STANDBY ATTITUDE DIRECTION INDICATOR

for the F-16 SIMULATOR

Version: V1.0 March 2018

History

01-2018 : Draft version, re-using 18F2550 PIC design

Initial work started for the Simtek Standby Attitude Direction Indicator (SADI). The Simtek SADI is a simulator instrument and the roll and pitch indication each are controlled by a sine/cosine pair of signals. The sine and cosine signals are generated using DACs with 10 bits resolution.

02-2018 : Breadboard setup of the design.

The interface hardware seems to work, but the SADI is not working. The sphere rotates continuously (both pitch and roll). After opening the SADI (this time I had to open the instrument), a defective CD4047 IC was found. After removing the defective IC, replacing it by an IC socket with a CD4047, the SADI did not move anymore after power was applied. Even better, It responded as expected when connected to the breadboard setup.

03-2018 : Version 1.0, describing the “Quad Sin/Cos interface” board and firmware

Remaining space on the PCB is used to implement two additional sine/cosine signal pairs. With these two sine/cosine pairs other similar simulated instruments can be controlled, for example the sphere of a simulated ADI or two small “needle-pointer” based instruments such as EPU etc. Each sine and cosine output can also be used as a simple analog output in the range –10V .. +10V (symmetrical adjustable) with 10-bit resolution. The PULL TO CAGE switch is also implemented for the SADI.
The command set has been defined.

Erratum See “Appendix A – Component locator QSC circuit board”.

Contents

1. **Introduction**
2. **Overview of the Simtek SADI solution**
    1. **Block diagram**
    2. **Power supplies**
    3. **Interface connections**
3. **Connections & jumpers on the QSC**
    1. **EXT5V connector**
    2. **PWR connector**
    3. **SADI connector**
    4. **DVC2.3 connector**
    5. **DOA connector**
    6. **USB connector**
    7. **USB/DOA jumper**
    8. **RST jumper**
4. **Assembly & adjustments of the QSC**
5. **Connection wiring**
    1. **Introduction**
    2. **Power supply wiring**
    3. **Simtek SADI wiring**
    4. **Interface wiring**
6. **QSC commands**
    1. **How to control the SADI**
        1. **SADI roll and pitch setting (angle)**
        2. **SADI roll and pitch setting (value)**
    2. **IDENTIFY command**
    3. **USB debug command**
    4. **Watchdog functionality**
7. **Parts list QSC circuit board**

Appendix A – Component locator QSC circuit board
Appendix B – Connection of the Simtek SADI
Appendix C – QSC SADI Demonstrator program

1. Introduction

In the top right corner of the instrument panel is the Standby Attitude Direction Indicator, from now on abbreviated to “SADI”. The real instrument in the F-16 is based on a gyro. The gyro uses 115V AC 400 Hz 3-phase power to spin the motor of the gyro. Unless your F-16 simulator is installed on a *full* motion platform, the real SADI will only make the spin up sound and then will remain continuously “horizontal leveled” positioned (as your pit does not roll or pitch). If you want the SADI to move and show a correct indication, you have two options. The first option requires mechanical work: remove the gyro and come up with some sort of servo or stepper motor construction to create the roll and pitch movement of the sphere. The other solution is finding a SADI which is specifically made for “simulator use only”. Those SADIs do not have a gyro, but already contain some drive mechanism for the roll and pitch indication. There are several variations in the type of interface required for these “simulator use only” instruments. As I have a SADI made by Simtek, I am developing hardware for that SADI. The roll movement is based on two signals, a sine and a cosine voltage each with a specific amplitude that results in a specific indicator position. Likewise for the pitch indication. So, the interface must generate two sine/cosine pairs of signals.

As the PIC has several unused I/O pins, and some board space was still available after designing the interface for the Simtek SADI, I decided to use the I/O pins and board space for four additional analog outputs. These 4 analog signal outputs can be used to drive indicators that need an analog voltage, but the outputs can also generate sine/cosine voltages if the to-be-connected instrument needs that type of signals.

The Simtek SADI requires two power supply voltages, +15V DC and –15V DC. It is logical to design an interface that will work with the same supply voltages, thus eliminating additional power supplies and wiring.

The following parts are needed to control the Simtek SADI.

- The “Quad Sin/Cos interface” circuit board V1.0
  The “Quad Sin/Cos interface” circuit board (from now on abbreviated to “QSC”) requires the same power supply voltages as the Simtek SADI, +15V DC and –15V DC. An on-board DC/DC converter generates the required +5V DC power supply voltage for the control logic. However, if you have a *clean* +5V DC already available in your pit, you can omit the DC/DC converter (that beast is almost $20). On the QSC is a programmed PIC processor that generates all signals based on the received commands. Communication uses either USB or PHCC DOA.
- +15V DC and –15V DC power supplies
  These power supply voltages are connected to the QSC. A dedicated connector on the QSC connects the Simtek SADI.

The QSC V1.0 circuit board has the following features.

- Small PCB (3.8” x 3.9”), easy to build, no SMD components
- Generation of 2 sine/cosine signal pairs with 10 bits resolution (1024 discrete values)
- One input for the “PULL TO CAGE” switch of the Simtek SADI
- One output for the “OFF” flag of the Simtek SADI
- 2 additional sine/cosine signal pairs, also usable as 4 separate analog output signals, adjustable symmetrical range –10V...+10V (10-bit resolution)
- 1 additional digital output to control an external device (ON/OFF)
- Amplitude adjustment of each sine and cosine output to exactly 10V DC
- On-board +5V power supply
- All required connections for the SADI application at one side of the circuit board.
- All adjustment trim potentiometers at one side of the circuit board.

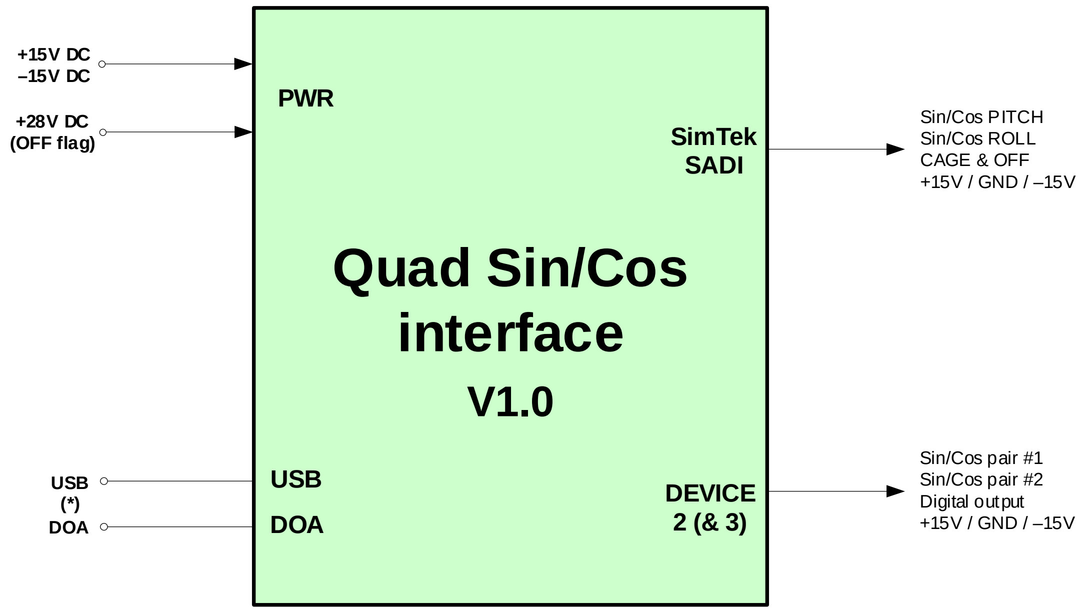
Disclaimer

All use of the QSC and all mentioned hardware are solely *your* responsibility. Any damage to the QSC, 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 Simtek SADI solution

2.1 Block diagram

The block diagram shows the complete Simtek SADI solution.



Not visible from the block diagram is that all connections are on one side of the PCB.

The "PWR" connector connects the +15V / GND / –15V DC power supply to the QSC. The "SADI" connector has all connections required for the SADI. Those are the 2 pairs of sin/cos signals for pitch and roll indication, a digital input for the "PULL TO CAGE" switch, and a digital output for the FLAG. Note that (according to Simtek specification) the OFF flag needs +28V DC, but +24V DC is fine. The SADI connector also supplies the +/–15V DC to the SADI. Internal illumination of the instrument must be wired separately.

The DVC2.3 connector has also 2 pairs of sin/cos signals, a digital output and the +/–15V DC connection. DVC2.3 can be used for one indicator that needs two pairs of sin/cos signals, or two indicators that each need one pair of sin/cos signal, or as 4 separate analog outputs. However, there is only 1 digital output available on DVC2.3. This output is directly connected to the PIC, can sink/source at maximum 20 mA, and has no protection circuitry.

2.2 Power supplies

The QSC requires +15V and –15V DC for proper operation. If you have *clean* +5V DC available in your pit, you do not need the on-board DC/DC converter. The DC/DC converter generates the +5V required for the PIC processor and the DACs. The OFF flag of the SADI needs a separate 28V DC power supply, but 24V DC works fine.

2.3 Interface connections

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

3. Connections & jumpers on the QSC

The QSC has 6 connectors and 2 jumpers. Pin #1 is indicated on the PCB.

3.1 EXT5V connector

The EXT5V connector pins are power supply **input** pins to connect external power supply.
The EXT5V connector and on-board DC/DC converter installation is mutual exclusive.

pin number	Signal	Usage
1	GND	Ground (0V)
2	GND	Ground (0V)
3	+5V	+5V input *(only if TMR-1211 is **not** installed)*

3.2 PWR connector

The PWR connector connects a the +15Volt and –15V DC power supply to the QSC. Make sure that you connect the power supply correctly. However, if you plug the connector in the opposite way everything is fine as the connections are “mirrored duplicated”.

→ ***The QSC has no provisions against wrong power supply connections !***

pin number	Signal	Usage
1	–15V	–15V DC power supply
2	+15V	+15V DC power supply
3	GND	Ground (0V)
4	GND	Ground (0V)
5	+15V	+15V DC power supply
6	–15V	–15V DC power supply

3.3 SADI connector

The SADI connector connects the Simtek SADI.

pin number	Signal	Usage
1	+15V	+15V DC power supply
2	SIN ROLL	Sine signal roll
3	–15V	–15V DC power supply
4	SIN PITCH	Sine signal pitch
5	GND	Ground (0V)
6	COS ROLL	Cosine signal roll
7	GND	Ground (0V)
8	GND	Ground (0V)
9	CAGE	PULL TO CAGE switch
10	COS PITCH	Cosine signal pitch
11	OFF+	OFF flag
12	+28V	+28V (only for OFF flag needed)

3.4 DVC2.3 connector

The DVC2.3 connector has 2 pairs of sin/cos signals and one digital output.

pin number	Signal	Usage
1	GND	Ground (0V)
2	COS2	Cosine signal device #2
3	GND	Ground (0V)
4	SIN2	Sine signal device #2
5	–15V	–15V DC power supply
6	COS3	Cosine signal device #3
7	+15V	+15V DC power supply
8	SIN3	Sine signal device #3
9	GND	Ground (0V)
10	OUT	Digital output

3.5 DOA connector

The DOA connector is the standard PHCC DOA 10-pin connector. See also “3.8 RST jumper”.

3.6 USB connector

The USB connector is a standard USB Type B connector.

3.7 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

3.8 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.

4. Assembly & adjustments of the QSC

Only a fine low-wattage soldering iron is required. Read through the steps below to have an idea of the work you are about to do. 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 or wrong component placement (especially resistors).

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 A – Component locator QSC circuit board” for reference.

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

1. Solder diode D1. Observe the orientation. The “bar” is the cathode.
2. Solder all resistors, except the trim potentiometers. Make sure that the resistors with the different values are in their correct position. Check chapter 7 and Appendix A for reference. If you are not sure that you read the color code correctly, use an Ohm meter.

   → Do **NOT** solder R26. This is an error in the design! R26 must be soldered as indicated in “Appendix A – Component locator QSC circuit board”.
3. 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.
4. Solder the crystal and the capacitors.
5. Solder the LED. The longer wire of the LED (+ anode) is “towards” the RST header.
6. Solder the tantalum polarized capacitor. Observe polarity!
7. Solder all pin headers, then the DOA connector and finally the USB connector.
   Note: if you use the DC/DC converter **do not** solder the header EXT5V.
8. Solder the 9 trim potentiometers.
9. Solder the DC/DC converter (if used).
10. Solder the 2 transistors. T1 is BC640, T2 is BC547.
11. Solder the “patched” R26 at the solder side of the board. See Appendix A.

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.*

© First check.

1. Connect the +15V, GND, and –15V DC power supply to the header PWR.
   Observe correct polarity!

2. Set the multimeter to measuring DC voltages.
   Connect the –lead to GND (0V of the +15/–15V power supply).
   a. Hold the +lead on pin 7 of IC12. The meter should show +15V.
   b. Hold the +lead on pin 4 of IC12. The meter should show –15V.

3. Switch OFF the power supply (and multimeter ☺).

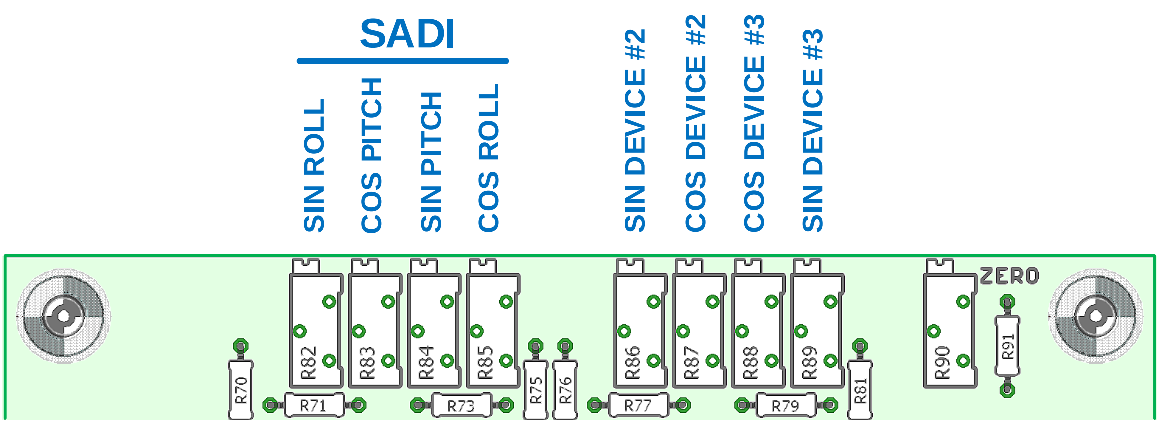
Adjustments

The QSC has 9 (!) trim potentiometers. The ZERO trim potentiometer is used to adjust the amplitude symmetry of the generated sine and cosine signals. The other 8 trim potentiometers set the maximum amplitude of each sine and cosine signal.
Do the following steps to adjust the symmetry (ZERO) and signal maximum amplitudes.

1. Make sure all power supplies are switched off.
2. Install IC12 (µA741) in its socket. Note correct orientation!
3. Connect the header "PWR" on the SQC to +15V, GND, and –15V DC power supplies.
   - Check once more that these connections are correct!
4. Connect a multimeter.
   - – lead connected to the power supply GND.
   - + lead connected to the TP1 (below IC12), or pin #2, or pin #6 of the OpAmp.
5. Set the multimeter to measuring DC voltages.
6. Switch on the +15V and –15V DC power supplies.
7. Adjust the trim potentiometer R90 "ZERO" so that the multimeter reads 2.04 V.
   This is a coarse adjustment, but it will be close. Fine adjustment can be done in step 20.
8. Switch off the power supplies.
9. Disconnect the header "PWR" and disconnect the multimeter.
10. Put all other ICs in their sockets. See chapter 7 and Appendix A 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!
11. If you use the PHCC Motherboard, connect the QSC "DOA" header to the Motherboard, and make sure that the jumper is not installed on the header "DOA/USB".
    If you use the USB connection, make sure that the jumper is installed on the "DOA/USB" header.
12. Connect the multimeter.
    - – lead connected to the power supply GND.
13. Connect the header "PWR" on the QSC to +15V, GND, and –15V DC power supplies.
14. Switch on the +15V and –15V DC power supplies.
15. If you use the QSC "USB" connection, connect the USB cable to the PC.
16. Start the "Simtek SADI Demonstrator" program.
17. Select the USB or PHCC radio button, and select the COM port in the application.

18. Checkmark the “Align / Test” checkbox.
19. Perform the amplitude adjustments as described in the following table.

Action in the Align / Test section		Explanation
- Select radio button “min” - Select SIN/COS channel 1 – click SEND - Select SIN/COS channel 2 – click SEND - Select SIN/COS channel 3 – click SEND - Select SIN/COS channel 4 – click SEND		These commands set the sine and cosine outputs of all “channels” to the lowest possible negative value.
Measurement: connect multimeter + lead to		Adjustment
Header “SADI”, pin #2	(SIN ROLL)	Trim R82 to meter reading –10V.
Header “SADI”, pin #6	(COS ROLL)	Trim R85 to meter reading –10V.
Header “SADI”, pin #4	(SIN PITCH)	Trim R84 to meter reading –10V.
Header “SADI”, pin #10	(COS PITCH)	Trim R83 to meter reading –10V.
Header “DVC2.3”, pin #4	(SIN DVC #2)	Trim R86 to meter reading to required value.
Header “DVC2.3”, pin #2	(COS DVC #2)	Trim R87 to meter reading to required value.
Header “DVC2.3”, pin #8	(SIN DVC #3)	Trim R89 to meter reading to required value.
Header “DVC2.3”, pin #6	(COS DVC #3)	Trim R88 to meter reading to required value.



You can check the symmetry by sending either “min” or “max” and verify that the output voltage is identical except for the polarity. A small tweak of the ZERO trim potentiometer might be required.

20. Terminate the “Simtek SADI Demonstrator“ application.
21. Switch off all power supplies.
22. Disconnect the multimeter.

After these adjustments you can connect the Simtek SADI to the QSC, see chapter 5.

1. Connect SADI header to the Simtek SADI.
2. Switch on all power supplies.
3. Start the “Simtek SADI Demonstrator“ application.
4. Play with it ☺

5. Connection wiring

5.1 Introduction

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

- power supply (+24V DC, +15V DC, –15V DC, GND) → QSC
  (+24V only needed for OFF flag)
- QSC → Simtek SADI
- QSC → PC (USB) or PHCC Motherboard (DOA).

5.2 Power supply wiring

The power supply wiring for the "SADI solution" is simple, just +15V and –15V DC to the QSC. On the QSC is a DC/DC converter that generates the +5V for the logic.

*If you have a clean +5V DC power supply voltage available in the pit, you can omit the DC/DC converter. If you want to use the EXT5V header to connect a 5V DC power supply, 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
+15V DC	+15V DC power supply	QSC "PWR" header
–15V DC	–15V DC power supply	QSC "PWR" header
GND	power supplies GND	QSC "PWR" header

⇨ *Always double-check the connections before you switch on the power supplies!*

5.3 Simtek SADI wiring

The SADI is connected to the QSC via the SADI connector. This connector connects +15V / –15V DC power, the pitch and roll sine and cosine signals, the "PULL TO CAGE" switch, and the OFF flag. The power supply for the OFF flag (Simtek specification says +28V DC) only needs to be connected to the SADI connector on the QSC. A test with 24V DC demonstrated that this lower voltage worked equally well.

Connection	From	to
+15V DC	QSC "SADI" header	Simtek SADI
–15V DC	QSC "SADI" header	Simtek SADI
GND	QSC "SADI" header	Simtek SADI (several connections)
SINE & COSINE ROLL	QSC "SADI" header	Simtek SADI (two connections)
SINE & COSINE PITCH	QSC "SADI" header	Simtek SADI (two connections)
PULL TO CAGE	QSC "SADI" header	Simtek SADI
OFF flag	QSC "SADI" header	Simtek SADI
+28V DC	+28V DC power supply	QSC "SADI" header

5.4 Interface wiring

The QSC must be connected to the PC using the USB connection, or must be connected to the PHCC Motherboard using the DOA connection.

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 3.8 for details) from the QSC to the PHCC Motherboard. The QSC 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 QSC 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 QSC circuit board accordingly!

Connection	**From**	**To**
DOA	QSC “DOA” header	PHCC Motherboard board DOA
USB	QSC “USB” connector	PC

6. QSC commands

If the QSC 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 QSC 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 command and 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 commands with their possible data (byte) values. Remember that the QSC is a “listen-only” device, thus if you want to know what data was sent, the sending program must keep a local copy.

STANDBY ATTITUDE INDICATOR CONTROL

sub-address	data byte	function / description
0	0x00 … 0xFF	ROLL_0 – set ROLL indicator in “range” 0000 – 0255
1	0x00 … 0xFF	ROLL_1 – set ROLL indicator in “range” 0256 – 0511
2	0x00 … 0xFF	ROLL_2 – set ROLL indicator in “range” 0512 – 0767
3	0x00 … 0xFF	ROLL_3 – set ROLL indicator in “range” 0768 – 1023
4	0x00 … 0xFF	PITCH_0 – set PITCH indicator in “range” 0000 – 0255
5	0x00 … 0xFF	PITCH_1 – set PITCH indicator in “range” 0255 – 0511
6	0x00 … 0xFF	PITCH_2 – set PITCH indicator in “range” 0512 – 0767
7	0x00 … 0xFF	PITCH_3 – set PITCH indicator in “range” 0767 – 1023
8	0 or 1	SADI “OFF” flag (0 ≡ flag visible // 1 ≡ flag hidden)

DIRECT SINE/COSINE SIGNAL LOADING

sub-address	data byte	function / description
9	0x00 … 0xFF	SIN_0 – set SINE value 0000 – 0255
10	0x00 … 0xFF	SIN_1 – set SINE value 0256 – 0511
11	0x00 … 0xFF	SIN_2 – set SINE value 0512 – 0767
12	0x00 … 0xFF	SIN_3 – set SINE value 0768 – 1023
13	0x00 … 0xFF	COS_0 – set COSINE value 0000 – 0255
14	0x00 … 0xFF	COS_1 – set COSINE value 0255 – 0511
15	0x00 … 0xFF	COS_2 – set COSINE value 0512 – 0767
16	0x00 … 0xFF	COS_3 – set COSINE value 0767 – 1023
17	0 – 1 – 2 – 3	LOAD SINE/COSINE command for specified device 0 ≡ SADI ROLL // 1 ≡ SADI PITCH 2 ≡ DEVICE #2 // 3 ≡ DEVICE #3
18	0 or 1	Digital output DEVICE 2 / DEVICE 3

PULL to CAGE setpoint preset values

sub-address	data byte	function / description
19	0x00 … 0xFF	SIN_R0 – set ROLL sin value P2C in "range" 0000 – 0255
20	0x00 … 0xFF	SIN_R1 – set ROLL sin value P2C in "range" 0256 – 0511
21	0x00 … 0xFF	SIN_R2 – set ROLL sin value P2C in "range" 0512 – 0767
22	0x00 … 0xFF	SIN_R3 – set ROLL sin value P2C in "range" 0768 – 1023
23	0x00 … 0xFF	COS_R0 – set ROLL cos value P2C in "range" 0000 – 0255
24	0x00 … 0xFF	COS_R1 – set ROLL cos value P2C in "range" 0255 – 0511
25	0x00 … 0xFF	COS_R2 – set ROLL cos value P2C in "range" 0512 – 0767
26	0x00 … 0xFF	COS_R3 – set ROLL cos value P2C in "range" 0768 – 1023
27	0x00 … 0xFF	SIN_P0 – set PITCH sin value P2C in "range" 0000 – 0255
28	0x00 … 0xFF	SIN_P1 – set PITCH sin value P2C in "range" 0256 – 0511
29	0x00 … 0xFF	SIN_P2 – set PITCH sin value P2C in "range" 0512 – 0767
30	0x00 … 0xFF	SIN_P3 – set PITCH sin value P2C in "range" 0768 – 1023
31	0x00 … 0xFF	COS_P0 – set PITCH cos value P2C in "range" 0000 – 0255
32	0x00 … 0xFF	COS_P1 – set PITCH cos value P2C in "range" 0256 – 0511
33	0x00 … 0xFF	COS_P2 – set PITCH cos value P2C in "range" 0512 – 0767
34	0x00 … 0xFF	COS_P3 – set PITCH cos value P2C in "range" 0768 – 1023

WATCHDOG functionality

sub-address	data byte	function / description
35	(*)	Disable watchdog functionality **--** See text for description.
36	0x00 … 0xFF	**Watchdog control --** See text for description.

MISCELLEANOUS

sub-address	data byte	function / description
37	0 – 1 – 2 – 3	**DIAG** LED operation mode 0 – LED always OFF 1 – LED always ON 2 – LED flashes at heart beat rate (power-up default) 3 – LED toggles ON/OFF state per *accepted* command
38	don't care	**IDENTIFY** USB only: send identification "QSC vA.B $xy"
39	'N' or 'Y'	**USB** debug command (USB only ☺)

6.1 How to control the SADI

The sin/cos based actuators need two signal voltages. These are steady DC voltages, not sine waves as the signal name (sine/cosine) might suspect. Sine and cosine only relate to the voltage level of both signals as a pair for a given angle value. The position of the actuator (for example the roll or pitch indication of the SADI) is expressed as an angle. To position the actuator at the specified (mechanical) angle *alpha* you send a voltage that is the sine value of *alpha* to the “sin” input, and a voltage that is the cosine value of *alpha* to the “cos” input of the actuator. The value of a mathematical sine or cosine of an angle is always a value between -1 ... +1. The SADI requires a voltage between -10V … +10V. The level shifting and scaling to the required output voltage is realized by on-board amplifiers.

6.1.1 SADI roll and pitch setting (angle)

The first 8 commands (sub-address 0 ~ 7) set the sphere of the SADI to the desired position, based on the pitch angle value and the roll angle value. The roll angle value is sent to one of the 4 sub-addresses 0 ~ 3. The pitch angle value is sent to one of the 4 sub-addresses 4 ~ 7. The value of the angle is any value between 0° and 360°. However, you do not send an angle value expressed in “degrees”, but a numeric value. The numeric value lies between 0 and 1023 and this represents the 0° to 360° range. Thus a numeric increment of 1 represents approx. 0.35°. As the data byte can only hold 8 bits which gives a range of 0 – 255 values, you use the sub-address for the values larger than 255. Sub-address 0 (or 4) is for values 0 – 255. Sub-address 1 (or 5) is for values 256 – 511. Sub-address 2 (or 6) is for values 512 – 767. Sub-address 3 (or 7) is for values 768 – 1023. Example. Suppose you want to position roll at 140°. The numeric value is 140*1024/360 equals 398. The command to send is sub-address 1, data byte 142 (142=398–256).

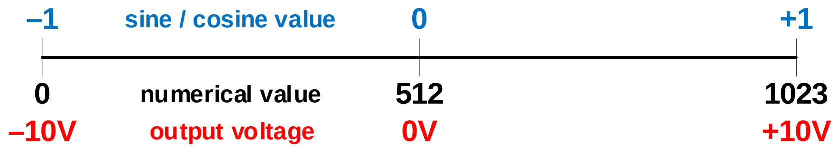
6.1.2 Sine and cosine setting (value)

In chapter 6.1.1 the positioning of the roll and pitch indication is described using the angle as input value. The firmware converts the specified angle (in degrees) to a numerical value between 0 and 1023. However, you can calculate the numerical values on the PC and then send the results to the QSC. In that case the firmware should (of course) not do a sine/cosine conversion, but accept the sent data “as is” and drive the indicators directly.

In case of sending an angle value, you send one “message” (the angle) and the QSC will generate the sine and cosine signal which is loaded *at the same time* to the instrument inputs. However, if you send numerical data, you must send *two* messages. One message for the sine value and one message for the cosine value. As time elapses between sending these two messages, the QSC cannot set the sine output and (some time later) set the cosine output. That could result in brief erratic “random in between” positioning of the indicator. So, a command is needed to indicate that the sine and cosine setpoint values are available and can be loaded. This is the function of command 17.

What is the numerical value?
Sub-addresses 9 ~ 12 set the sine value (range 0–255, 256–511, 512–767, 768–1023). Sub-addresses 13 ~ 16 set the cosine value (range 0–255, 256–511, 512–767, 768–1023). Sub-address 17 specifies for which axis (roll, pitch, device #2, device #3) the specified sine and cosine values are meant *and* executes the actual loading of these values.

The mathematical sine and cosine can have a value between –1 and +1. As the numerical data is “just a number” between 0 and 1023 (without sign), you must interpret the numerical value as follows. Numerical value 0 represents sine/cosine value of "–1" (output level is –10V). Numerical value 512 represents sine/cosine value of 0 (output level is 0 V). Numerical value 1023 represents sine/cosine value of "+1" (output level is +10 V).



Example. Suppose you want to position roll at 140°. The sine value of 140° is +0.643. This value is multiplied by 512 = 329. The numeric value is 512 + 329 = 841. The cosine value of 140° is –0.766. This value is multiplied by 512 = –392. The numeric value is 512 – 392 = 120. Thus, the commands to send are sub-address 12, data 73 (841 – 768). Then command sub-address 13, data 120. And finally to load these values sub-address 17, data 0 (for the roll axis of the SADI).

6.2 IDENTIFY command

The IDENTIFY command is only available when the USB communication connection is used. When the IDENTIFY command with an "any value" data byte is sent, the QSC returns a message in the form of "`QSC 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 QSC 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 '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 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 QSC 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 comamnd”, because it does not

require the data byte. This solution only works for USB 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’).

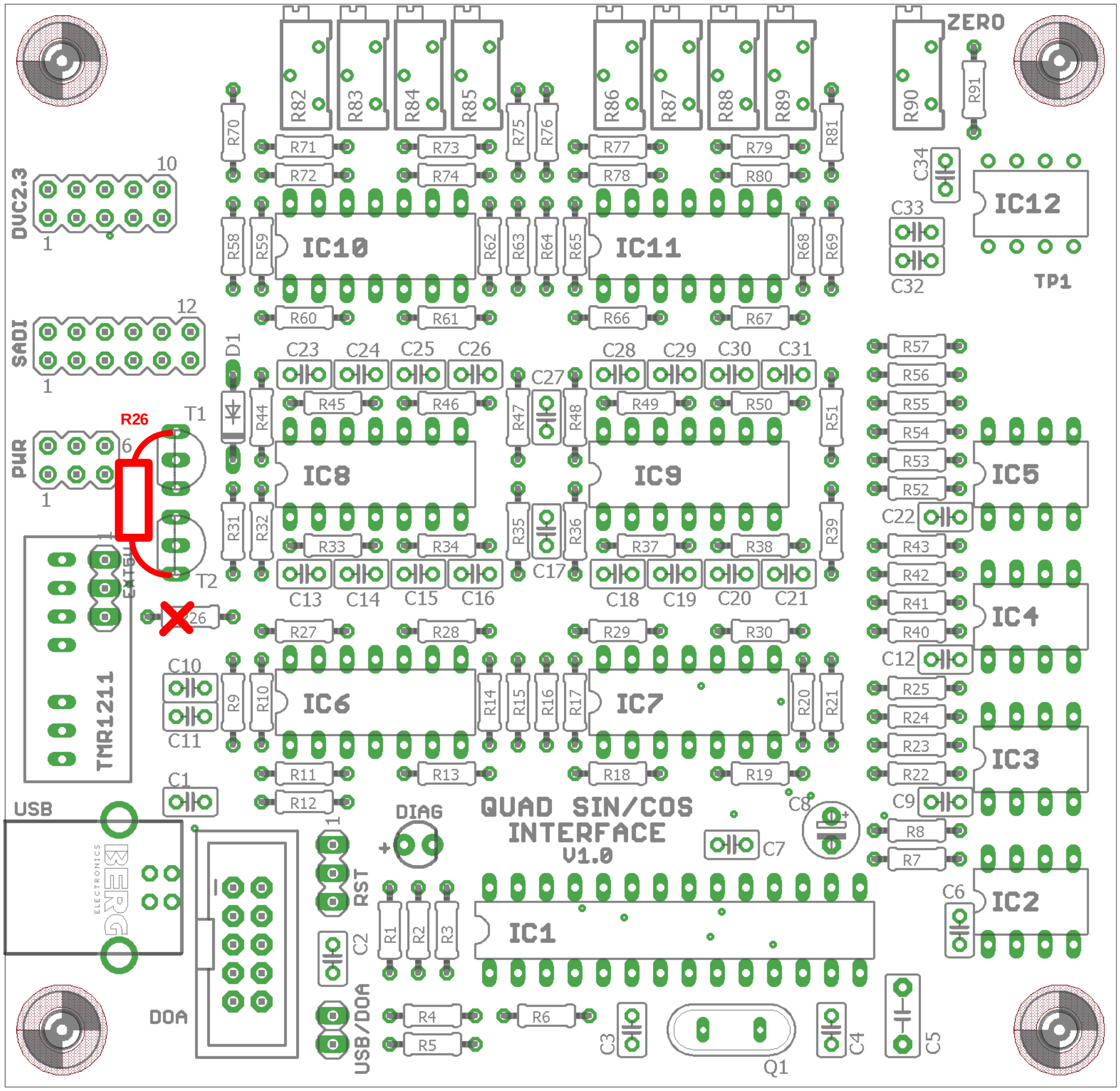
7. Parts list QSC circuit board

Quantity	Component	Description
1	IC1	PIC 18F2550
4	IC2, IC3, IC4, IC5	MCP4812
4	IC6, IC7, IC8, IC9	TL084
2	IC10, IC11	LM324
1	IC12	µA741
5	IC socket DIL, 8-pin	high-quality machined pin socket
6	IC socket DIL, 14-pin	high-quality machined pin socket
1	IC socket DIL, 28-pin	high-quality machined pin socket
1	LED	3 mm, any color
1	D1	1N4148
1	Q1	20Mhz crystal
1	T1	BC640
1	T2	BC547
1	TMR2-1211	Traco Power DC/DC converter +5V 2W
14	C1, C2, C6, C7, C9, C10, C11, C12, C17, C22 C27, C32, C33, C34	100 nF
2	C3, C4	18 pF
1	C5	220 nF
16 ⌘	C13, C14, C15, C16, C18, C19, C20, C21 C23, C24, C25, C26, C28, C29, C30, C31	470 nF (non-polarized) – see ⌘ below
1	C8	10 µF tantalum, polarized
9	R82, R83, R84, R85, R86, R87, R88, R89, R90	50k trim potentiometer RTRIM64Z
60	R1, R2, R3, R7, R8, R9, R10, R11, R13, R14, R15, R16, R17, R18, R19, R20, R21, R22, R23 R24, R25, R26, R27, R28, R29, R30, R40, R41 R42, R43, R52, R53, R54, R55, R56, R57, R58 R59, R60, R61, R62, R63, R64, R65, R66, R67 R68, R69, R70, R71, R72, R73, R74, R75, R76 R77, R78, R79, R80, R81	10k
3	R4, R12, R31	1k
2	R5, R91	100k
1	R6	470Ω
16	R32, R33, R34, R35, R36, R37, R38, R39 R44, R45, R46, R47, R48, R49, R50, R51	150k
1	USB/DOA	2-pin male header
1	RST	3-pin male header
1 ✝	EXT5V	3-pin male header
1	PWR	2x3-pin male header
1	DVC2.3	2x5-pin male header
1	SADI	2x6-pin male header
1	DOA	2x5pin male header with shroud
1	USB	USB Type B
1	PCB	QSC V1.0

⌘ These capacitors determine the filter response. They may be lowered *all* to decrease possible lag on the indication.
✝ Header only installed if DC/DC converter is *not* installed.
💣 Component values may change without notice.

Appendix A. Component locator QSC circuit board

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



ERRATUM

R26 as indicated on the PCB must NOT be soldered (see red cross). Instead, R26 must be soldered as indicated – "top connection" of T1 to "bottom connection" of T2. If R26 is soldered as indicated in the silkscreen on the PCB, no damage will occur, but the OFF flag will not work. To avoid a mistake, the solder "eyes" for R26 will be filled with solder.

Appendix B – Connection of the Simtek SADI

➔ Make sure that the table with connections that you use matches your SADI model number!

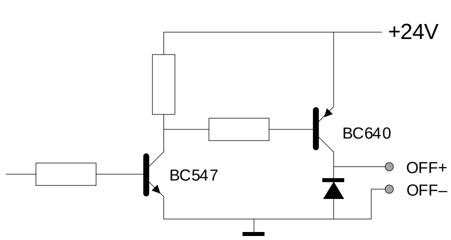
Model : 10-0672-02	
Pin	**Function**
A	+15V power supply
B	–15V power supply
C	Pitch sine signal
D	Pitch cosine signal
E	Pitch GND signal
CC	Roll sine signal
DD	Roll cosine signal
EE	Roll GND signal
K	GND power supply
L	Roll null test
M	Power Failure flag (+28V)
N	Power Failure flag (GND)
R	Chassis GND
S	Lightning +5V
T	Lighting GND
LL	Pitch null test
V	Cage switch input
W	Cage switch circuit make

Model : 10-0335-01	
Pin	**Function**
A	+15V power supply
B	GND power supply
C	–15V power supply
D	Spare
E	Spare
F	Spare
G	Spare
H	5V lighting
J	5V lighting
K	Case GND
L	Roll sine signal
M	Roll GND signal
N	Roll cosine signal
P	Pitch sine signal
R	Pitch GND signal
S	Pitch cosine signal
T	Power Failure flag (+28V)
U	Power Failure flag (GND)
V	Cage GND out

Model : 10-1084	
Pin	**Function**
A	+15V power supply
B	GND power supply
C	–15V power supply
D	Lighting GND
E	Lightning +5V
F	Pitch/Roll GND signal
G	Power Failure flag (+28V)
H	Chassis GND
J	Spare
K	Roll sine signal
L	Roll cosine signal
M	+24V power return
N	Spare
P	Cage switch out common
R	Cage switch out n.o.
S	Spare
T	Spare
U	Pitch sine signal
V	Pitch cosine signal

NOTE:
I can only say that the table for Model 10-0335-01 is correct, because I have that SADI. Therefore, I am not accepting any responsibility for correctness of the tables. Please check the SADI connections yourself.

OFF flag connection to the QSC



OFF flag output circuit of the QSC

As you can see, one side of the coil of the OFF flag is connected to GND and the positive connection of the coil must be connected to the collector of the BC640. If the SADI has a diode connected across the coil, the diode D1 on the QSC can be omitted (but it does not harm to have the diode installed).

Appendix C – QSC SADI Demonstrator program

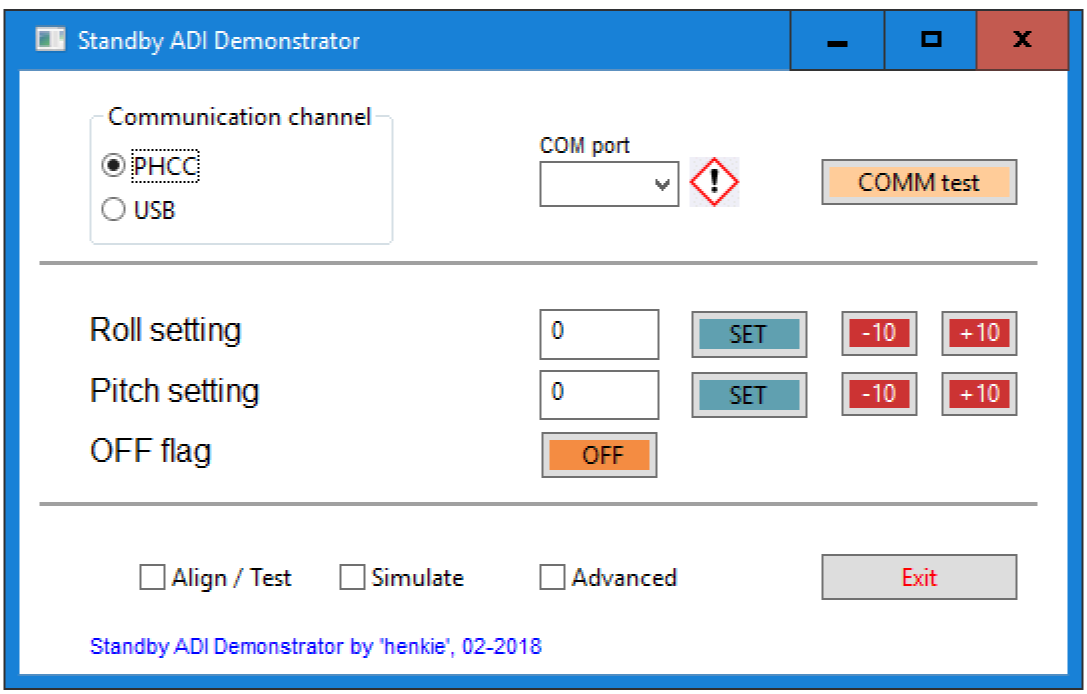
I wrote a test application in Python. The language allows for fast program development and with available libraries you can create programs with a user-friendly graphical interface. I do not pretend my Python program is well-written – it gets the job done ☺

sqc-testappl.zip

The “Package” is a zip file that contains the Python application “qsc-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 “qsc-testappl.py” with a text editor (notepad is fine). In the first lines you can read a description of the steps you must do to install the Python environment, wxPython and the Python serial package, needed to run the QSC test application. After the installations you can double-click the .py file to start the application, if all went well ☺.

▶ 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.



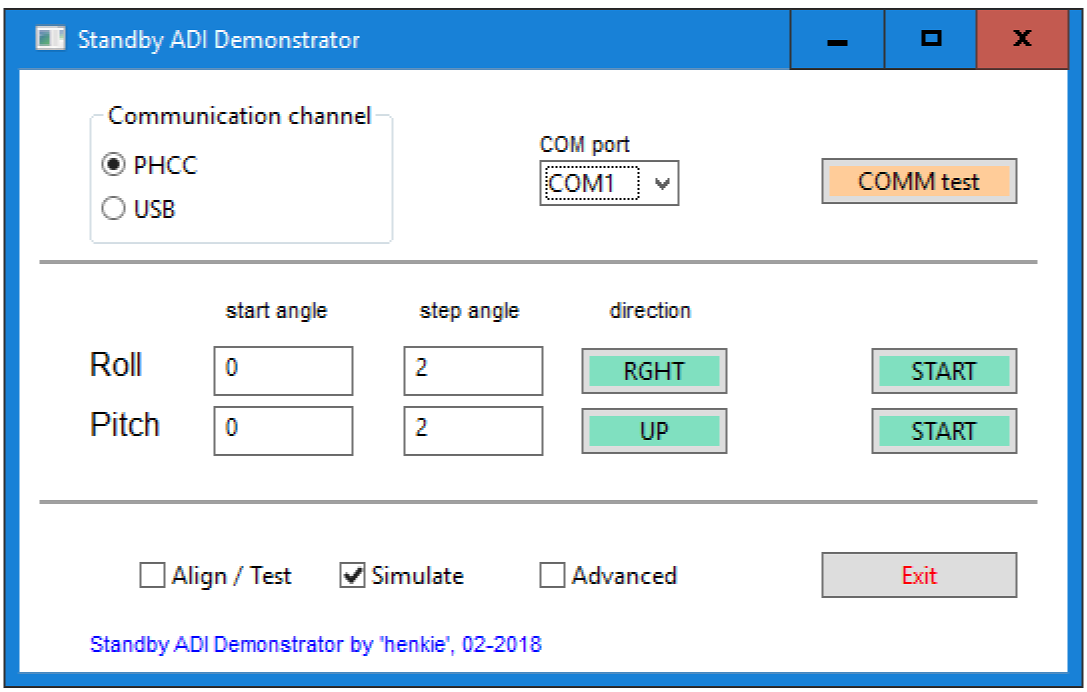
The first action you must do is to tell the program how you connect to the QSC interface. This can be using the PHCC Motherboard (DOA) or a USB connection. This is selected with the radio buttons in the section “Communication channel”. To the right of that section you can select the COM port which is 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 fields "Roll setting" and "Pitch setting" you can enter a value in the edit box. Any number ≤ 360 is allowed. Click the "SET" button to send the entered value to the QSC interface. You can also use the "+10" and "–10" buttons to change the value. Note that the "+10" and "–10" buttons imply the send command. The "OFF" button toggles the OFF flag of the SADI.

In the bottom section you see three checkboxes and the program "Exit" button. When you click a checkbox the middle section changes accordingly. Clicking the same checkbox again removes the checkmark and the middle section shows the Roll and Pitch entry fields again. If you click another checkbox, the middle section changes to that selection. The three checkboxes are mutually exclusive.

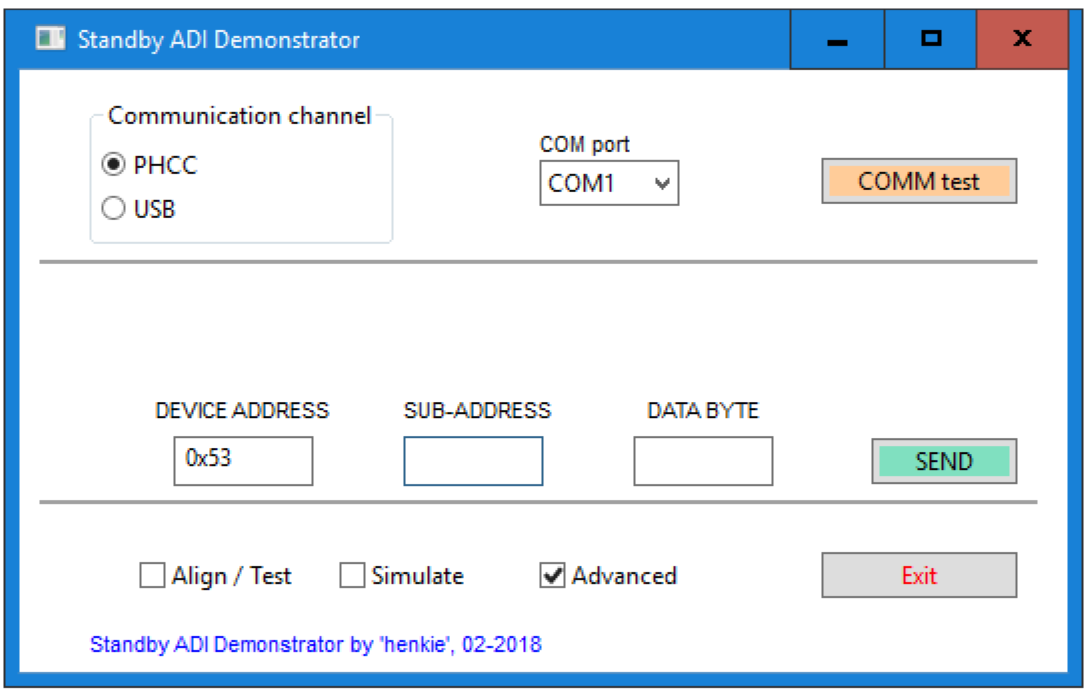
☑ Simulate



When you select the checkmark "Simulate", the middle section shows the edit fields "start angle" and "step angle", the two buttons for the update direction ("LEFT"/"RGHT" and "UP"/"DOWN") and "START" buttons to start the simulated movement. The "Simulate" dialog enables you to send a value setting to the SADI in time intervals of approx. 1 second. If you already changed the roll and/or pitch setting with the setting edit fields, or the "+10" or "–10" buttons, the set value is automatically copied into the "start angle" edit field. This value setting is used as the start for the simulated value change. The “step angle” edit box allows you to set the value change per interval. With the direction buttons 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 SADI interface. When clicked, “START” changes to “STOP”. When the update reaches the maximum value (either 0 or 360), the update is stopped and the button automatically changes to “START”.

☑ Advanced

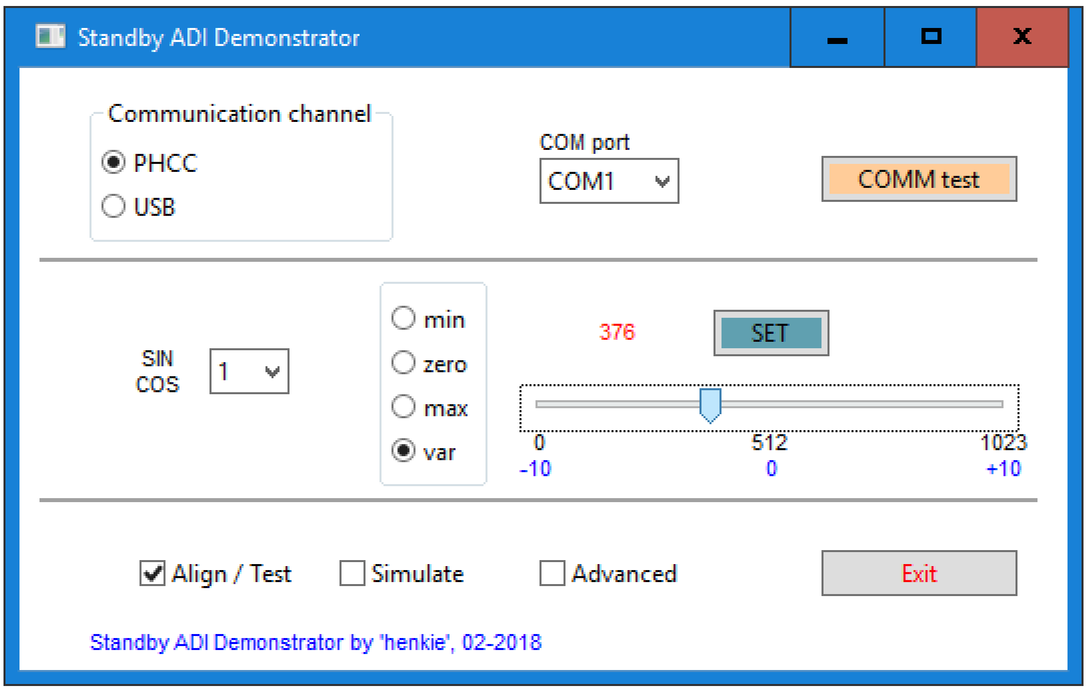
The “Advanced” section allows you to send any command. The edit field “DEVICE ADDRESS” defines the device address for which the command is intended. If you use a USB connection, this byte is not sent; it is only relevant if you use PHCC. The device address is the first byte of a DOA command, and effectively 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 QSC, the “DEVICE ADDRESS” edit field is preloaded with the device address of the SADI interface (ASCII capital letter “S”, 53 hexadecimal). When you click the “SEND” button the data is sent to the QSC interface.



☑ **Align / Test**

The "Align / Test" section is useful when executing the adjustment of the trim potentiometers on the QSC. With the radio buttons "min", "zero", "max", and "var" you send the *same* value to both the sine and cosine output of a signal pair. The signal pair is selected with the SIN/COS drop-down list. "1" is the SADI ROLL sin/cos outputs, "2" is the SADI PITCH sin/cos outputs, "3" is the DEVICE #2 sin/cos outputs, and "4" is the DEVICE #3 sin/cos outputs.

When you select the radio button "min" and click the "SET" button the value 0 is sent (to the sine and cosine output of the selected pair). Likewise, "zero" will send the value 512, and "max" will send the value 1023. When you select the radio button "var" a slider appears, and at the left of the "SET" button is in red shown the value that will be sent when "SET" is clicked. With the slider you can set the value to be sent to any value. For convenience, the numerical value range and the output voltage range is shown below the slider.



With the "Exit" button you terminate the application.