

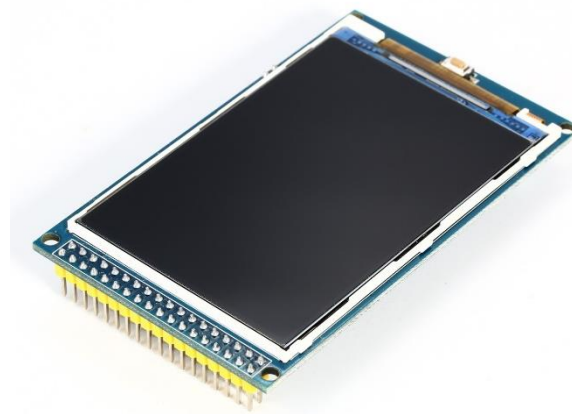
TFT LCD shield pro Arduino Mega 2560

1. POPIS

Jedná se o TFT LCD displej s úhlopříčkou 3,2 palce. Shield je určen především pro platformu Arduino Mega 2560 a její klony. Lze jej použít také s dalšími vývojovými kity, ovšem nikoliv jako shield (a navíc po úpravě nastavení pinů v kódu). Produkt obsahuje slot na SD kartu, jenž rozšiřuje paměť zařízení pro složitější aplikace, a také tlačítko reset.

Základní charakteristika:

- Přímá kompatibilita s Arduino Mega 2560
- Rozlišení 480 x 320 bodů
- Zobrazuje až 65 tisíc barev (RGB)
- Slot pro SD kartu

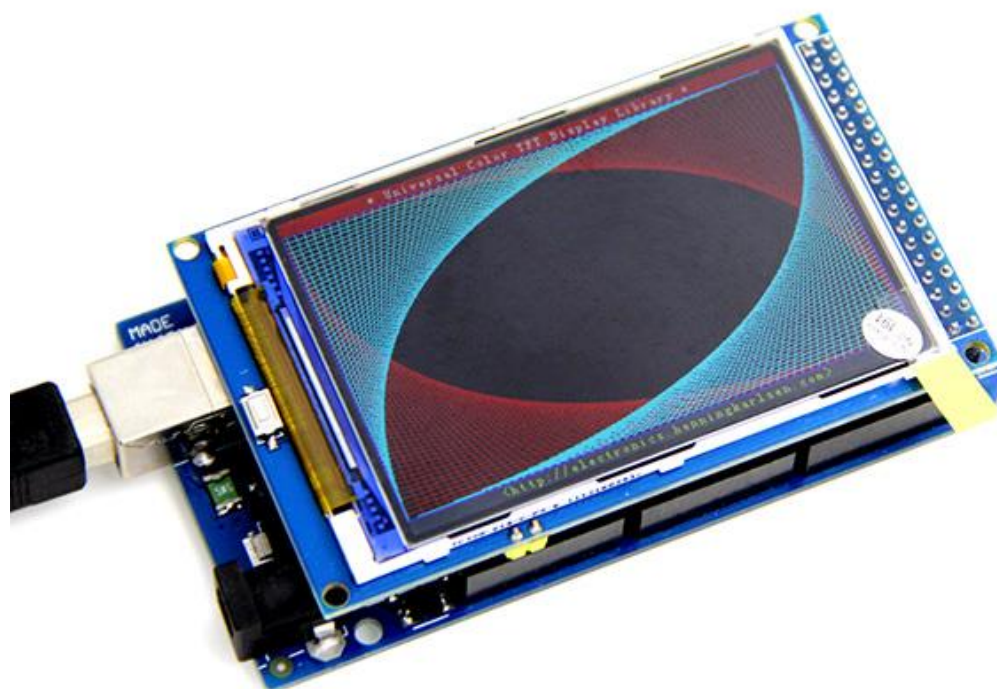
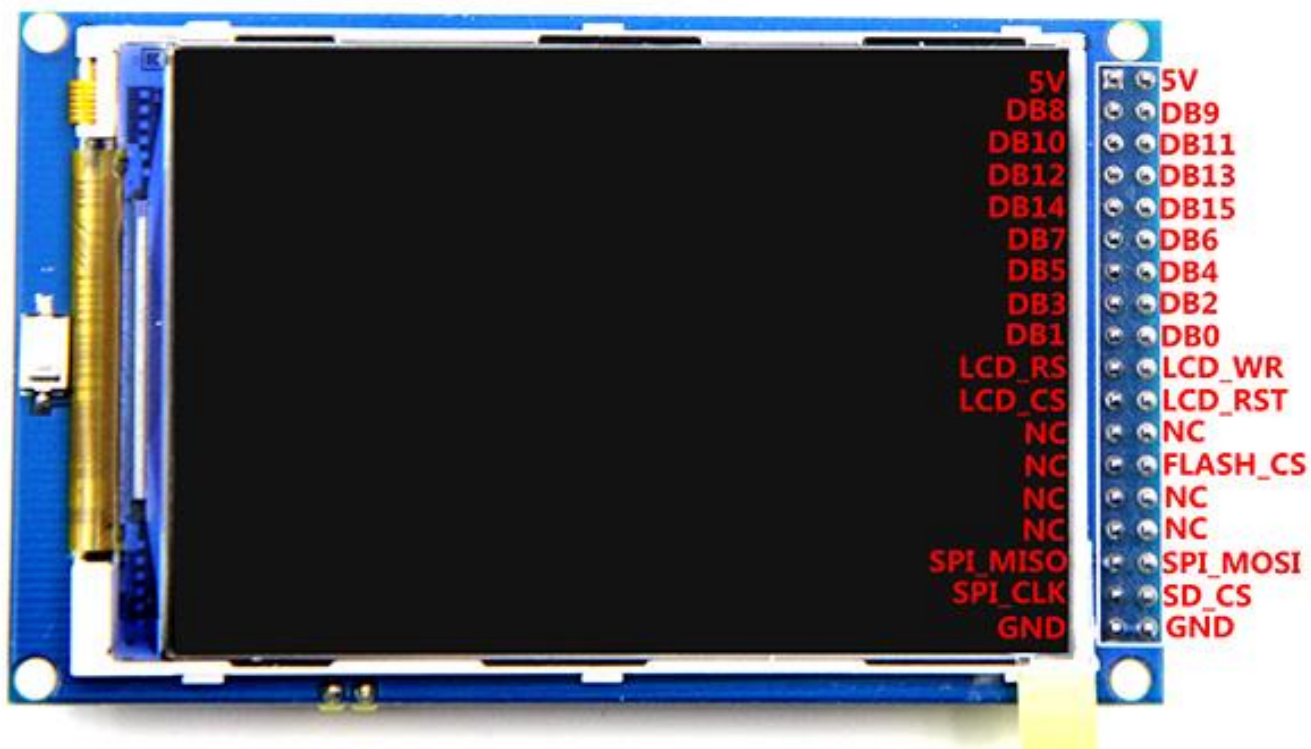


2. SPECIFIKACE

Grafický driver	HX8357B	Napájecí napětí	3,3 nebo 5 V
Typ displeje	TFT IPS	Logické napětí	3,3 nebo 5 V
Rozlišení	480 x 320	Proud	80 až 110 mA
Úhlopříčka	3,2"	Typ paměťové karty	SD
Podsvícení	6 LED diody	Rozměry (mm)	51 x 78 x 5
Barevné hloubka	16 bitů	Hmotnost	36 g

3. Zapojení

Shield stačí připojit do vývojové desky Arduino Mega 2560. Žádné další zapojování není třeba.



00101
01001
00001

4. UKÁZKA PROGRAMU

Pro chod tohoto programu je zapotřebí nainstalovat knihovny [TFT_HX835.h](#). Toto je příklad z uvedené knihovny.

```
#define WAIT 2000 // Delay between tests, set to 0 to demo speed, 2000 to see what it does!

#define CENTRE 240

#include <TFT_HX8357.h> // Hardware-specific library

TFT_HX8357 tft = TFT_HX8357(); // Invoke custom library

#define TFT_GREY 0x7BEF

uint32_t runTime = 0;

void setup()
{
  randomSeed(analogRead(0));
  Serial.begin(38400);
  // Setup the LCD
  tft.init();
  tft.setRotation(1);
}

void loop()
{
  int buf[478];
  int x, x2;
  int y, y2;
  int r;

  runTime = millis();
  // Clear the screen and draw the frame
  tft.fillScreen(TFT_BLACK);

  tft.fillRect(0, 0, 480, 13, TFT_RED);

  tft.fillRect(0, 305, 480, 320, TFT_GREY);
  tft.setTextColor(TFT_BLACK, TFT_RED);

  tft.drawCentreString("* TFT_HX8357 *", CENTRE, 3, 1);
  tft.setTextColor(TFT_YELLOW, TFT_GREY);
  tft.drawCentreString("Adapted by Bodmer", CENTRE, 309, 1);

  tft.drawRect(0, 14, 479, 305-14, TFT_BLUE);

  // Draw crosshairs
  tft.drawLine(239, 15, 239, 304, TFT_BLUE);
  tft.drawLine(1, 159, 478, 159, TFT_BLUE);
  for (int i=9; i<470; i+=10)
    tft.drawLine(i, 157, i, 161, TFT_BLUE);
  for (int i=19; i<220; i+=10)
    tft.drawLine(237, i, 241, i, TFT_BLUE);

  // Draw sin-, cos- and tan-lines
```

```

tft.setTextColor(TFT_CYAN);
tft.drawString("Sin", 5, 15,2);
for (int i=1; i<478; i++)
{
  tft.drawPixel(i,159+(sin(((i*1.13)*3.14)/180)*95),TFT_CYAN);
}

tft.setTextColor(TFT_RED);
tft.drawString("Cos", 5, 30,2);
for (int i=1; i<478; i++)
{
  tft.drawPixel(i,159+(cos(((i*1.13)*3.14)/180)*95),TFT_RED);
}

tft.setTextColor(TFT_YELLOW);
tft.drawString("Tan", 5, 45,2);
for (int i=1; i<478; i++)
{
  tft.drawPixel(i,159+(tan(((i*1.13)*3.14)/180)),TFT_YELLOW);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);
tft.drawLine(239, 15, 239, 304,TFT_BLUE);
tft.drawLine(1, 159, 478, 159,TFT_BLUE);

// Draw a moving sinewave
int col = 0;
x=1;
for (int i=1; i<(477*15); i++)
{
  x++;
  if (x==478)
    x=1;
  if (i>478)
  {
    if ((x==239) || (buf[x-1]==159))
      col = TFT_BLUE;
    else
      tft.drawPixel(x,buf[x-1],TFT_BLACK);
  }
  y=159+(sin(((i*0.7)*3.14)/180)*(90-(i / 100)));
  tft.drawPixel(x,y, TFT_BLUE);
  buf[x-1]=y;
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some filled rectangles
for (int i=1; i<6; i++)
{
  switch (i)
  {
    case 1:
      col = TFT_MAGENTA;
      break;
    case 2:
      col = TFT_RED;

```

```

    break;
case 3:
    col = TFT_GREEN;
    break;
case 4:
    col = TFT_BLUE;
    break;
case 5:
    col = TFT_YELLOW;
    break;
}
tft.fillRect(150+(i*20), 70+(i*20), 60, 60,col);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some filled, rounded rectangles
for (int i=1; i<6; i++)
{
    switch (i)
    {
    case 1:
        col = TFT_MAGENTA;
        break;
    case 2:
        col = TFT_RED;
        break;
    case 3:
        col = TFT_GREEN;
        break;
    case 4:
        col = TFT_BLUE;
        break;
    case 5:
        col = TFT_YELLOW;
        break;
    }
    tft.fillRoundRect(270-(i*20), 70+(i*20), 60, 60, 3, col);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some filled circles
for (int i=1; i<6; i++)
{
    switch (i)
    {
    case 1:
        col = TFT_MAGENTA;
        break;
    case 2:
        col = TFT_RED;
        break;
    case 3:
        col = TFT_GREEN;
        break;
    case 4:

```

```

    col = TFT_BLUE;
    break;
case 5:
    col = TFT_YELLOW;
    break;
}
tft.fillCircle(180+(i*20),100+(i*20), 30,col);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some lines in a pattern

for (int i=15; i<304; i+=5)
{
    tft.drawLine(1, i, (i*1.6)-10, 303, TFT_RED);
}

for (int i=304; i>15; i-=5)
{
    tft.drawLine(477, i, (i*1.6)-11, 15, TFT_RED);
}

for (int i=304; i>15; i-=5)
{
    tft.drawLine(1, i, 491-(i*1.6), 15, TFT_CYAN);
}

for (int i=15; i<304; i+=5)
{
    tft.drawLine(477, i, 490-(i*1.6), 303, TFT_CYAN);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some random circles
for (int i=0; i<100; i++)
{
    x=32+random(416);
    y=45+random(226);
    r=random(30);
    tft.drawCircle(x, y, r,random(0xFFFF));
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some random rectangles
for (int i=0; i<100; i++)
{
    x=2+random(476);
    y=16+random(289);
    x2=2+random(476);
    y2=16+random(289);
    if (x2<x) {
        r=x;x=x2;x2=r;
    }
}

```

```

}
if (y2<y) {
    r=y;y=y2;y2=r;
}
tft.drawRect(x, y, x2-x, y2-y,random(0xFFFF));
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

// Draw some random rounded rectangles
for (int i=0; i<100; i++)
{
    x=2+random(476);
    y=16+random(289);
    x2=2+random(476);
    y2=16+random(289);
    if (x2<x) {
        r=x;x=x2;x2=r;
    }
    if (y2<y) {
        r=y;y=y2;y2=r;
    }
    tft.drawRoundRect(x, y, x2-x, y2-y, 3,random(0xFFFF));
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

for (int i=0; i<100; i++)
{
    x=2+random(476);
    y=16+random(289);
    x2=2+random(476);
    y2=16+random(289);
    col=random(0xFFFF);
    tft.drawLine(x, y, x2, y2,col);
}

delay(WAIT);

tft.fillRect(1,15,478-1,304-15,TFT_BLACK);

for (int i=0; i<10000; i++)
{
    tft.drawPixel(2+random(476), 16+random(289),random(0xFFFF));
}

delay(WAIT);

tft.fillRect(0, 0, 480, 320, TFT_BLUE);

tft.fillRoundRect(160, 70, 319-160, 169-70, 3,TFT_RED);

tft.setTextColor(TFT_WHITE,TFT_RED);
tft.drawCentreString("That's it!", CENTRE, 93,2);
tft.drawCentreString("Restarting in a", CENTRE, 119, 2);
tft.drawCentreString("few seconds...", CENTRE, 132, 2);

```

```
tft.setTextColor(TFT_GREEN,TFT_BLUE);
tft.drawCentreString("Runtime: (msecs)", CENTRE, 280, 2);
tft.setTextDatum(TC_DATUM);
runTime = millis()-runTime;
tft.drawNumber(runTime, CENTRE, 300,2);
tft.setTextDatum(TL_DATUM);
delay (10000);
}
```