

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm zur Ansteuerung eines Stepper-Motors



| | |
|-----------------|----------------------|
| Autor | Robin „Hummer“ Bruns |
| Dokumentversion | 1.0 |
| Softwareversion | 1.8.3 |
| BMS Version | 4.34u5 |
| Datum | 22.02.2022 |

1. Überblick

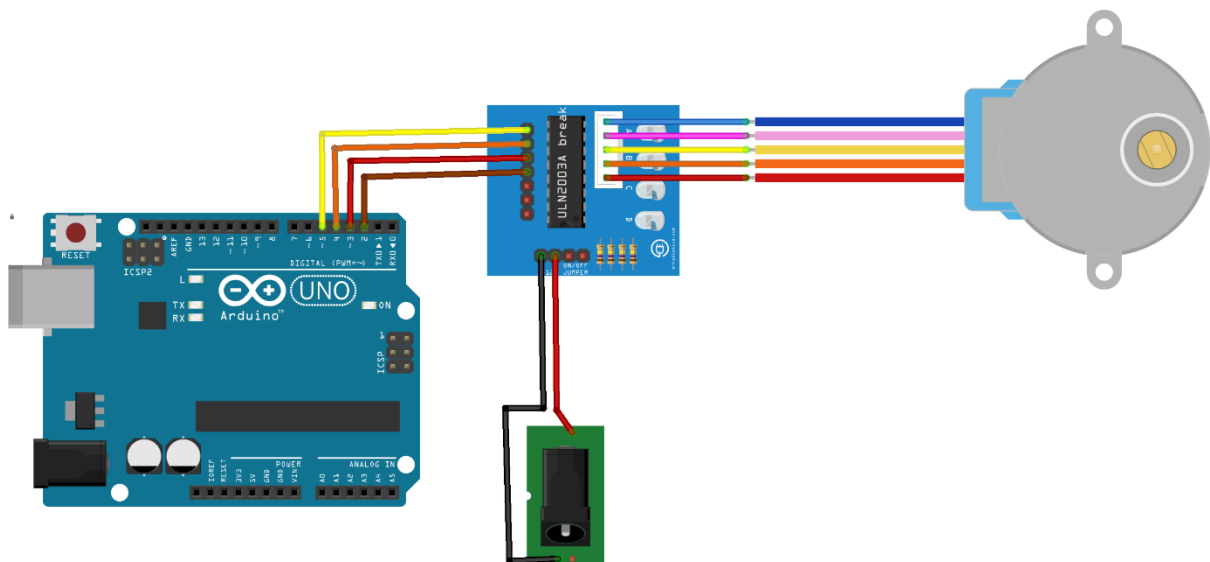
Das vorliegende Beispielprogramm demonstriert die Ansteuerung eines einfachen StepperMotors. Als Beispiel soll der Motor eine Kompassrose bewegen und das aktuelle Heading anzeigen.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein UNO)
- Einen Stepper Motor 28BYJ-48
- Ein ULN2003 Controller Board

Verbindungskabel

2. Verkabelung



fritzing

| Arduino | ULN2003 |
|---------|---------|
| 2 | IN1 |
| 3 | IN2 |
| 4 | IN3 |
| 5 | IN4 |

Schließt den Stepper wie dargestellt an die PINs 2,3,4 und 5 des Arduino Board an. Beachtet, dass Ihr Motoren möglichst nie direkt mit dem Arduino verbindet, da Spannungsspitzen diesen beschädigen können. Nutzt Controllerplatinen, die eine externe Spannungsversorgung für den Motor erlauben und über notwendige Schutzdioden verfügen.

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_Stepper28BYJ\ 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-28BJY-48 Demo.ini). BMSAIT sollte nun die geladene Definition anzeigen (ein COM-Port und eine Variable).

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).

Führt nun eine Kalibrierung durch, um die Anzeige auf einen definierten Nullpunkt zu setzen. Im vorliegenden Beispiel wird das durch einen Button erzeugt, der an PIN 11 angeschlossen ist. Solange der Button gedrückt ist, wird der Motor sich im Kreis drehen. Sobald der Nullpunkt erreicht ist, muss der Button losgelassen werden.

Startet nun die Verarbeitung des BMSAIT und startet Falcon BMS im Freien Flug.

Beim Einstieg in die 3D Welt wird die Anzeige sich in Richtung des aktuellen Headings drehen und dies auch beim Kurvenflug der F-16 beibehalten. Testet vor allem, ob der Übergang von 359° zu 0° und anders herum problemlos funktioniert.