CALCULATOR

DISPLAY UNIT 2.8” TFT LCD ILI9341

Download following A,B,&,C 3 Libraries on your desktop Folder name “TFT LCD”

(A) Adafruit GFX.h Library
http://www.areptone.in/arduino/arduino-soft/Adafruit_GFX_Library-1.4.7.zip
(B) Adafruit TFTLCD.h Library
http://www.areptone.in/arduinoi/arduino-soft/TFTLCD-Library-master.zip
(C) Adafruit TOUCH SCREEN.h Library
http://www.areptone.in/arduino/arduino-soft/Adafruit_TouchScreen-master.zip

TFT LCD ILI9341 DRIVER
http://www.areptone.in/arduino/arduino-soft/Adafruit_ILI9341-master.zip
http://www.areptone.in/arduino/arduino-soft/CALCULATOR.rar

RUN ARDUINO IDE Software On your PC
1 Go to Sketch Menu & Click Mouse Button
2 Go to Include Library Go to Add ZIP Library
3 Then Click Enter Button
4 Find These Libraries on your desktop TFT LCD Folder CKECKE all 3 LIBRARIES
5 ADDED to Adruino IDE Software
6 On your PC Go to Sketch Menu & Click Mouse Button
7 Go to Include Library then right side Then Click UP/DOWN Button
Find These Libraries on your list

NOW DOWNLOAD CALCULATOR Sketch
http://www.areptone.in/arduino/calculator.pdf
http://www.areptone.in/arduino/arduino-soft/CALCULATOR.rar
SCAKAGE FOR TFT LCD 2.8” INTERFACING WITH ARDUINO UNO
Most important part is this identifier 9341 number of TFT LCD

CALCULATOR SKETCH

```
// Arduino TFT LCD 2.8" DISPLAY UNIT
// CALCULATOR Design By Mr. Adeeb Raza for Arduino UNO
// USB Socket on Top Side Of Calculator
// Row Counting from Right to Left
//*************** 25-03-2019 ****************************
//*************************************************
#include <Adafruit_GFX.h>    // Core graphics library
#include <Adafruit_TFTLCD.h> // Hardware-specific library
#include <TouchScreen.h>

#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 140
#define TS_MINY 120
#define TS_MAXX 910
#define TS_MAXY 920

// 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
#define LCD_RESET A4

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

int32_t answer = 0;
char lastchar = ' ';
char keyarray1[200];
char keyarray2[200];
int32_t keyarray1i = 0;
int32_t keyarray2i = 0;
char operation = ' ';
long Time = 0;
long millicount = 0;
int interval = 1000;
int screenTimout = 15;

// Assign human-readable names to some common 16-bit color values:
#define BLACK   0x0000
int32_t BLUE = tft.color565(50, 50, 255);
#define DARKBLUE 0x0010
#define VIOLET 0x8888
#define RED     0xF800
#define GREEN   0x07E0
#define CYAN    0x07FF
#define MAGENTA 0xF81F
#define YELLOW  0xFFE0
#define WHITE   0xFFFF
#define GREY   tft.color565(64, 64, 64);
#define GOLD 0xFEA0
#define BROWN 0xA145
#define SILVER 0xC618
#define LIME 0x07E0
#define ORANGE 0xFD20
#define ORANGERED 0xFA20
```

```
#define minpressure 10
#define maxpressure 1000

#define row1x 0
#define box1y 260
#define boxsize 60

#define r1x 190
#define b1y 55

int32_t tftheight = 0;
int32_t tftwidth = 0;
int32_t extray = 43;
int32_t x, y = 0;

char button[4][4] = {
 { '7', '8', '9', '/' },
 { '4', '5', '6', '*' },
 { '1', '2', '3', '-' },
 { 'C', '0', '=', '+' }

};

void draw()
{
 tft.fillScreen(BLACK);

 tft.drawRoundRect(row1x, box1y, boxsize, boxsize, 8, RED);
 tft.drawRoundRect(row1x, box1y - boxsize, boxsize, boxsize, 8, WHITE);
 tft.drawRoundRect(row1x, box1y - boxsize * 2, boxsize, boxsize, 8, WHITE);
 tft.drawRoundRect(row1x, box1y - boxsize * 3, boxsize, boxsize, 8, WHITE);

 for (int32_t b = box1y; b >= 80; b -= boxsize)
 { tft.drawRoundRect  (180, b, boxsize, boxsize, 8, BLUE);
  tft.drawRoundRect  (row1x + boxsize, b, boxsize, boxsize, 8, WHITE);
 }
 tft.drawRoundRect(row1x + boxsize * 2, box1y, boxsize, boxsize, 8, GREEN);
 tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize, boxsize, boxsize, 8, WHITE);
 tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 2, boxsize, boxsize, 8, WHITE);
 tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 3, boxsize, boxsize, 8, WHITE);

 for (int32_t j = 0; j < 4; j++) {
  for (int32_t i = 0; i < 4; i++) {
   tft.setCursor(22 + (boxsize * i), 100 + (boxsize * j));
   tft.setTextSize(3);
   tft.setTextColor(WHITE);
   tft.println(button[j][i]);
  }
 }
 tft.drawRoundRect(2, 5, 235, 70, 8, ORANGERED);
}
```

```
void drawintro()
{
 tft.fillScreen(BLACK);
 tft.setTextSize(3);
 tft.drawRoundRect(0, 0, 240, 319, 20, RED);

 tft.setTextColor(BLUE);
 tft.setCursor(30, 100);
 tft.print("Welcome To");
 tft.setCursor(30, 130);
 tft.setTextColor(YELLOW);
 tft.print("Areptone");
 tft.setCursor(30, 160);
 tft.setTextColor(RED);
 tft.print("CALCULATOR");
 tft.setCursor(45, 250);
 tft.setTextColor(WHITE);
 tft.print("Touch To");
 tft.setCursor(45, 280);
 tft.print("Continue");
 tft.setCursor(30, 50);
 tft.fillRoundRect(20, 40, 200, 40, 8, YELLOW);
 tft.setTextColor(RED);
 tft.print("Adeeb Raza");
 tft.fillRoundRect(25, 215, 180, 5, 8, GREEN);

 digitalWrite(13, HIGH);
 TSPoint p = ts.getPoint();     // Read touchscreen
 digitalWrite(13, LOW);

 pinMode(XM, OUTPUT);
 pinMode(YP, OUTPUT);

 int i = 1;

 while (i == 1)
 {
  digitalWrite(13, HIGH);
  TSPoint p = ts.getPoint();     // Read touchscreen
  digitalWrite(13, LOW);

  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  if (p.z > minpressure && p.z < maxpressure)
  {
   return;
  }
 }
}
```

```
void screenSaver()
{
  digitalWrite(13, HIGH);
  TSPoint p = ts.getPoint();     // Read touchscreen
  digitalWrite(13, LOW);
  //
  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  tft.fillScreen(BLACK);
  tft.setCursor(30, 100);
  tft.print("HAPPY HOLI");
  tft.setCursor(10, 150);
  tft.print("TO ALL OF");
  tft.setCursor(50, 200);
  tft.print("YOU");
  delay(1000);
  tft.fillScreen(RED);

  int i = 0;
  while (i == 0)
  {
    for (int i = 0; i < 150; i++)
    {
      int color = tft.color565(random(255), random(255), random(255));
      int x = random(350);
      int y = random(350);
      int r = random(30);

      tft.fillCircle(x, y, r, color);

      TSPoint p = ts.getPoint();     // Read touchscreen
      digitalWrite(13, LOW);
      //
      pinMode(XM, OUTPUT);
      pinMode(YP, OUTPUT);

      if (p.z > minpressure && p.x < 1000)
      {
        draw();
        tft.setCursor(10, 30);
        if (keyarray1[0] != '\0')
        {
          tft.print(keyarray1);

          if (operation != ' ')
          {
            tft.print(operation);

            if (keyarray2[0] != '\0')
```

```
          {
            tft.print(keyarray2);
          }
         }
        }
        else
         tft.setCursor(10, 30);
        return;
      }
    }
    tft.fillScreen(tft.color565(random(255), random(255), random(255)));
  }
}


void setup()
{
  keyarray1[0] = '\0';
  keyarray2[0] = '\0';

  Serial.begin(9600);
  Serial.println("Calculator");
  tft.begin(0x9341);
  tft.setRotation(2);
  drawintro();
  tftheight = tft.height();
  tftwidth = tft.width();
  draw();
  tft.setCursor(10, 30);
}

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

  pinMode(XM, OUTPUT);
  pinMode(YP, OUTPUT);

  x = map(p.x, TS_MINX, TS_MAXX, tft.width(), 0);
  y = map(p.y, TS_MINY, TS_MAXY,  tft.height(), 0) + 30;

  if ((millis() - millicount) > interval)
  {
    millicount = millis();
    Time += 1;
  }
  if (Time >= screenTimout && (p.z < 10 || p.z > 1000))
  {
    screenSaver();
```

```
  Time = 0;
 }

 if (p.z > 10 && p.z < 1000)
 {
  Time = 0;
  millicount = millis();
 }

 if (p.z > minpressure && p.z < maxpressure)
 {

  lastchar = ' ';
  lastchar = idbutton();

  if (lastchar >= '0' && lastchar <= '9')
  {
   if (operation == ' ')
   {
    addchar(keyarray1, lastchar);
    tft.print(lastchar);
    Serial.println(keyarray1);
   }

   else if (operation != ' ')
   {
    addchar(keyarray2, lastchar);
    tft.print(lastchar);
    Serial.println(keyarray2);
   }
  }

  if (lastchar == '+' || lastchar == '-' || lastchar == '/' || lastchar == '*' && keyarray2[0] == '\0' &&
keyarray1[0] != '\0')
  {
   if ( operation != ' ')
   {
    operation = lastchar;
    tft.fillRoundRect(2, 5, 235, 70, 8, BLACK);
    tft.drawRoundRect(2, 5, 235, 70, 8, ORANGERED);
    tft.setCursor(10, 30);
    tft.print(keyarray1);
    tft.print(operation);
   }
   else
   {
    if (operation == ' ')
    {
     operation = lastchar;
     tft.print(operation);
    }
```

```
  }
}

if (lastchar == 'C')
{
  keyarray1[0] = '\0';
  keyarray2[0] = '\0';
  answer = 0;

  keyarray1i = 0;
  keyarray2i = 0;

  operation = ' ';
  draw();
  tft.fillRoundRect(2, 5, 235, 70, 8, BLACK);
  tft.drawRoundRect(2, 5, 235, 70, 8, ORANGERED);
  tft.setCursor(10, 30);
}

if (lastchar == '=')
{
  if (keyarray1[0] != '\0' && keyarray2[0] != '\0')
  {
    Serial.println("Calculate");
    if (keyarray2[0] == '0' && operation == '/')
    {
      tft.setCursor(10, 30);
      tft.fillRoundRect(2, 5, 235, 70, 8, BLACK);
      tft.drawRoundRect(2, 5, 235, 70, 8, ORANGERED);
      tft.print("=");
      tft.setTextColor(RED);
      tft.print("ERROR");
      tft.setTextColor(WHITE);
      keyarray1[0] = '\0';
      keyarray2[0] = '\0';
      operation = ' ';
      return;
    }
    keyarray1i = 0;
    keyarray2i = 0;
    keyarray1i = chararraytoint(keyarray1);
    keyarray2i = chararraytoint(keyarray2);

    answer = calc(keyarray1i, keyarray2i, operation);

    tft.fillRoundRect(2, 5, 235, 70, 8, BLACK);
    tft.drawRoundRect(2, 5, 235, 70, 8, ORANGERED);
    tft.setCursor(10, 30);
    tft.print('=');
    tft.print(answer);
    keyarray1i = 0;
```

```
      operation = ' ';
      keyarray2i = 0;
     }
    }
    delay(500);
  }
}

char idbutton()
{
  //Row 1 identification
  //******************** Right To Left Calculator 1St Row ***************************
  if ((tftwidth >= x) && (x >= (tftwidth - boxsize)))
  {
    //    Serial.println("Row 1  ");
    //+
    if (((extray + boxsize) >= y) && (y >= extray))
    {
     tft.drawRoundRect(row1x + boxsize * 3, box1y, boxsize, boxsize, 8, GREEN);
     delay(100);
     tft.drawRoundRect(row1x + boxsize * 3, box1y, boxsize, boxsize, 8, BLUE);
     return '+';
    }
    //-
    if (((extray + (boxsize * 2)) >= y) && (y >= (extray + boxsize)))
    {
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize, boxsize, boxsize, 8, GREEN);
     delay(100);
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize, boxsize, boxsize, 8, BLUE);
     return '-';
    }
    //*
    if (((extray + (boxsize * 3)) >= y) && (y >= (extray + (boxsize * 2))))
    {
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize * 2, boxsize, boxsize, 8, GREEN);
     delay(100);
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize * 2, boxsize, boxsize, 8, BLUE);
     return '*';
   }
    // /
    if (((extray + (boxsize * 4)) >= y) && (y >= (extray + (boxsize * 3))))
    {
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize * 3, boxsize, boxsize, 8, GREEN);
     delay(100);
     tft.drawRoundRect(row1x + boxsize * 3, box1y - boxsize * 3, boxsize, boxsize, 8, BLUE);
     return '/';

    }
  }

  //Row 2 identification
```

```
//***************** Right To Left Calculator 2nd Row ****************************
if (((tftwidth - boxsize) >= x) && (x >= (tftwidth - (boxsize * 2))))
{
 //    Serial.print32_tln("Row 2  ");
 //=
 if (((extray + boxsize) >= y) && (y >= extray))
 {
  tft.drawRoundRect(row1x + boxsize * 2, box1y, boxsize, boxsize, 8, BLUE);
  delay(100);
  tft.drawRoundRect(row1x + boxsize * 2, box1y, boxsize, boxsize, 8, GREEN);
  return '=';
 }
 //3
 if (((extray + (boxsize * 2)) >= y) && (y >= (extray + boxsize)))
 {
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize, boxsize, boxsize, 8, WHITE);
  return '3';
 }
 //6
 if (((extray + (boxsize * 3)) >= y) && (y >= (extray + (boxsize * 2))))
 {
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 2, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 2, boxsize, boxsize, 8, WHITE);
  return '6';
 }
 //9
 if (((extray + (boxsize * 4)) >= y) && (y >= (extray + (boxsize * 3))))
 {
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 3, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize * 2, box1y - boxsize * 3, boxsize, boxsize, 8, WHITE);
  return '9';
 }
}

//Row 3 identification
//*************** Right To Left Calculator 3rd Row ******************************
if (((tftwidth - boxsize * 2) >= x) && (x >= (tftwidth - (boxsize * 3))))
{
 //    Serial.print32_tln("Row 3  ");
 //0
 if (((extray + boxsize) >= y) && (y >= extray))
 {
  tft.drawRoundRect(row1x + boxsize, box1y, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize, box1y, boxsize, boxsize, 8, WHITE);
  return '0';
 }
```

```
 //2
 if (((extray + (boxsize * 2)) >= y) && (y >= (extray + boxsize)))
 {
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize, boxsize, boxsize, 8, WHITE);
  return '2';
 }
 //5
 if (((extray + (boxsize * 3)) >= y) && (y >= (extray + (boxsize * 2))))
 {
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize * 2, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize * 2, boxsize, boxsize, 8, WHITE);
  return '5';
 }
 //8
 if (((extray + (boxsize * 4)) >= y) && (y >= (extray + (boxsize * 3))))
 {
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize * 3, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x + boxsize, box1y - boxsize * 3, boxsize, boxsize, 8, WHITE);
  return '8';
 }
}

//Row 4 identification
//*************** Right To Left Calculator 4th Row ******************************
if (((tftwidth - boxsize * 3) >= x) && (x >= (tftwidth - (boxsize * 4))))
{
 //   Serial.print32_tln("Row 4  ");
 //C
 if (((extray + boxsize) >= y) && (y >= extray))
 {
  tft.drawRoundRect(row1x, box1y, boxsize, boxsize, 8, WHITE);
  delay(100);
  tft.drawRoundRect(row1x, box1y, boxsize, boxsize, 8, RED);
  return 'C';
 }
 //1
 if (((extray + (boxsize * 2)) >= y) && (y >= (extray + boxsize)))
 {
  tft.drawRoundRect(row1x, box1y - boxsize, boxsize, boxsize, 8, RED);
  delay(100);
  tft.drawRoundRect(row1x, box1y - boxsize, boxsize, boxsize, 8, WHITE);
  return '1';
 }
 //4
 if (((extray + (boxsize * 3)) >= y) && (y >= (extray + (boxsize * 2))))
 {
  tft.drawRoundRect(row1x, box1y - boxsize * 2, boxsize, boxsize, 8, RED);
```

```
     delay(100);
     tft.drawRoundRect(row1x, box1y - boxsize * 2, boxsize, boxsize, 8, WHITE);
     return '4';
    }
    // 7
    if (((extray + (boxsize * 4)) >= y) && (y >= (extray + (boxsize * 3))))
    {
     tft.drawRoundRect(row1x, box1y - boxsize * 3, boxsize, boxsize, 8, RED);
     delay(100);
     tft.drawRoundRect(row1x, box1y - boxsize * 3, boxsize, boxsize, 8, WHITE);
     return '7';
    }
  }
}

void addchar(char *array1, char inchar)
{
  int32_t input_str_len = strlen(array1);  //get current length of string
  array1[input_str_len]   = inchar;  //assign received character to the last position in the string
  array1[input_str_len + 1] = '\0';
}

int32_t chartoint(char num)
{

  return num - '0';
}

int32_t calc(int32_t num1, int32_t num2, char op)
{
  if (op == '+')
  {
    Serial.print(num1);
    Serial.print("   ");
    Serial.print(num2);
    return (num1 + num2);
  }
  if (op == '-')
  {
    return (num1 - num2);
  }
  if (op == '*')
  {
    return (num1 * num2);
  }
  if (op == '/')
  {
    return (num1 / num2);
  }
}
```

```
int32_t chararraytoint(char *a)
{
 int32_t len = strlen(a);
 int32_t finalnum = 0;
 int32_t intval = 0;
 int32_t placeval = 1;

 if (len > 0)
 {
  for (int i = len - 1; i >= 0; i--)
  {
   intval = chartoint(a[i]);
   intval = intval * placeval;
   finalnum = finalnum + intval;
   placeval *= 10;
  }
 }
 return finalnum;
```

}}END of Programme

Copy code from here or download from link

http://www.areptone.in/arduino/arduino-soft/CALCULATOR.rar

http://www.areptone.in/arduino/

OUTPUT

(By Mr. Adeeb Raza)