

```

// Paint example specifically for the TFTLCD breakout board.
// If using the Arduino shield, use the tftpaint_shield.pde sketch instead!
// DOES NOT CURRENTLY WORK ON ARDUINO LEONARDO

// I replace Magenta with Black

#include <Adafruit_GFX.h> // Core graphics library
#include <TftSpfd5408.h> // Hardware-specific library
#include <TouchScreen.h>

#if defined(__SAM3X8E__)
  #undef __FlashStringHelper::F(string_literal)
  #define F(string_literal) string_literal
#endif

// When using the BREAKOUT BOARD only, use these 8 data lines to the LCD:
// For the Arduino Uno, Duemilanove, Diecimila, etc.:
// D0 connects to digital pin 8 (Notice these are
// D1 connects to digital pin 9 NOT in order!)
// D2 connects to digital pin 2
// D3 connects to digital pin 3
// D4 connects to digital pin 4
// D5 connects to digital pin 5
// D6 connects to digital pin 6
// D7 connects to digital pin 7

// For the Arduino Mega, use digital pins 22 through 29
// (on the 2-row header at the end of the board).
// D0 connects to digital pin 22
// D1 connects to digital pin 23
// D2 connects to digital pin 24
// D3 connects to digital pin 25
// D4 connects to digital pin 26
// D5 connects to digital pin 27
// D6 connects to digital pin 28
// D7 connects to digital pin 29

// For the Arduino Due, use digital pins 33 through 40
// (on the 2-row header at the end of the board).
// D0 connects to digital pin 33
// D1 connects to digital pin 34
// D2 connects to digital pin 35
// D3 connects to digital pin 36
// D4 connects to digital pin 37
// D5 connects to digital pin 38
// D6 connects to digital pin 39
// D7 connects to digital pin 40

#define YP A3 // must be an analog pin, use "An" notation!
#define XM A2 // must be an analog pin, use "An" notation!
#define YM 9 // can be a digital pin
#define XP 8 // can be a digital pin

```

```

#define TS_MINX 150
#define TS_MINY 120
#define TS_MAXX 920
#define TS_MAXY 940

// For better pressure precision, we need to know the resistance
// between X+ and X- Use any multimeter to read it
// For the one we're using, its 300 ohms across the X plate
TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300);

#define LCD_CS A3
#define LCD_CD A2
#define LCD_WR A1
#define LCD_RD A0
// optional
#define LCD_RESET A4

// Assign human-readable names to some common 16-bit color values:
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

TftSpfd5408 tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);

#define BOXSIZe 40
#define PENRADIUS 3
int oldcolor, currentcolor;

void setup(void) {
  Serial.begin(9600);
  Serial.println(F("Paint!"));

  tft.reset();

  tft.begin(0x9341);

  tft.fillScreen(BLACK);

  tft.fillRect(0, 0, BOXSIZe, BOXSIZe, RED);
  tft.fillRect(BOXSIZe, 0, BOXSIZe, BOXSIZe, YELLOW);
  tft.fillRect(BOXSIZe*2, 0, BOXSIZe, BOXSIZe, GREEN);
  tft.fillRect(BOXSIZe*3, 0, BOXSIZe, BOXSIZe, CYAN);
  tft.fillRect(BOXSIZe*4, 0, BOXSIZe, BOXSIZe, BLUE);
  tft.fillRect(BOXSIZe*5, 0, BOXSIZe, BOXSIZe, BLACK);
  // tft.fillRect(BOXSIZe*6, 0, BOXSIZe, BOXSIZe, WHITE);

```

```

tft.drawRect(0, 0, BOXSIZE, BOXSIZE, WHITE);
currentcolor = RED;

pinMode(13, OUTPUT);
}

#define MINPRESSURE 10
#define MAXPRESSURE 1000

void loop()
{
  digitalWrite(13, HIGH);
  TSPoint p = ts.getPoint();
  digitalWrite(13, LOW);

  // if sharing pins, you'll need to fix the directions of the touchscreen pins
  //pinMode(XP, OUTPUT);
  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);
  //pinMode(YM, OUTPUT);

  // we have some minimum pressure we consider 'valid'
  // pressure of 0 means no pressing!

  if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {
    /*
    Serial.print("X = "); Serial.print(p.x);
    Serial.print("\tY = "); Serial.print(p.y);
    Serial.print("\tPressure = "); Serial.println(p.z);
    */

    if (p.y < (TS_MINY-5)) {
      Serial.println("erase");
      // press the bottom of the screen to erase
      tft.fillRect(0, BOXSIZE, tft.width(), tft.height()-BOXSIZE, BLACK);
    }
    // scale from 0->1023 to tft.width
    p.x = map(p.x, TS_MINX, TS_MAXX, tft.width(), 0);
    p.y = map(p.y, TS_MINY, TS_MAXY, tft.height(), 0);
    /*
    Serial.print("("); Serial.print(p.x);
    Serial.print(", "); Serial.print(p.y);
    Serial.println(")");
    */
    if (p.y < BOXSIZE) {
      oldcolor = currentcolor;

      if (p.x < BOXSIZE) {
        currentcolor = RED;
        tft.drawRect(0, 0, BOXSIZE, BOXSIZE, WHITE);
      } else if (p.x < BOXSIZE*2) {

```

```

    currentcolor = YELLOW;
    tft.drawRect(BOXSIZE, 0, BOXSIZE, BOXSIZE, WHITE);
} else if (p.x < BOXSIZE*3) {
    currentcolor = GREEN;
    tft.drawRect(BOXSIZE*2, 0, BOXSIZE, BOXSIZE, WHITE);
} else if (p.x < BOXSIZE*4) {
    currentcolor = CYAN;
    tft.drawRect(BOXSIZE*3, 0, BOXSIZE, BOXSIZE, WHITE);
} else if (p.x < BOXSIZE*5) {
    currentcolor = BLUE;
    tft.drawRect(BOXSIZE*4, 0, BOXSIZE, BOXSIZE, WHITE);
} else if (p.x < BOXSIZE*6) {
    currentcolor = BLACK;
    tft.drawRect(BOXSIZE*5, 0, BOXSIZE, BOXSIZE, WHITE);
}

if (oldcolor != currentcolor) {
    if (oldcolor == RED) tft.fillRect(0, 0, BOXSIZE, BOXSIZE, RED);
    if (oldcolor == YELLOW) tft.fillRect(BOXSIZE, 0, BOXSIZE, BOXSIZE, YELLOW);
    if (oldcolor == GREEN) tft.fillRect(BOXSIZE*2, 0, BOXSIZE, BOXSIZE, GREEN);
    if (oldcolor == CYAN) tft.fillRect(BOXSIZE*3, 0, BOXSIZE, BOXSIZE, CYAN);
    if (oldcolor == BLUE) tft.fillRect(BOXSIZE*4, 0, BOXSIZE, BOXSIZE, BLUE);
    if (oldcolor == BLACK) tft.fillRect(BOXSIZE*5, 0, BOXSIZE, BOXSIZE, BLACK);
}
}
if (((p.y-PENRADIUS) > BOXSIZE) && ((p.y+PENRADIUS) < tft.height())) {
    tft.fillCircle(p.x, p.y, PENRADIUS, currentcolor);
}
}
}
}

```