

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm DED (Data Entry Display)



Autor	Robin „Hummer“ Bruns
Dokumentversion	1.1
Softwareversion	1.4.4
BMS Version	4.37
Datum	23.04.2025

1. Überblick

Dieses Projekt beschreibt, wie ihr das DED auf einem OLED Display zur Anzeige bringt.

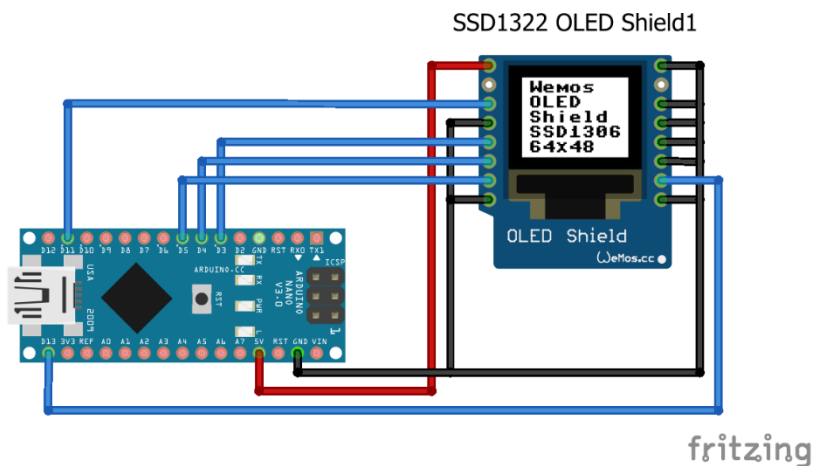
Dies kann sehr einfach gehen, wenn ihr die gleiche Hardware verwendet, die ich auch zum Testen dieses Arduino-Codes verwendet habe. Bei Abweichungen des Displays sind Anpassungen erforderlich, die sich aber in Grenzen halten sollten. Lest dazu bitte das Kapitel zum Modul DED in der Dokumentation zu BMSAIT.

Benötigte Hardware:

- Ein Arduino-Board (z.B. ein UNO)
- Ein OLED Display mit 256x64 Zeichen (in diesem Beispiel: SSD1322 1.3" mit 256x64 Zeichen; SPI 4Wire Verbindung)
- Verbindungskabel

2. Verkabelung

U8G2Lib	Arduino	SSD1322
	GND	1 GND
	5V	2 V _{SS}
	-	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. Programmierung des Arduino

Falls die Arduino IDE noch nicht installiert ist, lest bitte das Kapitel 4.1.4 der BMSAIT Dokumentation.

Ruft nun die .ino in der gewünschten Variante aus dem Ordner \Arduino Sketch\BMSAIT_DED\ mit einem Doppelklick auf. Das Sketch wird in der Arduino IDE geladen. Wenn ihr die Verkabelung gem. Kapitel 2 durchgeführt habt, sind hier keine Anpassungen erforderlich.

Beachtet bitte, dass Ihr für dieses Programm eine zusätzliche Bibliothek für die Arduino Software laden müsst (U8G2) **und in der Bibliothek der 16-Bit-Modus aktiviert werden muss!**

Nachdem ihr das richtige Arduino-Board ausgewählt habt, ladet ihr das Sketch auf den Arduino hoch.

4. Einstellung des Windows-Programms

Installiert und startet BMSAIT und stellt sicher, dass die Basiseinstellungen richtig vorgenommen wurden. Wichtig ist insbesondere, dass der Verweis auf die Variablendefinition (BMAIT-Variablen.csv) hergestellt wird. Schaltet den Autostart aus.

Ladet anschließend die beiliegende Konfiguration (BMSAIT_demoDED.ini). BMSAIT sollte nun die geladene Definition anzeigen (ein COM-Port und mehrere Variablen).

Achtung: Bei der Abbildung des DED entsteht viel Datenverkehr zum Arduino. Dies kann schnell dazu führen, dass Datenpakete verloren gehen, wenn der Arduino die Daten nicht schnell genug einliest. Um eine gute Synchronisation bei der Datenübertragung zu gewährleisten, sollte der Datenverkehr bei diesem Programm **über das PULL-Prinzip laufen**.

Macht einen Rechtsklick auf den COM-Port und bearbeitet diesen. Wählt den COM-Port aus, an dem euer Arduino angeschlossen ist. Wenn ihr nicht sicher seid, welcher COM-Port dies ist, dann wählt entweder die SCAN Funktion und schaut, auf welchem COM-Port der Arduino eine Antwort sendet oder ihr schaut in dem Windows-Gerätemanager nach.

Ich empfehle die Änderungen nun zu sichern („Speichern unter“ und Auswahl einer neuen Datei).

Startet die Verarbeitung. Wenn alles geklappt hat, wird das DED nach wenigen Sekunden ein Testbild anzeigen.

5. Ergebnis

Das DED wird nur angezeigt, wenn der Arduino aktuelle Daten zur Anzeige erhält. Werden keine Daten empfangen, wird das DED nach 5 Sekunden abgeschaltet.

Startet FalconBMS und betretet die 3D Welt. Das DED sollte nun wieder starten und die aktuellen Daten aus der 3D anzeigen.

Hinweis: Besteht keine Verbindung zu BMSAIT, wird der Arduino das OLED nach 10 Sekunden in einen Ruhemodus versetzen. Die Anzeige wird wieder aktiv, sobald eine Verbindung zu BMSAIT hergestellt wurde. Das Testbild verschwindet, sobald sich das Spiel in der 3D Welt befindet.