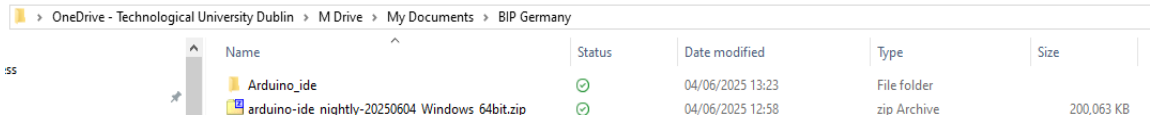
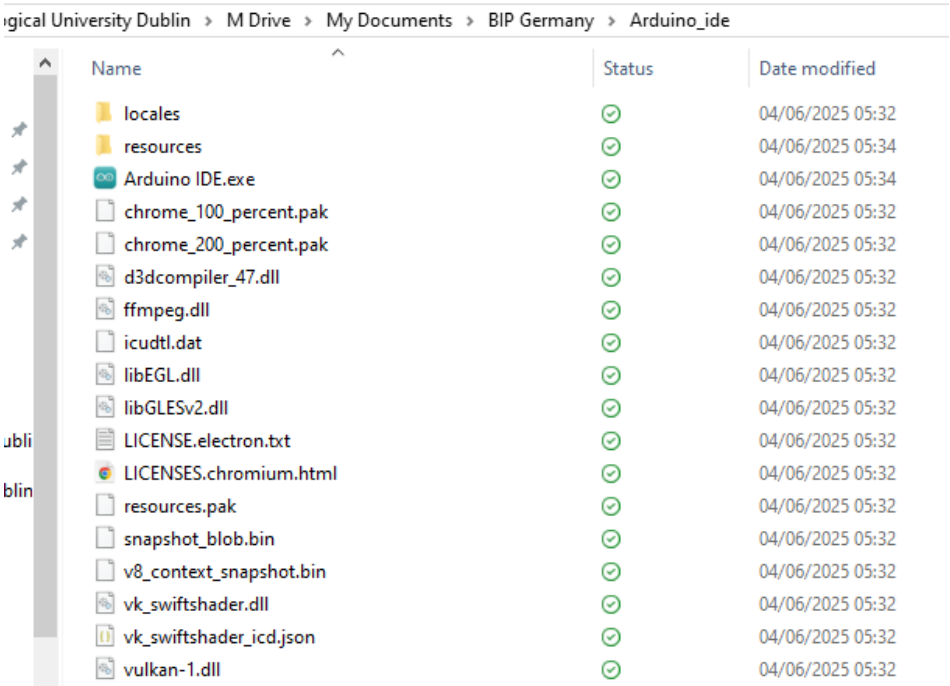
Setting up Arduino IDE for the Tenstar Robot ESP32S3-1.14TFTBoard (SENSATE-X BIP)

1. Getting the IDE.
    a. Go to https://www.arduino.cc/en/software/
    b. Download the current version of the IDE for your system. On the 04/06/2025 this was “arduino-ide_nightly-20250604_Windows_64bit.zip” for a windows system.
    c. Create a personal folder on your computer for the files you are going to use for this Blended Intensive Programme (BIP) called for example “BIP Germany”.
    d. Create a subfolder to this folder called for instance “Arduino_ide”. Place the zip file into this subfolder.
    e. Unzip the file. You can use **7Zip** as I have discovered that **Windows Explorer** comes up with a “path too long” error. **7Zip** is available for download at https://7-zip.org/ Open the zip file using **7Zip** and click extract. When prompted about a destination choose the default.
    f. Your main folder should look something like:



and your subfolder should look something like:

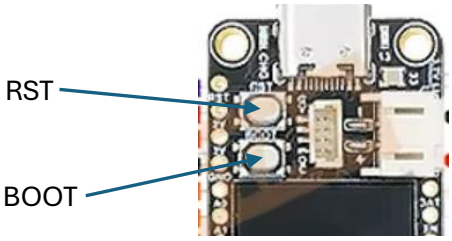


2. Running the IDE
    a. Double click on “Arduino IDE.exe” in the subfolder. The IDE should open with an empty sketch.
    b. Click on **Select Board** and choose **Select Other Board and Port**. Select “ESP32S3 Dev Module” and hit **OK**.
    c. Plug in the development board using a USB data cable. If the board is new it should run through some preprogrammed code to test the display and displays some sensor data.
    d. Go back to the ide and select **Tools->USB CDC on Boot** and set this to “Enabled”. CDC stands for communications device class.
    e. Select **Tools->Port** and select the “ESP32 Family Device” port. In my case it was “COM8”.
    f. Select **File->Examples->Basics->Blink** and the Blink sketch should open.
    g. Add the line “#define LED_BUILTIN 13” without a semicolon, just before the “setup()” function definition. The code section should look like:

```
#define LED_BUILTIN 13

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}
```

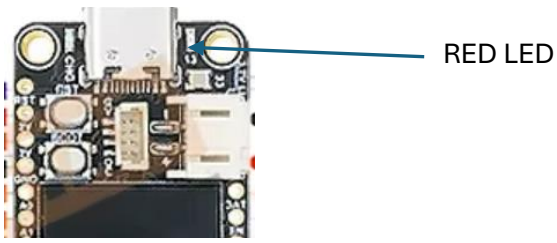
    Click on **File->Save** to save your updated program.

    h. On the development board **i)** press and hold the **BOOT** button. **ii)** Press and release the **RST** button. **iii)** Release the **BOOT** button. The board in now ready to programme.



    i. Press the upload button in the IDE to compile and upload the blink program. 

    j. The program should compile upload and run. The red led should be blinking with a cycle time of two seconds.



3. Getting the serial monitor working.
    a. Select **File->Examples->Communication->SerialEvent** and the SerialEvent sketch should open.
    b. For most Arduino boards the "SerialEvent()" function is called if there are bytes in the Serial input buffer. Alas this does not seem to be the case for ESP32 boards. To fix this we add the lines:

```
if (Serial.available()) {
    serialEvent();
}
```

    as the last instructions in the "loop()" function. The code section should look like:

```
    stringComplete = false;
  }

  if (Serial.available()) {
    serialEvent();
  }
}
```

    Click on **File->Save** to save your updated program.

    c. As before, on the development board **i)** press and hold the **BOOT** button. **ii)** Press and release the **RST** button. **iii)** Release the **BOOT** button. The board in now ready to programme.
    d. As before, press the upload button in the IDE to compile and upload the serialEvent program.
    e. After the program compiles and uploads, Click on **Tools->Serial Monitor**. This opens the Serial Monitor. Type some text into message section of the serial monitor and hit enter. The text should be echoed back from the board. For example:

    f. If this is not happening: Make sure that "USB CDC on Boot" is enabled (see 2.d). Make sure the port is set up correctly (see 2.e). Make sure your "loop()" function is modified correctly (see 3.b).

4. Getting the starter sketch working.
    a. Many thanks to Eiken Lübbers at Hochschule Darmstadt, who are running the SENSATE-X BIP, for supplying us with a Starter Sketch for the board.

```
/*
 * ------------------------------------------------------------------------------------------
 * Project Name:   Start Project
 * Author:         [Your Name]
 * Date:           [Insert Date]
 * Version:        0.1
 * Description:    Simple start project to initialize all onboard modules:
 *                 - TFT Display, BMP280, QMI8658C, RGB-LED
 *                 - Select "ESP32S3 Dev Module" in the Arduino Board Manager.
 *                 - Ensure all required libraries are installed.
 *
 * Hardware:       ESP32-S3 FH4R2, 1.14" TFT Display, BMP280, QMI8658C, WS2812 RGB-LED
 *
 * Initial Setup:  To enter boot mode and programm your ESP32S2:
 *                 1. Connect the ESP32S3 via USB.
 *                 2. Hold the "BOOT" button.
 *                 3. While holding "BOOT", press and release the "RST" button.
 *                 4. Release "BOOT".
 *
 * Warning:        The RGB LED and QMI8658C sensor cannot be used simultaneously
 *                 when connected to GPIO33, due to a conflict in the QMI library.
 * ------------------------------------------------------------------------------------------
 */

/*
 * ------------------------------------------------------------------------------------------
 * Libraries
 * ------------------------------------------------------------------------------------------
 */
 // Add or remove as needed
#include <Arduino.h>
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_NeoPixel.h>  // RGB_LED
#include <Adafruit_GFX.h>       // Display
#include <Adafruit_ST7789.h>    // Display
#include <Adafruit_BMP280.h>    // BMP
#include <SensorQMI8658.hpp>    // QMI

/*
 * ------------------------------------------------------------------------------------------
 * Pin Definitions
 * ------------------------------------------------------------------------------------------
 */
// RGB LED
// (GPIO33 conflicts with QMI8658C INT pin, do not use parallel)
#define LED_PIN       33
#define NUM_LEDS      1
// TFT Display Pins
#define TFT_CS        7
#define TFT_DC        39
#define TFT_RST       40
#define TFT_backlight 45
// SPI Pins
#define SPI_SCK       36
#define SPI_MISO      37
#define SPI_MOSI      35
// I2C Pins for Sensors
#define I2C_SDA       42
#define I2C_SCL       41
// Sensor Address
#define BMP_Addr      0x77
#define QMI_Addr      0x6B

/*
 * ------------------------------------------------------------------------------------------
```

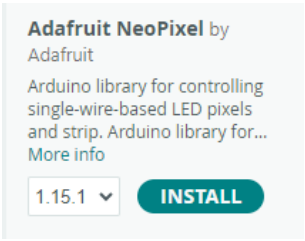
```
 * Global Objects/Varibles
 * ------------------------------------------------------------------------------------------
 */
// TFT-Display
Adafruit_ST7789 tft = Adafruit_ST7789(TFT_CS, TFT_DC, TFT_RST);
// RGB LED
Adafruit_NeoPixel RGB_LED = Adafruit_NeoPixel(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
// BMP Sensor (Temperature and Pressure Sensor)
Adafruit_BMP280 bmp;
// QMI Sensor (6-axis Gyroscope & Accelerometer)
SensorQMI8658 qmi;
// Data structures for QMI Sensor readings
IMUdata acc;
IMUdata gyr;

// Example Variables
int randomCounter = 0;
int randomTime = 500;

/*
 * ------------------------------------------------------------------------------------------
 * Function Prototypes
 * ------------------------------------------------------------------------------------------
 */
void initDisplay();
void initBMP();
void initQMI();

/*
 * ------------------------------------------------------------------------------------------
 * Setup Function – runs once at startup
 * @ brief Example Setup Code. Add and adjust Code as needed:
 * ------------------------------------------------------------------------------------------
 */
void setup() {
  // Start serial interface
  Serial.begin(115200);
  // write on serial interface
  Serial.println("Starting...");

  // Start SPI and I2C communication
  SPI.begin(SPI_SCK, SPI_MISO, SPI_MOSI, TFT_CS);
  Wire.begin(I2C_SDA,I2C_SCL);

  // Initialize TFT Display
  initDisplay();

  // write on TFT Display
  // Cursor position (x,y)
  tft.setCursor(0, 0);
  tft.print("Starting...");

  // Initialize BMP280
  initBMP();

  // Initialize QMI8658C
  initQMI();

  Serial.println("Setup finished");

  // Wait for 1000ms
  delay(1000);
}

/*
 * ------------------------------------------------------------------------------------------
 * Loop Function – runs repeatedly
 * @brief Example Code. Put your own Code here:
 * ------------------------------------------------------------------------------------------
 */
```

```
void loop() {
  // Refresh Display
  tft.fillScreen(ST77XX_BLACK);
  // print message on Display (x,y)
  if(randomCounter <= 140){
    tft.setCursor(randomCounter+20, randomCounter);
    tft.print("SENSATE-X");
    randomCounter += 10;
    // Wait for Random Time
    delay(randomTime);
  }
  else{
    randomCounter = rand() % 140;
    tft.setRotation(rand() % 8);
    randomTime = rand() % 500 + 100;
  }
}

/*
 * ---------------------------------------------------------------------------------------
 * Functions
 * ---------------------------------------------------------------------------------------
 */
/**
 * @brief Initializes the TFT display, sets rotation, background color, text color and
size.
 */
void initDisplay(){
  // Turn on TFT backlight
  pinMode(TFT_backlight, OUTPUT);
  digitalWrite(TFT_backlight, HIGH);
  // Initialize TFT Display
  tft.init(135, 240);   // Width x Height
  tft.setRotation(3);   // Adjust as needed
  tft.fillScreen(ST77XX_BLACK);
  tft.setTextColor(ST77XX_WHITE);
  tft.setTextSize(2);   // Adjust as needed
}

/**
 * @brief Initializes the BMP280 sensor and handles failure if not found.
 *
 * @note Uses I2C address defined by BMP_ADDR.
 */
void initBMP(){
  // Initialize BMP280 Sensor (address is 0x77)
  if (!bmp.begin(BMP_Addr)) {
    // if Sensor could not be found print error
    tft.setCursor(0, 20);
    tft.print("BMP not found");
    Serial.println("BMP not found");
    // endless loop
    while (1);
  }
}

/**
 * @brief Initializes the QMI8658C IMU sensor and configures accelerometer and gyroscope.
 *
 * @note Uses I2C address defined by QMI_ADDR.
 */
void initQMI(){
  // Initialize QMI8658C Sensor (address is 0x6B)
  if (!qmi.begin(Wire,QMI_Addr,I2C_SDA,I2C_SCL)) {
    // if Sensor could not be found print error
    tft.setCursor(0, 20);
    tft.print("QMI not found");
    Serial.println("QMI not found");
    // endless loop
    while (1);
  }
```

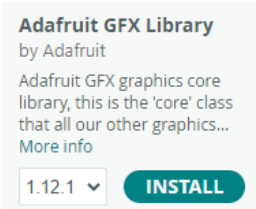
```
  // Configure QMI sensor
  qmi.enableGyroscope();
  qmi.enableAccelerometer();
  qmi.configAccelerometer(
  SensorQMI8658::ACC_RANGE_4G,
  SensorQMI8658::ACC_ODR_1000Hz,
  SensorQMI8658::LPF_MODE_0);
  qmi.configGyroscope(
  SensorQMI8658::GYR_RANGE_64DPS,
  SensorQMI8658::GYR_ODR_896_8Hz,
  SensorQMI8658::LPF_MODE_3);
}

/*
 * ------------------------------------------------------------------------------------------
 * End of file
 * ------------------------------------------------------------------------------------------
 */
```

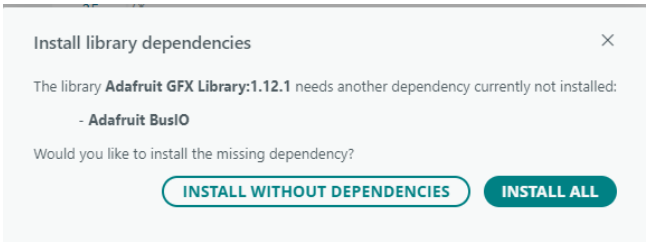
Click on **File->New Sketch** to open a new sketch. Overwrite the template code with the Starter Sketch above. Click on **File->Save** to save your updated program.

b. Click on **Tools->Manage Libraries...** which opens the Library Manager. Type in "Adafruit_Neopixel" into the search box and then click the **INSTALL** button on the "Adafruit NeoPixel" Library. This will install the library.

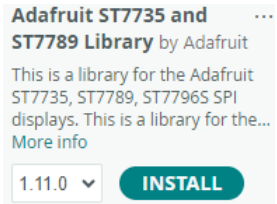


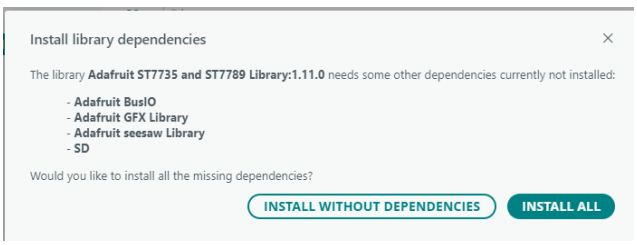
c. Next Search for "Adafruit_GFX" in the search box and then click the **INSTALL** button on the "Adafruit GFX" Library. Click on the **INSTALL ALL** button to also install all the dependencies of this graphics library.



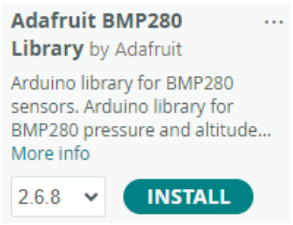


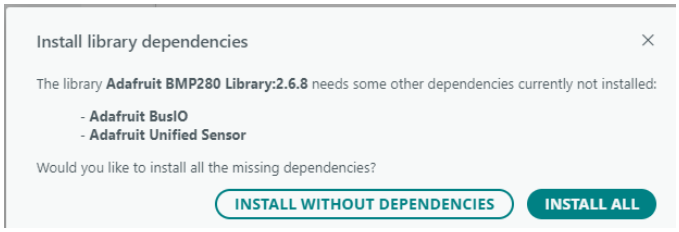
d. Next Search for "Adafruit_ST7789" in the search box and then click the **INSTALL** button on the "Adafruit ST77355 and ST7789" Library. Click on the **INSTALL ALL** button to also install all the dependencies of this library.



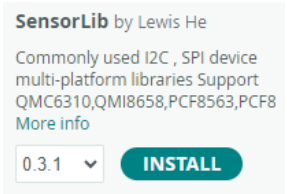


e. Next Search for “Adafruit_BMP280” in the search box and then click the **INSTALL** button on the “Adafruit BMP280” Library. Click on the **INSTALL ALL** button to also install all the dependencies of this library.





f. Next Search for “QMI8658” in the search box and then click the **INSTALL** button on the “Sensor Lib” Library.



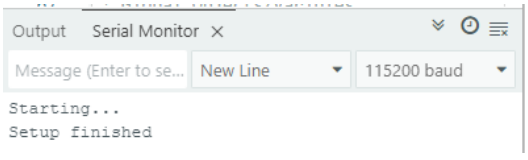
g. This should complete the installation of the libraries for the sketch: Adafruit NeoPixel, Adafruit GFX, Adafruit ST77355 and ST7789, Adafruit BMP280 and finally the Sensor Lib library. The sketch should now be able to compile and run.

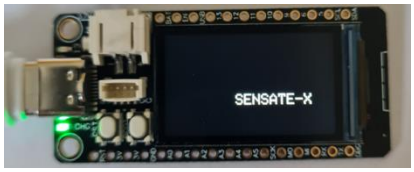
h. Click on **Tools->Serial Monitor**. This opens the Serial Monitor. Set the baud-rate to 115200.

i. As before, on the development board i) press and hold the BOOT button. ii) Press and release the RST button. iii) Release the BOOT button. The board in now ready to programme.

j. As before, press the upload button in the IDE to compile and upload the starter program.

k. The serial monitor should display messages from the development board and the TFT display should be displaying the word “SENSATE-X” at various locations and orientations.





GOOD LUCK

Arduino Glossary: https://edu-content-preview.arduino.cc/glossary