

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm Schaltermatrix



Autor	Robin „Hummer“ Bruns
Dokumentversion	1.0
Softwareversion	1.0
BMS Version	4.34u3
Datum	3.10.2020

1. Überblick

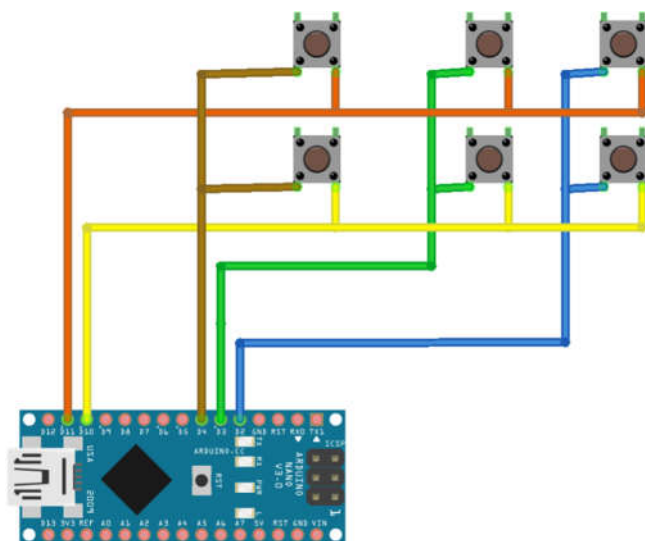
Das vorliegende Beispielprogramm demonstriert die Ansteuerung einer großen Anzahl von Schaltern über einen Arduino. Die Taster werden in diesem Beispiel nicht einzeln angeschlossen, sondern über eine Matrix verkabelt. Dies ermöglicht es, dass ein Arduino mit beispielsweise 20 freien Datenpins (Ein Uno kann 12 digitale und 8 analoge PINs bereitstellen) nicht nur 20, sondern bis zu 100 (10x10) Taster ansteuern kann. In diesem Beispiel werden nur 6 Taster angesteuert, was aber für die Erklärung des Prinzips dieses Moduls ausreichen sollte.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein NANO)
- 6 Taster
- Verbindungskabel

2. Verkabelung

Schließt die Taster wie dargestellt an.



fritzing

3. Programmierung des Arduino

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

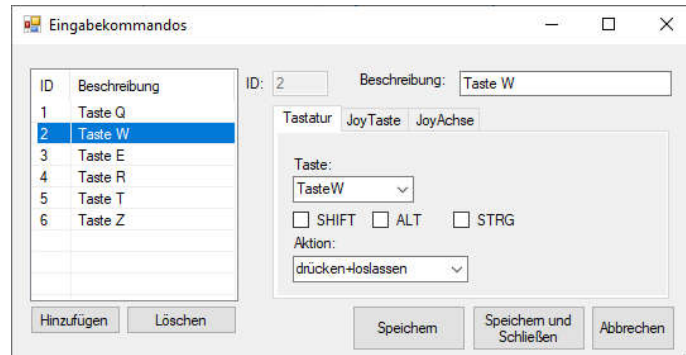
Ruft nun die .ino aus dem Ordner \Arduino Sketch\BMSAIT_ButtonMatrix\ mit einem Doppelklick auf. Das Sketch wird in der Arduino IDE geladen. Wenn ihr die Verkabelung gem. Kapitel 2 durchgeführt habt, sind hier keine Anpassungen erforderlich.

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

Ladet anschließend die beiliegende Konfiguration (BMSAIT-DemoButtonMatrix.ini). BMSAIT sollte nun die geladene Definition anzeigen (nur ein COM-Port, keine Variablen). Über einen Klick auf die Schaltfläche „Eingabekommandos bearbeiten“ sind die eingerichteten Kommandos sichtbar (siehe rechts).



Hier wurden sechs Kommandos für einfache Tastatureingaben (q,w,e,r,t,z) abgebildet. Wenn ihr wollt, könnt hier nun andere Tastaturbefehle hinterlegen, die einem Tastaturkommando in BMS entsprechen (z.B. ICP Tasten). Dann könnt ihr die Funktion auch in der 3D Welt des FalconBMS testen.

Schließt nun die Eingabeverarbeitung und kehrt zum Hauptfenster der BMSAIT Software zurück. 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 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 ihr einen Button oder Schalter betätigt, sollte in der Konsole des BMSAIT in Windows der Hinweis kommen, dass ein Kommando erkannt wurde. Ihr könnt zudem ein Schreibprogramm öffnen und prüfen, ob die Buttons die entsprechenden Buchstaben auslösen.

Deaktiviert den Testmodus und startet die Verarbeitung des BMSAIT.

5. Einstellungen FalconBMS

Startet Falcon BMS und öffnet das Setup der Eingabegeräte. Wenn ihr hier nun die Taster. Das Setup sollte die Eingaben erkennen und ihr könnt diese einem Callback in BMS zuordnen.

Alternativ könnt ihr natürlich auch bereits vergebene Tastaturkommandos herausuchen und diese den Kommandos in BMSAIT hinterlegen.

6. Ergebnis

Wurde den Tastenkommandos ein Tastatursignal hinterlegt, das einem Callback in BMS entspricht, könnt ihr nun einen freien Flug starten und prüfen, ob mit den Tastern der gewünschte Cockpitbefehl ausgelöst wird.