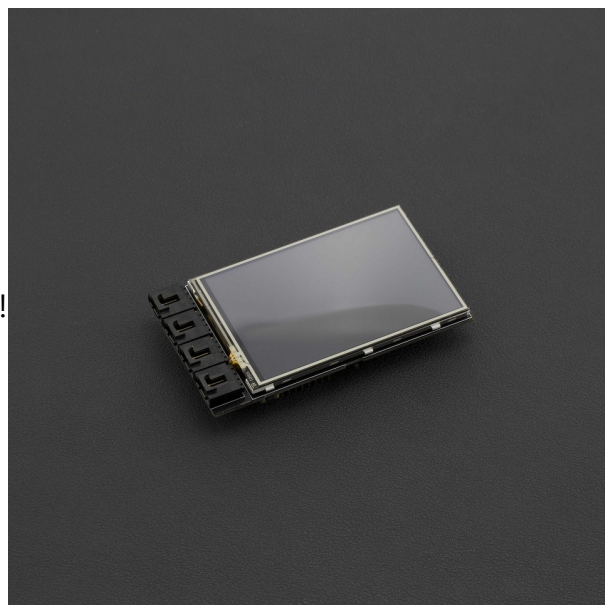


# TELEMATICS 3.5 TFT Touch LCD Shield SKU DFR0387



(<https://www.dfrobot.com/product-1500.html>)

## Introduction

---

For beginners to Arduino it can be daunting and risky to wire LCD driver circuits - one wrong connection and your component can be damaged. This LCD expansion board removes all the complications and risks. The TELEMATICS 3.5" TFT Touch LCD Shield is an Arduino compatible display designed by DFRobot, with a resolution of 320x480, three serial ports and one IIC interface. It is perfectly compatible with Arduino DUE, Mega1280/2560, and Bluno Mega1280/2560. There is an on-board voltage switch which supports changing the output voltage between 3.3V and 5V to ensure that it won't damage your DUE. In addition, the LCD shield comes with a MicroSD card slot on the back that can be used for data storage up to a maximum of 32GB. We also offer a variety of driver packages to help you implement different features.



**NOTE:** Since the operating voltage of DUE and MEGA are different, the backlight brightness will change if you replace your Arduino platform. Please adjust the D8 PWM signal to control the brightness of the backlight.

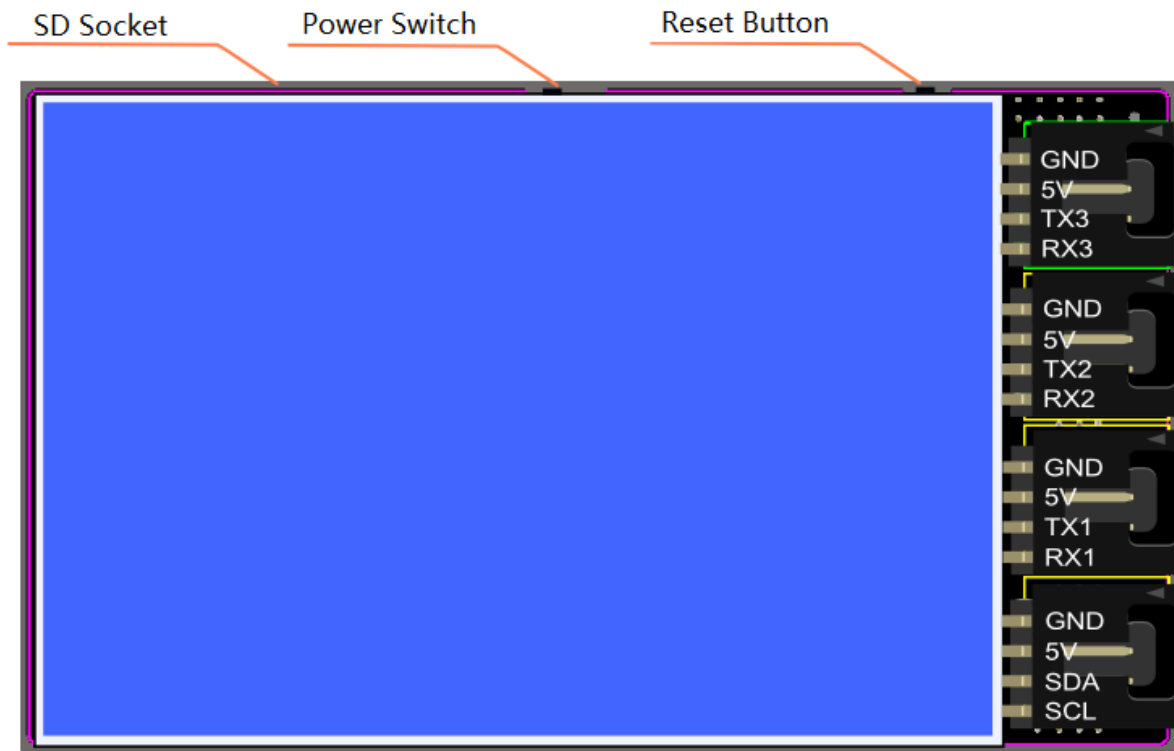
## Specification

---

- Screen Size: 3.5"
- Resolution Ratio: 320x480
- Power Supply: 5V
- Output Voltage: 3.3/5V
- Backlight Control Mode: D8 PWM signal
- Supports MicroSD card (Up to 32GB)
- Serial Ports: 3
- IIC interface: 1
- Compatible with Arduino DUE/Mega 1280/2560
- Compatible with the Bluno Mega 1280/2560
- Supports Touch Functionality
- Dimensions: 100x57mm/3.93x2.24 inches
- Weight: 70g

## Board Overview

---



Interface	Description
IIC Interface	IIC (SDA: 20; SCL: 21)
Serial1	Serial1
Serial2	Serial2
Serial3	Serial3
SD Driving Pins	D52->DATA0 D50->CMD D52->CLK D53->CD/DATA3

**Note:**

LCD Driving Pins: D2, D3, D4, D5, D6, D8, D22~D41, D50~D53

# Tutorial

---

## Requirements

- **Hardware**
  - Bluno Mega 2560 x 1
  - TELEMATICS 3.5" TFT Touch LCD Shield x 1
- **Software**
  - Arduino IDE (1.0.x) Click to Download Arduino IDE from Arduino® (<https://www.arduino.cc/en/Main/Software>)
  - Download related Libraries. ([https://github.com/Arduinolibrary/DFRobot\\_TELEMATICS\\_3.5-\\_TFT\\_Touch\\_LCD\\_Shield/raw/master/Arduino%20Libraries.zip](https://github.com/Arduinolibrary/DFRobot_TELEMATICS_3.5-_TFT_Touch_LCD_Shield/raw/master/Arduino%20Libraries.zip)) How to install libraries (<https://www.arduino.cc/en/Guide/Libraries#.UxU8mdzF9H0>) (Note: This library only works on Classic Arduino IDE 1.0.x)

## Sample Code 1

In this section, we'll explain how to initialize the LCD screen. You can use "**setFontSize**" and "**setColor**" function to change the font and the color of the characters.

```
/*  
This code will demonstrate how to make the LCD display string with the library  
If you need to set the font size, you can call this function "setFontSize()",and you can set the parameters:
```

```
    FONT_SIZE_SMALL  
    FONT_SIZE_MEDIUM  
    FONT_SIZE_LARGE  
    FONT_SIZE_XLARGE
```

```
If you want to set the font color, you can call this function "setColor()" with the parameters:
```

```
    RGB16_RED----->RED  
    RGB16_GREEN----->GREEN  
    RGB16_BLUE----->BLUE  
    RGB16_YELLOW----->YELLOW  
    RGB16_CYAN----->CYAN  
    RGB16_PINK----->PINK  
    RGB16_WHITE----->WHITE
```

```
Created 2016-4-8
```

```
By Andy zhou <Andy.zhou@dfrobot.com>
```

```
version:V1.0
```

```
*/
```

```
#include <Arduino.h>  
#include <SPI.h>  
#include <MultiLCD.h>
```

```
LCD_R61581 lcd;
```

```
void setup(){  
    lcd.begin();  
    lcd.setFontSize(FONT_SIZE_MEDIUM); //set font size  
    lcd.setColor(RGB16_RED); //set strings color  
    lcd.print(" ");
```

```
lcd.println();  
lcd.println();  
lcd.println("DFRobot!!!");  
lcd.println("TELEMATICS LCD SHIELD V1.0");  
lcd.println();  
  
lcd.setColor(RGB16_WHITE);  
}  
  
void loop(){  
}
```

## Sample Code 2

In this section, we'll explain how to use the SD card with LCD screen.

```
/*
/*
This code is to teach you how to use SD library.
```

```
Created 2016-4-8
```

```
By Andy zhou <Andy.zhou@dfrobot.com>
```

```
version:V1.0
```

```
*/
```

```
#include <Arduino.h>
#include <SPI.h>
#include <MultiLCD.h>
#include <SD.h>
#include "datalogger.h"
```

```
#define STATE_SD_READY 0x1
#define STATE_OBD_READY 0x2
#define STATE_GPS_CONNECTED 0x4
#define STATE_GPS_READY 0x8
#define STATE_MEMS_READY 0x10
#define STATE_GUI_ON 0x20
```

```
LCD_R61581 lcd;
CDataLogger logger;
```

```
byte state = 0;
```

```
bool checkSD()
```

```
{
    Sd2Card card;
    SdVolume volume;
    state &= ~STATE_SD_READY;
    pinMode(CC_OUTPUT,

```

```

pinMode(SS, OUTPUT);

lcd.setFontSize(FONT_SIZE_MEDIUM);
if (card.init(SPI_HALF_SPEED, SD_CS_PIN)) {
    const char* type;

    switch(card.type()) {
    case SD_CARD_TYPE_SD1:
        type = "SD1";
        break;
    case SD_CARD_TYPE_SD2:
        type = "SD2";
        break;
    case SD_CARD_TYPE_SDHC:
        type = "SDHC";
        break;
    default:
        type = "SDx";
    }

    lcd.print(type);
    lcd.write(' ');
    if (!volume.init(card)) {
        lcd.print("No FAT!");
        return false;
    }

    uint32_t volumesize = volume.blocksPerCluster();
    volumesize >>= 1; // 512 bytes per block
    volumesize *= volume.clusterCount();
    volumesize >>= 10;

    lcd.print((int)volumesize);
    lcd.print("MB");
} else {
    lcd.println("No SD Card");
    return false;
}

```



```

    if (!SD.begin(SD_CS_PIN)) {
        lcd.println("Bad SD");
        return false;
    }

    state |= STATE_SD_READY;
    return true;
}

void setup(){
    lcd.begin();
    lcd.setFontSize(FONT_SIZE_MEDIUM); //set font size
    lcd.setColor(RGB16_RED); //set strings color
    lcd.println();
    lcd.println();
    lcd.println("DFRobot!!!");
    lcd.println("TELEMATICS LCD SHIELD V1.0");
    lcd.println();
    lcd.setColor(RGB16_WHITE);
    if (checkSD()) {
        uint16_t index = logger.openFile();
        lcd.println();
        if (index > 0) {
            lcd.print("File ID:");
            lcd.println(index);
        } else {
            lcd.print("No File");
        }
    }
}

void loop(){
}

```

### Sample Code 3

In this section, we'll explain how to use the touch function

```
#include <Arduino.h>
#include <SPI.h>
#include <MultiLCD.h>
#include "touch.h"

LCD_R61581 lcd;

void setup(){
  lcd.begin();
  lcd.setFontSize(FONT_SIZE_MEDIUM); //set font size
  lcd.setColor(RGB16_YELLOW); //set strings color
  lcd.println("DFRobot");
  lcd.println("TELEMATICS LCD SHIELD V1.0");
  lcd.println();
  lcd.setColor(RGB16_WHITE);

  touch.init();
}

void loop(){
  lcd.setCursor(0, 8);
  int x, y;
  if ( touch.read(x, y) ){
    lcd.print("X:");
    lcd.print(x);
    lcd.print(" Y:");
    lcd.print(y);
    lcd.print(" ");
  } else {
    lcd.print("          ");
  }
}
```

## Sample Code 4

Display a picture "ladar.bmp" from SD card.

```

#include <Arduino.h>
#include <SPI.h>
#include <MultiLCD.h>
#include <SD.h>
#include "datalogger.h"

#define STATE_SD_READY 0x1
#define STATE_OBD_READY 0x2
#define STATE_GPS_CONNECTED 0x4
#define STATE_GPS_READY 0x8
#define STATE_MEMS_READY 0x10
#define STATE_GUI_ON 0x20

LCD_R61581 lcd;
CDataLogger logger;

byte state = 0;

// These read 16- and 32-bit types from the SD card file.
// BMP data is stored little-endian, Arduino is little-endian too.
// May need to reverse subscript order if porting elsewhere.

uint16_t read16(File & f) {
    uint16_t result;
    ((uint8_t *)&result)[0] = f.read(); // LSB
    ((uint8_t *)&result)[1] = f.read(); // MSB
    return result;
}

uint32_t read32(File & f) {
    uint32_t result;

```

```

uint32_t result;
((uint8_t *)&result)[0] = f.read(); // LSB
((uint8_t *)&result)[1] = f.read();
((uint8_t *)&result)[2] = f.read();
((uint8_t *)&result)[3] = f.read(); // MSB

return result;
}

bool checkSD()
{
    Sd2Card card;
    SdVolume volume;
    state &= ~STATE_SD_READY;
    pinMode(SS, OUTPUT);

    lcd.setFontSize(FONT_SIZE_MEDIUM);
    if (card.init(SPI_FULL_SPEED, SD_CS_PIN)) {
        const char* type;
        switch(card.type()) {
            case SD_CARD_TYPE_SD1:
                type = "SD1";
                break;
            case SD_CARD_TYPE_SD2:
                type = "SD2";
                break;
            case SD_CARD_TYPE_SDHC:
                type = "SDHC";
                break;
            default:
                type = "SDx";
        }

        lcd.print(type);
        lcd.write(' ');
        if (!volume.init(card)) {
            lcd.print("No FAT!");
            return false;
        }
    }
}

```

```

    }

    uint32_t volumsize = volume.blocksPerCluster();
    volumsize >>= 1; // 512 bytes per block
    volumsize *= volume.clusterCount();

    volumsize >>= 10;

    lcd.print((int)volumsize);
    lcd.print("MB\n");
} else {
    lcd.println("No SD Card");
    return false;
}

if (!SD.begin(SD_CS_PIN)) {
    lcd.println("Bad SD");
    return false;
}

state |= STATE_SD_READY;
return true;
}

uint16_t Color565(uint8_t r, uint8_t g, uint8_t b) {
    return ((r & 0xF8) << 8) | ((g & 0xFC) << 3) | (b >> 3);
}

#define BUFFPIXEL 20

void bmpDraw(char *fileName, uint16_t x, uint16_t y){
    File    bmpFile;
    int     bmpWidth, bmpHeight;   // W+H in pixels
    uint8_t bmpDepth;              // Bit depth (currently must be 24)
    uint32_t bmpImageoffset;      // Start of image data in file
    uint32_t rowSize;              // Not always = bmpWidth; may have padding
    uint8_t  sdbuffer[3*BUFFPIXEL]; // pixel buffer (R+G+B per pixel)
    uint8_t  buffidx = sizeof(sdbuffer); // Current position in sdbuffer

```

```

boolean goodBmp = false;      // Set to true on valid header parse
boolean flip    = true;      // BMP is stored bottom-to-top
int          w, h, row, col;
uint8_t      r, g, b;
uint32_t     pos = 0, startTime = millis();

if((x >= 480) || (y >= 320))
    return;

if((bmpFile = SD.open(fileName)) == NULL){
    lcd.println("File not found");
    return;
}

if(read16(bmpFile) == 0x4d42){
    Serial.print("File size:");
    Serial.println(read32(bmpFile));
    (void)read32(bmpFile);
    bmpImageoffset = read32(bmpFile); // Start of image data
    Serial.print("Image Offset: ");
    Serial.println(bmpImageoffset, DEC);
    // Read DIB header
    Serial.print("Header size: ");
    Serial.println(read32(bmpFile));
    bmpWidth  = read32(bmpFile);
    bmpHeight = read32(bmpFile);

    if(read16(bmpFile) == 1){
        bmpDepth = read16(bmpFile);
        Serial.print("Bit Depth: ");
        Serial.println(bmpDepth);
        if((bmpDepth == 24) && (read32(bmpFile) == 0)) { // 0 = uncompressed 没有压缩
            goodBmp = true; // Supported BMP format -- proceed!
            Serial.print("Image size: ");
            Serial.print(bmpWidth);
            Serial.print('x');
            Serial.println(bmpHeight);
        }
    }
}

```

```

// BMP rows are padded (if needed) to 4-byte boundary
rowSize = (bmpWidth * 3 + 3) & ~3;

if(bmpHeight < 0) {
    bmpHeight = -bmpHeight;
    flip      = false;
}

// Crop area to be loaded
w = bmpWidth;
h = bmpHeight;

if((x+w-1) >= 480)
    w = 480 - x;
if((y+h-1) >= 320)
    h = 320 - y;

for (row=0; row<h; row++) { // For each scanline...
    if(flip) // Bitmap is stored bottom-to-top order (normal BMP)
        pos = bmpImageoffset + (bmpHeight - 1 - row) * rowSize;
    else // Bitmap is stored top-to-bottom
        pos = bmpImageoffset + row * rowSize;

    if(bmpFile.position() != pos) { // Need seek?
        bmpFile.seek(pos);
        buffidx = sizeof(sdbuffer); // Force buffer reload
    }

    for (col=0; col<w; col++) { // For each pixel...
        // Time to read more pixel data?
        if (buffidx >= sizeof(sdbuffer)) { // Indeed
            bmpFile.read(sdbuffer, sizeof(sdbuffer));
            buffidx = 0; // Set index to beginning
        }

        // Convert pixel from BMP to TFT format, push to display

```



```

        b = sdbuffer[buttidx++];
        g = sdbuffer[buffidx++];
        r = sdbuffer[buffidx++];
        lcd.drawPixel(col + x, row + y, Color565(r,g,b));
    } // end pixel

} // end scanline
Serial.print("Loaded in ");
Serial.print(millis() - startTime);
Serial.println(" ms");
}
}
}
bmpFile.close();
if(!goodBmp)
    Serial.println("BMP format not recognized.");

}

static uint32_t startTime = 0;

void dataIdleLoop(){
    // called while initializing
    char buf[10];
    unsigned int t = (millis() - startTime) / 1000;

    sprintf(buf, "%02u:%02u", t / 60, t % 60);
    lcd.setFontSize(FONT_SIZE_XLARGE);
    lcd.setColor(RGB16(0,0,0));
    lcd.setBackColor(RGB16_WHITE);
    lcd.setCursor(0, 0);
    lcd.print(buf);
}

void setup(){
    Serial.begin(115200);
    lcd.begin();
}

```

```
lcd.setBackLight(255);  
lcd.setFontSize(FONT_SIZE_MEDIUM); //set font size  
lcd.println();  
  
if (checkSD()) {  
    bmpDraw("ladar.bmp", 0, 0);  
}  
}  
  
void loop(){  
}
```

## FAQ

Q&A	Some general Arduino Problems/FAQ/Tips
Q	It only shows a "T" on the screen
A	This library is only compatible with classis Arduino IDE 1.0.x, please change the IDE
A	For any questions, advice or cool ideas to share, please visit the <b>DFRobot Forum</b> ( <a href="https://www.dfrobot.com/forum/">https://www.dfrobot.com/forum/</a> ).

## More

- Github Repository ([https://github.com/Arduinolibrary/DFRobot\\_TELEMATICS\\_3.5-\\_TFT\\_Touch\\_LCD\\_Shield](https://github.com/Arduinolibrary/DFRobot_TELEMATICS_3.5-_TFT_Touch_LCD_Shield))
- Arduino libraries ([https://github.com/Arduinolibrary/DFRobot\\_TELEMATICS\\_3.5-\\_TFT\\_Touch\\_LCD\\_Shield/raw/master/Arduino%20Libraries.zip](https://github.com/Arduinolibrary/DFRobot_TELEMATICS_3.5-_TFT_Touch_LCD_Shield/raw/master/Arduino%20Libraries.zip))
- Schematic ([https://github.com/Arduinolibrary/DFRobot\\_TELEMATICS\\_3.5-\\_TFT\\_Touch\\_LCD\\_Shield/raw/master/DFR0387%20%20OBD-II%20LCD%20Shield\\_V1.0\\_.pdf](https://github.com/Arduinolibrary/DFRobot_TELEMATICS_3.5-_TFT_Touch_LCD_Shield/raw/master/DFR0387%20%20OBD-II%20LCD%20Shield_V1.0_.pdf))

 Shopping from **TELEMATICS 3.5" TFT Touch LCD Shield** (<https://www.dfrobot.com/product-1500.html>) or **DFRobot Distributor**.

(<https://www.dfrobot.com/index.php?route=information/distributorslogo>)