

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm zur Abbildung des Speedbrake Indicator



Autor	Robin „Hummer“ Bruns
Dokumentversion	1.1
Softwareversion	1.3.1
BMS Version	4.35
Datum	17.02.2021

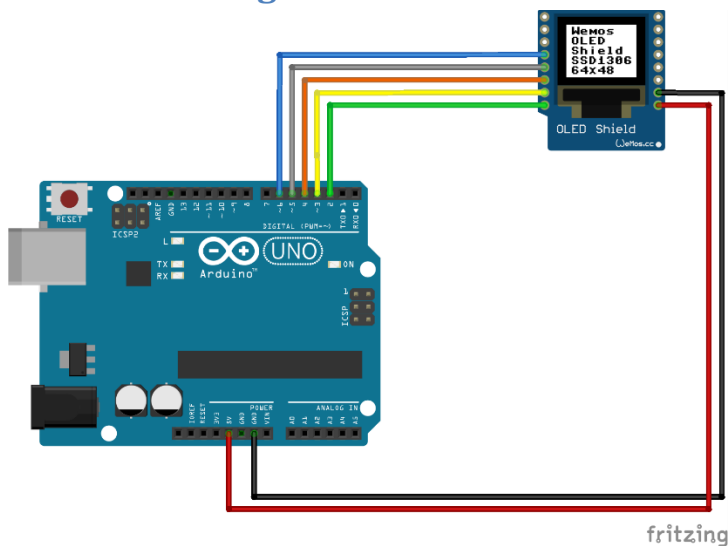
1. Überblick

Das vorliegende Beispielprogramm demonstriert die Abbildung des Speedbrake Indicator auf einem monochromen OLED display.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein UNO)
- Ein OLED display (in diesem Beispiel: SSD1306 128x64, Größe 0.96“ mit 4-Kabel SPI Verbindung).
- Verbindungskabel

2. Verkabelung



PIN Arduino	PIN OLED
GND	GND
5V	Vcc
2	D0
3	D1
4	RES
5	DC
6	CS

Beachtet, dass die Verkabelung des OLED displays davon abhängt, auf welche Kommunikationsform das OLED eingestellt ist. Ihr könnt dies durch Setzen/Entfernen von Kontakten auf der Rückseite des OLED beeinflussen.

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 aus dem Ordner \Arduino Sketch\BMSAIT_SBI\ 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.

Bei OLEDs ist es wichtig den richtigen „Konstruktor“ für die Ansteuerung des OLEDs zu wählen. Das Vorgehen habe ich in der Dokumentation von BMSAIT beschrieben.

Nachdem der Konstruktor im Sketch geprüft wurde und ihr das richtige Arduino-Board ausgewählt habt, ladet ihr das Sketch auf den Arduino hoch.

Nach dem Hochladen sollte das Display kurz das „off“ Anzeige (Streifen) anzeigen. Diese verschwinden nach 10 Sekunden.

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. Wählt das PUSH-Prinzip und schaltet den Autostart aus.

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

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).

Aktiviert den Testmodus und startet die Verarbeitung. Wenn alles geklappt hat, sollte nun die „CLOSED“ Anzeige erscheinen.

5. Ergebnis

Deaktiviert den Testmodus und startet Falcon BMS im Freien Flug. Öffnet die Luftbremse und prüft, ob die Anzeige den geöffneten Status anzeigt (9 Kreise). Schließt die Luftbremse und prüft, ob die Anzeige wieder zu „CLOSED“ wechselt.

Die „off“ Anzeige (gestrichelte Anzeige) erscheint kurzzeitig beim Öffnen/Schließen und wenn die Stromversorgung des Flugzeugs nicht hergestellt ist (z.B. beim Rampstart).

Hinweis: Solange keine Verbindung mit BMSAIT besteht, wird der Arduino das OLED in einen Ruhezustand versetzen und das Display nach 10 Sekunden abschalten. Zum Test muss daher eine Verbindung hergestellt werden.