Falcon BMS to Arduino Interface Tool (BMSAIT)

Example program for the backup radio

Author	Robin „Hummer“ Bruns
Document version	1.3
Software version	1.0
BMS version	4.34u4
Date	16.12.2020

1. Overview

Other examples described isolated functions of the different BMSAIT modules. This document shows an advanced example and combines several modules to drive the backup UHF panel.

BMSAIT is a flexible tool that is designed to allow different approaches to use input and output devices for the use in Falcon BMS and DCS. Therefore, this document describes three variants with different hardware settings.

All variants have the common goal to provide the functions of a F-16 backup radio:

- Display of the backup UHF preset channel
- Display of the backup UHF manual frequency
- Enable the required buttons and switches of the radio panel for internal purposes and relay of signals to the simulation.

BMSAIT allows the implementation of some advanced features of the real backup radio that will exceed the simulated functions of the BMS backup radio.

- If in 3D and no a/c power is present, the displays will shut down
- If the radio is turned off, the displays will shut down
- If the COM1 volume is turned down, the displays will shut down
- If the radio is set to manual mode, the preset channel will not be shown
- If the test button is pressed, all displays will light up all segments
- If the status button is pressed, the manual frequency display will show the stored frequency of the selected channel.

To enable these functions, some modifications were implemented in the BMSAIT windows app. In order to use the status button to show a stored frequency, BMSAIT needs to access your Falcon BMS pilot.ini, as the sim will store all preset frequency in this file. The path to this file needs to be stored in the basic settings of the BMSAIT windows app.

Variant 1:

Variant 1 is a minimalistic approach to drive the backup radio. It only requires a single Max7219 8-digit 7-segment tube to drive both the channel and the manual frequency display (the tubes first 6 digits will show the frequency, the remaining two will show the channel).

This variant requires some soldering work to dismount the 7-segment digits from the Max7219 tube to rearrange them behind the radio panel. You can reduce your work by getting a Max7219 tube variant that has detachable 7-segment digits.

Required Hardware:

- one arduino-board (i.e. UNO or NANO (maybe with breakout board))
- one Max7219 8 digit 7-Segment-Tube (with detachable digits)
- two pushbuttons (6mm)

- one rotary switch (at least 3 detents)
- one switch (ON-OFF-ON)
- connector wires

Variant 2:

Variant 2 uses two different display controller boards (one Max7212 with 8 digits (or a TM1637 with 6 digits) and one TM1637 with 4 digits)

The additional display tube will allow to use one tube each for the preset channel and the manual frequency. Therefore, you will not need to unsolder the digits from the Max7219 tube to split them up between the two displays. On the other hand, you will have to find a solution to hide the additional, unneeded digits from both displays.

As a further addition to variant 1, this solution will incorporate the COM1 volume switch, as this will also influence the display behavior.

You need all hardware from variant 1 plus:

- One TM1637 7-segment-tube with 4 digits or one TM1637 7-segment-tube with 6 digits
- A potentiometer with build in ON/OFF detent

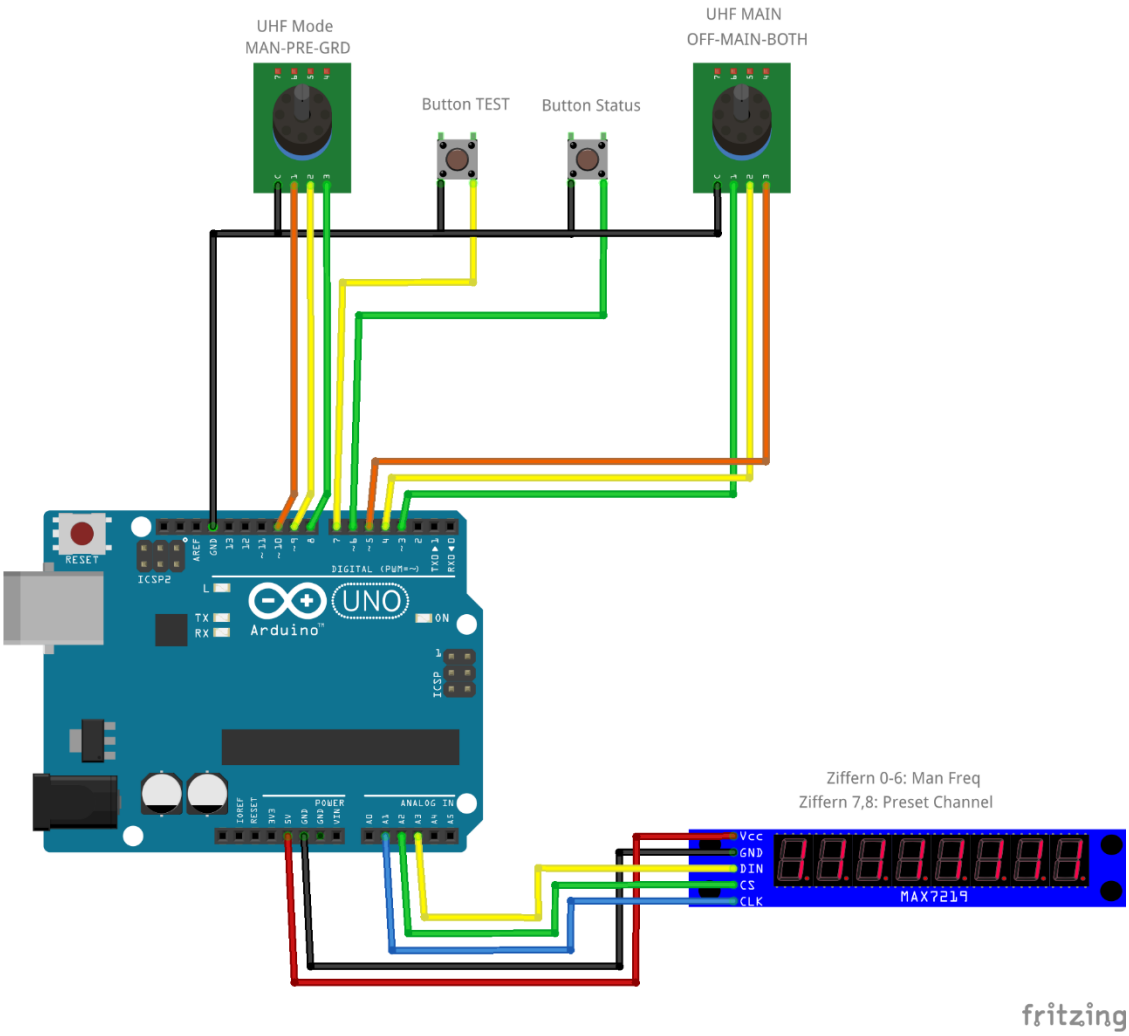
Variant 3:

Variant 3 is an upgrade of variant 1 and incorporates the full setup of all switches of the backup radio. The difference to variant 2 is the encoder for the backup channel control and the implementation of rotary switches to set the manual frequencies.

- One rotary encoder
- A potentiometer with build in ON/OFF detent
- one rotary switch (2 detents)
- one rotary switch (4 detents)
- one rotary switch (10 detents)
- 1 resistor 4.7kOhm
- 5 resistor 2.2kOhm
- 36 resistor 1kOhm

2. Setup

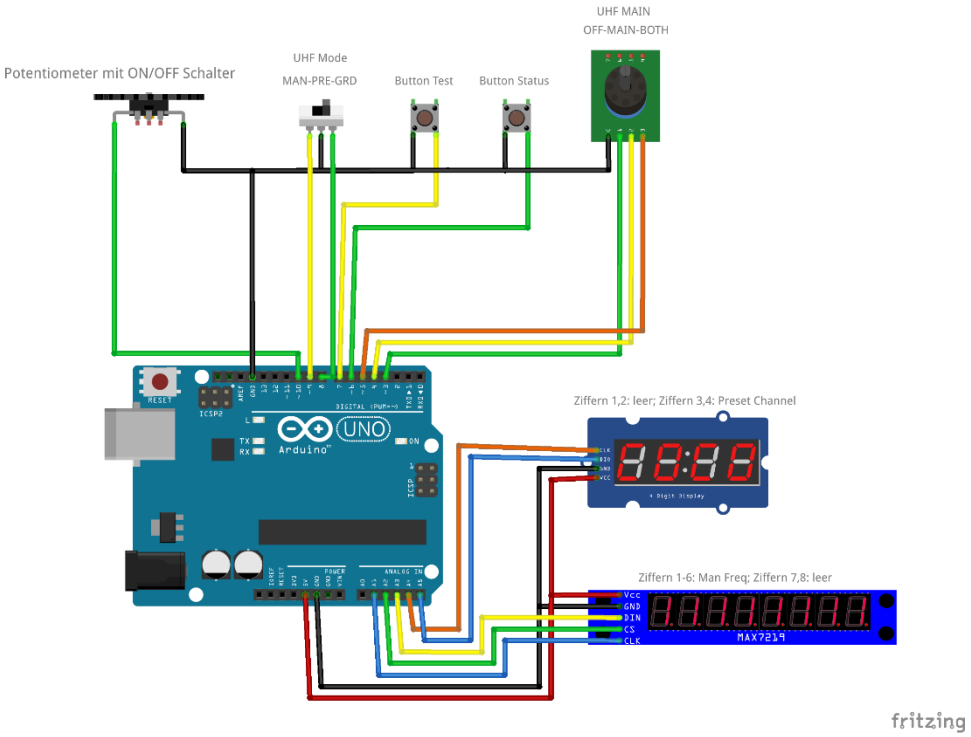
2.1. Variant 1



Arduino	MAX7219
A1	CLK
A2	CS
A3	DIN

Button/Switch	Arduino PIN	Connect to
Button Test	7	GND
Button Status	6	GND
UHF MAIN - OFF	3	GND
UHF MAIN - MAIN	4	GND
UHF MAIN - BOTH	5	GND
UHF MODE – MANUAL	8	GND
UHF MODE – PRE	9	GND
UHF MODE – GUARD	10	GND

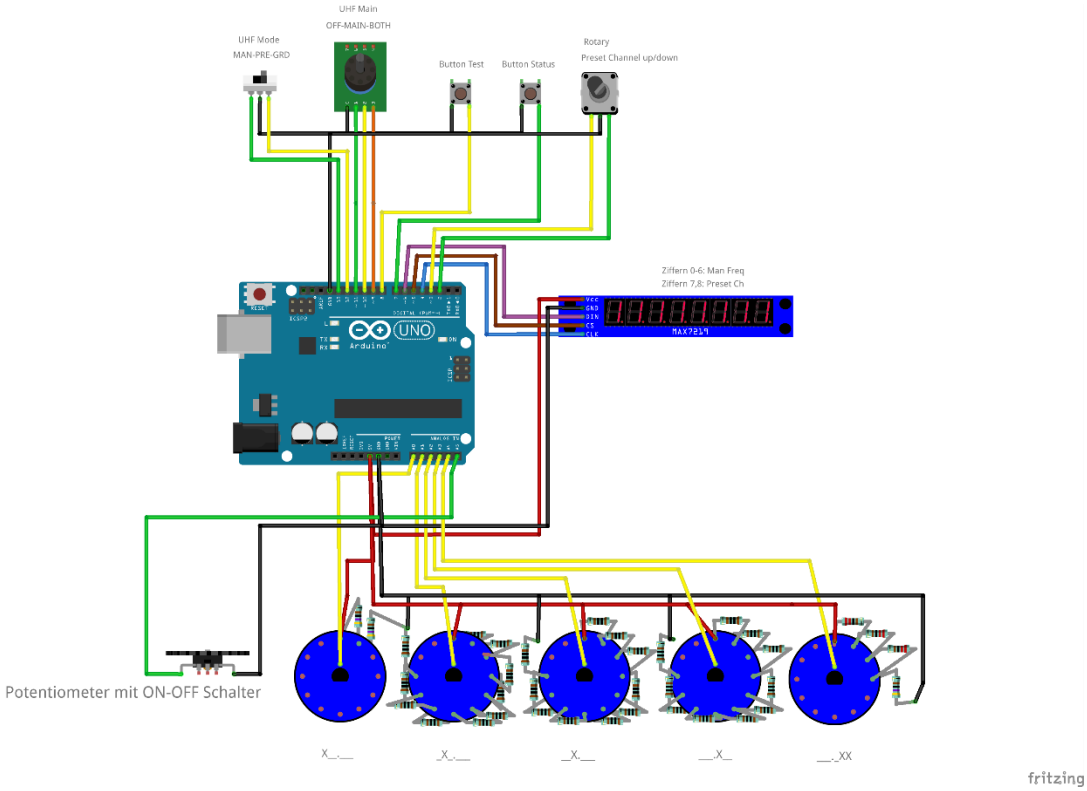
2.2. Variant 2



Arduino	MAX7219	TM1637
A1	CLK	
A2	CS	
A3	DIN	
A4		DIO
A5		CLK

Button / Switch	Arduino PIN	Connect to
Button Test	7	GND
Button Status	6	GND
UHF MAIN - OFF	3	GND
UHF MAIN - MAIN	4	GND
UHF MAIN - BOTH	5	GND
UHF MODE – MANUAL	8	GND
UHF MODE – GUARD	9	GND
UHF1 VOLUME ON/OFF	10	GND

2.3. Variant 3



Arduino	MAX7219
4	CLK
5	CS
6	DIN

Button/Switch	Typ	Arduino PIN	Connect to
PresetChannelKnob Links	RotaryEncoder1	2	GND
PresetChanellKnob Rechts	RotaryEncoder1	3	GND
Button Test	Button1	8	GND
Button Status	Button2	7	GND
UHF MAIN - OFF	Rotary Switch1	9	GND
UHF MAIN - MAIN	Rotary Switch 1	10	GND
UHF MAIN - BOTH	Rotary Switch 1	11	GND
UHF MODE – MANUAL	Switch1	12	GND
UHF MODE – GUARD	Switch1	13	GND
UHF1 Volume ON/OFF	Potentiometer with detent	A5	GND
MNL Freq X__.___	Rotary Switch2	A0	GND
MNL Freq _X_.___	Rotary Switch3	A1	GND
MNL Freq __X.___	Rotary Switch4	A2	GND
MNL Freq ___.X__	Rotary Switch5	A3	GND
MNL Freq ___._XX	Rotary Switch8	A4	GND

Please note that rotary switch 1 has a digital connection (every detent is read by individual data connection).

As the Arduino does not have enough data ports (Pins) to connect every switch position, this sketch will use an analog connection of the rotary switch. The detents of the rotary switches will be connected to each other with a resistor. If the rotary switch is moved to a different detent, the

Arduino will encounter a different number of resistors and therefore will be able to determine the current switch position. The resistors that are used on one rotary switch should sum up to about 10k Ohms.

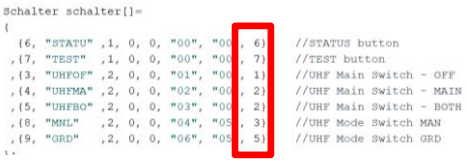
2.4. Internal commands

In order to get the backup radio to work, it is mandatory to incorporate the buttons/switches UHF MAIN, UHF MODE, STATUS und TEST. This is required as the switch positions will affect the display behavior.

BATT PWR	COM1 VOLUME	UHF MAIN	UHF Mode	Test	Status	Preset Anzeige	Manual Anzeige
OFF	xxx	xxx	xxx	xxx	Xxx	No display	No display
xxx	OFF	xxx	xxx	xxx	xxx	No display	No display
xxx	xxx	OFF	xxx	xxx	xxx	No display	No display
ON	ON	MAIN/BOTH	MNL	OFF	OFF	No display	Display MNL freq
ON	ON	MAIN/BOTH	PRESET	OFF	OFF	Display Channel	Display MNL freq
ON	ON	MAIN/BOTH	GUARD	OFF	OFF	No display	Display 243.000
ON	ON	MAIN/BOTH	xxx	OFF	ON	Display Channel	Display frequency of preset channel
ON	ON	MAIN/BOTH	xxx	ON	xxx	Display 88	Display 888888

The power status of the a/c in 3D mode of BMS will be read, transmitted and saved on the Arduino. It is not required to connect the battery switch of your cockpit to the Arduino. But it is mandatory to incorporate the data variable of the power status in the Arduino programming sketch (ID 1242 „Emergency Power Bus“).

The switches will be assigned to an internal code that will allow to trigger actions within the Arduino.



The special BUPRadio code will read these internal codes (functions *BUPRadio_Update* and *CheckPowerOn*).

3. Program the Arduino

If you haven't installed the Arduino IDE yet, please refer to chapter 4.1.4 of the main BMSAIT documentation.

Load the BUPRadio Arduino sketch \Arduino Sketch\BMSAIT_BUPRadio\. Into the Arduino IDE. After selecting the correct Arduino board and COM port from the Arduino IDE menu, you can upload the Arduino code to the Arduino.

4. Settings in BMSAIT windows app

Download and install the BMSAIT windows app. Make sure that the correct basic settings are set. For this example, it is imperative that the path to the variable setting and the pilot.ini are set. Choose PUSH-profile and disable auto-launch.

In BMSAIT, use the load button to load the setting file from this example (BMSAIT_demoBUPRADIO.ini). BMSAIT should show some information (one COM-port, multiple data variables and commands).

Right click the COM-Port and select „change“. Make sure that the correct COM-port of your Arduino is set.

I recommend to save both the basic settings and the user configuration.

Now activate the „testmode“ radio button and hit „run“. If the setup was successful, the displays should briefly show some figures.

5. Result

Stop the connection and deselect testmode. Now run BMSAIT again and run Falcon BMS. Once in 3D mode, check the effect of the connected switches and the displays (the table from chapter 2.4 shows the desired behavior).

Outside of the 3D sim enviroment, only the switch positions from UHF MAIN und UHF Mode switches will be checked to determine the display status. BMSAIT will check the power status upon entering 3D.

The “status” feature of the frequency display will only work when in 3D of BMS and will only show the underlying preset-frequency if the pilot.ini was referenced in the basic settings.

Thanks to my buddy Bumerang who offered to make a video of his successful setup:

BUPRadio F-16 Home Cockpit