Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm zur Abbildung des Speedbrake Indicator über einen ServoMotor

Autor	Robin „Hummer“ Bruns
Dokumentversion	1.0
Softwareversion	1.3.8
BMS Version	4.35u3
Datum	21.04.2022

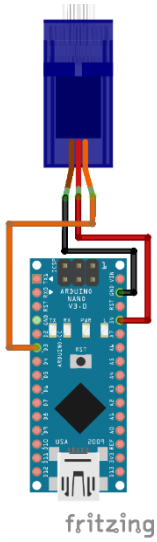
1. Überblick

Das vorliegende Beispielprogramm demonstriert die Abbildung des Speedbrake Indicator, wobei ein Servo-Motor benutzt wird, um die Symbolik eines Speedbrake-Indicator zur Anzeige zu bringen.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein UNO)
- Einen 5V Mico-Servo motor (bspw. SG90 9g Servo)
- Verbindungskabel

2. Verkabelung



PIN Arduino	PIN Servo
GND	GND (braun)
5V	Vcc (rot)
3	Signal (orange)

Bei der Verkabelung des Servo-Motors ist zu beachten, dass die Signalübertragung über das sogenannte PWM (Pulse Width Modulation) erfolgt. Nut bestimmte PINs des Arduino sind in der Lage diese Signalart zu übertragen. Prüft daher, welche PINs hier bei eurem Arduino-Board in Frage kommen[1]. Beim Arduino Uno oder Nano sind dies beispielsweise die PIN D3,D5,D6,D9,D10 oder D11.

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_SBIServo\ mit einem Doppelklick auf. Das Sketch wird in der Arduino IDE geladen. Wenn ihr die Verkabelung gem. Kapitel 2 durchgeführt habt, sind hier erst einmal keine Anpassungen erforderlich.

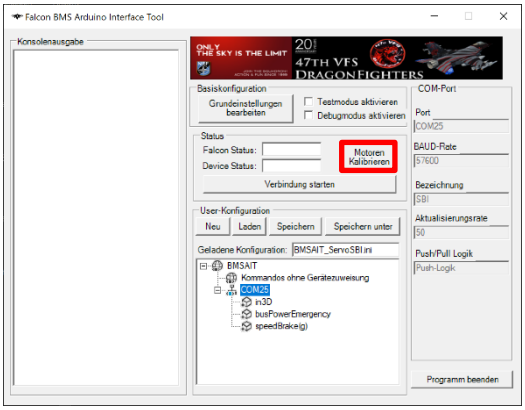
Soll der Motor an einen anderen PIN angeschlossen werden, muss das im Arduino Code bei der Option SBI_PIN angepasst werden:

[1] https://www.arduino.cc/reference/de/language/functions/analog-io/analogwrite/

```
#include <Servo.h>

#define SERVODELAY 200          //time (ms) the servo will pause after movement
#define SBIDELAY 500            //time (ms) the off-flag will show up when the

#define SBI_PIN              3  //Arduino PIN the servo is conncted to
#define SBI_OFF_ANGLE       90  //servo movement angle for the "off" symbol
#define SBI_CLOSED_ANGLE   135  //servo movement angle for the "closed" symbol
#define SBI_OPEN_ANGLE      55  //servo movement angle for the "open" symbol

```

Nach dem Hochladen sollte der Servo kurzzeitig die drei möglichen Stellungen des Speedbrake Indicator anzeigen und abschließend auf der „off“ Anzeige (Streifen) stehen bleiben.

Mit der Windows-App kann eine Motor-Kalibrierung durchgeführt werden.



Prüft dabei, ob die Anzeigen präzise funktionieren. Sollte eine der drei SBI Positionen nicht korrekt dargestellt werden, da das Symbol zu hoch oder zu tief dargestellt wird, dann ist eine manuelle Anpassung erforderlich, die im Arduino-Code hinterlegt wird.

```
#include <Servo.h>

#define SERVODELAY 200          //time (ms) the servo will pause after movement
#define SBIDELAY 500            //time (ms) the off-flag will show up when the

#define SBI_PIN              3  //Arduino PIN the servo is conncted to
#define SBI_OFF_ANGLE       90  //servo movement angle for the "off" symbol
#define SBI_CLOSED_ANGLE   135  //servo movement angle for the "closed" symbol
#define SBI_OPEN_ANGLE      55  //servo movement angle for the "open" symbol

```

Ändert bei Bedarf den Winkel für die Symbole und ladet den Code erneut auf den Arduino. Prüft anschließend, ob die Anzeige nun wie gewünscht funktioniert.

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.

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

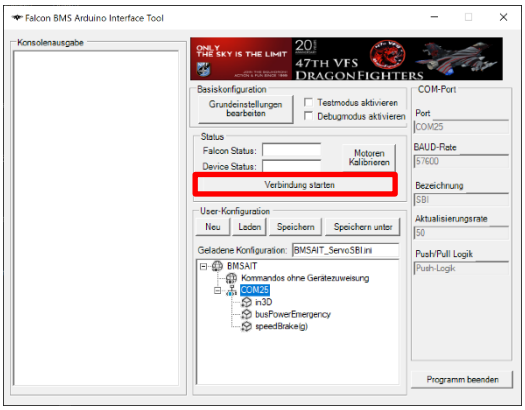
Macht einen Rechtsklick auf den COM-Port und bearbeitet diesen. Wählt den COM-Port aus, an dem euer Arduino angeschlossen ist. Wenn ihr nicht sicher seid, welcher COM-Port dies ist, dann wählt

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).

Startet nun die Verbindung zum Arduino.

Der Speedbrake Indicator wird sich kurz bewegen und dann in der „off“ Stellung stehen bleiben.



5. Ergebnis

Startet Falcon BMS im Freien Flug. Mit dem Einstieg in die 3D Welt wird der Speedbrake Indicator den Wert in der Simulation annehmen.

Im freien Flug sollte die Speedbrake geschlossen sein und der entsprechende Wert auf dem Speedbrake Indicator erscheinen. Öffnet die Luftbremse und prüft, ob die Anzeige in den geöffneten Status wechselt. 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).