FUEL FLOW INDICATOR for the F-16 SIMULATOR

Version: V1.0 November 2017

History

10-2017 : Draft version, based on the Altimeter circuit board V3.0.

Initial work for the Fuel Flow Indicator (FFI). The hardware is based on the Altimeter circuit board, Version 3.0. Preliminary tests using the altimeter firmware are very promising. The firmware only needs modifications specifically to the functionality offered/needed to control the FFI.

11-2016 : Version 1.0, describing the FFI firmware V1.0.

Some commands implemented for the altimeter were not applicable or needed for the FFI, so they have been removed. As a result, command numbers have changed. Further, a higher PPH setpoint value resulted in a lower shown PPH on the indicator. Finally, the full range of the FFI (0 – 80000 PPH) was obtained with only half of the possible setpoint values from the Altimeter circuit board, thus losing resolution. The stator sine wave table now has 12 bit accuracy which results in a full range of the FFI with almost 12 bits, thus the resolution is 80000 / 4096 which is ~ 19.5 PPH. Whether the hardware and FFI can live up to this resolution remains to be seen …

Contents

1. Introduction

If you have an original Fuel Flow Indicator (FFI) in the cockpit and want to bring it to "life", you have two options. Either you open the FFI, remove some parts of the mechanism, and mount a stepper motor, or you leave the FFI intact, unopened, and use an electronic solution to drive the FFI. This manual is all about the second approach.

The FFI is a rather simple instrument as far as controlling it is concerned. It only needs 115V AC 400 Hz power supply voltage and the signals to drive one synchro. Note that 115V AC 400 Hz is ***not*** 115V 50/60 Hz from your wall socket!

The synchro in the FFI gets its rotor drive voltage from the 115V AC 400 Hz power supply. The FFI interface (based on the Altimeter circuit board V3.0 with minor modifications) only needs to generate the three stator voltages in phase with the rotor voltage.

The following parts are needed to control the FFI.

- The Altimeter circuit board V3.0
  The Altimeter circuit board requires only one power supply voltage, +24V DC. On-board is a DC/DC converter that generates the required power supply voltages for the control hardware and synchro stator signals. However, if you have the required voltages already available in your pit, you can omit the DC/DC converter (that beast is expensive). On the Altimeter circuit board is a programmed PIC processor that generates the synchro stator signals based on the received commands. Communication uses either USB or PHCC DOA.
- 115V AC 400 Hz power supply
  The 115V AC 400 Hz is connected to the FFI and the Altimeter circuit board.
- 24V DC power supply
  The +24V DC power supply is needed for the Altimeter circuit board.
- 3 amplifier stages and symmetrical DC power supply for the amplifiers
  The Altimeter circuit board generates the 3 stator signals for the synchro, but these signals are not capable of driving the coils of the FFI synchro directly. Three external (audio) amplifiers are needed to drive the stator coils of the synchro.
  Further, a symmetrical +/– 18 .. 24V DC dual (positive and negative) power supply is needed to power the amplifiers.

The Altimeter V3.0 circuit board has the following features.

- Small PCB (3.1" x 3.9"), easy to build, no SMD components
- Generation of 3 precision synchro stator signals
- Spare outputs to (optionally) control external devices (ON/OFF)
- Amplitude adjustment of the input reference signal
- Amplitude adjustment of the 3 output stator signals
- On-board +5V and –5V power supplies (optional), in that case only +24VDC needed
- All required connections at one side of the circuit board.

I spent quite some time developing and testing the hardware. Therefore, I put some “copyright” on my work. You are allowed to build and use the Altimeter circuit board design for your own purpose, but you are not allowed to make commercial profit building and/or selling the work. This manual gives detailed descriptions, because I feel that this information is useful when another device is used. However, it is not the intention to enable easy copying of my work and try to make money. The Altimeter circuit board is developed for hobbyists who want to ‘bring to life’ a real instrument without modifications to the delicate mechanisms inside the instrument.

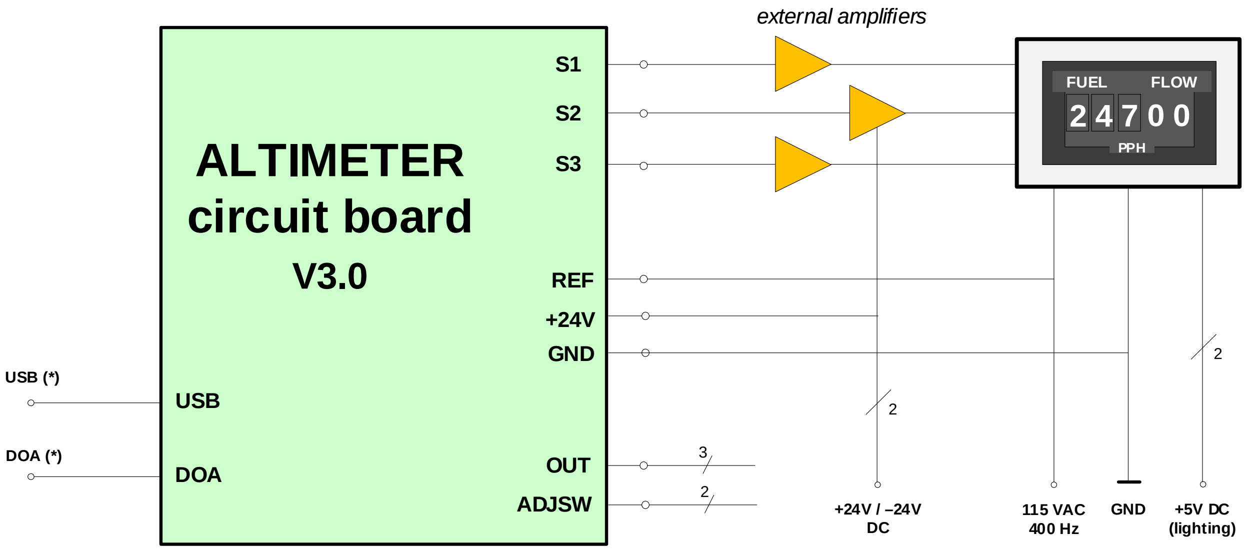
Disclaimer

All uses of the Altimeter circuit board and all mentioned hardware are solely *your* responsibility. Any damage to the Altimeter circuit board, other hardware or connected instrument(s) is ***not*** my responsibility. I have tried to remove any typo or error, but you cannot hold me responsible for errors if any error causes a defect.

2. Overview of the Fuel Flow Indicator solution

2.1 Block diagram

The block diagram shows the complete F-16 Fuel Flow Indicator solution.



The 3 external amplifiers are needed to boost the stator signals from the Altimeter circuit board to drive the stator coils of the FFI synchro. The amplifiers can be based on the LM1875 (for example). An important aspect of the external amplifiers is that they should be powered by a symmetrical supply voltage, for example +24V DC and –24V DC. If you use external amplifiers operating on a single supply voltage you must check that the output of the amplifiers is 0V with respect to GND when no input signal is applied. Note that with a single supply voltage the output amplitude might be insufficient. The amplifiers can be simple DIY kits found on eBay.

OUT connects 3 digital outputs that can freely be used for any purpose. ADJSW were inputs in the altimeter solution, but are not needed in the FFI solution. Therefore, the FFI firmware programs these two inputs as outputs, and these two outputs can also be used freely for any purpose. Via the communication interface each output can be individually switched ON/OFF. The 5 outputs are directly connected to the PIC. These outputs can source/sink at maximum 20 mA.

2.2 Power supplies

The Altimeter circuit board requires +5V, and –5V DC for proper operation. If you have clean +5V and –5V DC available in your pit, you do not need the on-board DC/DC converter. The DC/DC converter generates the +5V and –5V required for the PIC processor and the analog synchro signal generation. With the DC/DC converter installed, you only need +24V DC.

2.3 Interface connections

The Altimeter circuit board has two interface connections, mutual exclusive, either DOA (PHCC) or USB. An on-board jumper defines which communication connection is selected.

3. Connections & jumpers on the Altimeter circuit board

The Altimeter circuit board has 7 connectors and 3 jumpers. Pin #1 is indicated on the PCB.

3.1 24V.POWER connector

The 24V.POWER connector connects a +24 Volt DC power supply to the Altimeter circuit board. Make sure that you connect the power supply correctly.

→ *The Altimeter circuit board has no provisions against wrong power supply connections !*

pin number	Signal	Usage
1	GND	Ground (0V)
2	+24V	+24V DC power supply
3	+24V	+24V DC power supply
4	GND	Ground (0V)

3.2 ALTIMETER connector

The ALTIMETER connector connects the synchro stator inputs and reference output.

pin number	Signal	Usage
1	GND	Ground (0V)
2	RF	Synchro reference signal (input)
3	TP	Test point reference input voltage
4	SY	Test point signal symmetry
5	S3	Synchro stator S3 (output)
6	S2	Synchro stator S2 (output)
7	S1	Synchro stator S1 (output)
8	SR	Synchro rotor (output, not used for FFI)

3.3 OUT connector

The OUT connector has 3 freely usable digital outputs.

pin number	Signal	Usage
1	Output #1	Freely usable
2	Output #2	Freely usable
3	Output #3	Freely usable

3.4 ADJSW connector

The ADJSW connector has 2 freely usable digital outputs.

pin number	Signal	Usage
1	Output #4	Freely usable
2	Output #5	Freely usable
3	GND	Ground (0V) – common connection

3.5 DC/DC connector

The DC/DC connector pins are power supply **input** pins to connect external power supplies.
Not to be used to power external electronics if the on-board DC/DC converter is installed!

pin number	Signal	Usage
1	+5V	+5V input *(only if TMR3-2421 is **not** installed)*
2	GND	Ground (0V)
3	–5V	–5V input *(only if TMR3-2421 is **not** installed)*

3.6 DOA connector

The DOA connector is the standard PHCC DOA 10-pin connector. See also “3.9 RST jumper”.

3.7 USB connector

The USB connector is a standard USB Type B connector.

3.8 USB/DOA jumper

The USB/DOA jumper selects which communication interface is used.

pin number	Signal	Usage
1 – 2	USE_USB	The USB connector is used for communication
no jumper	USE_DOA	The DOA connector is used for communication

3.9 RST jumper

The RST jumper selects what the function is of pin #1 of the DOA connector.

pin number	Signal	Usage
1 – 2	DOA_GND	Pin #1 of DOA connector tied to GND (original DOA specification)
2 – 3	DOA_RST	Pin #1 of DOA connector tied to RST* of PIC Pin 1 of the DOA connector can be used as external reset signal for the PIC. Note that the original ribbon cable from the PHCC Motherboard for DOA **cannot** be used, because it will keep the PIC in reset.

3.10 SRC jumper

The SRC jumper selects the synchro reference signal source. For the FFI usage, the jumper must be installed on pins #1 and #2.

pin number	Signal	Usage
1 – 2	EXT_REF	Reference signal from external source (FFI in this case)
2 – 3	LOC_REF	Reference signal from local 400 Hz sine wave generator (not used)

4. Connection wiring

4.1 Introduction

This chapter describes the interconnections per identified functionality.
The required connections are the following.

- power supplies (115V AC 400 Hz and 24V *or* +/–5V) → Altimeter circuit board
- power supplies (115V AC 400 Hz) → Fuel Flow Indicator
- power supplies (+/–18..24V) → External power amplifiers
- Altimeter circuit board → External power amplifiers inputs
- External power amplifiers outputs → Fuel Flow Indicator
- Altimeter circuit board → PC (USB) or PHCC Motherboard (DOA).

The optional connections are the following.

- digital outputs of the Altimeter circuit board → user purpose ON/OFF control

4.2 Power supply wiring

The power supply wiring for the "FFI solution" is simple, just +24VDC to the Altimeter circuit board and 115VAC 400 Hz to the Altimeter circuit board and the FFI. On the Altimeter circuit board is a DC/DC converter that generates the required voltages (+5V and –5V).

If you have clean +5V / –5V power supply voltages available in the pit, you can omit the DC/DC converter. If you connect a power supply to the DC/DC header, the DC/DC converter must ***not*** *be installed!*

➔ **Always consider safety as the most important aspect with everything you do.**

The table shows the power supply connections that must be made.

Connection	from	to
115V AC [400 Hz] HOT	115V AC power supply	to Fuel Flow Indicator to "RF" pin on Altimeter circuit board
115V AC [400 Hz] GND	115V AC power supply	to Fuel Flow Indicator
		to common GND connection
+24V DC	24V DC power supply	24V.POWER pin #2 or #3
+24 GND	24V DC power supply common GND connection	24V.POWER pin #1 or #4

NOTE: the 115VAC is not the 50/60Hz from your household mains power! The 115VAC must be 400 Hz.

⇨ *Always double-check the connections before you switch on the 24V power supply! If you know of yourself that you might inadvertently get the polarity wrong, add a diode in series with the +24V power supply connection and the (optional) fuse to protect the hardware against wrong polarity. This wrong-polarity protection will cause a small voltage drop. That drop is not a problem for the electronics.*

4.3 Fuel Flow Indicator wiring

The FFI is connected to the Altimeter circuit board via the ALTIMETER connector. This connector has a GND connection, the RF reference signal input and the 3 stator signal outputs.

Connection	from	to
Synchro S1	ALTIMETER #7 (S1)	Fuel Flow Indicator (*)
Synchro S2	ALTIMETER #6 (S2)	Fuel Flow Indicator (*)
Synchro S3	ALTIMETER #5 (S3)	Fuel Flow Indicator (*)
Reference	ALTIMETER #2 (RF)	115V AC [400 Hz] HOT
GND	ALTIMETER #1 (GND)	System GND

Pin #3 (TP) and pin #4 (SY) of the ALTIMETER connector are test points. If the 115V AC 400 Hz is connected and powered you can measure the reference input voltage to the Altimeter circuit board on TP, and adjust the amplitude for symmetry measuring SY. See chapter 5 for details.

(*) The stator outputs of the Altimeter circuit board cannot drive the synchro inputs of the Fuel Flow Indicator directly. External amplifiers are required.

4.4 Interface wiring

The Altimeter circuit board must be connected to the PC using the USB connection, or must be connected to the PHCC Motherboard using the DOA connection. Start a communications program on the PC if you use USB (I use the *"Fuel Flow Indicator"* test application), or start *lighting*'s PHCC TestTool program if you use DOA. In the PHCC TestTool, select the DOA tab and enter the Fuel Flow Indicator device code (default 0x46).

If you use the PHCC connection you must connect the DOA clock, DOA data, GND, and optionally the RESET pin (not standard PHCC, see chapter 3.9 for details) from the Altimeter circuit board to the PHCC Motherboard. The Altimeter circuit board does *not* use the +5V from the DOA bus.

If you use the USB connection, you need a USB Type B to USB Type A cable to connect the PC. The Altimeter circuit board does *not* use the +5V from the USB connection.

Note that you can use only one connection, either DOA or USB.
Set the DOA/USB jumper on the Altimeter circuit board accordingly!

Connection	From	To
DOA	Altimeter circuit board DOA	PHCC Motherboard board DOA
USB	Altimeter circuit board USB	PC

5. First time use

The Altimeter circuit board has 6 trim potentiometers. The SENS trim potentiometer is used to adjust the amplitude of the 115V AC 400 Hz reference signal. The amplitude is way too high and cannot be handled by the electronics on the Altimeter circuit board. Therefore, the amplitude of the reference signal must be adjusted.

To prevent damage to the electronics, do the following steps to adjust the amplitude safely.

1. Make sure all power supplies are switched off.
2. The Altimeter circuit board, FFI and power supplies are of course correctly wired.
    - The FFI must be connected to 115VAC 400 Hz (pin 7 [HOT], pin 6 [GND]).
    - 115VAC 400 Hz [HOT] must be connected to the ALTIMETER header pin #2, RF.
    - The stator signal inputs (pin 11, 4, 12) of the FFI are **not** connected.
3. ***Remove*** IC3 (TL082) from its socket.
   Remove the jumper from the SRC header.
4. Connect a multimeter.
    - – lead connected to the ALTIMETER header pin #1, GND.
    - + lead connected to the SRC header pin #1.
5. Set the multimeter to measuring range AC Volts.
6. Switch on the 115VAC 400 Hz power supply (the +24VDC is not needed).
7. Adjust the trim potentiometer “SENS” so that the multimeter reads 1.0 V.
8. Switch off the 115VAC 400 Hz power supply.
9. Disconnect the multimeter.
10. Put IC3 (TL082) in its socket. Note correct orientation!
    Put the jumper on the SRC header pins #1 – #2.
11. Connect the multimeter.
    - – lead connected to the ALTIMETER header pin #1, GND.
    - + lead connected to the ALTIMETER header pin #3, TP.
12. Set the multimeter to measuring range AC Volts.
13. Switch on the 115VAC 400 Hz power supply and the +24VDC power supply.
14. Measure accurately the voltage on test pin TP (approx. 2.2 Volts).
15. Connect the + lead of the multimeter to the ALTIMETER header pin #4, SY.
16. Adjust the trim potentiometer “SYMM” so that the multimeter reads exactly the same voltage that was measured on test pin TP.
17. Switch off all power supplies.
18. Disconnect the multimeter.

After these steps the “input side” is adjusted.
The following procedure is for the “output side”, the connection to the stator coils of the FFI. You need control software to set the output amplitudes of the stator signals.

1. The power supply wiring must be complete, that is, 115VAC 400 Hz is connected to the FFI, and +24V DC is connected to the Altimeter circuit board (or +5V DC and –5V DC if you do not use the DC/DC converter).

2. The 115VAC 400 Hz is connected to the pin #2, “RF” on the Altimeter circuit board.
3. The inputs of the external amplifiers are connected to pins S1, S2, and S3 of the Altimeter circuit board. Note that the outputs of the amplifiers are **not** connected to the FFI.
4. Connect +24V DC and –24V DC to power the external amplifiers.
5. Install the Fuel Flow Indicator test application on your PC, see Appendix E.
6. If you use USB, connect the USB connection of the Altimeter circuit board to the PC. If you use PHCC, connect your PHCC Motherboard to the PC (USB or serial COM port), and connect the Altimeter circuit board DOA header to the PHCC Motherboard.
7. Switch all power supplies on.
8. Wait until the PC has detected the COM port (if USB is used).
9. Start the Fuel Flow Indicator test application.
10. Select the USB or PHCC radio button, and select the COM port in the application.
11. Checkmark the “Advanced” checkbox.
12. Send the following commands.
    a. Enter sub-address 16, data 0, click “SEND”.
    b. Enter sub-address 17, data 0, click “SEND”.
    c. Enter sub-address 18, data 0, click “SEND”.
    d. Enter sub-address 19, data 0, click “SEND”.
13. Set the multimeter to measuring range AC Volts.
14. Connect the multimeter.
    - – lead connected to the ALTIMETER header pin #1, GND.
    - \+ lead connected to the “S1” external amplifier output.
15. Adjust the trim potentiometer S1 until the multimeter reads 9.000V.
16. Connect the + lead connected to the “S2” external amplifier output.
17. Adjust the trim potentiometer S2 until the multimeter reads 9.000V.
18. Connect the + lead connected to the “S3” external amplifier output.
19. Adjust the trim potentiometer S3 until the multimeter reads 9.000V.
20. Terminate the application.
21. Switch off all power supplies.
22. Connect ALTIMETER header pin #7, #6, #5 (S1, S2, S3) to the FFI.
23. Switch on all power supplies.
24. Start the Fuel Flow Indicator test application.
25. Play with it ☺

6. Fuel Flow Indicator commands

If the Altimeter circuit board is used as a PHCC slave (DOA communication) you must send the DOA address byte, the sub-address byte and a data byte. The DOA address byte identifies that the data packet sent is for the FFI firmware. The value of the DOA address is hard-coded in the PIC firmware, and can be any value as long as the value is unique on the entire DOA bus. The sub-address and data byte define for which functionality the data is intended.

If you use the USB connection it is clear what the destination is, and the address byte is not needed. The USB data packet thus consists of the sub-address and the data byte (in that order).

This chapter describes the implemented sub-addresses with their possible data byte values. Remember that the Altimeter circuit board is a “listen-only” device, thus if you want to know what data was sent, the sending program must keep a local copy.

FUEL FLOW INDICATOR CONTROL

sub-address	data byte	function / description
0	0x00 … 0xFF	SFFI_0 – set indicator in “range” 0000 – 0255
1	0x00 … 0xFF	SFFI_1 – set indicator in “range” 0256 – 0511
2	0x00 … 0xFF	SFFI_2 – set indicator in “range” 0512 – 0767
3	0x00 … 0xFF	SFFI_3 – set indicator in “range” 0768 – 1023
4	0x00 … 0xFF	SFFI_4 – set indicator in “range” 1024 – 1279
5	0x00 … 0xFF	SFFI_5 – set indicator in “range” 1280 – 1535
6	0x00 … 0xFF	SFFI_6 – set indicator in “range” 1536 – 1791
7	0x00 … 0xFF	SFFI_7 – set indicator in “range” 1792 – 2047
8	0x00 … 0xFF	SFFI_8 – set indicator in “range” 2048 – 2302
9	0x00 … 0xFF	SFFI_9 – set indicator in “range” 2304 – 2559
10	0x00 … 0xFF	SFFI_A – set indicator in “range” 2560 – 2815
11	0x00 … 0xFF	SFFI_B – set indicator in “range” 2816 – 3071
12	0x00 … 0xFF	SFFI_C – set indicator in “range” 3072 – 3327
13	0x00 … 0xFF	SFFI_D – set indicator in “range” 3328 – 3583
14	0x00 … 0xFF	SFFI_E – set indicator in “range” 3584 – 3839
15	0x00 … 0xFF	SFFI_F – set indicator in “range” 3840 – 4095
	Commands 16 ~ 18 are “deferred”. They prepare a new setting for direct control of all 3 stators, and are loaded synchronously with command 19. *Make sure all 3 amplitudes are set* ***and then*** *set the 3 polarity values.*	
16	0x00 … 0xFF	S1AMPL – set amplitude (coarse) of stator signal S1 deferred
17	0x00 … 0xFF	S2AMPL – set amplitude (coarse) of stator signal S2 deferred
18	0x00 … 0xFF	S3AMPL – set amplitude (coarse) of stator signal S3 deferred
19	0x00 … 0x07	SxPOL – set polarity of stator signals & load S[1,2,3]AMPL

WATCHDOG functionality

sub-address	data byte	function / description
20	(*)	Disable watchdog functionality **--** See text for description.
21	0x00 … 0xFF	**Watchdog control --** See text for description.

USER-DEFINED DIGITAL OUTPUTS

sub-address	data byte	function / description
22	0 or 1	Standard digital output #1. 0 :: “OFF” , 1 :: “ON”.
23	0 or 1	Standard digital output #2. 0 :: “OFF” , 1 :: “ON”.
24	0 or 1	Standard digital output #3. 0 :: “OFF” , 1 :: “ON”.
25	0 or 1	Standard digital output #4. 0 :: “OFF” , 1 :: “ON”.
26	0 or 1	Standard digital output #5. 0 :: “OFF” , 1 :: “ON”.

MISCELLEANOUS

sub-address	data byte	function / description
27	0, 1, 2, 3	**DIAG** LED operation mode 0 – LED always OFF 1 – LED always ON 2 – LED flashes at heart beat rate 3 – LED toggles ON/OFF state per accepted command
28	0x00 … 0xFF	Stator S1 base angle value LSB *(must be set before MSB)*
29	0x00 … 0x1F	Stator S1 base angle value MSB *(must be set after LSB)*
30	0x00 … 0xFF	Stator S2 base angle value LSB *(must be set before MSB)*
31	0x00 … 0x1F	Stator S2 base angle value MSB *(must be set after LSB)*
32	0x00 … 0xFF	Stator S3 base angle value LSB *(must be set before MSB)*
33	0x00 … 0x1F	Stator S3 base angle value MSB *(must be set after LSB)*
34	0x00 … 0xFF	Stator S1 high-accuracy angle value LSB *(must be set before MSB)*
35	0x00 … 0x1F	Stator S1 high-accuracy angle value MSB *(must be set after LSB)*
36	0x00 … 0xFF	Stator S2 high-accuracy angle value LSB *(must be set before MSB)*
37	0x00 … 0x1F	Stator S2 high-accuracy angle value MSB *(must be set after LSB)*
38	0x00 … 0xFF	Stator S3 high-accuracy angle value LSB *(must be set before MSB)*
39	0x00 … 0x1F	Stator S3 high-accuracy angle value MSB *(must be set after LSB)*
40	0x00 … 0x07	Set polarity high-accuracy angle values AND load
41	don’t care	**IDENTIFY** USB only: send identification “FFI vA.B $xy”
42	‘N’ or ‘Y’	**USB** debug command (USB only ☺)

6.1 How to control the Fuel Flow Indicator

The three stator signals must have a correct voltage amplitude to set the mechanical position of a synchro axis. The correct (coarse) voltage amplitude for each stator coil can be set *immediately* with the commands S1AMPL, S2AMPL, S3AMPL, and SxPOL. S#AMPL sets the amplitude of the stator signal. SxPOL sets the polarity of the three stator signal, which can be positive or negative, *and* loads the stator amplitudes synchronously. Thus, the three stator amplitudes commands must be issued *before* the polarity command is sent. The algorithm is simple. For a setpoint angle α of the synchro axis, the algorithm (on the PC) calculates the sine values of α, (α+120°), and (α+240°). These values are made absolute and then scaled (value between 0 and 255). The absolute value is sent with the SxAMPL commands, the sign of the values is sent with the SxPOL command. As the indication range of the synchro is covered by 256 "positions", and the full range represents 80000 PPH on the FFI, one coarse step is ~312 PPH !

Note that the update of all three stator amplitudes is executed synchronously when the command SxPOL is received. The last received S#AMPL amplitude values are used, if several values are sent to a specific stator. If no value is received, the last sent values are reloaded.

The SFFI_x commands (where x can be 0 – 9, A – F) move the indicator to the "position" set by the 8-bit data byte. Value 0 represents a position at the start of the range, value 255 represents a position at the end of the range. An increasing value increases the indicated PPH value.
Using the single byte data support by the DOA protocol would limit the resolution of a synchro to 8 bits. Full range using 8 bits would result in a rather coarse resolution of ~312 PPH. The solution is *16* commands where each command specifies with 8 bits a position in a *range*. The range is defined by the command address. As there are 16 commands, 4 bits "identify" the range. Effectively, the full range definition is now 12 bits: 4 bits range plus 8 bits data. This approach makes the PPH resolution ~19.5 PPH, which is adequate. Higher resolution is probably not achievable, due to limitations in the hardware.

The math is simple to calculate the sub-address and data value from a given PPH value setpoint.

$$\texttt{indication} = \texttt{int}\left(\frac{\texttt{fuelflow}[\texttt{PPH}] * 4096}{80000}\right)$$

$$\texttt{subAddress} = \texttt{int}\left(\frac{\texttt{indication}}{256}\right)$$

$$\texttt{dataValue} = \texttt{indication} - \left(\texttt{subAddress} * 256\right)$$

Example 1. Value is 2400 PPH => subAddress = 0 // dataValue = 221.
Example 2. Value is 21300 PPH => subAddress = 3 // dataValue = 33.

However, it turns out that the Altimeter (or FFI) behavior is not linear ☹ The simple formulas can be used to create an initial lookup table in the control software (which obtains the value from BMS and sends the setpoint commands to the FFI. You can get a fairly accurate value indication by making a lookup table for every 500 PPH increment. Then send a “500 PPH incremented” position to the FFI and read what the actual indication is. Make a correction in the table to obtain an accurate indication. This is a “trial & correction” process, but it only needs to be done once. Using 500 PPH steps will result in a 160 entries table, and then you can use linear interpolation to get fairly accurate value indications. The accuracy depends mostly on how accurate you determine the table entries.

6.2 IDENTIFY command

The IDENTIFY command is only available when the USB communication connection is used. When the IDENTIFY command and an “any value” data byte is sent, the Altimeter circuit board returns a message in the form of “`FFI vA.B $xx`” followed by a CR (0x13) and LF (0x0A). The “`A`” and “`B`” represent the major and minor version number of the firmware. “`$xx`” is the hexadecimal device code of the FFI firmware, and used for recognition of DOA PHCC commands. Although the DOA device code is irrelevant for the USB communication, it can still be useful. Suppose you have more than one USB-connected PHCC device (for example the SDI boards of the ADI). Using the IDENTIFY command you can find out which specific instrument is connected to a virtual COM port, because the device code (should be) unique. Note that because of this, the firmware for each PHCC device is unique, but that is always the case with PHCC DOA.

6.3 USB debug command

The USB debug command enables or disables debug output sent to the USB connection. Obviously, this command only operates when you use a USB connection. Default USB debug is disabled. To enable USB debugging output you must send the command with the data byte ASCII character ‘Y’. Likewise, to disable the USB debugging output send the command with the data byte ASCII ‘N’. When USB debug is enabled a specific character is returned every time a data byte is received via the USB connection, followed by the data byte itself (in hexadecimal format). Depending on the interpretation (by the firmware) of the data byte one of the following specific characters is returned leading the echoed (hexadecimal) data byte.

returned character	SDI interpretation
–	following 2 characters are hexadecimal command (first byte)
=	following 2 characters are hexadecimal data (second byte)
x	"disable watchdog" command received
#	invalid sub-address in command
!	USB receive data state machine in illegal state (firmware error ☹)

6.4 Watchdog functionality

The watchdog functionality makes sure that the data communication stays "synchronized". Commands sent via the PHCC DOA channel are 3 "bytes" (device address, sub-address and data byte, where the sub-address is 6 bits!). Commands sent via the USB channel are 2 bytes. The firmware uses a state machine to keep track of the received data bits and bytes. If, due to some external disturbance, one byte is not correctly received the receive routine may treat the next byte as the first, because the state machine is in the wrong state. If the firmware gets in this state, all subsequent commands received will be wrong. This condition is not recognized by the sender (PHCC Motherboard or the PC), but you will notice that all functionality of the Altimeter circuit board no longer seems to work (because the "commands" are wrong). You could say that the firmware has become "deaf".

To solve this problem a so-called watchdog timer is implemented in the firmware. In a normal condition the bytes sent via USB that belong together (they form the command) are sent one after the other without much delay. Every time a byte is received the watchdog timer is reloaded to some value. Every millisecond the value is decreased by one. When the watchdog timer reaches 0, the firmware state variables that control the data reception are reset to the initial state. Thus, as long as a command transmission is active the watchdog will not expire. If there is some (predefined) time no communication activity the watchdog will expire and by resetting the state variables that control the data reception the communication channel is forced to be "in sync". For DOA the story is similar, but here the data *bits* are sent one after the other, and the watchdog timer is reloaded at every received *bit*. If some disturbance causes the communication to go out of sync, it is corrected as soon as a short pause between commands occurs. That the mechanism works is proved by "deaf" PHCC daughter boards after power-up. Without the watchdog functionality, they remain "deaf", but with the watchdog functionality they work OK. (Power-up can cause spurious pulses which can mistakenly be detected as data bits).

However, experiments have shown that this watchdog solution is not perfect either ☹ To solve the most common DOA communication problem at power-up start, a one second delay is implemented if DOA communication is selected. After the one second delay it is assumed that all noise spikes on the DOA communication lines (specifically clock) are gone, and the state machine is initialized. During the one second delay the diagnostic LED flashes at a high rate.

The watchdog functionality is by default disabled. If you are *manually* testing commands (for example using PuTTY), data bytes that belong to each other to form a complete command using the USB connection are transmitted with delay between them. If the watchdog mechanism is enabled you will never succeed sending a complete command, because the time delay between the first and second data byte is sufficient to let the watchdog do its work. Thus, your second data byte will be treated as if it is the first data byte of a next command. For this reason, it must be possible to disable the watchdog. But here you get into a "chicken – egg" problem.
The watchdog command itself consists of 2 data bytes. As long as the watchdog is active, a manually sent command to disable the watchdog will fail, because of the described watchdog action. Therefore, the "disable watchdog" command is "special", because it does not require the

data byte. This solution only works for USB (only command and data byte) as it will make the command a single byte. Sending commands via the USB connection with a PC terminal communication program (for example PuTTY) will work fine. For DOA the solution will not work. However, if you use the PHCC TestTool for testing via the DOA connection, the watchdog activity will not be a problem. (Note that with the PHCC TestTool a message always consists of 3 data bytes).

A separate command is available to define the watchdog timer count-down value, and optionally, enable/disable the watchdog. The data format is as follows.

7	6	5	4	3	2	1	0
ENA	**0**	6 data bits representing the watchdog count-down value					

Bit 7 (“ENA”) controls whether the watchdog is enabled or disabled. When bit 7 is ‘1’ the watchdog is enabled. When bit 7 is ‘0’ the watchdog is disabled. So, you can disable the watchdog with this command and you can disable the watchdog with the special “disable watchdog” command. The 6 bits count-down value allows a less strict setting than default set by the firmware (8). If the 6 bits are all zero (‘000000’), the firmware default value is set. Note that you can set the count-down value also when the watchdog is disabled (“ENA” bit is ‘0’).

6.5 Direct control of Fuel Flow Indicator setpoint

With the commands 0 ~ 15 you can set the indication to any value within the 80000 PPH range. The firmware converts the command data to amplitude and polarity settings for the three stator coils in the FFI. However, with the “direct access” commands (35 ~ 41) you can control the amplitude and polarity settings for the three stator coils yourself. Maybe you come up with a better (more linear) control of the FFI ! The commands are available for direct control ☺

7. Assembly of the Altimeter circuit board

Only a fine low-wattage soldering is required. Read through the steps below to have an idea of the work you are about to do.

The on-board 400 Hz oscillator and “SR” (rotor) output are not used for the FFI.
You do not need to install D1, D2, IC10, C30, C31, R30 and R37 up to R45 (including).

Take your time to solder the components on the PCB. Better spend a few more minutes working accurately now, than searching for that little solder excess that causes a short circuit.
Soldering the components in order from smallest height to higher has the advantage that the board lays stable on your desk while soldering and keeps the component against the PCB. Therefore the following soldering order is proposed. All components are placed on the component side of the PCB. The component side has the white text painted on the PCB (the so-called silkscreen). See the “Appendix B – Component locator” for reference.

Observe ESD safety measures to prevent static discharge damage.
(This applies to the LED and ICs)

1. Solder all resistors, except the trim potentiometers. Make sure that the resistors with the different values are in their correct position. Check chapter 8 and Appendix B for reference. If you are not sure that you read the color code correctly, use an Ohm meter.
2. Solder the IC sockets, if you want to use sockets. I always use sockets, not to protect the ICs, but the PCB! If an IC is defective, it can easily be swapped. A soldered IC is difficult to remove and you likely damage the PCB traces or the through-hole plating. The PCB is more valuable than any of the ICs. Make sure that the notch, which indicates pin #1 location, is at the correct side. The silkscreen shows the notch.
3. Solder the crystal and the capacitors.
4. Solder the LED. The longer wire of the LED (anode) is “towards” the identification text.
5. Solder the tantalum polarized capacitor. Observe polarity!
6. Solder all pin headers, then the DOA connector and finally the USB connector.
7. Solder the 6 trim potentiometers.
8. Solder the DC/DC converter.

Before you proceed, do a visual inspection of the board with a bright light and magnifying glass.

- ✓ Are all soldering joints clean and shiny? A dull soldering joint may be a bad soldered joint.
- ✓ No small droplets of solder near the soldering joints?

TIP *You can use an old tooth brush to brush off solder residue and tiny solder droplets.*

© Functionality check.

1. Connect the +24 Volt power supply to the header 24V.POWER.
   Observe correct polarity!

2. Set the multimeter to measuring DC voltages.
   Connect the –lead to GND (0V of the 24V power supply).
   a. Hold the +lead tip on pin 8 of IC3. The meter should show +5V.
   b. Hold the +lead on pin 4 of IC3. The meter should show –5V.
3. Switch OFF the power supply (and multimeter ☺).

Put a jumper on pin #1 and pin #2 of the SRC header.

Install the ICs in their socket. See chapter 8 and Appendix B for reference.
Observe the correct orientation; pin #1 is at the notch side.

→ ***ICs with wrong orientation on the PCB will be damaged when power is applied!***

8. Parts list Altimeter circuit board

Quantity	Component	Description
1	IC1	PIC 18F2550
2	IC3, IC6, IC8, IC10*	TL082
1	IC4	CD4053
2	IC2, IC5	MCP4812
1	IC8	V2164D
1	IC9	TL084
6	IC socket DIL, 8-pin	high-quality machined pin socket
1	IC socket DIL, 14-pin	high-quality machined pin socket
2	IC socket DIL, 16-pin	high-quality machined pin socket
1	IC socket DIL, 28-pin	high-quality machined pin socket
1	LED	3 mm, any color
2	D1*, D2*	1N4148
1	Q1	20Mhz crystal
1	TMR3-2421	Traco Power dual DC/DC converter +/-5V 3W
13	C1, C5, C7, C9, C10, C11, C16, C17, C18, C19, C20, C24, C27	100 nF
2	C2, C3	18 pF
1	C4	220 nF
6 †	C8, C12, C13, C26, C28, C29	1 µF (non-polarized) – see † below
3	C14, C15, C25	560 pF
3	C21, C22, C23	470 pF
2	C30*, C31*	10 nF
1	C6	10 µF tantalum, polarized
1	R13	10k trim potentiometer RTRIM64Z
5	R18, R30, R31, R32, R33	20k trim potentiometer RTRIM64Z
3	R1, R4, R41*	1k
1	R2	390Ω
6	R3, R6*, R7*, R9, R11, R45*	10k
2	R5, R8⌘	100k
1	R10	12k
2	R12, R40*	6k8
6	R14, R15, R17, R27, R28, R29	150k
8	R20, R22, R23, R34, R35, R36, R37, R38	47k
3	R19, R21, R24	560Ω
3	R16, R25, R26	2k7
1	R39*	2k2
2	R42*, R43*	39k
1	R44*	56k
1	USB/DOA	2-pin male header
5	RST, OUT, ADJSW, SRC, DC/DC	3-pin male header
1	24V.POWER	4-pin male header
1	ALTIMETER	8-pin male header
1	DOA	2x5pin male header with shroud
1	USB	USB Type B
1	PCB	ALTIMETER V3.0

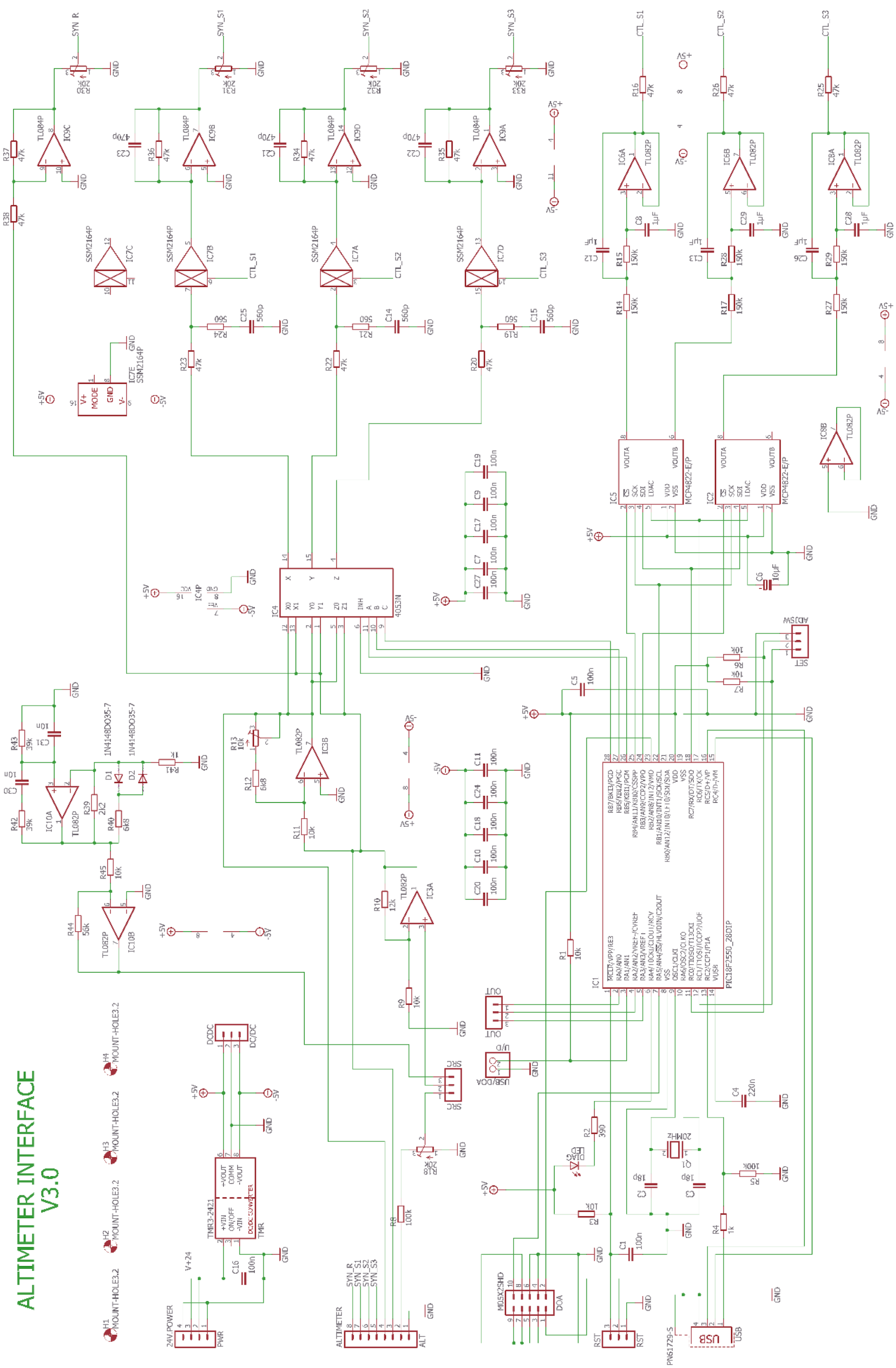
† These capacitors (C8, C12, C13, C26, C28, C29) determine the filter response. They may be lowered to decrease the lag on the indication, but may increase the “jitter” on the indication. They must have the same value.

* The components marked with an asterisk are NOT needed for the Fuel Flow Indicator.

⌘ R8 is 100k if the reference voltage is 26V AC -- R8 is 560k if the reference voltage is 115V AC.

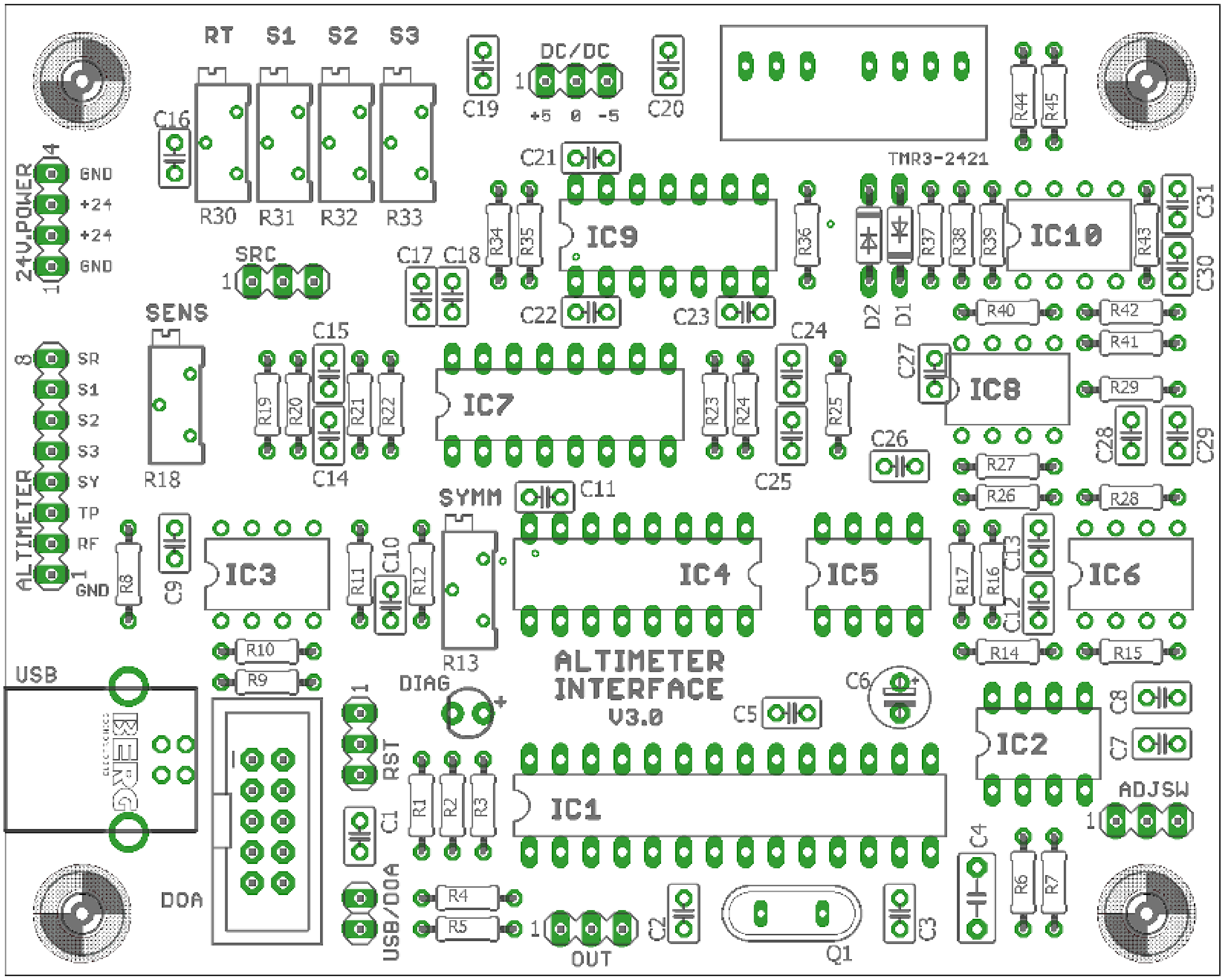
💣 Component values may change without notice.

Appendix A – Schematic diagram Altimeter circuit board

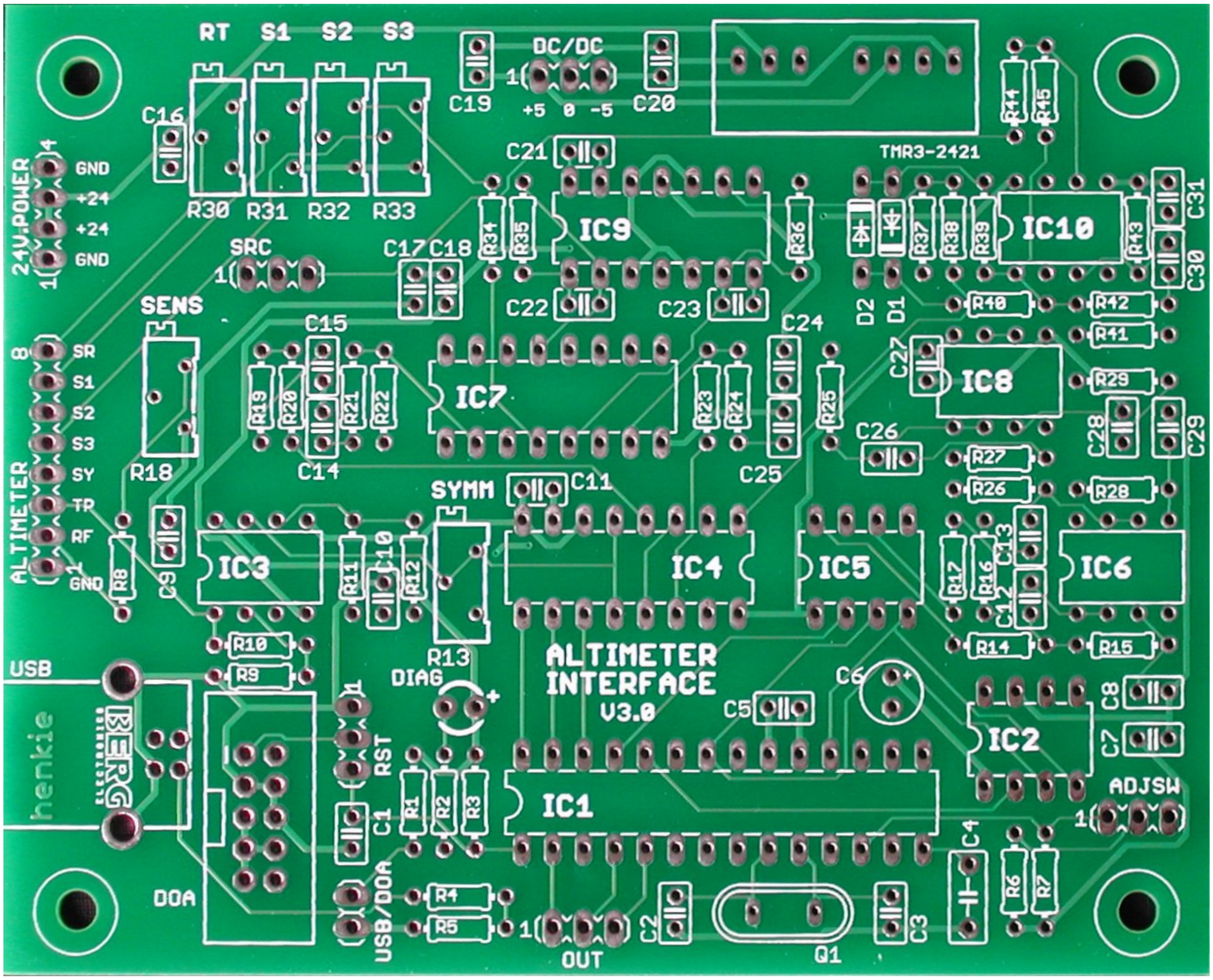


Appendix B. Component locator Altimeter circuit board

This image is retrieved from the actual Eagle .BRD file. For clarity all PCB traces are removed.



Appendix C – PCB component side Altimeter circuit board



(actual board picture)

Appendix D – Connection of the Fuel Flow Indicator

pin #	function
5	GND
6	115V AV 400 Hz “GND”
7	115V AC 400 Hz “HOT”
8	Internal illumination
9	Internal illumination
4	Synchro stator S2
11	Synchro stator S1
12	Synchro stator S3

Required are the connections 5, 6, 7, 4, 11, 12.

For internal instrument illumination pin 8 and 9 are needed.

Appendix E – Fuel Flow Indicator Demonstrator program

I wrote a test application in Python. The language allows for fast program development and with available libraries you can create programs with a user-friendly graphical interface. I do not pretend my Python program is well written – but it gets the job done ☺

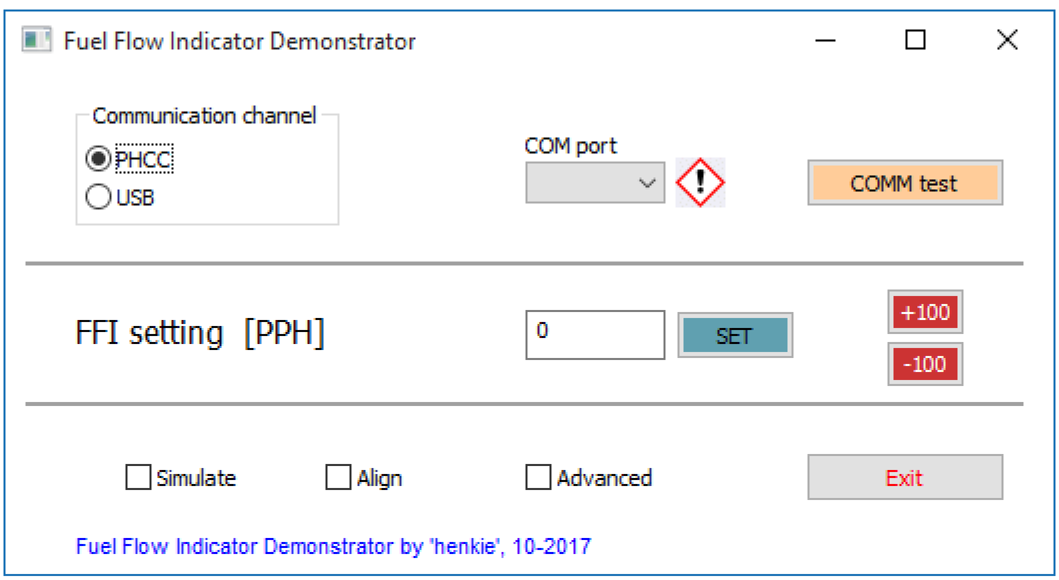
ffi-testappl.zip

The “Package” is a zip file that contains the Python application “ffi-testappl.py” and a small “error.gif” icon used by the application. Extract and put the Python and GIF file together in a folder. Open the file “ffi-testappl.py” with a text editor (notepad is fine). In the first lines you find a description of the steps you must do to install Python, wxPython and the Python serial package, needed to run the Fuel Flow Indicator test application. After the installations you can double-click the .py file to start the application, if all went well ☺

In the code you can also find the lookup table, and use that as your first “accuracy improvement”.

▶ Before you start the application program, you must have the interface connected to the PC. That is either a connection to the PHCC Motherboard or a USB cable.

Two windows appear when the program starts. The CMD window, that shows diagnostic output, and the application window.



The first action you must do is to tell the program how you connect to the altimeter interface. This can be using the PHCC Motherboard (DOA) or a USB connection. This is selected with the radio button in the section “Communication channel”. To the right of that section you can select the COM port which is used to connect (either DOA or USB). You can see in the diagnostic window which COM ports the program detected, but you can also check the Windows Device

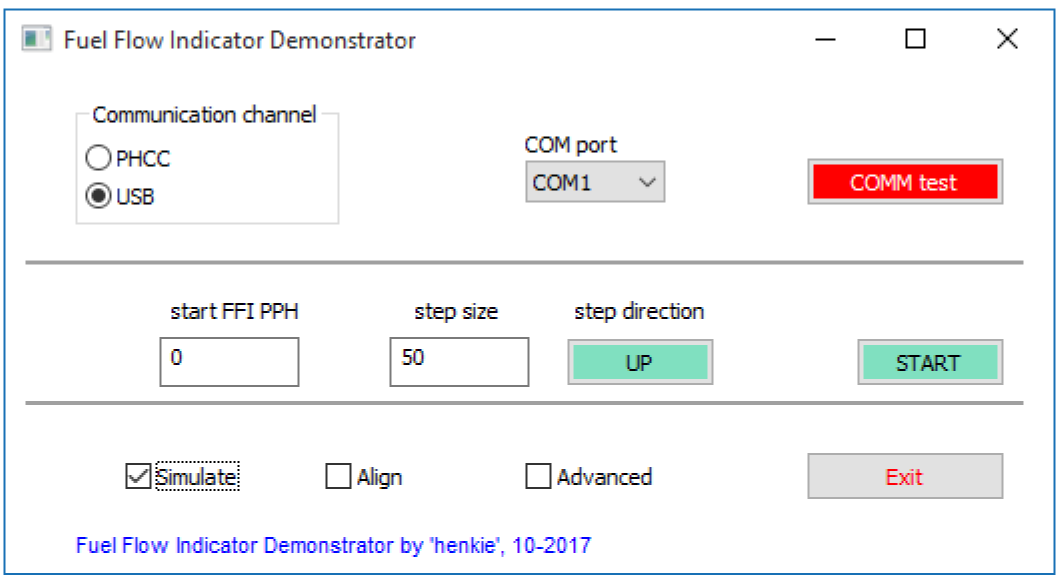
Manager under “Ports (COM & LPT)”. As long as you do not select a COM port the error sign will stay up blinking as a warning that you must select the COM port.

The “COMM test” button is an attempt to check the communications connection. This is only supported when USB is selected, but for now, it does not work properly, so forget this button ☺

In the “FFI setting [PPH] you can enter a value in the edit box. Any number < 80000 is allowed, but keep in mind that the entered number should be within say 2000 PPH of the actual indication of the FFI, otherwise you may stress the indicator needlessly a lot. Click the “SET” button to send the entered value to the FFI interface. You can also use the “+100” and “–100” buttons to change the value. Note that these two buttons imply the send command.

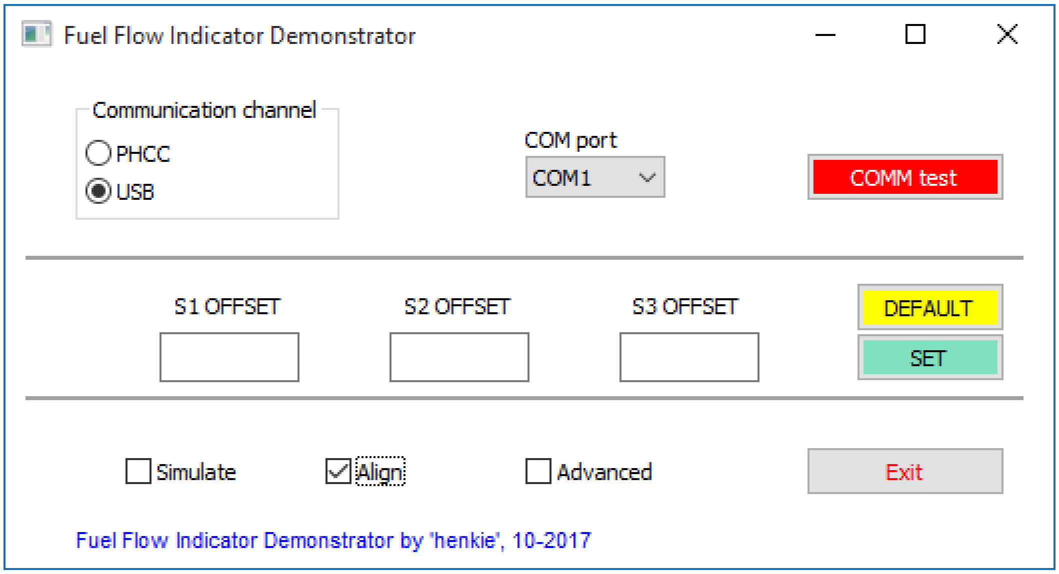
In the bottom section you see three checkboxes and the program “Exit” button. When you click a checkbox the “FFI setting [PPH]“ section changes accordingly. Clicking the same checkbox again removes the checkmark and the middle section shows the “FFI setting [PPH]“ again. If you click another checkbox, the middle section changes to that selection. The three checkboxes are mutually exclusive.

☑ **Simulate**



When you select the checkmark “Simulate”, the middle section shows the edit fields “start FFI PPH” and “step size”, and the two buttons “UP” and “START”. The “Simulate” dialog enables you to send a value setting to the FFI in time intervals of approx. 0.5 seconds. If you already changed the FFI setting with the “FFI setting [PPH]“ edit field, the “+100” or “–100” button, the set value is automatically copied into the “start FFI PPH” edit field. This value setting is used as the start for the simulated value change. The “step size” edit box allows you to set the value change per interval. With the button “UP” / “DOWN” you can set the value change “direction”. When this button is clicked, it automatically changes to the opposite direction for the next selection. Finally, the “START” buttons starts the sending of values to the FFI interface. When clicked, “START” changes to “STOP”.

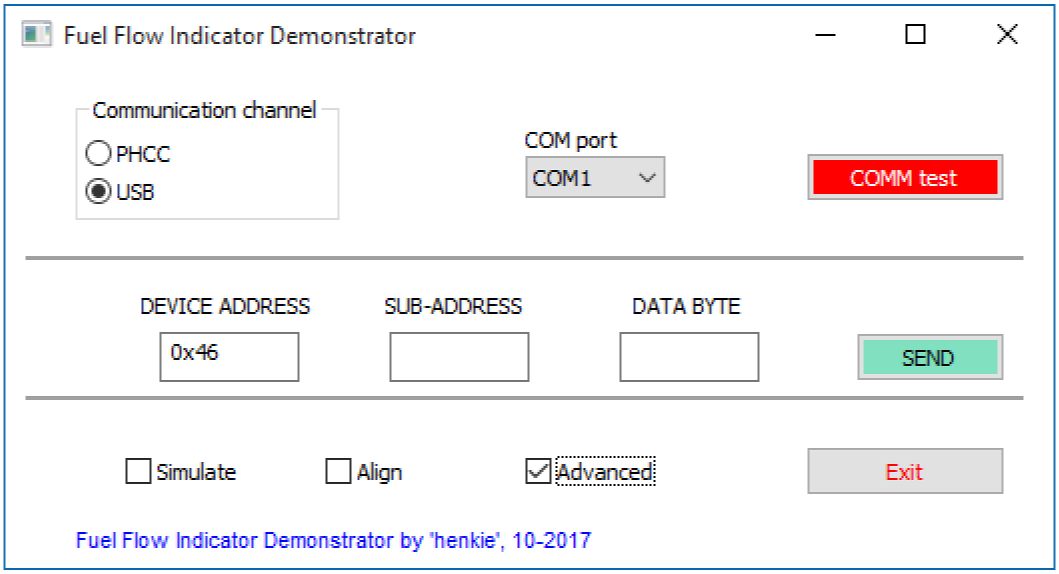
☑ Align



With the "Align" dialog, the middle section allows you to set different "offset" values for the 3 synchro stator coils of the FFI. The value must be in the range 0 .. 4095 for all three edit fields, and they must have an electrical phase relation of 120°. With the offset you can "zero-align" the instrument, that is, define an indicated value of 00000 PPH when value "0" is sent. If you changed the offset values, but are not sure of what you did you can click the "DEFAULT" button. The default offset values are then loaded in the edit fields. Only when you click the "SET" button the (new) offsets are sent to the FFI interface. To see the effect of the new offset, you must send, for example, a setpoint "0" command. Although you cannot cause any damage (as far as I know), making errors in the offset entry fields will surely lead to strange FFI settings or behavior.

☑ Advanced

The "Advanced" section allows you to send any command. The edit field "DEVICE ADDRESS" defines the device address for which the command is intended. If you use a USB connection, this byte is not sent; it is only relevant if you use PHCC. The device address is the first byte of a DOA command, and effectively defines which PHCC daughterboard must react on the received message. The "SUB-ADDRESS" and "DATA BYTE" edit fields are used for USB and for DOA. These two bytes define the command and, if applicable, the data required by the command. As this is the test application for the altimeter, the "DEVICE ADDRESS" edit field is preloaded with the device address of the FFI interface (ASCII capital letter "F", 46 hexadecimal).



When you click the "SEND" button the data is sent to the FFI interface.

With the "Exit" button you terminate the application.