

# Falcon BMS to Arduino Interface Tool (BMSAIT)

## Beispielprogramm zur Abbildung des ECM Panel



Autor	Robin „Hummer“ Bruns
Dokumentversion	1.0
Softwareversion	1.3.12
BMS Version	4.37
Datum	23.01.2023

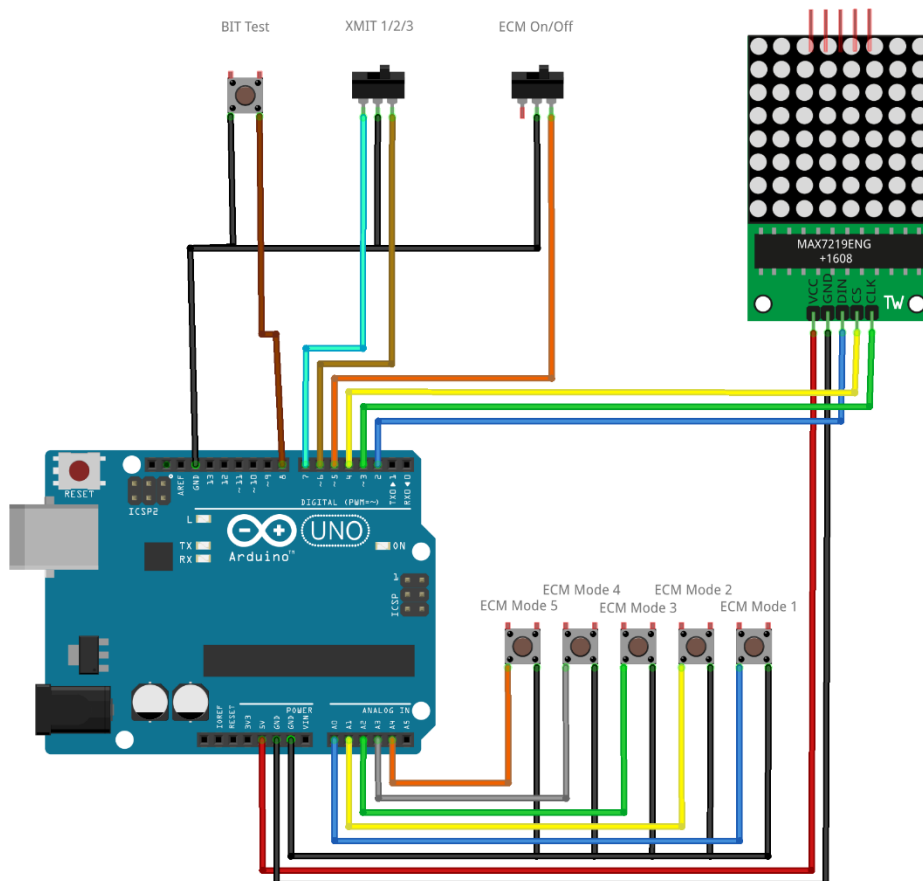
## 1. Überblick

Das vorliegende Beispielprogramm stellt eine Möglichkeit dar, um das ECM Panel mit einem Arduino anzusteuern.

Um dieses Beispiel nachzubauen benötigt ihr:

- Ein Arduino-Board (z.B. ein UNO)
- Einen Kippschalter ON/OFF
- Einen Kippschalter ON/OFF/ON
- Sechs Taster
- Eine 8x8 LEDMatrix Platine mit Max7219 Controller
- 4 rote LED
- 4 blaue LED
- 4 grüne LED
- 4 gelbe LED
- Widerstände (ca.220 Ohm)
- Verbindungskabel

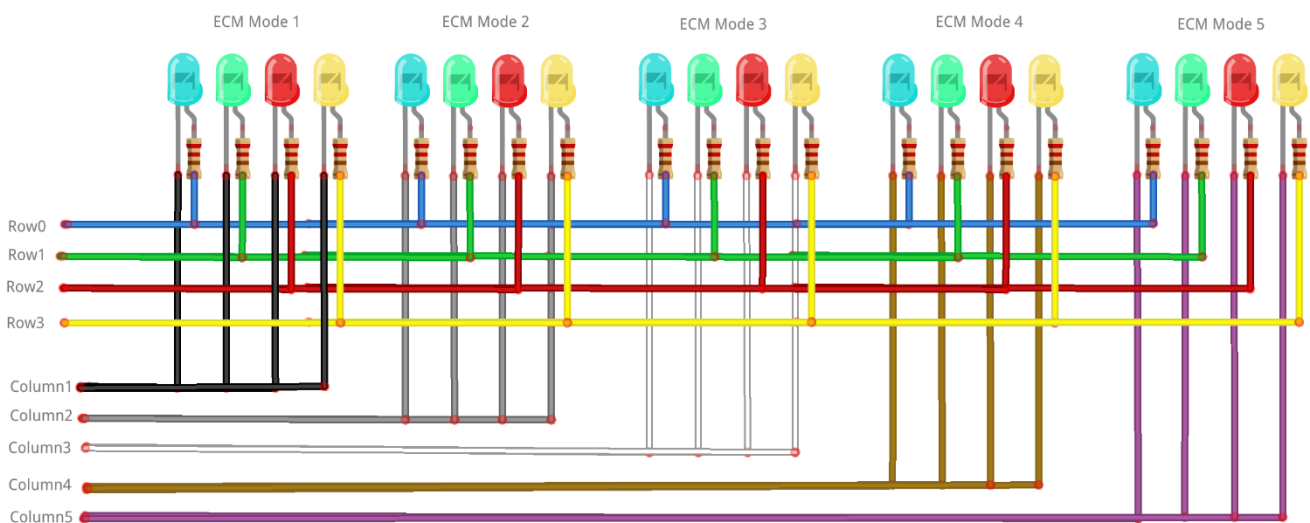
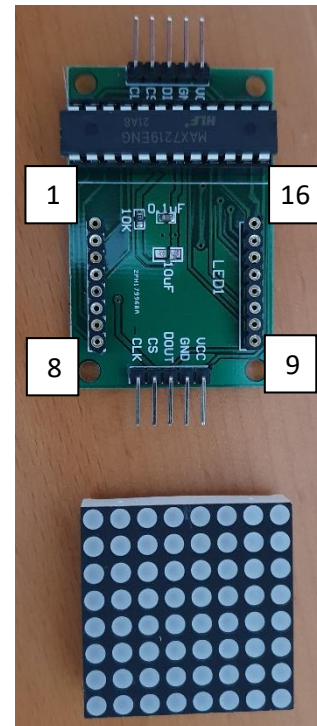
## 2. Verkabelung



PIN Arduino	Anschluss Schalter
GND	GND
5	Schalter ECM on/off
6	Schalter XMIT 1 – 2
7	Schalter XMIT 3 – 2
8	Taster BIT Test

PIN Arduino	Anschluss Max7219
GND	GND
5V	Vcc
2	DIN
4	CS
3	CLK

PIN MAX7219	Zuordnung Matrix	Beispiel
1	Spalte 2 (-)	ECM Mode 2
2	Spalte 5 (-)	ECM Mode 5
3	Zeile 4 (+)	
4	Spalte 7 (-)	
5	Zeile 2 (+)	Rote LED (F)
6	Zeile 1 (+)	Grüne LED (A)
7	Spalte 6 (-)	
8	Spalte 4 (-)	ECM Mode 4
9	Spalte 0 (-)	
10	Zeile 3 (+)	Gelbe LED (S)
11	Zeile 5 (+)	
12	Spalte 3 (-)	ECM Mode 3
13	Zeile 0 (+)	Blaue LED (T)
14	Spalte 1 (-)	ECM Mode 1
15	Zeile 6 (+)	
16	Zeile 7 (+)	



### 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\_ECM\ 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.

Es ist möglich eine andere Verkabelung als in Kapitel 2 gezeigt zu wählen, allerdings sind dann Anpassungen an der Konfiguration des Arduino Sketch erforderlich, bevor dieses auf den Arduino geladen wird.

## 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.

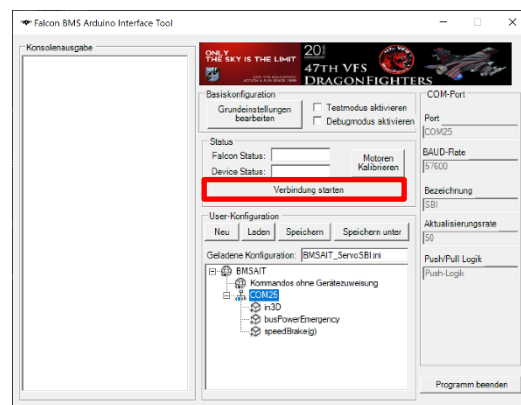
Ladet anschließend die beiliegende Konfiguration (BMSAIT\_demoECM.ini). BMSAIT sollte nun die geladene Definition anzeigen (ein COM-Port, 20 Variablen und 11 Kommandos).

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.

In der Kommandoverwaltung sind die Signale zu prüfen, die bei Tastendruck oder Schalterbetätigung auszulösen sind. Weist hier geeignete Tastatursignale oder einen DirectX Button zu.

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

Startet nun die Verbindung zum Arduino.



## 5. Einstellungen in Falcon BMS

Mit aktivierter Verbindung ist im Setup von Falcon BMS zu prüfen, ob bei Schalterbewegung oder Tasterbetätigung die richtigen Eingabesignale registriert werden. Sollten ihr in BMSAIT DirectX-Buttons vergeben haben, müssen diese im Setup von Falcon BMS noch den Callbacks für das ECM zugeordnet werden.

## 6. Ergebnis

Startet Falcon BMS in der Instant Action. Mit dem Einstieg in die 3D Welt werden die LED des ECM Panels die Anzeige der Simulation übernehmen (wenn nichts leuchtet, dann prüft, ob eure Maschine in der Instant Action mit einem ECM ausgestattet ist – Im Balkan Theater ist das z.B. nicht der Fall; wechselt zum Testen zu KTO oder erstellt eine TE, in der die Maschine ein ECM trägt).

Nutzt die CMS-Tasten am HOTAS, um den ECM Consent ein-/auszuschalten. Die Anzeigen auf dem ECM Panel sollten zwischen S(tandby) und (A)ctive sowie ggf. T(ransmit) wechseln. Beim Ausschalten des ECM sollten alle LED erlöschen und beim ECM-BIT-Test alle Lampen aufleuchten.