

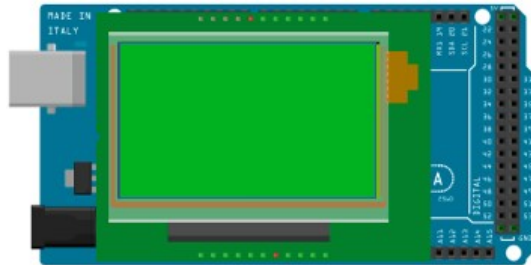
Οθόνη LED

Για να μπορούμε να ενημερωνώμαστε κάθε χρονική στιγμή για τις καιρικές συνθήκες που επικρατούν χρησιμοποιήθηκε μια οθόνη TFT LCD 2,8".



Κύκλωμα Ελέγχου Λειτουργίας

Η οθόνη αυτή τοποθετείται "πάνω" στην πλακέτα του Arduino.



Για να λειτουργήσει σωστά η οθόνη θα πρέπει να κατεβάσουμε και να εγκαταστήσουμε τις απαραίτητες βιβλιοθήκες από το github:

https://github.com/prenticedavid/MCUFRIEND_kbv

Στη συνέχεια να τρέξουμε το διαγνωστικό πρόγραμμα **TouchScreen_Calibr_native.ino** αυτής της βιβλιοθήκης για να γίνει "καλιμπράρισμα" της οθόνης.

Μετά το στάδιο αυτό, η οθόνη είναι έτοιμη προς χρήση.

Θα πρέπει, στον κώδικα που θα χρησιμοποιήσετε, να αντικαταστήσετε τις τιμές του πρωτότυπου με αυτές που σας προτείνει το πρόγραμμα.

Αυτές είναι οι παράμετροι που αντιστοιχούν στην οθόνη που έχετε.

Στον κώδικα που παρατίθεται παρακάτω, είναι υπογραμμισμένος με κίτρινο φόντο.

Στο τεστ πρόγραμμα θα δημιουργήσουμε ένα μενού επιλογών

- Θερμοκρασία
- Υγρασία
- Ατμοσφαιρική Πίεση
- Όλα

Χρησιμοποιώντας τη γραφίδα ή τα δάχτυλα των χεριών μας

- επιλέγουμε τη μέτρηση που θέλουμε να δούμε,
- Η οθόνη
 - εμφανίζει τη μέτρηση για 5 δευτερόλεπτα
 - επιστρέφει στο μενού επιλογών

Αναλυτικά

```
//Απαραίτητες Βιβλιοθήκες
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>
MCUFRIEND_kbv tft;
#include <TouchScreen.h>
#define MINPRESSURE 200
#define MAXPRESSURE 1000

const int XP=8,XM=A2,YP=A3,YM=9; // Pin της οθόνης για διάφορες λειτουργίες
const int TS_LEFT=878,TS_RT=119,TS_TOP=107,TS_BOT=896; // Διαστάσεις οθόνης

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300); // Ορισμός οθόνης Αφής (TouchScreen)

Adafruit_GFX_Button t_btn, h_btn, p_btn, all_btn; // τα κουμπιά επιλογών

int pixel_x, pixel_y; // Μεταβλητές σημείου πίεσης στην Οθόνη

// Ορισμός Χρωμάτων
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

void setup(void){ // Αρχικοποίηση

  uint16_t ID = tft.readID(); // Τύπος Οθόνης
  if (ID == 0xD3D3) ID = 0x9486; // Αλλαγή τύπου αν είναι ο 0xD3D3

  tft.begin(ID); // Έναρξη λειτουργίας Οθόνης
  tft.setRotation(0); // Οθόνη σε κατακόρυφη θέση
  tft.fillRect(BLACK); // Μαύρο φόντο οθόνης
  tft.setFont(NULL); // Δεν έχει τεθεί είδος γραμματοσειράς
  // Κόκκινα Κουμπιά του Μενού
  t_btn.initButton(&tft, 120, 60, 200, 50, WHITE, RED, WHITE, "TEMP(C)", 2); // Θερμοκρασία
  h_btn.initButton(&tft, 120, 130, 200, 50, WHITE, RED, WHITE, "HUMM(%)", 2); // Υγρασία
  p_btn.initButton(&tft, 120, 200, 200, 50, WHITE, RED, WHITE, "PRESS(mb)", 2); // Πίεση
  all_btn.initButton(&tft, 120, 270, 200, 50, WHITE, RED, WHITE, "ALL", 2); // Όλα

  t_btn.drawButton(false); // Μην σχεδιάζεις
  h_btn.drawButton(false); // ακόμα
```

```
p_btn.drawButton(false); // κανένα
all_btn.drawButton(false); // κουμπί
}

void loop(void){ // Κώδικας επανάληψης

  bool down = Touch_getXY(); // Σημείο στην οθόνη που πατήθηκε
  t_btn.press(down && t_btn.contains(pixel_x, pixel_y)); // έλεγξε
  h_btn.press(down && h_btn.contains(pixel_x, pixel_y)); // αν πατήθηκε
  p_btn.press(down && p_btn.contains(pixel_x, pixel_y)); // κάποιο από
  all_btn.press(down && all_btn.contains(pixel_x, pixel_y)); // τα κουμπιά

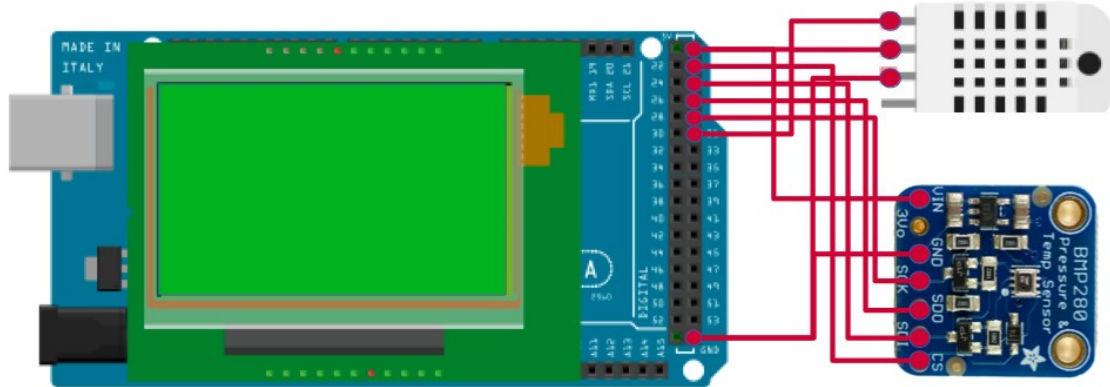
  // αν αφέθηκε το σημείο που πατήθηκε
  if (t_btn.justReleased() || h_btn.justReleased() || p_btn.justReleased() || all_btn.justReleased()){
    t_btn.drawButton(); // Εμφάνισε
    h_btn.drawButton(); // όλα
    p_btn.drawButton(); // τα
    all_btn.drawButton(); // κουμπιά
  }
  if (t_btn.justPressed()) { // αν πατήθηκε Θερμοκρασία
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
    showmsgXY(20, 100, 3, NULL, "Temperature"); // γράψε: Temperature
    showmsgXY(80, 150, 3, NULL, "27 C"); // άλλαξε γραμμή και γράψε: 27 C
    delay(5000); // αναμονή για 5sec
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
  }
  if (h_btn.justPressed()) { // αν πατήθηκε Υγρασία
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
    showmsgXY(20, 100, 3, NULL, "Hummidity"); // γράψε: Hummidity
    showmsgXY(80, 150, 3, NULL, "87 %"); // άλλαξε γραμμή και γράψε: 87 %
    delay(5000); // αναμονή για 5sec
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
  }
  if (p_btn.justPressed()) { // αν πατήθηκε Πίεση
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
    showmsgXY(20, 100, 3, NULL, "Pressure"); // γράψε: Pressure
    showmsgXY(80, 150, 3, NULL, "1023 mbar"); // άλλαξε γραμμή και γράψε: 1023 mbar
    delay(5000); // αναμονή για 5sec
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
  }
  if (all_btn.justPressed()) { // αν πατήθηκε Όλα
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
    showmsgXY(20, 100, 3, NULL, "A L L"); // γράψε: ALL
    delay(5000); // αναμονή για 5sec
    tft.fillScreen(BLACK); // κάνε το φόντο της οθόνης μαύρο
  }
}
```

```
bool Touch_getXY(void){ // Συνάρτηση χειρισμού αφής (παρέχεται από τον κατασκευαστή)
    TSPoint p = ts.getPoint();
    pinMode(YP, OUTPUT);
    pinMode(XM, OUTPUT);
    digitalWrite(YP, HIGH);
    digitalWrite(XM, HIGH);
    bool pressed = (p.z > MINPRESSURE && p.z < MAXPRESSURE);
    if (pressed) {
        pixel_x = map(p.x, TS_LEFT, TS_RT, 0, tft.width());
        pixel_y = map(p.y, TS_TOP, TS_BOT, 0, tft.height());
    }
    return pressed;
}

void showmsgXY(int x, int y, int sz, const GFXfont *f, const char *msg){ // Συνάρτηση προβολής μηνύματος
                                                                    // (παρέχεται από τον κατασκευαστή)
    tft.setFont(f);
    tft.setCursor(x, y);
    tft.setTextColor(RED);
    tft.setTextSize(sz);
    tft.print(msg);
}
```

Arduino Κώδικας Προβολής Μετρήσεων

Σε περίπτωση που θέλετε να υλοποιήσετε ένα έργο προβολής μετρήσεων θερμοκρασίας, υγρασίας και πίεσης στην οθόνη TFT LCD χρησιμοποιήστε το ακόλουθο κύκλωμα.



Προσοχή! Έχει χρησιμοποιηθεί *Arduino Mega*.

Το πρόγραμμα διαβάζει συνεχώς τις τιμές θερμοκρασίας, υγρασίας και πίεσης.

Ταυτόχρονα, εμφανίζει ένα μενού επιλογών με τις μετρήσεις των μεγεθών που θέλουμε να δούμε – υπάρχει και η επιλογή να δούμε όλες τις μετρήσεις. Όταν επιλέξουμε ένα μέγεθος η οθόνη γίνεται μαύρη και εμφανίζει για 5 δευτερόλεπτα τη μέτρηση. Στη συνέχεια, επιστρέφει στο αρχικό μενού επιλογών.

Ο κώδικας υλοποίησης παρατίθεται ακολούθως.

```
#include <Adafruit_GFX.h>
#include <MCUFRIEND_kbv.h>
#include <TouchScreen.h>
#include <SPI.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include "DHT.h"

#define BMP_SCK 29 // Έξοδος SCK του αισθητήρα στο Pin 29 του Arduino
#define BMP_MISO 27 // Έξοδος SDO του αισθητήρα στο Pin 27 του Arduino
#define BMP_MOSI 25 // Έξοδος SDI του αισθητήρα στο Pin 25 του Arduino
#define BMP_CS 23 // Έξοδος CS του αισθητήρα στο Pin 23 του Arduino

#define MINPRESSURE 200
#define MAXPRESSURE 1000
```

```
#define DHTPIN 31 // Το Ψηφιακό Pin 31 θα χρησιμοποιηθεί για διάβασμα δεδομένων
#define DHTTYPE DHT11 // Ο Αισθητήρας DHT είναι τύπου DHT11

DHT dht(DHTPIN, DHTTYPE); // Βασικό Αντικείμενο για αποθήκευση Μετρήσεων

Adafruit_BMP280 bmp(BMP_CS, BMP_MOSI, BMP_MISO, BMP_SCK); // Βασικό Αντικείμενο Μετρήσεων
float temperature;
float pressure;
float hum;

String msg = "";
char buff[20];
int counter = 0;

const int XP=8,XM=A2,YP=A3,YM=9; // Ειδικές Ρυθμίσεις για την
const int TS_LEFT=878,TS_RT=119,TS_TOP=107,TS_BOT=896; // οθόνη που χρησιμοποιούμε

MCUFRIEND_kbv tft; // Μεταβλητή Οθόνης

TouchScreen ts = TouchScreen(XP, YP, XM, YM, 300); // Μεταβλητή Οθόνης Αφής (TouchScreen)

Adafruit_GFX_Button t_btn, h_btn, p_btn, all_btn; // Τα κουμπιά που θα χρησιμοποιηθούν

int pixel_x, pixel_y; // Συντεταγμένες σημείου αφής
// Ορισμός Χρωμάτων
#define BLACK 0x0000
#define BLUE 0x001F
#define RED 0xF800
#define GREEN 0x07E0
#define CYAN 0x07FF
#define MAGENTA 0xF81F
#define YELLOW 0xFFE0
#define WHITE 0xFFFF

void setup(void){
  Serial.begin(9600);
```

```
bmp.begin();
dht.begin();

uint16_t ID = tft.readID();
if (ID == 0xD3D3) ID = 0x9486;
tft.begin(ID);
tft.setRotation(0);
tft.fillScreen(BLACK);
tft.setFont(NULL);

t_btn.initButton(&tft, 120, 60, 200, 50, WHITE, RED, WHITE, "TEMP(C)", 2);
h_btn.initButton(&tft, 120, 130, 200, 50, WHITE, RED, WHITE, "HUMM(%)", 2);
p_btn.initButton(&tft, 120, 200, 200, 50, WHITE, RED, WHITE, "PRESS(mb)", 2);
all_btn.initButton(&tft, 120, 270, 200, 50, WHITE, RED, WHITE, "ALL", 2);

t_btn.drawButton(false);
h_btn.drawButton(false);
p_btn.drawButton(false);
all_btn.drawButton(false);

}

void loop(void){
  pressure = bmp.readPressure(); // Αποθήκευση τιμής Ατμοσφαιρικής Πίεσης
  temperature = bmp.readTemperature(); // Αποθήκευση τιμής Θερμοκρασίας
  hum = dht.readHumidity();      // Αποθήκευση τιμής Υγρασίας (DHT.humidity) στη μεταβλητή hum
  bool down = Touch_getXY();
  t_btn.press(down && t_btn.contains(pixel_x, pixel_y));
  h_btn.press(down && h_btn.contains(pixel_x, pixel_y));
  p_btn.press(down && p_btn.contains(pixel_x, pixel_y));
  all_btn.press(down && all_btn.contains(pixel_x, pixel_y));

  if (t_btn.justReleased() || h_btn.justReleased() || p_btn.justReleased() || all_btn.justReleased()){
    t_btn.drawButton();
    h_btn.drawButton();
    p_btn.drawButton();
  }
}
```

```
    all_btn.drawButton();
}
if (t_btn.justPressed()) {
    msg = String(temperature) + " C"; // Μήνυμα που θα εμφανιστεί: <τιμή θερμοκρασίας> C
    tft.fillScreen(BLACK);
    msg.toCharArray(buff, 50); // Μετατροπή μηνύματος σε πίνακα χαρακτήρων (ΑΠΑΡΑΙΤΗΤΟ)
    showmsgXY(10, 100, 3, NULL, buff); // Εμφάνιση Μηνύματος
    delay(5000);                      // Καθυστέρηση 5 δευτερολέπτων
    tft.fillScreen(BLACK);
}
if (h_btn.justPressed()) {
    msg = String(hum) + " %"; // Μήνυμα που θα εμφανιστεί: <τιμή υγρασίας> %
    tft.fillScreen(BLACK);
    msg.toCharArray(buff, 50); // Μετατροπή μηνύματος σε πίνακα χαρακτήρων (ΑΠΑΡΑΙΤΗΤΟ)
    showmsgXY(10, 100, 3, NULL, buff); // Εμφάνιση Μηνύματος
    delay(5000);                      // Καθυστέρηση 5 δευτερολέπτων
    tft.fillScreen(BLACK);
}
if (p_btn.justPressed()) {
    msg = String(pressure/100) + " mbar"; // Μήνυμα που θα εμφανιστεί: <τιμή πίεσης> mbar
    tft.fillScreen(BLACK);
    msg.toCharArray(buff, 50); // Μετατροπή μηνύματος σε πίνακα χαρακτήρων (ΑΠΑΡΑΙΤΗΤΟ)
    showmsgXY(10, 100, 3, NULL, buff); // Εμφάνιση Μηνύματος
    delay(5000);                      // Καθυστέρηση 5 δευτερολέπτων
    tft.fillScreen(BLACK);
}
if (all_btn.justPressed()) {
    /*
    Μήνυμα που θα εμφανιστεί:
    <τιμή θερμοκρασίας> C
    <τιμή υγρασίας> %
    <τιμή πίεσης> mbar
    */
    msg = String(temperature) + " C\n\n" + String(hum) + " %\n\n" + String(pressure/105.035) + " mbar";
    tft.fillScreen(BLACK);
    msg.toCharArray(buff, 50); // Μετατροπή μηνύματος σε πίνακα χαρακτήρων (ΑΠΑΡΑΙΤΗΤΟ)
```



```
showmsgXY(10, 100, 3, NULL, buff); // Εμφάνιση Μηνύματος
delay(5000);                        // Καθυστέρηση 5 δευτερολέπτων
tft.fillScreen(BLACK);
}
}

bool Touch_getXY(void){
    TSPoint p = ts.getPoint();
    pinMode(YP, OUTPUT);
    pinMode(XM, OUTPUT);
    digitalWrite(YP, HIGH);
    digitalWrite(XM, HIGH);
    bool pressed = (p.z > MINPRESSURE && p.z < MAXPRESSURE);
    if (pressed) {
        pixel_x = map(p.x, TS_LEFT, TS_RT, 0, tft.width());
        pixel_y = map(p.y, TS_TOP, TS_BOT, 0, tft.height());
    }
    return pressed;
}

void showmsgXY(int x, int y, int sz, const GFXfont *f, const char *msg){
    //int16_t x1, y1;
    //uint16_t wid, ht;
    tft.setFont(f);
    tft.setCursor(x, y);
    tft.setTextColor(RED);
    tft.setTextSize(sz);
    tft.print(msg);
}
```