

Falcon BMS to Arduino Interface Tool (BMSAIT)

Example program for DED (Data Entry Display)



Author	Robin „Hummer“ Bruns
Document version	1.1
Softwareversion	1.4.4
BMS Version	4.37
Date	23.04.2025

1. Overview

This project describes how to display the DED on an OLED display.

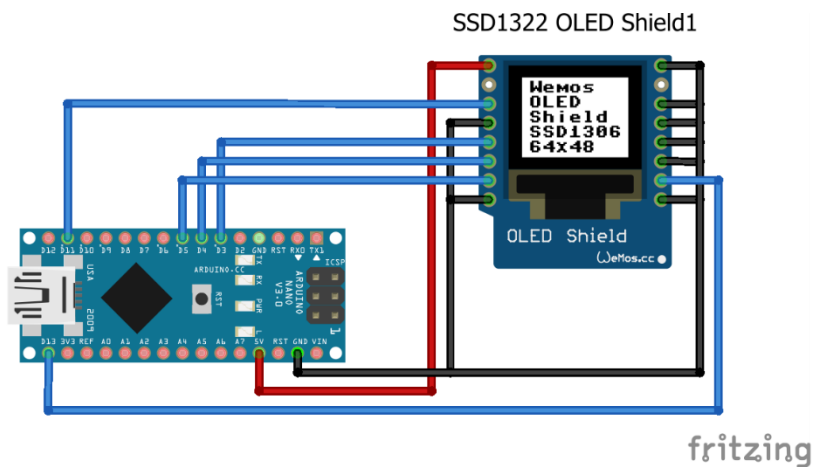
This can be done very easily if you use the same hardware that I used to test this Arduino code. Adjustments are only necessary if you use different hardware. Please read the chapter on the DED module in the BMSAIT documentation before you start to implement this example.

Required hardware:

- An Arduino board (e.g. a UNO) or a compatible controller (e.g. ESP32). I strongly recommend to use an ESP32 board as these offer some decent computing power and will allow a very smooth and fast update of the DED
- An OLED display with 256x64 pixels (in this example: SSD1322 1.3" with 256x64 pixels; SPI 4Wire connection)
- Connecting cables

2. Wiring

U8G2Lib	Arduino	SSD1322
	GND	1 GND
	5V	2VSS
	-	3
CLOCK	2	4 SCLK
DATA	3	5 SDIN
	-	6 D7
	GND	7 D6
	GND	8 D5
	GND	9 D4
	GND	10 D3
	GND	11 D2
	GND	12 D1
	GND	13 D0
DC	5	14 DC
RESET	6	15 RES
CS	4	16 CS



3. Programming the Arduino

If the Arduino IDE is not yet installed, please refer to chapter 4.1.4 of the BMSAIT documentation.

Double-click the .ino file in the desired variant from the \Arduino Sketch\BMSAIT_DED\ folder. The sketch will be loaded in the Arduino IDE. If you have done the wiring according to Chapter 2, no adjustments are required here.

Please note that this module requires an additional library for the Arduino software (U8G2). In U8G2 library, **the 16-bit mode has to be activated!**

Once you've chosen the correct Arduino board, upload the sketch to the Arduino.

4. Setting up the Windows program

Install and start BMSAIT and make sure that the basic settings have been set correctly. It is particularly important that the reference to the variable definition (BMAIT-Variablen.csv) is set. Turn off autostart.

Then load the enclosed configuration (BMSAIT_demoDED.ini). BMSAIT should now display the loaded definition (one COM port and several variables).

Attention: The DED generates a lot of data traffic between the PC and the Arduino. This can quickly cause data packets being lost if the Arduino can't read the data fast enough. In order to ensure a stable data transmission, the data traffic in this program should **run via the PULL mode**.

Right-click on the COM port and edit it. Select the COM port to which your Arduino is connected. If you are not sure which COM port this is, then either select the SCAN function and see on which COM port the Arduino sends a response or you can look in the Windows Device Manager.

I recommend saving the changes now ("save as" and selecting a new file).

Start the processing. If everything has worked, the DED will display a test image after a few seconds.

5. Result

The DED is only displayed when the Arduino receives current data to display. If no data is received, the DED is switched off after 5 seconds.

Start FalconBMS and enter the 3D world. The DED should now start again and display the current data from the simulation.

Note: If there is no connection to BMSAIT, the Arduino will put the OLED into a sleep mode after 10 seconds. The indicator becomes active again as soon as a connection to BMSAIT is established. The test image disappears as soon as the game is in the 3D world.