

Falcon BMS to Arduino Interface Tool (BMSAIT)

Beispielprogramm Servoansteuerung über ein PWM Shield



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

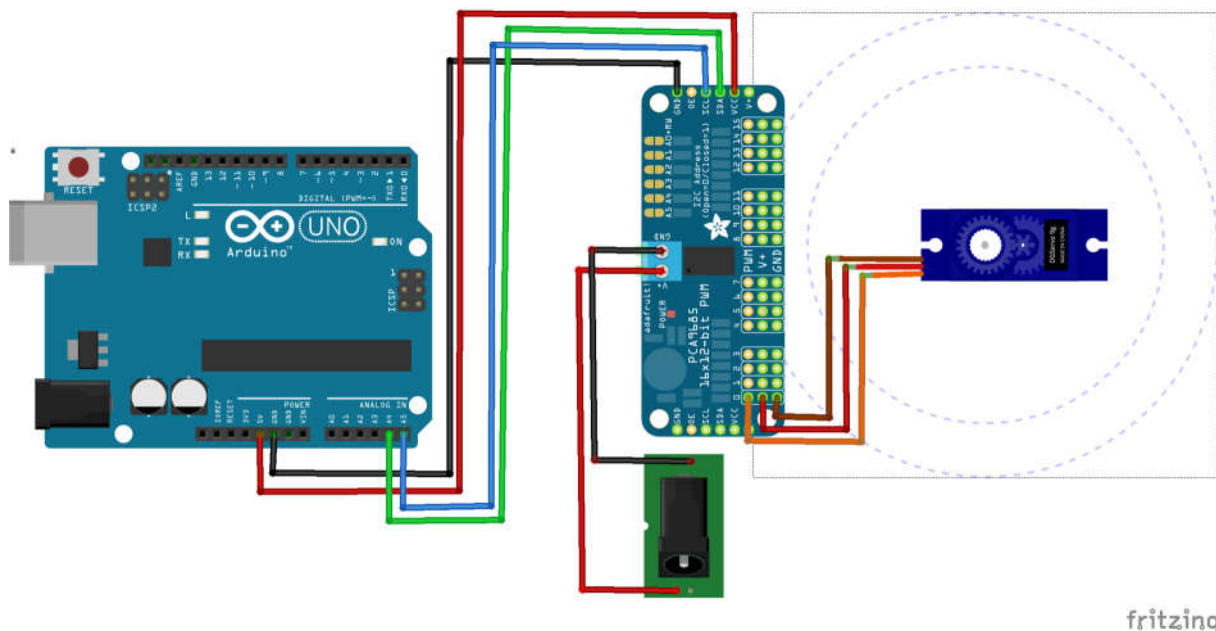
1. Überblick

Das vorliegende Beispielprogramm demonstriert die Ansteuerung von Servos, die über ein PWM Shield angeschlossen werden. Als Beispiel soll die RPM angezeigt werden.

Um das Beispiel auszuprobieren benötigt ihr:

- Ein Arduino-Board (z.B. ein UNO)
- Ein PWM Shield (Adafruit PCA9685)
- Ein Servomotor (9g SG90)
- Eine externe Spannungsversorgung (5-6V)
- Verbindungskabel

2. Verkabelung



Schließt das PWM Shield wie dargestellt an die PINs A4 und A5 sowie V_{cc} (+5V) und GND des Arduino Board an. Das PWM Shield kann an eine externe Spannungsversorgung angeschlossen werden. Bei nur einem angeschlossenen Motor reicht die Spannungsversorgung, die der Arduino mitbringt, aus. Wenn ihr meint, dass die Spannungsversorgung des Arduino ausreicht, dann verbindet den V+ PIN des PWM Board mit der V_{cc} des Arduino (in der Grafik nicht dargestellt).

Wenn ihr aber mehrere Motoren ansteuern wollt, dann schließt bitte eine externe Versorgung an (siehe Grafik).

Das Board bietet Platz für bis zu 16 Motoren. Schließt für das hier vorliegende Beispiel einen Motor an den Port 0 des PWM Shield an.

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_ServoPWM\ 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.

Beachtet, dass ihr für dieses Programm eine Arduino-Bibliothek benötigt, die installiert werden muss (Adafruit_PWMServoDriver).

Nachdem ihr das richtige Arduino-Board ausgewählt und die erforderliche Bibliothek installiert 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-PWMServo Demo.ini). BMSAIT sollte nun die geladene Definition anzeigen (ein COM-Port und eine Variable).

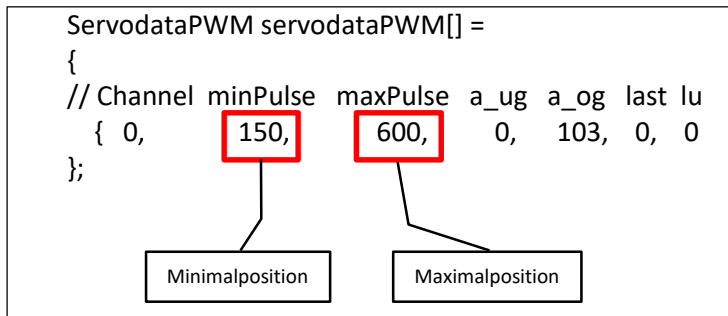
Macht einen Rechtsklick auf den COM-Port und bearbeitet diesen. Wählt den COM-Port aus, an dem euer Arduino angeschlossen ist. Wenn ihr nicht sicher seid, welcher COM-Port dies ist, dann wählt entweder die SCAN Funktion oder ihr schaut in dem Windows-Gerätemanager nach.

Ich empfehle die Änderungen nun zu sichern („Speichern unter“ und Auswahl einer neuen Datei).

5. Begrenzungen einrichten/anpassen

Startet eine Kalibrierung über die Schaltfläche in der BMSAIT Windows App. Der Motor sollte nun in die Minimal- und Maximalposition fahren und dort jeweils eine Sekunde verbleiben, bevor er in die Mittelstellung zurückkehrt. Prüft in dieser Zeit, ob der Arduino dort fühlbar oder hörbar vibriert.

Wenn ja, dann versucht der Servo eine Position einzunehmen, die aufgrund der Hardwarebeschränkungen nicht erreicht werden kann. Dies wird den Servo wohl nicht beschädigen, sollte aber grundsätzlich vermieden werden. Eine Anpassung der Software erlaubt die Minimal- und Maximalwerte individuell festzulegen und Anschläge an den Hardwarebegrenzungen zu verhindern. Dies wird im Arduino-Sketch in der Datei BMSAIT_Servo.h gespeichert. Ihr müsst die dort angegebenen Werte manuell korrigieren. Wenn der Arduino in der Minimalposition (erste Drehung im Uhrzeigersinn nach Initialisierung) vibriert, ist der Minimalwert zu erhöhen (um ca. 5). Wenn der Arduino in der Maximalposition (Drehung gegen Uhrzeigersinn nach Initialisierung) vibriert, ist der Maximalwert zu verringern (um ca. 5). Als ersten Ansatz solltet ihr 150 als Minimalwert und 600 als Maximalwert verwenden.



6. Ergebnis

Startet die Verarbeitung der BMSAIT Software und startet den Falcon in einem freien Flug. Bewegt die Throttle um zu prüfen, ob der Motor der Änderung der RPM folgt. Um eine genaue Prüfung durchzuführen, könnt ihr den Servo an eine Nadel anschließen und eine Skala darunterlegen. Aufgrund der Winkelbeschränkung des Servos ist dabei in diesem einfachen Beispiel nur eine lineare Skala von 180° möglich (0° entspricht 0% RPM, 180° entspricht 110% RPM).

