}

void loop()
{
  int buf[318];
  int x, x2;
  int y, y2;
  int r;

  // Clear the screen and draw the frame
  myGLCD.clrScr();

  myGLCD.setColor(255, 0, 0);
  myGLCD.fillRect(0, 0, 319, 13);
  myGLCD.setColor(64, 64, 64);
  myGLCD.fillRect(0, 226, 319, 239);
  myGLCD.setColor(255, 255, 255);
  myGLCD.setBackColor(255, 0, 0);
  myGLCD.print("* Universal Color TFT Display Library *", CENTER, 1);
  myGLCD.setBackColor(64, 64, 64);
  myGLCD.setColor(255,255,0);
  myGLCD.print("<http://electronics.henningkarlsen.com>", CENTER, 227);

  myGLCD.setColor(0, 0, 255);
  myGLCD.drawRect(0, 14, 319, 225);

  // Draw crosshairs
  myGLCD.setColor(0, 0, 255);
```

```

myGLCD.setBackgroundColor(0, 0, 0);
myGLCD.drawLine(159, 15, 159, 224);
myGLCD.drawLine(1, 119, 318, 119);
for (int i=9; i<310; i+=10)
    myGLCD.drawLine(i, 117, i, 121);
for (int i=19; i<220; i+=10)
    myGLCD.drawLine(157, i, 161, i);

// Draw sin-, cos- and tan-lines
myGLCD.setColor(0,255,255);
myGLCD.print("Sin", 5, 15);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(sin(((i*1.13)*3.14)/180)*95));
}

myGLCD.setColor(255,0,0);
myGLCD.print("Cos", 5, 27);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(cos(((i*1.13)*3.14)/180)*95));
}

myGLCD.setColor(255,255,0);
myGLCD.print("Tan", 5, 39);
for (int i=1; i<318; i++)
{
    myGLCD.drawPixel(i,119+(tan(((i*1.13)*3.14)/180)));
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);
myGLCD.setColor(0, 0, 255);
myGLCD.setBackgroundColor(0, 0, 0);
myGLCD.drawLine(159, 15, 159, 224);
myGLCD.drawLine(1, 119, 318, 119);

// Draw a moving sinewave
x=1;
for (int i=1; i<(318*20); i++)
{
    x++;
    if (x==319)
        x=1;
    if (i>319)
    {
        if ((x==159) || (buf[x-1]==119))
            myGLCD.setColor(0,0,255);
        else
            myGLCD.setColor(0,0,0);
        myGLCD.drawPixel(x,buf[x-1]);
    }
}

```

```

myGLCD.setColor(0,255,255);
y=119+(sin(((i*1.1)*3.14)/180)*(90-(i / 100)));
myGLCD.drawPixel(x,y);
buf[x-1]=y;
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled rectangles
for (int i=1; i<6; i++)
{
  switch (i)
  {
    case 1:
      myGLCD.setColor(255,0,255);
      break;
    case 2:
      myGLCD.setColor(255,0,0);
      break;
    case 3:
      myGLCD.setColor(0,255,0);
      break;
    case 4:
      myGLCD.setColor(0,0,255);
      break;
    case 5:
      myGLCD.setColor(255,255,0);
      break;
  }
  myGLCD.fillRect(70+(i*20), 30+(i*20), 130+(i*20), 90+(i*20));
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);
// Draw some filled, rounded rectangles
for (int i=1; i<6; i++)
{
  switch (i)
  {
    case 1:
      myGLCD.setColor(255,0,255);
      break;
    case 2:
      myGLCD.setColor(255,0,0);
      break;
    case 3:
      myGLCD.setColor(0,255,0);
      break;
    case 4:

```

```

    myGLCD.setColor(0,0,255);
    break;
    case 5:
    myGLCD.setColor(255,255,0);
    break;
}
myGLCD.fillRect(190-(i*20), 30+(i*20), 250-(i*20), 90+(i*20));
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some filled circles
for (int i=1; i<6; i++)
{
    switch (i)
    {
        case 1:
        myGLCD.setColor(255,0,255);
        break;
        case 2:
        myGLCD.setColor(255,0,0);
        break;
        case 3:
        myGLCD.setColor(0,255,0);
        break;
        case 4:
        myGLCD.setColor(0,0,255);
        break;
        case 5:
        myGLCD.setColor(255,255,0);
        break;
    }
    myGLCD.fillCircle(100+(i*20),60+(i*20), 30);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some lines in a pattern
myGLCD.setColor (255,0,0);
for (int i=15; i<224; i+=5)
{
    myGLCD.drawLine(1, i, (i*1.44)-10, 224);
}
myGLCD.setColor (255,0,0);
for (int i=224; i>15; i-=5)
{
    myGLCD.drawLine(318, i, (i*1.44)-11, 15);
}
}

```

```

myGLCD.setColor (0,255,255);
for (int i=224; i>15; i-=5)
{
  myGLCD.drawLine(1, i, 331-(i*1.44), 15);
}
myGLCD.setColor (0,255,255);
for (int i=15; i<224; i+=5)
{
  myGLCD.drawLine(318, i, 330-(i*1.44), 224);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,225);

// Draw some random circles
for (int i=0; i<100; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=32+random(256);
  y=45+random(146);
  r=random(30);
  myGLCD.drawCircle(x, y, r);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

```

```

// Draw some random rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(207);
    x2=2+random(316);
    y2=16+random(207);
    myGLCD.drawRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

// Draw some random rounded rectangles
for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(207);
    x2=2+random(316);
    y2=16+random(207);
    myGLCD.drawRoundRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

for (int i=0; i<100; i++)
{
    myGLCD.setColor(random(255), random(255), random(255));
    x=2+random(316);
    y=16+random(209);
    x2=2+random(316);
    y2=16+random(209);
    myGLCD.drawLine(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,318,224);

for (int i=0; i<10000; i++)

```



```
{
  myGLCD.setColor(random(255), random(255), random(255));
  myGLCD.drawPixel(2+random(316), 16+random(209));
}

delay(2000);

myGLCD.fillScr(0, 0, 255);
myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect(80, 70, 239, 169);

myGLCD.setColor(255, 255, 255);
myGLCD.setBackgroundColor(255, 0, 0);
myGLCD.print("That's it!", CENTER, 93);
myGLCD.print("Restarting in a", CENTER, 119);
myGLCD.print("few seconds...", CENTER, 132);

myGLCD.setColor(0, 255, 0);
myGLCD.setBackgroundColor(0, 0, 255);
myGLCD.print("Runtime: (msecs)", CENTER, 210);
myGLCD.printNumI(millis(), CENTER, 225);

delay (10000);
}
```