

```

#include <SPI.h>
#include "Adafruit_ILI9341.h" // Display Treiber
#include <Wire.h>
#include "BluetoothSerial.h"
#include <Adafruit_SSD1306.h>
#include <SoftwareSerial.h>
#include "VeDirectFrameHandler.h"

VeDirectFrameHandler myve;

// SoftwareSerial
#define rxPin 26 // RX using Software Serial so we can
use the hardware UART to check the output
#define txPin 14 // TX Not used

SoftwareSerial veSerial(rxPin, txPin);

//TFT-Display
#define TFT_CS 5
#define TFT_DC 4
#define TFT_MOSI 23
#define TFT_CLK 18
#define TFT_RST 22
#define TFT_MISO 19
#define TFT_LED 15

//I2C
#define I2C_SDA 17
#define I2C_SCL 16

#define DISPLAY_2_I2CBME_ADDRESS 0x3C
#define DISPLAY_2_WIDTH 128 // Breite in Pixeln
#define DISPLAY_2_HEIGHT 64 // Höhe in Pixeln

#define GREEN 0x07E0
#define BLACK 0x0000
#define WHITE 0xFFFF
#define YELLOW 0xFFE0
#define ORANGE 0xFD20

TwoWire I2CBME = TwoWire(0);
Adafruit_SSD1306 display2(DISPLAY_2_WIDTH, DISPLAY_2_HEIGHT, &I2CBME, -1);

float SOC, SOC2, V2, I2, T22, P2, H11;

BluetoothSerial SerialBT;

#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 32 // OLED display height, in pixels
#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)

Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC, TFT_RST);

```

```

void setup() {

    I2CBME.begin(I2C_SDA, I2C_SCL, 100000); // Frequenz I2C-Bus 400000
    bool status;
    status = display2.begin(SSD1306_SWITCHCAPVCC, DISPLAY_2_I2CBME_ADDRESS);
    delay(200);

    SerialBT.begin("Frischwasser"); //BT Name

    tft.begin();
    tft.setRotation(1);
    tft.fillScreen(ILI9341_BLACK);

    Serial.begin(115200);                // output serial port
    veSerial.begin(19200);               // input serial port (VE device)
    veSerial.flush();

}

void loop() {
    ReadVEData();
    EverySecond();
}

void ReadVEData() {
    while ( veSerial.available() ) {
        myve.rxData(veSerial.read());
    }
    yield();
}

void EverySecond() {
    static unsigned long prev_millis;

    if (millis() - prev_millis >1000) {
        PrintData();
        prev_millis = millis();
    }
}

void PrintData() {

    Serial.print(myve.veName[1]);
    Serial.print(" = ");
    Serial.println(myve.veValue[1]);

    // Daten abrufen und prüfen
    V2 = (atoi(myve.veValue[1]));
    if (V2 < 15000) {
        I2 = (atoi(myve.veValue[3]));
        P2 = (atoi(myve.veValue[4]));
        T22 = (atoi(myve.veValue[2]));
        SOC = (atoi(myve.veValue[6]));
        SOC2 = SOC*200/1000;
    }
}

```

```

//TFT Display
pinMode(TFT_LED, OUTPUT);
tft.setTextColor(ILI9341_GREEN, BLACK);
tft.setCursor(10, 10);
tft.setTextSize(2);
tft.println("Batterie LiFePO4 200Ah");

tft.setTextColor(ILI9341_YELLOW, BLACK);
tft.setCursor(10, 42);
tft.setTextSize(3);
tft.print(int(SOC/10)); tft.print(" % "); tft.print(int(SOC2));
tft.print(" Ah ");

tft.setTextColor(ILI9341_WHITE, BLACK);
tft.setCursor(10, 80);
tft.setTextSize(2);
tft.print((V2/1000),1); tft.print(" V "); tft.print((I2/1000),1);
tft.print(" A "); tft.print(int(P2)); tft.println(" W ");

int volt = analogRead(34); // Spannung Starterbatterie
float Spannung;
double voltage = map(volt, 0,4093, 0,1700 );// + offset;
voltage /=100;

tft.setTextColor(ILI9341_WHITE, BLACK);
tft.setCursor(10, 110);
tft.setTextSize(2);
tft.print(int(T22)); tft.print(" C SB ");
tft.print(((voltage)-0.1),1); tft.print(" V "); ;

tft.setTextColor(GREEN, BLACK);
tft.setCursor(10, 145);
tft.setTextSize(2);
tft.println("Frischwasser");

float S2 = (float)analogRead(35); // Niveau Frischwassertank
S2=((S2 * 3.3)/4095); // offset +0.13

tft.setTextColor(ILI9341_YELLOW, BLACK);
tft.setCursor(10, 175);
tft.setTextSize(4);
tft.print(int((130/2.2)*S2)); tft.println(" L "); //Tanksonde

// OLE Display
display2.setRotation(2);
display2.clearDisplay();
display2.setTextColor(SSD1306_WHITE);
display2.setTextSize(2);
display2.setCursor(0,0);
display2.println("Wasser");
display2.setTextSize(4);
display2.setCursor(0,32);
display2.print(int((130/2.2)*S2));

```

```
display2.display();

//Bluetooth
SerialBT.print("Frischwasser: ");
SerialBT.print(int((130/2.2)*S2));
SerialBT.println(" L");
}
```