

MMI Display Interface Panel

11-01-001-6

Contents

Abstract.....	2
Components.....	2
System Level Block Diagram	2
Hardware Setup	2
for usage	2
for programming.....	2
Programming Instructions	2
Arduino Uno Setup.....	3
Upload our program	6
Upload bmp in memory card	6
Hardware setup	6
Software Explanation.....	7

Document Version	Release Date	Author
1.0	22 nd Dec 2017	Sonal Gupta

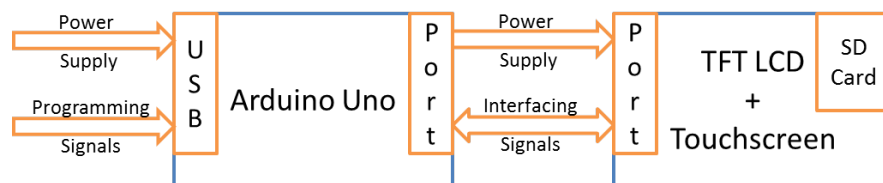
Abstract

This module consists of a TFT touchscreen shield mounted on Arduino Uno. This is used to display the name and manufacturer of the product.

Components

1. 2.4" TFT Touch Shield for Arduino
2. Arduino Uno
3. USB Type A (male) to Type B (male) cable
4. 5.0V, 1A power supply with USB Type A (female) connector
5. micro-SD card (less than or equal to 4GB)

System Level Block Diagram



Hardware Setup

for usage

As shown in figure above, the TFT Touch Shield is plugged onto Arduino Uno ports for interfacing. Vdd and Gnd pins on both boards should be matched for proper mounting. *If you are setting up arduino uno first time on this computer, do not mount TFT touchscreen shield on the Arduino board yet. It will be mounted after arduino uno setup section.*

for programming

The USB cable's Type B end is connected to the Arduino Uno board and this cable provides both the power and programming signals. In case one wants to program the module, USB Type A end or cable should be connected to a computer else it should be connected to a 5V, 1A power supply. In both cases Arduino and LCD screen should power up and you should be able to see red lights on Arduino board and backlight or any random display on LCD.

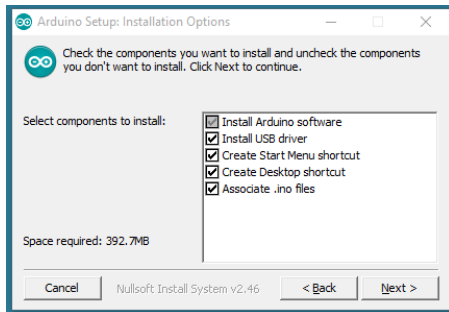
Programming Instructions

If the module needs to be reprogrammed, following instructions should to be followed

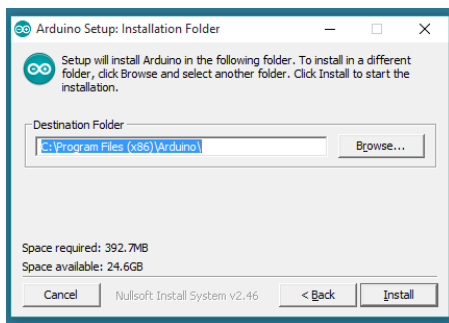
Arduino Uno Setup

Installing software

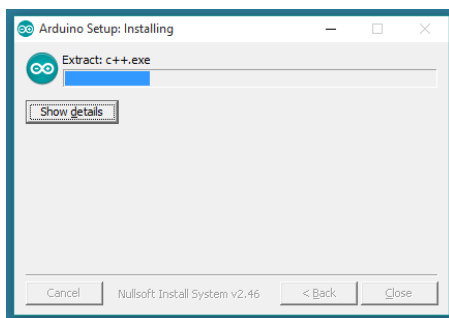
The Arduino Uno is programmed using the Arduino Software (IDE), Arduino's Integrated Development Environment. Double click on arduino-1.8.5-windows.exe provided in the CD or zip folder and please allow the driver installation process when you get a warning from the operating system.



Choose the components to install



Choose the installation directory (we suggest to keep the default one)



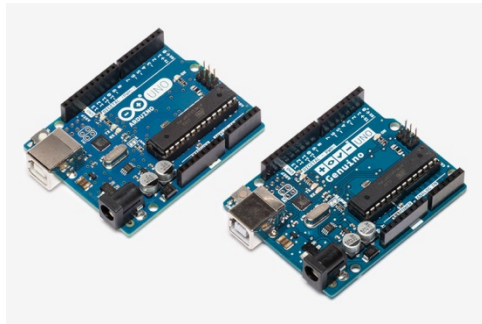
The process will extract and install all the required files to execute properly the Arduino Software (IDE)

Installing additional libraries

Unzip libraries.rar from the provided folder and copy the folders SPFD5408 - master etc to
C : \ProgramFiles(x86)\Arduino\libraries

Connecting board

Connect your Uno board with an A B USB cable; sometimes this cable is called a USB printer cable



The USB connection with the PC is necessary to program the board and not just to power it up. The Uno automatically draw power from either the USB or an external power supply. Connect the board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

Install the board drivers

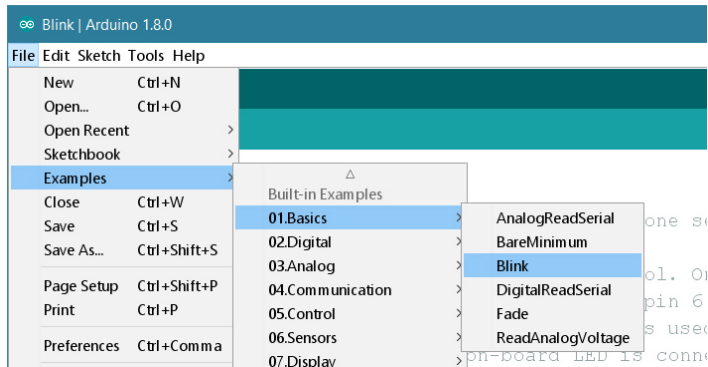
If you downloaded and expanded the Zip package or, for some reason, the board wasn't properly recognized, please follow the procedure below.

- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"
- Windows will finish up the driver installation from there.

Open your first sketch

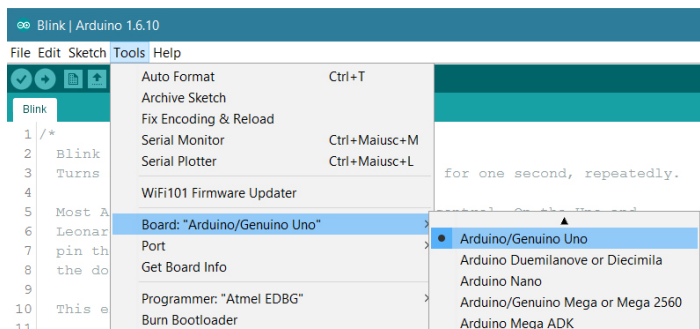
Open the LED blink example sketch:

File > Examples > 01.Basics > Blink.

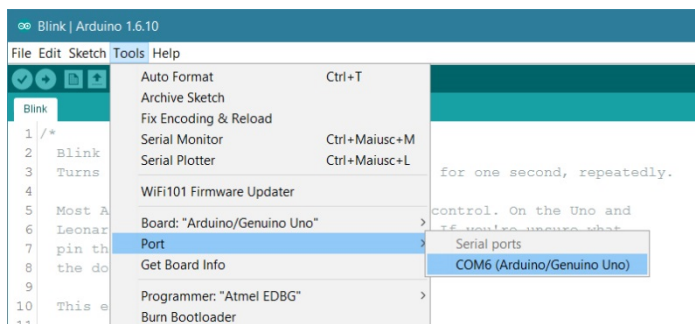


Select your board type and port

You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino or Genuino board.

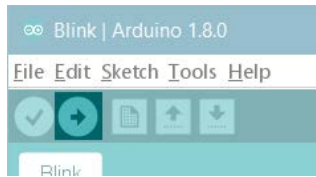


Select the serial device of the board from the Tools | Serial Port menu. This is likely to be COM3 or higher (COM1 and COM2 are usually reserved for hardware serial ports). To find out, you can disconnect your board and re-open the menu; the entry that disappears should be the Arduino or Genuino board. Reconnect the board and select that serial port.



Upload the program

Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.



A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino or Genuino up-and-running. If you have problems, please see the [troubleshooting suggestions](#).

Upload our program

- Open tftpaint.ino in Arduino by double clicking the file
- Click the upload button (looks like forward arrow symbol)

Upload bmp in memory card

- Copy files screen.bmp, screen2.bmp, screen3.bmp to the memory card (less than or equal to 4 GB)
- Insert the memory card to slot in LCD

Hardware setup

- Plug in the TFT touchscreen shield (memory card inserted) onto the Arduino Uno board. Match Vdd and Gnd pins for accurate mounting. The final setup should look as follows



- Plug the board using A B USB cable into power supply or computer. Display should show the screen.bmp file that you loaded into the memory card. After long press of 5 seconds, display should change to screen2.bmp. Again after long press of 5 seconds, display should change to screen3.bmp. Then after long press of 5 seconds, display should change back to screen.bmp.
- To troubleshoot, connect the setup to computer and open serial monitor in Arduino IDE using ctrl+shift+M

Software Explanation

Headers:

```
#include <SPI.h>
#include <Adafruit_GFX.h>    // Core graphics library
#include <SWTFT.h> // Hardware-specific library
#include <TouchScreen.h>
#include <SD.h>
```

Definitions:

```
#define CS    10
#define DC    9
#define RESET 8
#define SD_CS 10
#define touch_threshold 25
// In the SD card, place 24 bit color BMP files (be sure they are 24-bit!)
// There are examples in the sketch folder
#define YP A1 // must be an analog pin, use "An" notation!
#define XM A2 // must be an analog pin, use "An" notation!
#define YM 7  // can be a digital pin
#define XP 6  // 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);
int a = 1;
// 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
SWTFT tft;
#define BOXSIZE 40
#define PENRADIUS 3
int oldcolor, currentcolor;
```

Setup function runs once in the beginning:

```
void setup(void) {
  Serial.begin(9600); // baud rate
  Serial.println(F("Paint!")); // Will be printed on serial monitor
  pinMode(10, OUTPUT); // For SD card interface
  digitalWrite(10, HIGH); // For SD card interface

  tft.reset(); // Reset tft LCD
  uint16_t identifier = tft.readID(); // Read ID of the TFT module
  Serial.print(F("LCD driver chip: ")); // Check if this gets printed on serial
  monitor of Arduino
  Serial.println(identifier, HEX);
```

```

tft.begin(identifier);    // Initializing tft LCD

Serial.print(F("Initializing SD card...")); // Initializing SD card
if (!SD.begin(SD_CS)) {
    Serial.println(F("failed!"));
    return;
}
Serial.println(F("OK!")); // Check for this while debugging

tft.setRotation(0);    // Change this if display is coming rotated
tft.fillScreen(0);    // Change this if you don't want to clear the previous display
bmpDraw("screen.bmp", 0, 0); // Drawing the default bmp from SD card
tft.println();    // Print on the LCD
currentcolor = RED;
pinMode(13, OUTPUT);    // For enabling read from touchscreen
}

#define MINPRESSURE 5    // Reduce this if you want softer touch. Softer touch may
                        // cause more unintentional change of display
#define MAXPRESSURE 1000

```

Loop function loops infinitely:

```

void loop()
{
    int press_count = 0;
    int b = 0;
    int timeout = 0;
    TSPoint p;

    while (press_count < touch_threshold) // Increase touch_threshold value in define
    section to increase the long press time to change
    // from one display to other
    {
        if (timeout > 5 * touch_threshold) // break the while loop if not pressed 20
times in 100 times
        {
            break;
        }
        digitalWrite(13, HIGH);
        p = ts.getPoint(); // input from touchscreen
        digitalWrite(13, LOW);
        // if sharing pins, you'll need to fix the directions of the touchscreen pins
        pinMode(XM, OUTPUT);
        pinMode(YP, OUTPUT);
        // Serial.println(p.z);
        if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {
            press_count++;
            delay(50);
        }
        timeout++;
        Serial.println(press_count);
    }

    if (press_count == touch_threshold)
    {
        b = a; // This will cause switching to the next case
    }
}

```



```

switch (b)
{
    case 1:
        if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {
            tft.setRotation(0);
            tft.fillScreen(0);

            bmpDraw("screen2.bmp", 0, 0);
            a = 2;    // Switch to case 2 if long press happens in this case statement
        }
        break;
    case 2:
        if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {

            tft.setRotation(0);
            tft.fillScreen(0);
            bmpDraw("screen3.bmp", 0, 0);
            a = 3;
        }
        break;
    case 3:
        if (p.z > MINPRESSURE && p.z < MAXPRESSURE) {

            tft.setRotation(0);
            tft.fillScreen(0);

            bmpDraw("screen.bmp", 0, 0);

            a = 1;    // Go back to case 1
        }
        break;
    default:
        break;
}
}

```