

TFT LCD Mega Shield V2.2

1. POPIS

Jedná se o shield pro platformu Arduino Mega 2560, který umožní pohodlně připojit displeje o větších rozměrech, které lze následně jednoduše programovat. Do shieldu lze připojit TFT displeje s podporou dotykového ovládání a SD karty (příklad TFT LCD s kódem produktu 1474276662).

Základní charakteristika:

- Podporované úhlopříčky: 3.2, 4.3, 5.0 a 7.0"
- Shield podporuje 16bitové datové rozhraní
- Napájení pro displej 3,3 V
- Přímá kompatibilita s Arduino Mega 2560
- Snadné připojení displeje



2. SPECIFIKACE

Datové rozhraní	16 bit
Vstupní napětí	5 VDC
Výstupní napětí	3,3 V
Kompatibilita	Arduino Mega 2560

00101
01001
00001

3. UKÁZKA PROGRAMU

K správné funkci musí uživatel nainstalovat knihovnu [UTFT.h](#). Tento kód je příkladem z této knihovny a kompatibilní s Arduino Mega 2560. Dále je pro správnou funkci nutné připojit do shieldu podporovaný TFT displej. V tomto případě byl použit displej s kódem produktu [1427735951](#).

```
#include <UTFT.h>

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

UTFT myGLCD(ITDB50,38,39,40,41);

void setup()
{
  randomSeed(analogRead(0));

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

void loop()
{
  int buf[798];
  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, 799, 13);
  myGLCD.setColor(64, 64, 64);
  myGLCD.fillRect(0, 466, 799, 479);
  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://www.RinkyDinkElectronics.com/>", CENTER, 467);

  myGLCD.setColor(0, 0, 255);
  myGLCD.drawRect(0, 14, 799, 465);

  // Draw crosshairs
  myGLCD.setColor(0, 0, 255);
  myGLCD.setBackColor(0, 0, 0);
  myGLCD.drawLine(399, 15, 399, 464);
  myGLCD.drawLine(1, 239, 798, 239);
  for (int i=9; i<790; i+=10)
    myGLCD.drawLine(i, 237, i, 242);
  for (int i=19; i<470; i+=10)
    myGLCD.drawLine(397, i, 402, i);
```

```

// Draw sin-, cos- and tan-lines
myGLCD.setColor(0,255,255);
myGLCD.print("Sin", 5, 15);
for (int i=1; i<798; i++)
{
  myGLCD.drawPixel(i,239+(sin(((i*1.13)*3.14)/180)*200));
}

myGLCD.setColor(255,0,0);
myGLCD.print("Cos", 5, 27);
for (int i=1; i<798; i++)
{
  myGLCD.drawPixel(i,239+(cos(((i*1.13)*3.14)/180)*200));
}

myGLCD.setColor(255,255,0);
myGLCD.print("Tan", 5, 39);
for (int i=1; i<798; i++)
{
  myGLCD.drawPixel(i,239+(tan(((i*0.9)*3.14)/180)));
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);
myGLCD.setColor(0, 0, 255);
myGLCD.setBackColor(0, 0, 0);
myGLCD.drawLine(399, 15, 399, 464);
myGLCD.drawLine(1, 239, 798, 239);

// Draw a moving sinewave
x=1;
for (int i=1; i<(798*20); i++)
{
  x++;
  if (x==799)
    x=1;
  if (i>799)
  {
    if ((x==399) | |(buf[x-1]==239))
      myGLCD.setColor(0,0,255);
    else
      myGLCD.setColor(0,0,0);
    myGLCD.drawPixel(x,buf[x-1]);
  }
  myGLCD.setColor(0,255,255);
  y=239+(sin(((i*1.65)*3.14)/180)*(200-(i / 100)));
  myGLCD.drawPixel(x,y);
  buf[x-1]=y;
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random filled rectangles
for (int i=0; i<50; i++)
{

```

```

myGLCD.setColor(random(255), random(255), random(255));
x=2+random(746);
y=16+random(397);
x2=x+50;
y2=y+50;
myGLCD.fillRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random filled, rounded rectangles
for (int i=0; i<50; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=2+random(746);
  y=16+random(397);
  x2=x+50;
  y2=y+50;
  myGLCD.fillRoundRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random filled circles
for (int i=0; i<50; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=27+random(746);
  y=41+random(397);
  myGLCD.fillCircle(x, y, 25);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some lines in a pattern
myGLCD.setColor (255,0,0);
for (int i=15; i<463; i+=5)
{
  myGLCD.drawLine(1, i, (i*1.66)-10, 463);
}
myGLCD.setColor (255,0,0);
for (int i=463; i>15; i-=5)
{
  myGLCD.drawLine(798, i, (i*1.66)+30, 15);
}
myGLCD.setColor (0,255,255);
for (int i=463; i>15; i-=5)
{
  myGLCD.drawLine(1, i, 770-(i*1.66), 15);
}
myGLCD.setColor (0,255,255);
for (int i=15; i<463; i+=5)

```

```

{
  myGLCD.drawLine(798, i, 810-(i*1.66), 463);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random circles
for (int i=0; i<250; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=32+random(736);
  y=45+random(386);
  r=random(30);
  myGLCD.drawCircle(x, y, r);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random rectangles
for (int i=0; i<250; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=2+random(796);
  y=16+random(447);
  x2=2+random(796);
  y2=16+random(447);
  myGLCD.drawRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

// Draw some random rounded rectangles
for (int i=0; i<250; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=2+random(796);
  y=16+random(447);
  x2=2+random(796);
  y2=16+random(447);
  myGLCD.drawRoundRect(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

for (int i=0; i<250; i++)
{
  myGLCD.setColor(random(255), random(255), random(255));
  x=2+random(796);
  y=16+random(447);
}

```

```

x2=2+random(796);
y2=16+random(447);
myGLCD.drawLine(x, y, x2, y2);
}

delay(2000);

myGLCD.setColor(0,0,0);
myGLCD.fillRect(1,15,798,464);

for (int i=0; i<10000; i++)
{
myGLCD.setColor(random(255), random(255), random(255));
myGLCD.drawPixel(2+random(796), 16+random(447));
}

delay(2000);

myGLCD.fillScr(0, 0, 255);
myGLCD.setColor(255, 0, 0);
myGLCD.fillRoundRect(320, 190, 479, 289);

myGLCD.setColor(255, 255, 255);
myGLCD.setBackColor(255, 0, 0);
myGLCD.print("That's it!", CENTER, 213);
myGLCD.print("Restarting in a", CENTER, 239);
myGLCD.print("few seconds...", CENTER, 252);

myGLCD.setColor(0, 255, 0);
myGLCD.setBackColor(0, 0, 255);
myGLCD.print("Runtime: (msecs)", CENTER, 450);
myGLCD.printNumI(millis(), CENTER, 465);

delay (10000);
}

```