

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm zur Ansteuerung eines Air Core Motors



Autor	Robin „Hummer“ Bruns
Dokumentversion	1.0
Softwareversion	1.0
BMS Version	4.34u3
Datum	24.07.2024

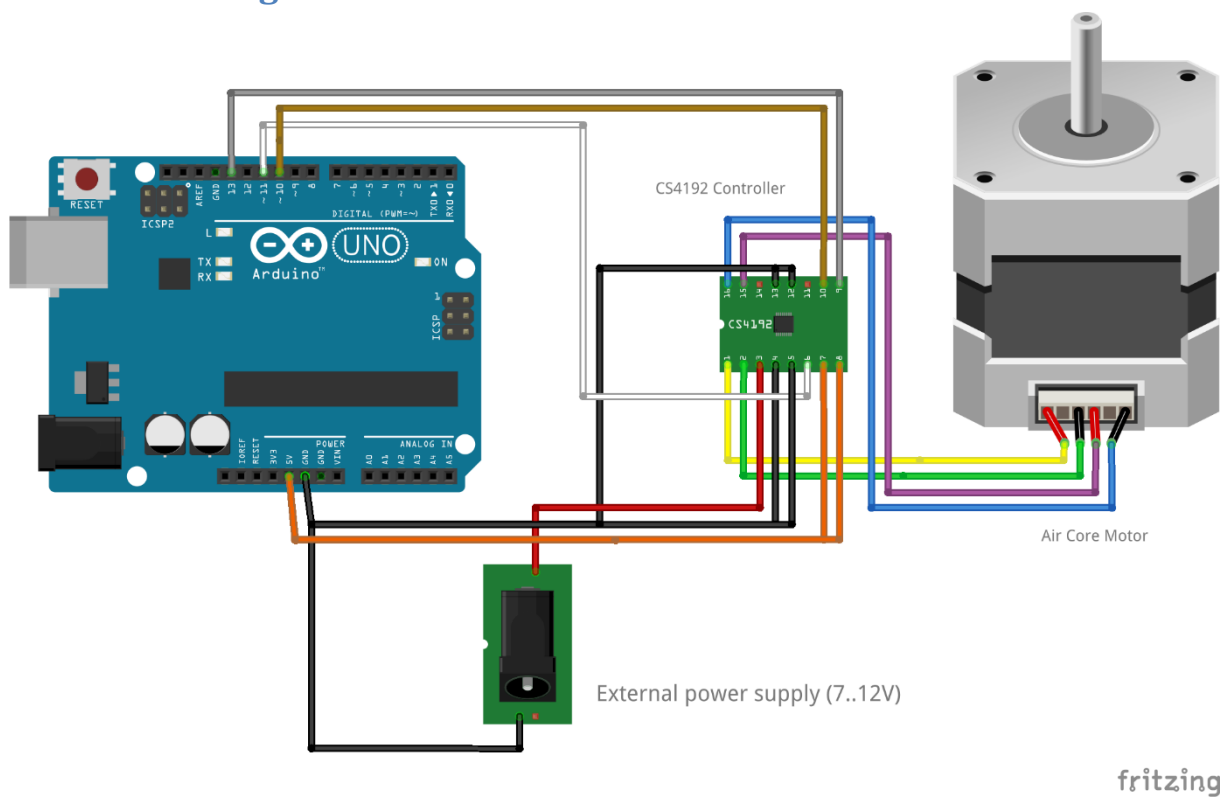
1. Überblick

Das vorliegende Beispielprogramm demonstriert die Ansteuerung eines Air Core Motors. Als Beispiel soll die RPM angezeigt werden.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein Uno)
- Einen Air Core Motor
- Einen CS4192 Controller zur Ansteuerung des Air Core Motors
- Verbindungskabel

2. Verkabelung



CS4192	Air Core	Arduino	Ext Power
1	SIN-	1	
2	SIN+	2	
3	V _{BB}		+
4	GND	GND	-
5	GND	GND	
6	SI	11	
7	V _{CC}	5V	
8	OE	5V	

CS4192	Air Core	Arduino	Ext Power
9	SCLK	13	
10	CS	10	
11	ST		
12	GND	GND	
13	GND	GND	
14	SO		
15	COS-	4	
16	COS+	3	

Der Motor wird über 4 Kabel mit den PINs Cos+, Cos-, Sin+ und Sin- des CS4192 Controller verbunden. Ggf. müsst ihr ausprobieren, wo diese Kabel am Motor verbunden werden müssen, da die Motoren oft Gebrauchtware sind und keine eindeutige Dokumentation für diese vorliegt.

Der CS 4192 Controller wird dann wie gezeigt mit dem SPI-PINs des Arduino verbunden. Solltet ihr mehrere Motoren/Controller an einem Arduino betreiben, werden alle Controller gemeinsam an die Daten-PINs 11 und 13 angeschlossen. Nur der CS-Anschluss muss für jeden Controller individuell vergeben werden.

Ich habe einen einzelnen Air Core Motor auch direkt über einen Arduino ansteuern können. Der Anschluss V_{BB} ist dazu ebenfalls mit dem 5V Ausgang des Arduino zu verbinden. Die Nutzung einer externen Spannungsquelle ist daher optional.

Sollten mehrere Motoren an einem Arduino betrieben werden, empfehle ich aber eine zusätzliche Spannungsquelle (7..12V). Achtet darauf, dass die GND-Anschlüsse der externen Spannungsquelle und des Arduino verbunden sind.

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 BMSAIT_AirCore.ino aus dem Ordner \Arduino Sketch\BMSAIT_AirCore\ 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- DemoAirCore.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).

Startet nun eine Kalibrierung über die entsprechende Schaltfläche der BMSAIT Windows App oder dem Menüeintrag bei Rechtsklick auf den COM-Port. Wenn alles geklappt hat, sollte der Motor nun einmal den gesamten Anzeigebereich abfahren und anschließend in die Nullstellung zurückkehren. Wenn nicht, dann versucht die Kalibrierung nochmal anzustoßen.

5. Ergebnis

Druckt euch ggf. eine RPM Anzeige aus, die ihr für den Test auf den Motor legt, um die Anzeige besser prüfen zu können.

Startet nun die Verarbeitung im BMSAIT. Startet Falcon BMS und ruft einen freien Flug auf. Bewegt die Throttle und prüft, ob der Motor die Bewegungen der RPM Anzeige richtig abbildet.