Falcon BMS to Arduino Interface Tool (BMSAIT)

Example program CMDS (Countermeasure Dispenser System)

Author	Robin "Hummer" Bruns
Document version	1.0
Version	1.4.4
BMS Version	4.37
Date	23.4.2025

1. Overview

In the previous examples, only individual functions were shown as examples. This example will combine several individual components into a complex project. The CMDS module is available in several variants, which differ in the used hardware.

The basic function, which is the same for all variants, is that the functions of the CMDS are to be mapped. This includes:

- Display of available chaff and flare
- Warning indicator when the Chaff/Flare Bingos fall below
- Control of status LEDs
- Control of the switches of the CMDS panel

This example has its own module in BMSAIT to map special features of the CMDS panel:

- If the aircraft's power supply is switched off, the displays are switched off
- If the CMDS is switched off, the displays are switched off
- If the switches for activating the chaff/flare are not activated, the respective displays are switched off

Variant 1 of the illustration of the CMDS:

Variant 1 represents the minimum variant for the implementation or outputs of the CMDS panel. A single Max7219 display tube is used to display both the chaff and the flare display.

figures 1-2	figures 3 to 4	figures 5-6	figures 7-8
Flare "LO" Warning	Flare Number (0-99)	Chaff "LO" Warning	Chaff Number (0-99)

If you want, you can remove the displays from the board of the MAX7219 and then install them separately in the panel. This causes some soldering and, admittedly, some cable clutter, but you can get by with one display and then place the displays independently - or even use your own display elements in other sizes or colours.

Required hardware:

- An Arduino board (e.g. a UNO)
- A Max7219 7-segment tube
- Four LED (2.3V, Green)
- Four resistors (220 ohms)
- One rotary switch with six detents
- Two toggle switches (ON-OFF)
- Connecting cable

Variant 2 of the illustration of the CMDS:

As an alternative, the displays to the Chaff and Flare are not made possible via 7-segment displays, but via an LCD display.

Required hardware:

- An Arduino board (e.g. a UNO)
- An LCD display (16x2) with i2c connection
- Four LED (2.3V, Green)
- Four resistors (220 ohms)
- One rotary switch with six detents
- Two toggle switches (ON-OFF)
- Connecting cable

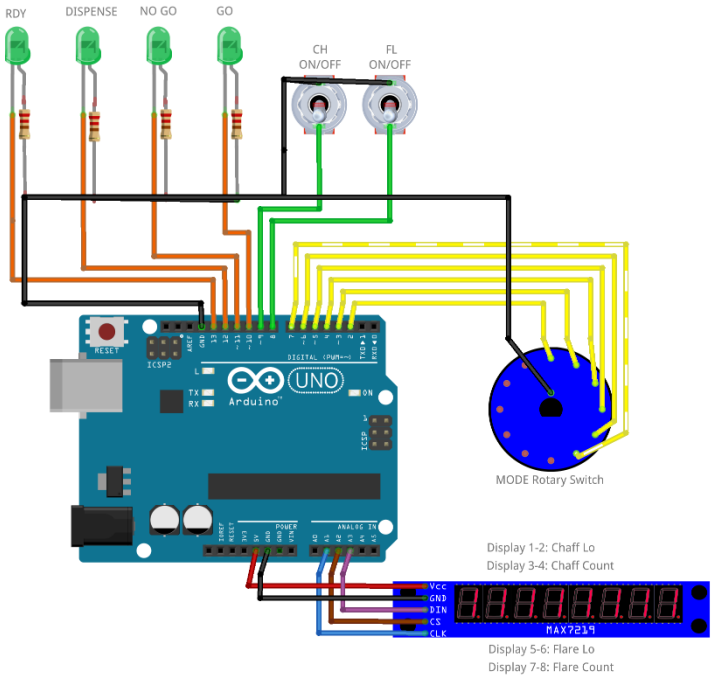
Variant 3 of the illustration of the CMDS:

Variant 3 represents a complete representation of all inputs and outputs of the CMDS mode via a single Arduino.

- An Arduino board (e.g. a UNO)
- One Max7212 7-segment tube
- Four LED (2.3V, Green)
- Four resistors (220 ohms)
- One rotary switch with six detents
- One rotary switch with four detents
- 10 resistors 2.2 kOhm
- Seven toggle switches (ON-OFF)
- Connecting cable

2. Wiring

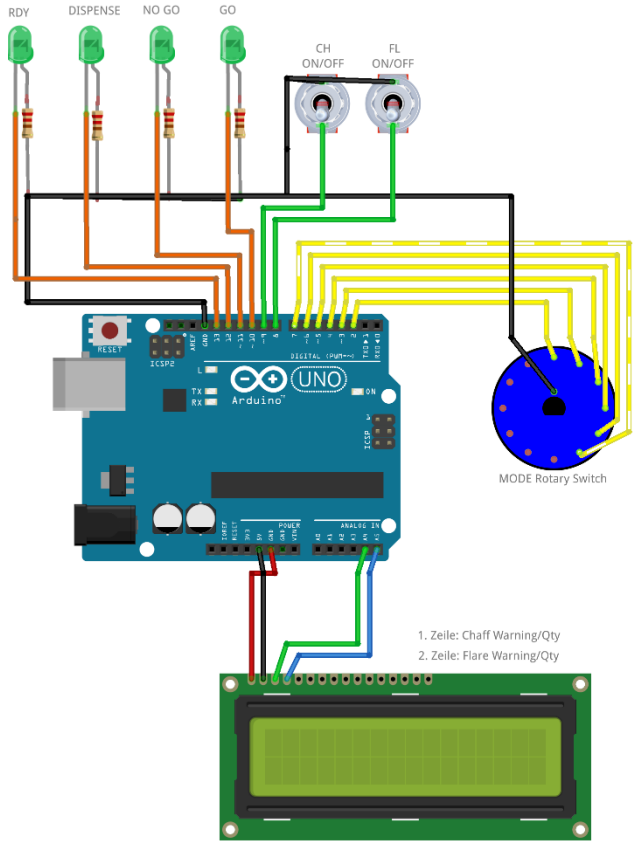
2.1. Variant 1



Arduino	MAX7219	LED	Opposition
A1	CLK		
A2	CS		
A3	DIN		
10		GO	GND
11		NOGO	GND
12		DISP	GND
13		OK	GND

Switch	Arduino PIN	Opposition
MODE-OFF	2	GND
MODE-STBY	3	GND
JMODE-MAN	4	GND
FASHION SEMI	5	GND
FASHION CAR	6	GND
FASHION BYP	7	GND
CHAFF	8	GND
FLARE	9	GND

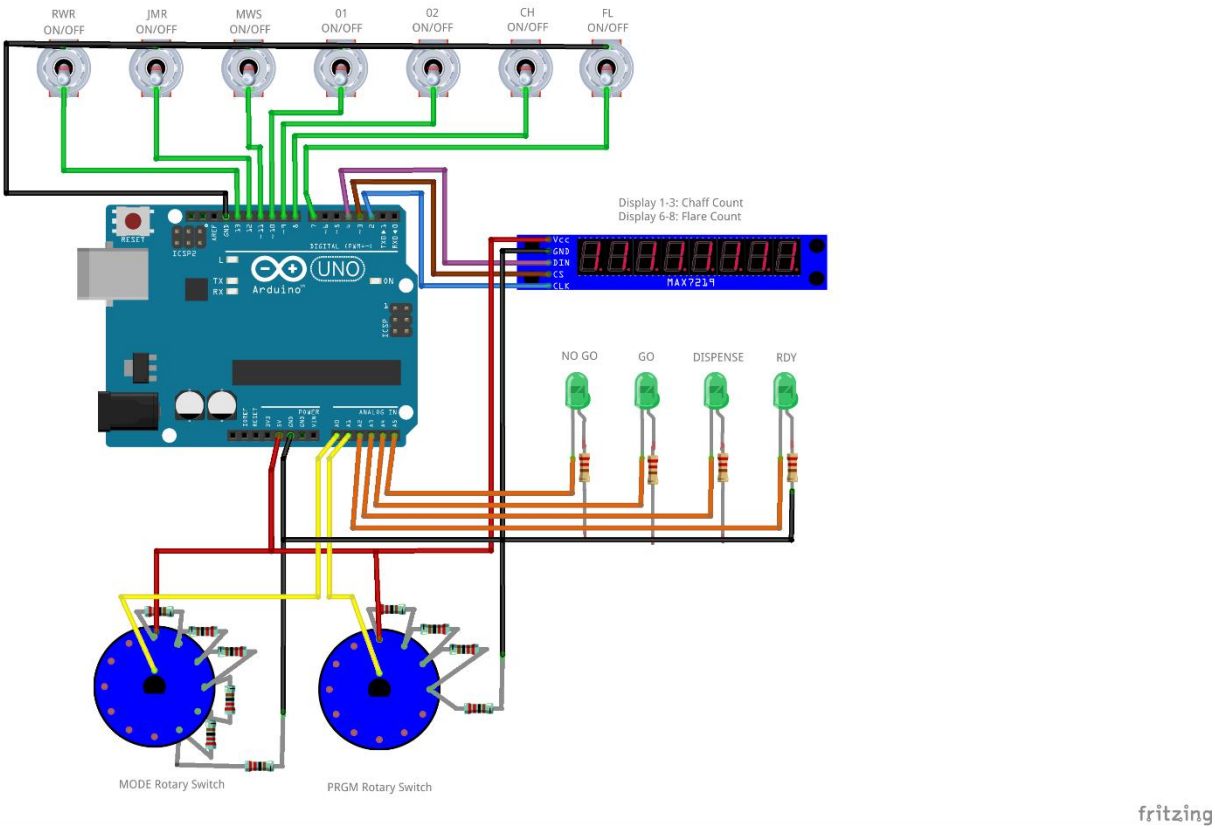
2.2. Variant 2



Arduino	LCD	LED	Opposition
A4(SDA)	SDA		
A5(SCL)	SCL		
10		GO	GND
11		NOGO	GND
12		DISP	GND
13		OK	GND

Switch	Arduino PIN	Opposition
MODE-OFF	2	GND
MODE-STBY	3	GND
JMODE-MAN	4	GND
FASHION SEMI	5	GND
FASHION CAR	6	GND
FASHION BYP	7	GND
CHAFF	8	GND
FLARE	9	GND

2.3. Variant 3



In the case of rotary switches, the positions are determined via the software. To do this, the detents of the rotary switch must be connected to resistors (see BMSAIT documentation in the explanation of the switch module). The sum of all resistors in a rotary switch should be about 10 kOhms (i.e. use 2.2 kOhm resistors in each case).

Arduino	MAX7219	LED	Opposition
2	CLK		
3	CS		
4	DIN		
A2		NOGO	GND
A3		GO	GND
A4		DISP	GND
A5		RDY	GND

Switch	Arduino PIN	Opposition
FASHION	A0	GND
PRGM	A1	GND
JET	6	GND
FL	7	GND
CH	8	GND
O2	9	GND
Ö1	10	GND
VAT	11	GND
JMR	12	GND
RWR	13	GND

In the case of rotary switches, the positions are determined via the software. To do this, the detents of the rotary switch must be connected to resistors (see BMSAIT documentation in the explanation of the switch module). The sum of all resistors in a rotary switch should be about 10 kOhms (i.e. use 2.2 kOhm resistors in each case).

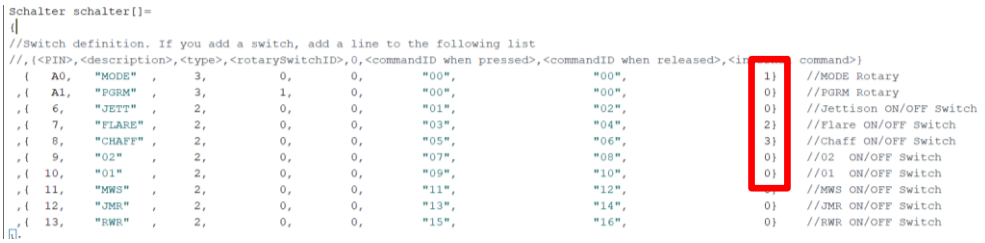
2.4. Internal command processing

In the CMDS module, it is necessary that the MODE, CHAFF and FLARE switches are connected to the Arduino on which this module is running. The reason for this is that the switch positions are used to influence functions of this module.

A/C PWR	CMDS mode	CHAFF	FLARE	CHAFF Ads	SHOW FLARE
OFF	Irrelevant	irrelevant	irrelevant	no display	no display
ON	OFF	Irrelevant	irrelevant	no display	no display
ON	Anything but OFF	ON	irrelevant	Display Number	
ON	Anything but OFF	OFF	Irrelevant	no display	
ON	Anything but OFF	Irrelevant	ON		Display Number
ON	Anything but OFF	Irrelevant	OFF		no display

The A/C power is read directly from the simulation by a BMSAIT variable. This switch therefore does not need to be connected to this Arduino. For the function of the module, however, the corresponding data variable must be included in the UserConfig area (ID 1260 "PWRST").

The way it works is that an internal code is added to the connected switches in the Switches module, which allows the CMDS module to detect the current switch position.



A coding system evaluates the positions and determines whether the displays are activated and which value should be displayed (function *checkPowerOn*).

3. Programming the Arduino

If the Arduino IDE is not yet installed, please refer to chapter 4.1.4 of the BMSAIT documentation.

Now call up the .ino in the desired variant from the \Arduino Sketch\BMSAIT_CMDS\ folder with a double-click. The sketch is loaded in the Arduino IDE. If you have done the wiring according to Chapter 2, no adjustments are required here.
Once you've chosen the right Arduino board, upload the sketch to the Arduino.

4. Setting up the Windows program

Install and start BMSAIT and make sure that the basic settings have been made correctly. It is particularly important that the reference to the variable definition (BMAIT-Variablen.csv) is made. Select the PUSH principle and turn off the autostart.

Then load the enclosed configuration (BMSAIT_demoCMDS.ini). BMSAIT should now display the loaded definition (one COM port and several variables).

Right-click on the COM port and edit it. Select the COM port to which your Arduino is connected. If you are not sure which COM port this is, then either select the SCAN function and see on which COM port the Arduino sends a response or you can look in the Windows Device Manager.

I recommend saving the changes now ("Save as" and selecting a new file).

Activate the test mode and start processing. If everything worked out, the 7-segment displays (variants 1 or 3) or the LCD (variant 2) should show a value.

5. Result

Exit the test mode and launche the BMSAIT processing. Start Falcon BMS and enter the 3D world. Now check the switch positions and the result of the display according to the table from chapter 2.2.

The check whether the power supply in the cockpit is also activated is only carried out in the 3D world. Outside the 3D world, only the position of the CMDS switches is checked to determine whether the indicators are activated or not.

Example of the implementation in the Homepit of Giovianni Pio Lombardi:

CMDS Homepit Lombardi.mp4