JRP 19RPT01

Quantum traceability for AC power standards

D1 - Summary report on the development of an open hardware multiplexer based on a single Josephson chip. The multiplexer must be sufficiently wideband to cover the required frequency range (20 Hz to 1 kHz)

Version control

Version	Date	Comment
2	June 2021	First Report D1

Introduction

This document explains the design and implementation of a multiplexer to be used on a power measurements system based on the Programmable Josephson Voltage Standard (PJVS). The following figures shows an example, the digitizer measures one signal at the time, the current, the voltage and the Josephson voltage, this last one can be adjusted to calibrate the digitizer at the same level than the current or voltage signal to be measured.
The digitizer takes a fixed number of periods of each channel, so a switching time in the order of ms will be required. In addition, the switching must be synchronized with the signals so the multiplexer must include an external trigger.
In the power measurements system, the digitizer can be e.g. a Keysight 3458 multimeter or a NI-5922 oscilloscope.

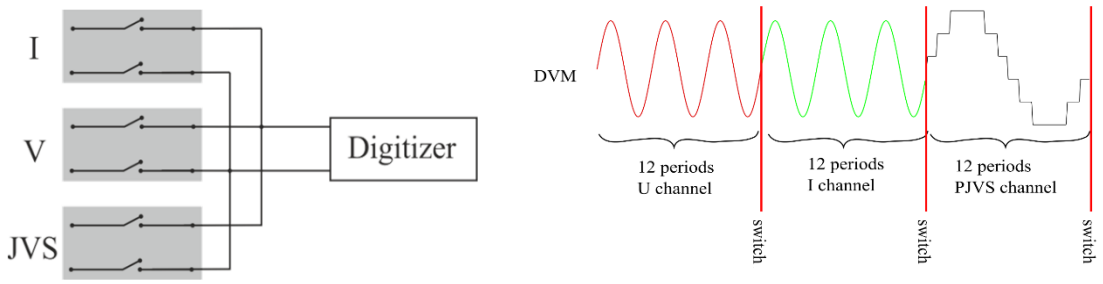


Figure 1: (left) Power measurement system and (right) signals measured by the digitizer. A digitizer measures the current, the voltage and the Josephson voltage in an alternating mode. Some periods of each signal are measured periodically and the PJVS can be adjusted to calibrate the corresponding range of the multimeter.

This multiplexer is designed to be flexible, the switching sequence and the trigger can be configured by a PC. In addition, it is built in a modular form, with a master board that controls slave boards and includes timing functionalities. The slave boards have a single pole and double throw (SPDT or 2-to-1), or a single pole and four throw (SPFT or 4-to-1). The user can use many of them controlled by the master, as the next figure shows.

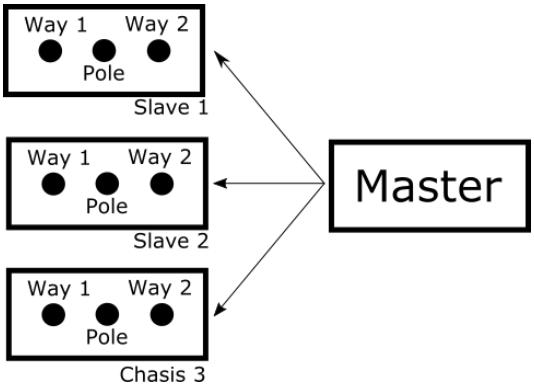


Figure 2: Three slave boards SPDT controlled by a single master board.

Specification

The modular design of the multiplexer allows multiple configurations that can be selected by the users connecting the boards between them. The multiplexer uses the strategies break-before-make in order to avoid short circuits between sources and it switches synchronized with an external signal.

The signals are considered differential, so the switch commutes positive and negative terminals of the voltage source. Two slave boards have been designed, one has two throws and one pole, as the next figure shows, and the other has four throws and one pole. The slave board is reversible; the SPDT board can be considered as 2 inputs - 1 output (2-to-1) or 1 input - 2 output (1-to-2).

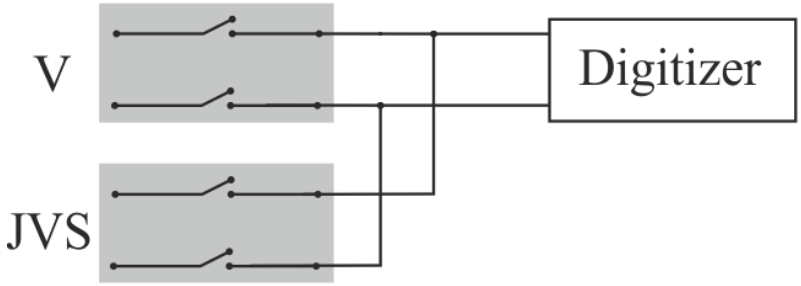


Figure 3: A slave board SPDT ready to measure a voltage source and the JVS.

Channel specification

- Maximum input voltage 60 V peak
- Turn on/off time: <3 ms
- Differential signal
- Coaxial connector for input and output signals
- Bounce-free operation
- Individual guard connection for each signals (binding post)
- Ground connection for the multiplexer
- Reversible operation

General specification

- Trigger input (BNC connector, opto-isolated)
- USB connection to PC for configuration and control
- 9 V DC External supply

Uncertainty contribution

- 0.5 Ω On-resistance
- Off-isolation at 1 MHz < -90 dB
- Crosstalk at 1 MHz < -100 dB
- Uncertainty contribution < 0.2 ppm at power line frequency
- Maximum uncertainty due to multiplexer: 1 ppm at 1 kHz.

Theory of operation

Slave board

Each switch is built in T-configuration using three solid-state relays to improve the overall off-isolation. The next figure shows a simple scheme in the blue box, S1 is connected to the voltage source and its output is connected with two relays. S3 connects the signal path to ground and S2 connects the signal path to the output. In the circuit schematic, the relays U3 - U7 are the signal relay for the high terminal and the U4 - U8 for the low terminal. The relays U5 and U6 are used to tie to ground the middle point, the ground is floating and can be connected to the source ground. The capacitances and resistor networks are used to improve transition time and reduced charge injection.

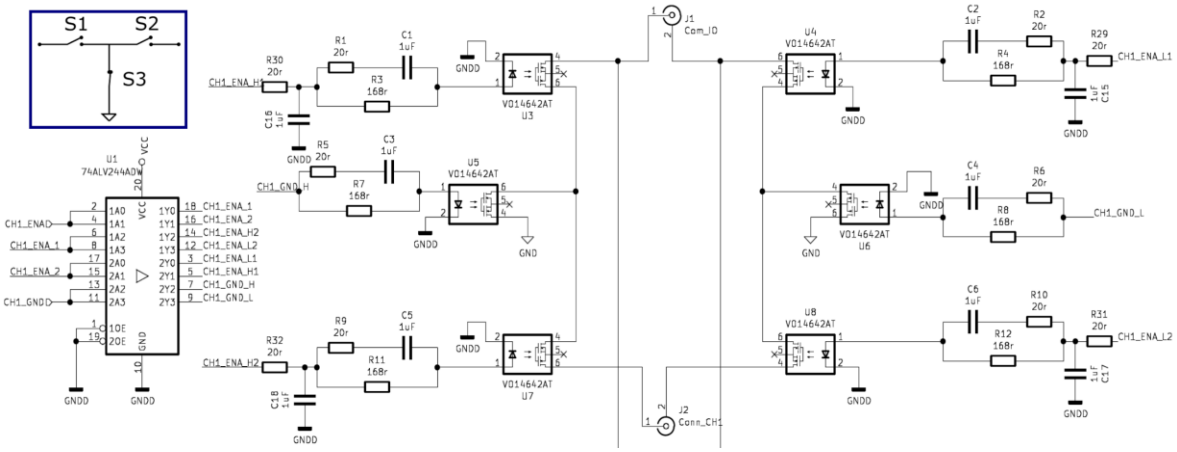


Figure 4: Circuit schematics for one differential channel, the switches are connected for each polarity in T configuration as the blue box shows.

The circuit board has a coaxial design, minimizing inductance path and stray capacitance. The figure 5 shows the top and bottom side of the slave-board, on one side are the high terminal relays and the other side has the low terminal relays. Both sides are organized equally to minimize the area enclosed by the current path. Three guarding rings are used to reduce the stray capacitance between channels and output, they can be connected to the instrument's guards. Several test points were included to measure the digital signal, they are indicated in the top side.

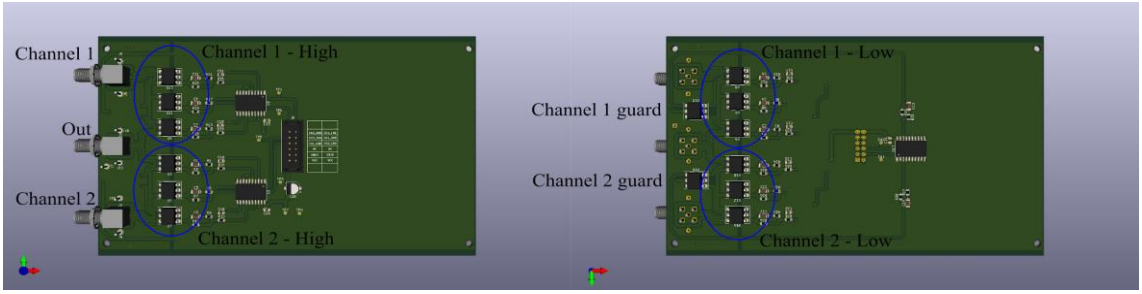


Figure 5: (left) Top and (right) bottom of the slave 2-to1 board.

During the open or close process, the activation of each relay must be done preventing a short circuit. A break-before-make strategy is implemented: first all switches are opened and then the correct configuration is set. A secure time is included to prevent a short circuit and will be set considering the relay specifications.

Source of errors

The multiplexer based on Solid-State relay has some source of error that were considered in the design and will introduce an uncertainty in the measurement system. The most important components are listed below.

On-resistance: The on-resistance *Ron* is the resistance of the conducting path of the relay and it builds a voltage divider with the load resistance *RL*. Thus, the measured voltage is affected by a systematic error, that can be calculated relative to input voltage as *Error = Ron/ (Ron +RL).* In addition, leakage current due to finite impedance to ground can flow for this resistance increasing the error or introducing distortion. In order to reduce the effects of the on-resistance a relay with very low channel resistance and large impedance to ground was selected by mean of simulations.

Offset voltage: the relay presents some thermoelectric voltages in series with the conduction channel, which can introduce low-frequency noise. For the tested semiconductor relay, this voltage is negligible due to its small temperature gradient inside the device. In addition, the device do not present dependence between the voltage offset and the activation circuit.

Off-isolation and crosstalk: Both characteristics describe the effects of the input voltage on a channel in off-state to its output voltage (off-isolation) or to the output of a neighborhood channels (crosstalk). This can introduce distortion on the output signal that increase with the number of channels. The design of the multiplexer includes guard ring to improve isolation between channels and a symmetric design of the High and Low path to reduce the overall magnetic field pick up. In order to increase the off-isolation, a T configuration of the relay was used.

Guard connection

The slave board has switches to connect the guard of each throw with the guard of the pole, and the connectors are surrounded by a ring at guard potential. The figure shows the example to the slave board SPDT.

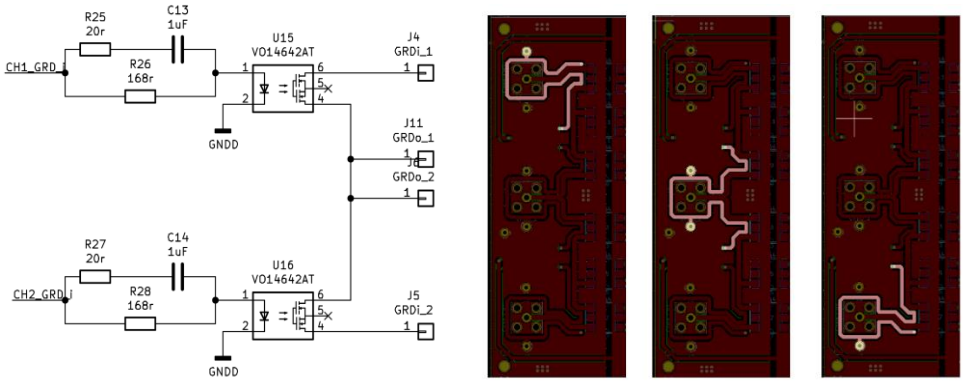


Figure 6: (left) Schematic circuit of the guard relays of the salve board SPDT. (right) guard rings on each connector.

PhotoMos relay

The following table shows the characteristic of the relay used on the first prototype. It was selected comparing by simulations elements from different manufacturer, and we found that this element has a balanced specification and it is available to buy.

Table 1: PhotoMos relay VO14642AABTR specification.

	VO14642AABTR
ON-Resistance	0.18 Ω (AC)
Max. voltage	60 V
On-time	370 µs / 800 µs
Off-time	50 µs / 800 µs
Stray capacitance	200 pF
Control current	0.5 mA to 20 mA
Family	Vishay

In the case that the application requires another specification on the relay, the market offers a large number of compatible devices with different capabilities. The following table shows some equivalents elements with its mains characteristic obtained from specifications.

Table 2: PhotoMos relay models and main characteristics.

Relay	**Main characteristics**
PVT2125SPbf	150 V input voltage
ASSR-1511-301E	Low stray capacitance
PVN012APbF	Low on-resistance

Firmware and Flow chart

Switching sequence matrix

The switching sequence is transferred to the multiplexer by USB port and internally stored for further uses. The switching sequence can be considered as a matrix where each row represents the state of all the slave boards. The transition between each row will be controlled by a switching event. The user can configure the row value and in practice no more than 32 rows will be necessary. For example, if 3 slave boards are connected in a 4-step process, the matrix is:

Table 3: Binary switching sequence matrix for SPDT slave boards.

Step	Slave 1 Ch1	Slave 1 Ch2	Slave 2 Ch1	Slave 2 Ch2	Slave 3 Ch1	Slave 3 Ch2	Delay time (number of external trigger)
1	1	0	1	0	1	0	10
2	0	1	0	1	0	1	15
3	0	1	0	1	0	1	15
4	1	0	1	0	1	0	10

In the binary matrix one bit indicates the state of one channel, 0 indicates open and 1 indicates close. The matrix can be built with two bytes for channel states, so a total of 16 switches can be controlled, plus an extra byte to the numbers of clocks.

Byte 1:

S4-CH2	S4-CH1	S3-CH2	S3-CH1	S2-CH2	S2-CH1	S1-CH2	S1-CH1

Byte 2:

				S6-CH2	S6-CH1	S5-CH2	S5-CH1

Byte 3: numbers of external trigger from 0 to 255

Remote operation

A trigger tree is considered to control the switching sequence, first the master board is armed and then, the state of the multiplexer channels change synchronized with the switching event. The arm and disarm message is provided by the PC. The switching event is defined as a trigger signal at trigger BNC plus a delay time based on integer numbers of trigger pulses. The following figure presents the trigger tree.

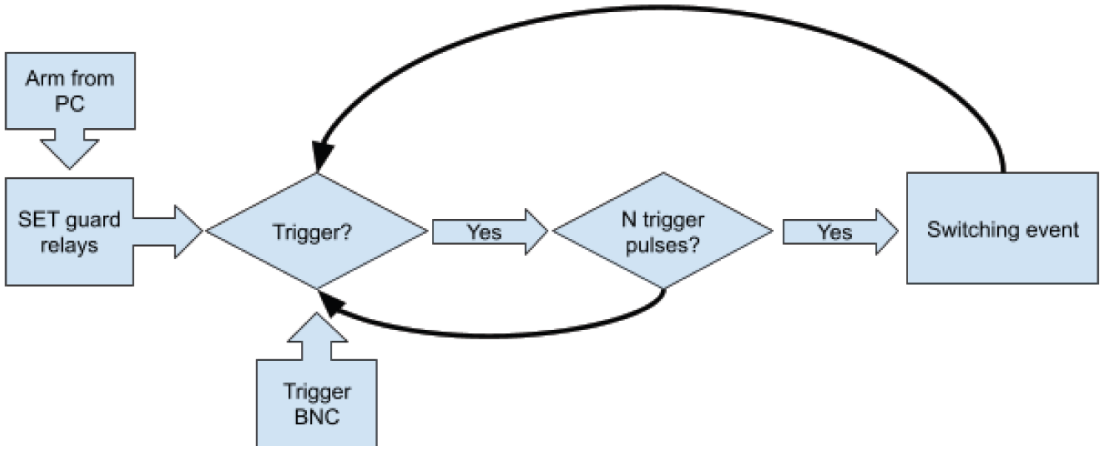


Figure 7: Trigger tree, the Arm message is provided by PC and "N" is the number of external trigger pulses configured for each row in the matrix sequence.

Switching pseudocode

The following sequence gives information about the switching procedure, for more information go to Annex 4.

1. The PC configures the multiplexer and loads the switching matrix.
2. The PC arm the multiplexer.
3. The switches that connect the source guard are activated and they are not modified during the sequence.
4. The controller detects a trigger signal; the user can select positive or negative slopes.
5. The controller counts the numbers of pulses. When the configured value is reached the switching event is produced.
6. The controller writes FALSE on the "Ready for Trigger" bit.
7. Switching event (see figure 9):
    a. All the channels are opened. (Signal relays are opened, GND relay are closed).
    b. The GND relays of the ON channels are turned off.
    c. An enable delay time is implemented to prevent a short circuit. The secure time is 2 times the turn-off time of the relay.
    d. The corresponding signal relays are configured based on the corresponding row of the sequence matrix.
    e. The switch remains in the actual state during the delay configured.
    f. A counter to indicate the following matrix row is incremented.
8. The controller returns to point 4 and writes TRUE on the "Ready for Trigger" bit.
9. If a disarm command is received the controller returns to configuration state.

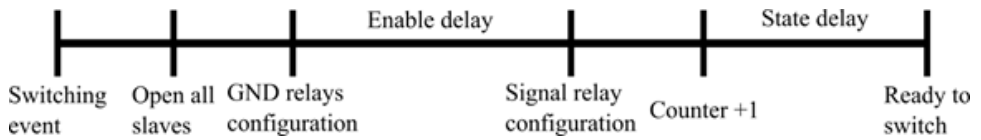


Figure 8: Time diagram of the operations after switching event, items 7a to 7f. The Enable delay must be larger than the maximum turn-off time of the relay.

Local operation

The multiplexer can be configured on local operation. The button “ENA CH” runs the sequence step by step, each time that the user pushes the button the multiplexer changes the state following the sequence matrix, which must be previously loaded, or the default sequence will be used.

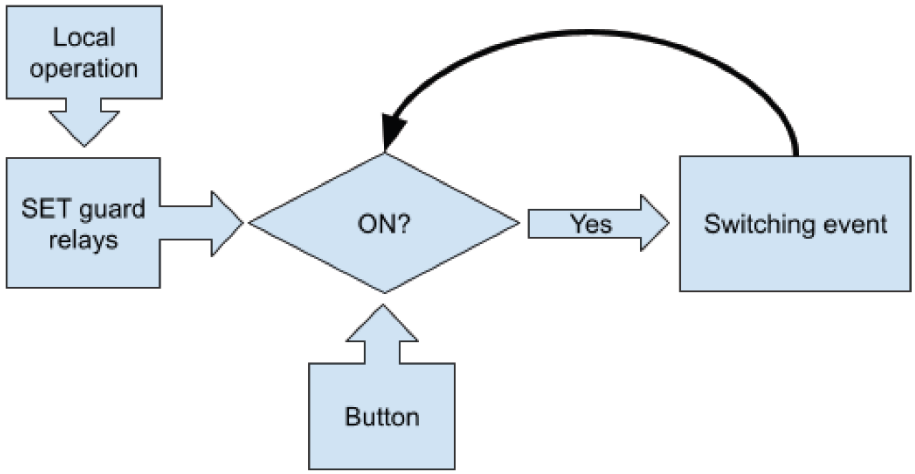


Figure 9: In local operation, the multiplexer changes between states with the ENA CH button.

Installation and connections

All the boards are designed to be fixed in a standard 19” case with a height of 3U (127 mm). The following figure shows one possible configuration with three SPDT slaves boards.

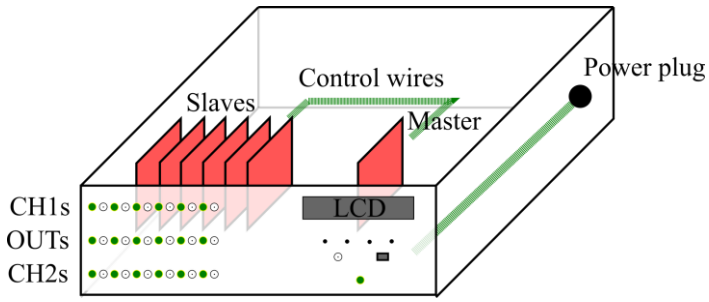


Figure 10: Recommended configuration of the case for SPDT slave boards.

Each slave can be connected to the front panel by means of coaxial wire for signals and unipolar wire for guard and ground. The guard voltage provides a shield ring around the inputs and the outputs, they help to reduce the stray capacitance between signals. The ground voltage is connected to the middle point of the T-configuration during turn-off state. In order to obtain high off-isolation, the ground terminal must be the source GND and it must be connected by means of a low impedance wire.

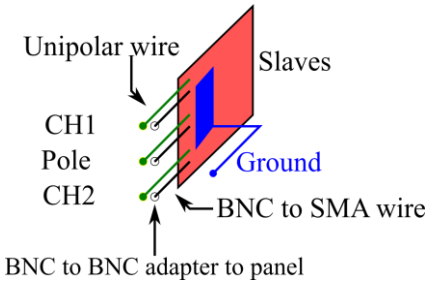


Figure 11: Connection diagram of the SPDT slave board.

The following figures show possible connections for the 2-to-1 board and the 4-to-1 board, and for different cases: 1, 2 and 3 multimeters. We also include the connection for the DC voltage measurement by differential method with the JVS. In all the cases, the diagram shows all the allowed slaves boards and the shadow BNCs are not used in that connection.

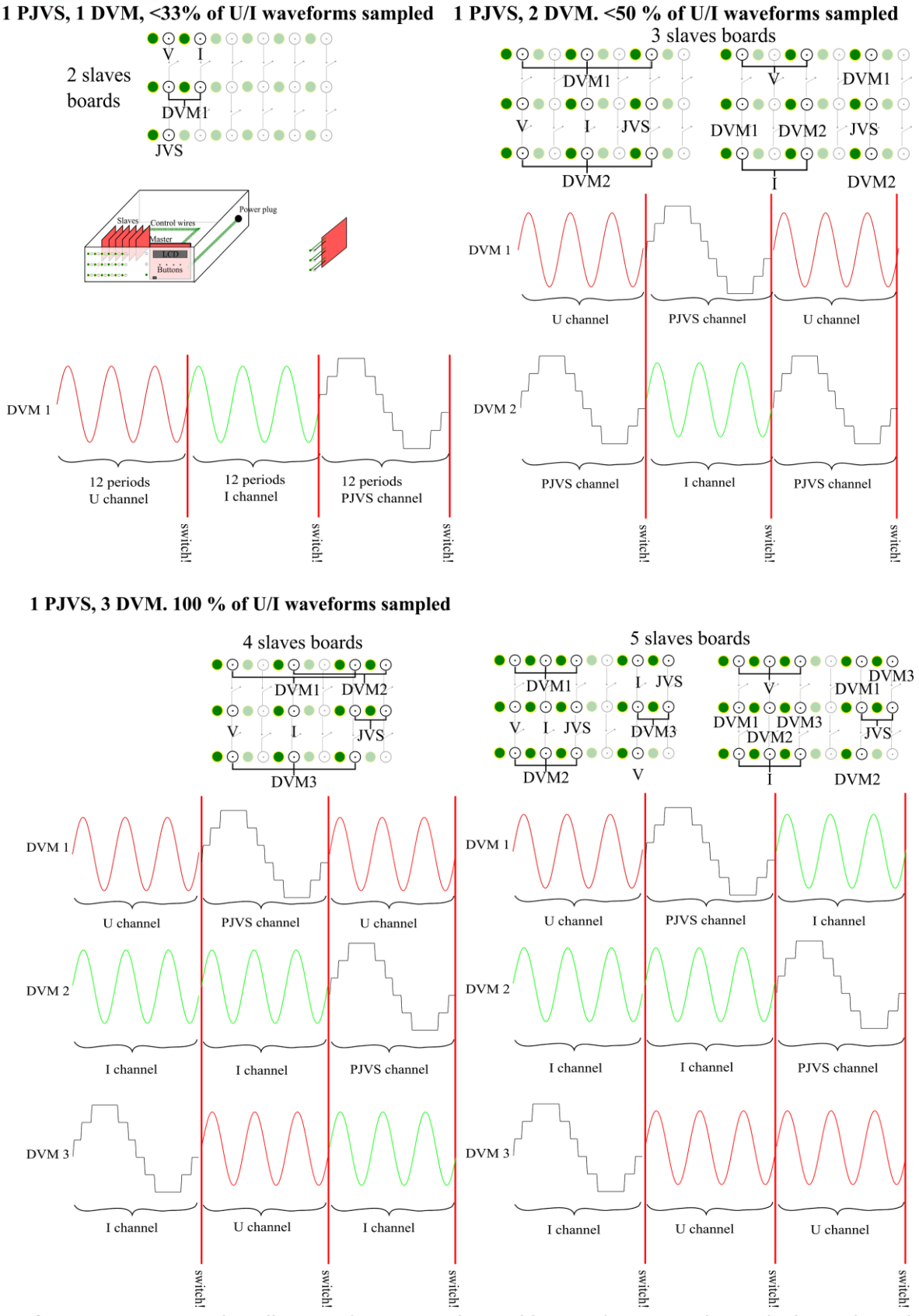


Figure 12: Connection diagram for 1, 2 and 3 multimeter for SPDT (2-to-1) slaves board.

1 PJVS, 1 DVM, differential DC measurement

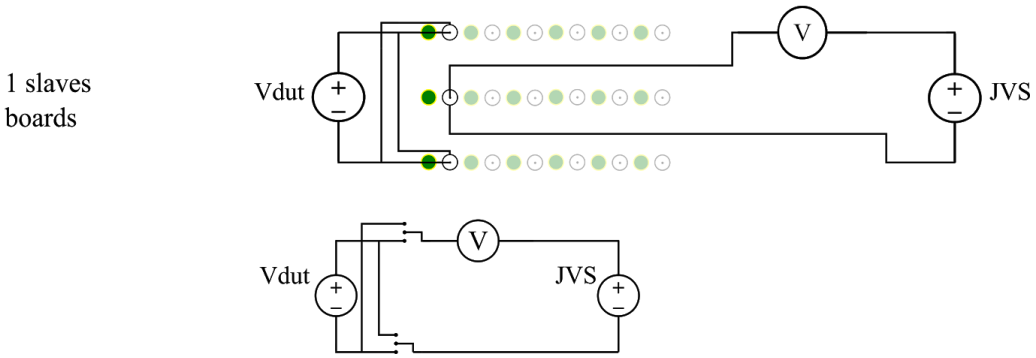


Figure 13: Connection diagram for DC measurement with SPDT slave board

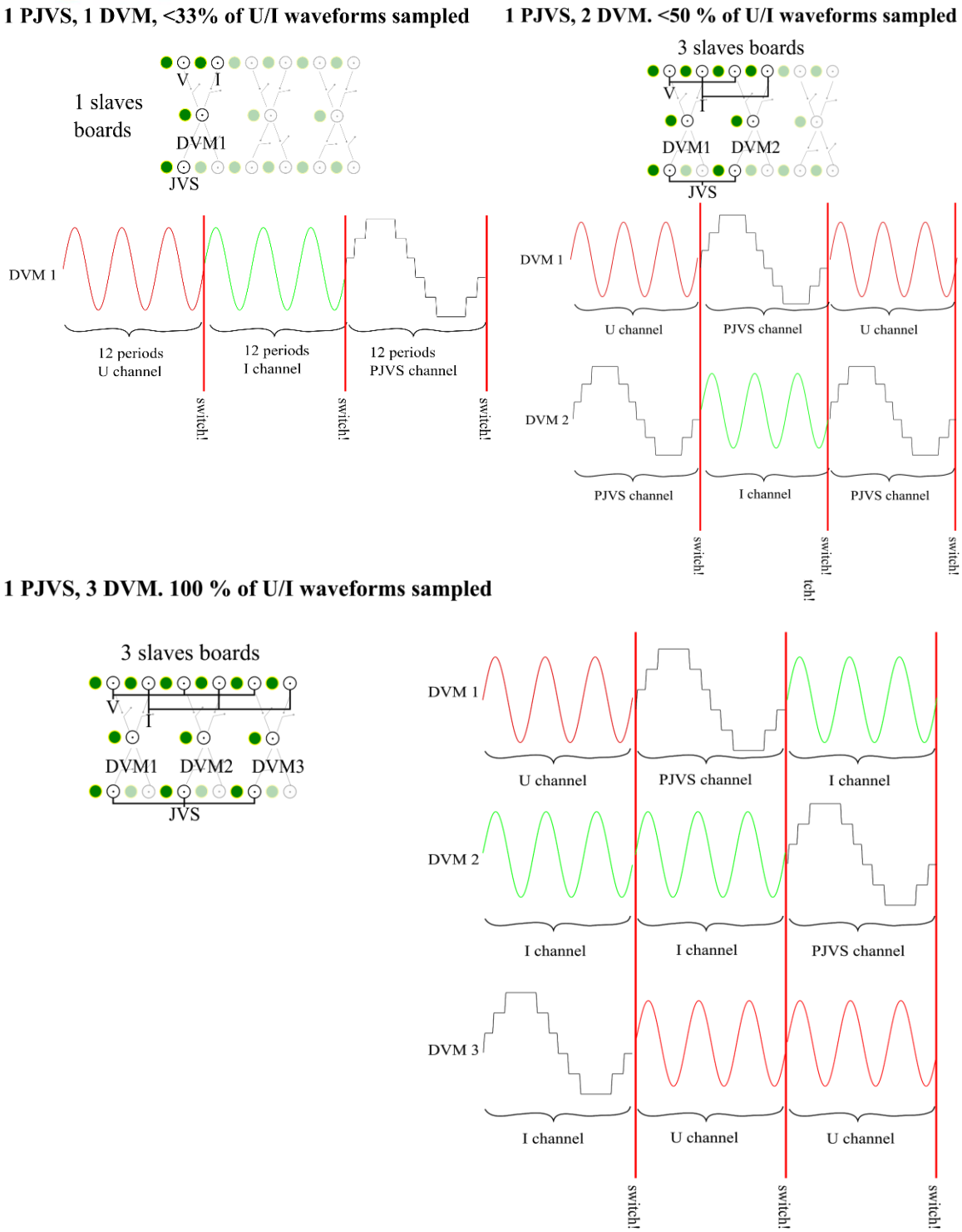


Figure 14: Connection diagram for 1, 2 and 3 multimeter for SPFT (4-to-1) slaves board.

LabVIEW driver

About

For implementation of the multiplexer into already developed software (TWM), a LabVIEW driver have been made. The driver can be accessed here:
https://github.com/KaeroDot/QPsw/tree/main/QuPMX/QuPMXLabVIEWdriver

Driver structure

Driver is made in typical Initialize-Use-Close method. VIs for configuration and setup have been created according commands implemented in the firmware of the multiplexer.
The VI Tree is shown in figure 16.

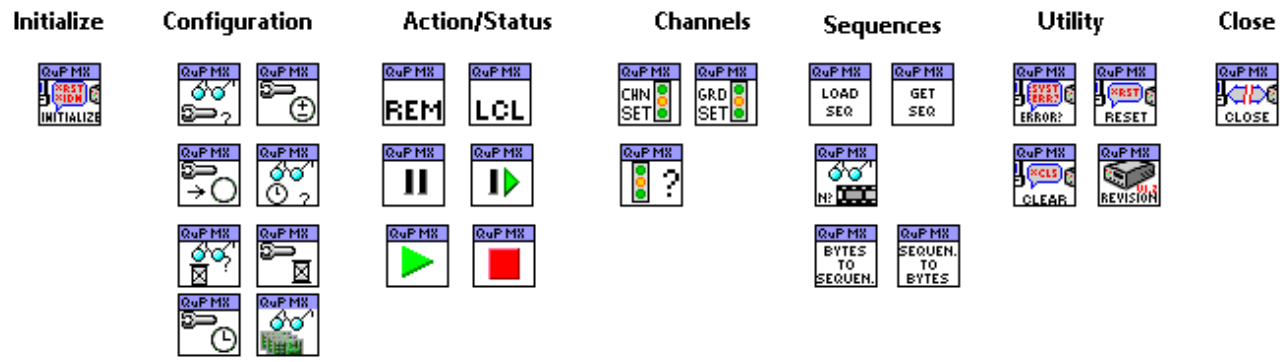


Figure 15: Overview of subprograms (VI) of the multiplexer driver.

Multiplexer Control software and Driver Example

As an example, for use of the driver, a multiplexer control software *QuP Multiplexer Control* have been developed. The control software gives ability to manually control the multiplexer and set and read properties. The graphical user interface is shown in figure 17.

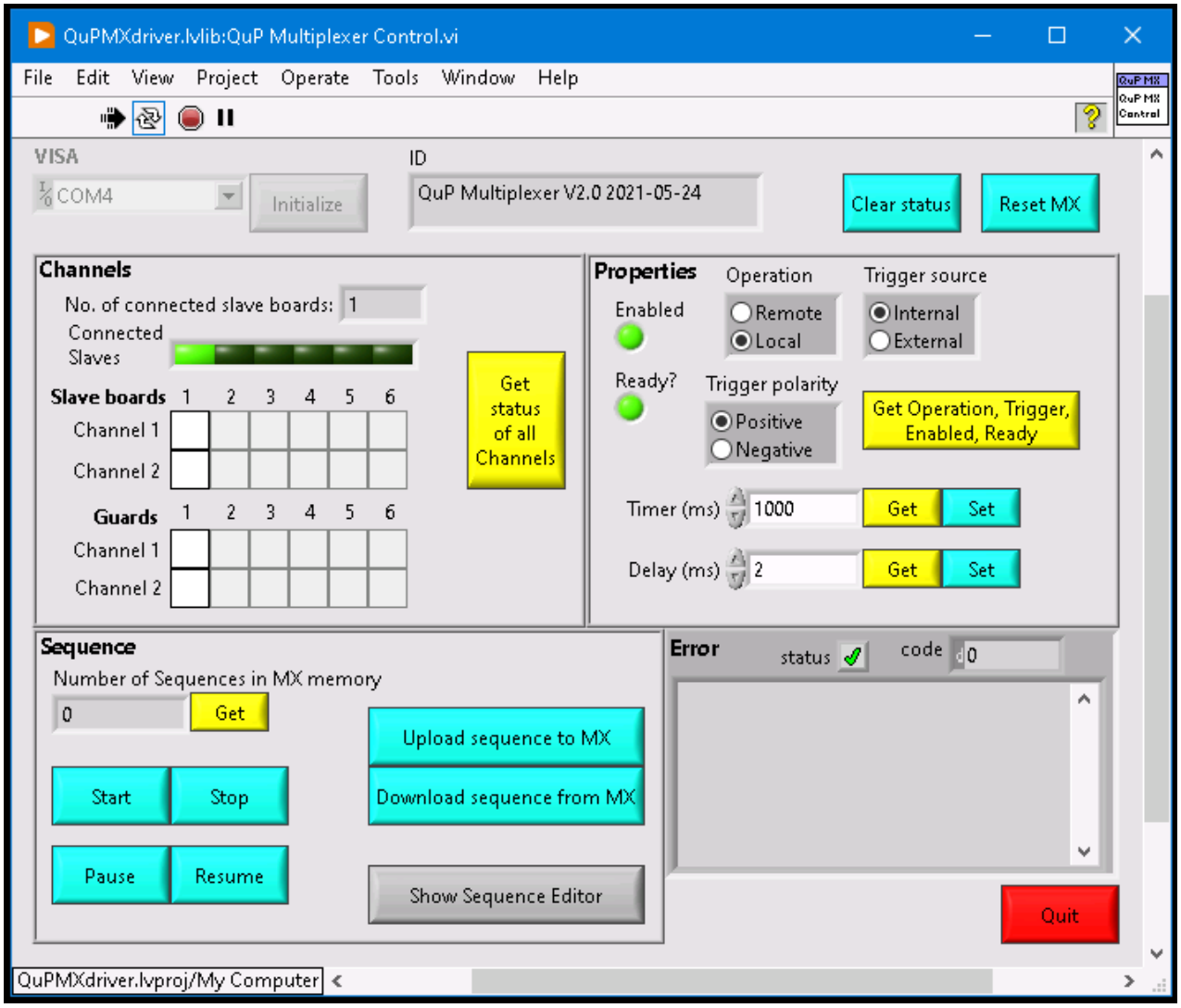


Figure 16: Graphical user interface of *QuP Multiplexer Control*.

The connection status of the slave boards can be changed by pressing appropriate buttons (white rectangles on the left side of graphical user interface).
The sequences that are to be loaded into multiplexer can be edited in the Sequence Editor (see figure 20) after pressing button *Show Sequence Editor*.

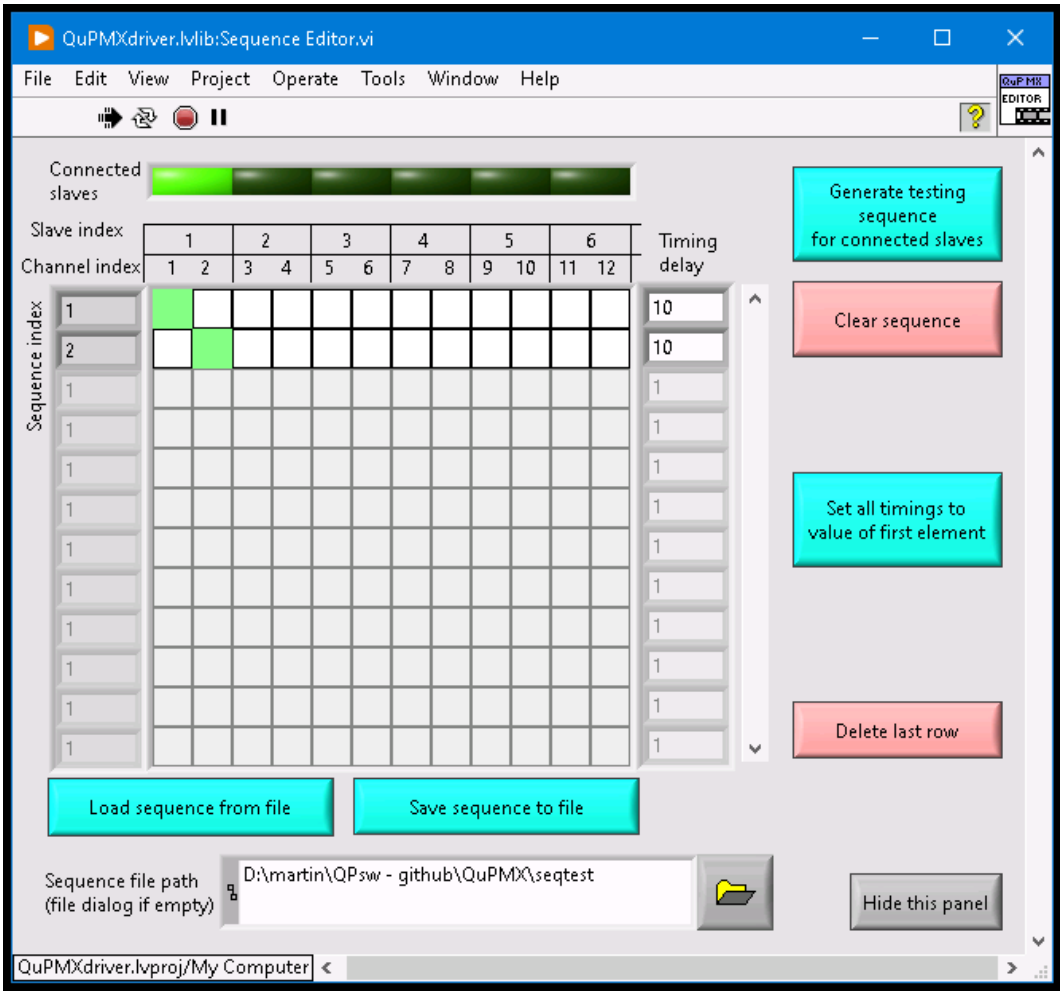


Figure 17: Graphical user interface of *QuP Multiplexer Control*, editor of sequences.

Any sequence can be manually created, loaded or saved from/to a file and uploaded or downloaded from/to multiplexer.

Compiled binaries

The latest version of the compiled binaries of the driver library and control software can be accessed here:
https://github.com/KaeroDot/QPsw/tree/main/QuPMX/QuPMXControl
To run the control software, a run time libraries for LabVIEW 2020 are required.

Measurements

Multiplexer amplitude error

The amplitude error of the multiplexer was measured using PJVS and Fluke calibrator 5720A. The PJVS measured amplitude of the calibrator, and amplitude of the calibrator signal with multiplexer connected in between. See figure 21 for measurement setup.

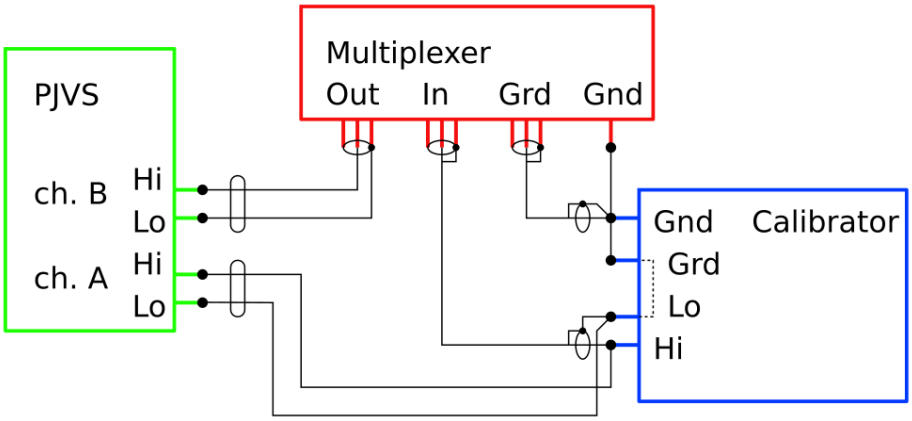


Figure 18: Measurement setup for multiplexer amplitude error estimation.

For every reading, the calibrator was set to 7 V at 1 kHz. First a calibrator voltage was measured by PJVS on channel A. Next the PJVS measured the signal on channel B. The error of the multiplexer was calculated as a difference of both measured values. This procedure was repeated.
The measured multiplexer error is (1.615 ± 2.383) µV for a 95 % level of confidence and 43 repetitions. That means the relative error is (0.231 ± 0.340) µV/V when referenced to the nominal source voltage of 7 V.
The figure 19 shows the measured values and figure 20 the Overlapped Allan deviation of the data. One can see the noise floor have not yet been reached and continued measurement would result in smaller uncertainty.

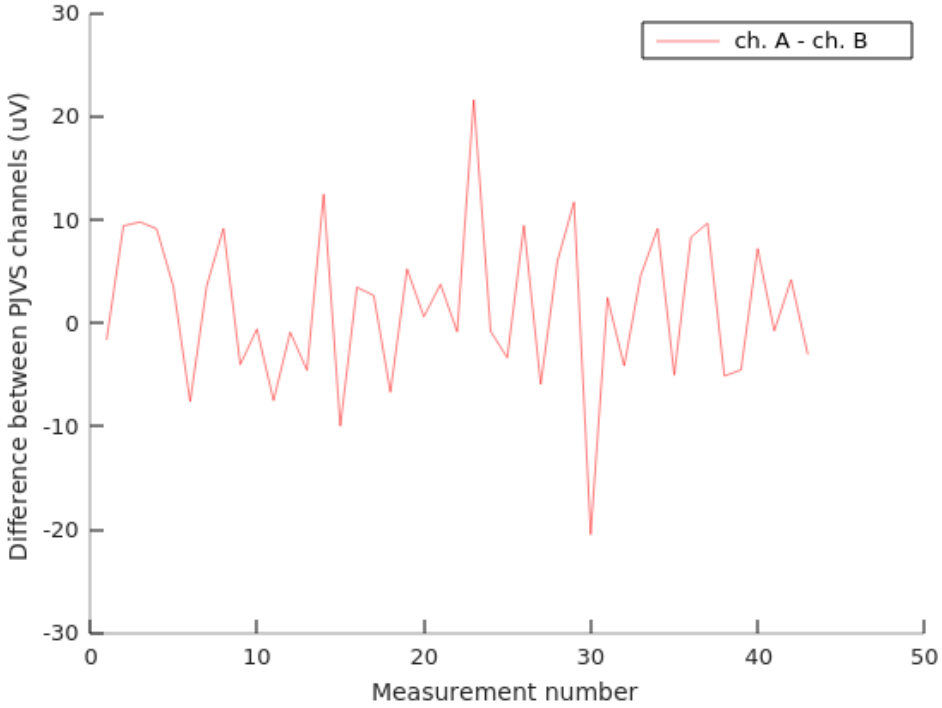


Figure 19: Difference between calibrator value measured directly and through multiplexer.

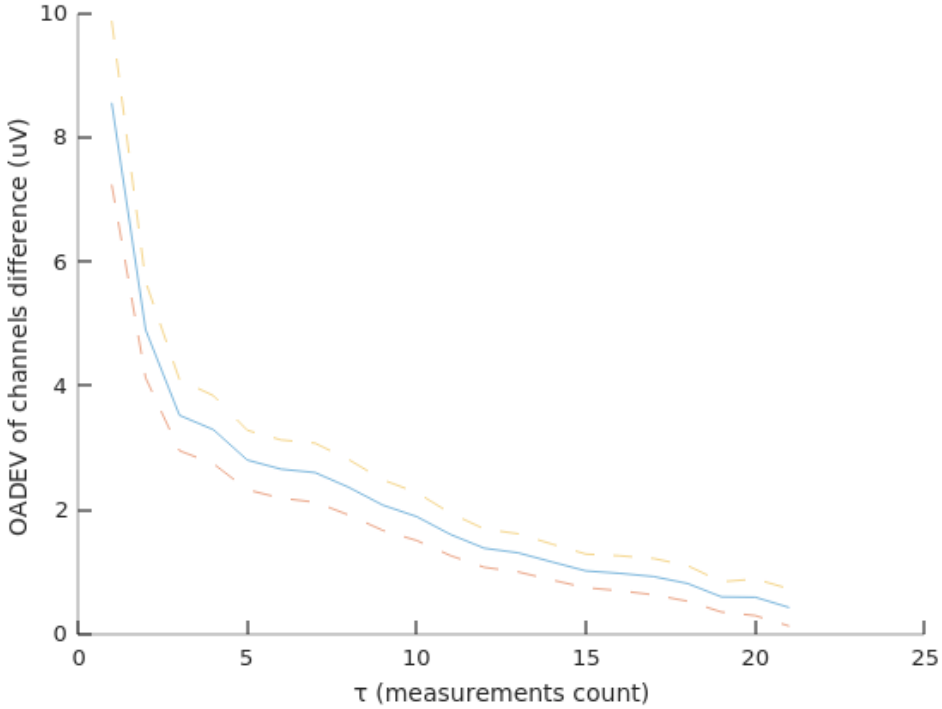


Figure 20: Overlapped Allan deviation of differences.

The uncertainty of the measurement is based only on type A uncertainty. Differential method of measurement was used in the PJVS. The type B uncertainties, such as PJVS digitizer errors should mostly cancel out by calculating difference between channels. The important type B uncertainty, that was not cancelled out, is the error of the inner multiplexer of the PJVS. This error was estimated by measuring calibrator output by both channels directly. The error was smaller than **0.39 μV**, unfortunately measured for insufficient number of repetitions.

The better method of measuring multiplexer offset would be measurement on single PJVS channel, with multiplexer switching during continuous sampling. This method will be possible after finishing whole measurement software to be developed in this project.

Phase delay

Setups

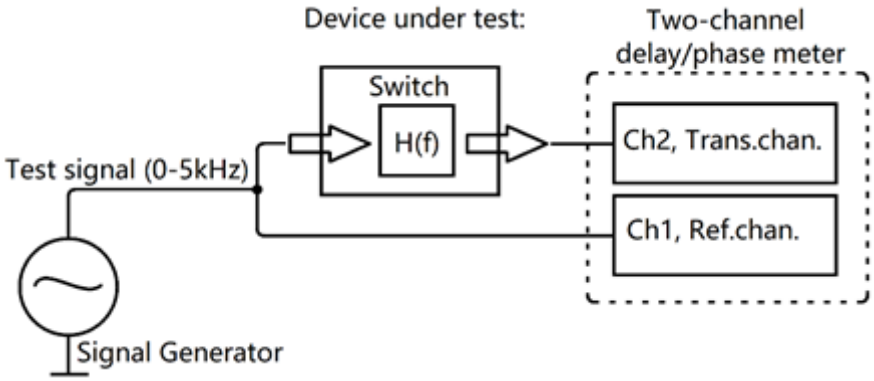


Figure 21: Setup for measuring phase delay through multiplexer.

- The delay/phase-meter consist of two 3458A, in a two-channel sampling setup.
- The device under test is the signal-path through the switch, Ch1-to-Out, and Ch2-to-Out separately.
- Signal-generator can either sweep a signal through 0-5 kHz, or generate a pink-noise, frequency-limited inside 0-5 kHz range.

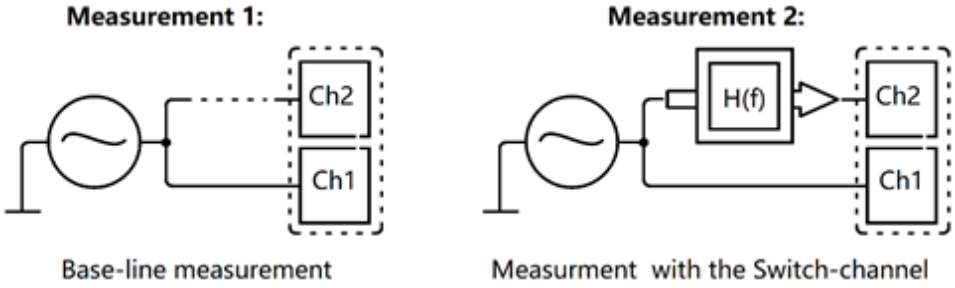


Figure 22: Base-line measurements to correct for the setup induces differences.

- For each channel of the switch, two measurements are done:
    1. A base-line measurement, to determine the contribution from the setup without the switch, and,
    2. Measurement when the signal go through the switch
- The delay/phase calculation is done by subtracting the base-line measurement from the switch-measurement.

Results

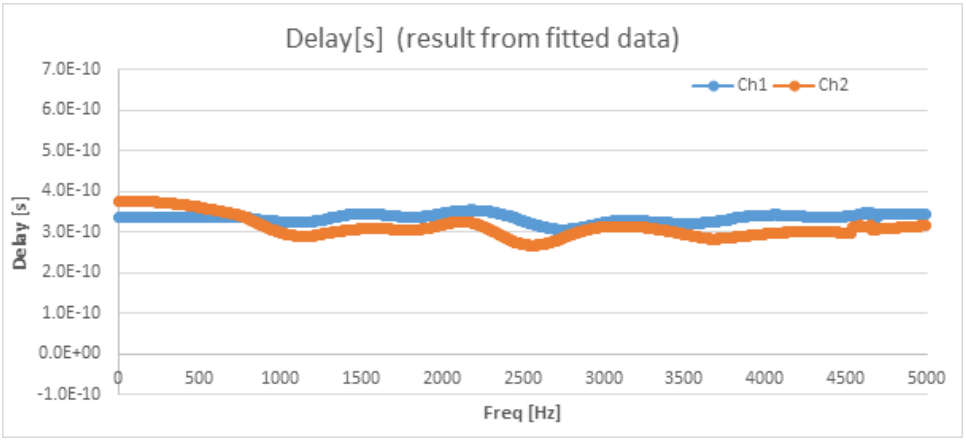


Figure 23: Delay measurements between the input and output of Ch1 and Ch2. The fitted data is an approximation to the local weighted average delay round the frequency.

The delay is flat over the tested frequency-range, within the uncertainty of the measurement. We see a little tendency that the delay of Ch1 is higher than Ch2, in the range of about 35 ps. The variations over the frequency-range is due to the noise. Better signal-to-noise can be achieved by longer measurements.

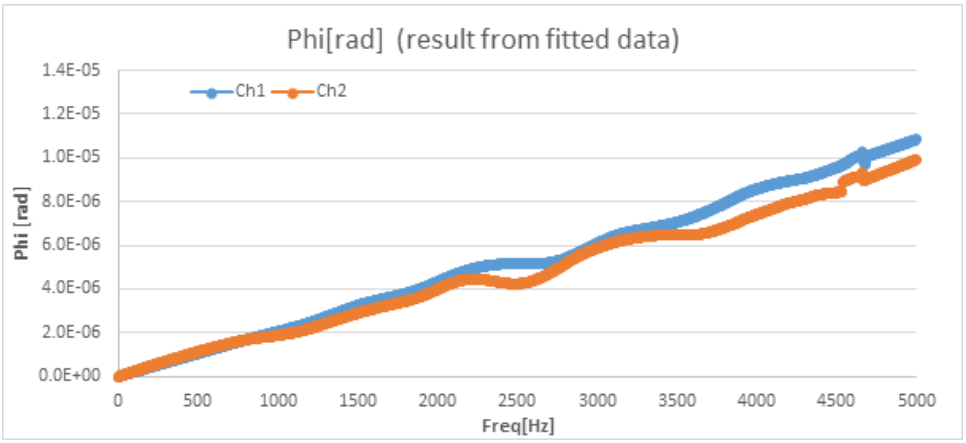


Figure 24: Phase difference between the input and output signal for Ch1 and Ch2. The fitted data is an approximation to the local weighted average delay round the frequency.

The phase is linear over the tested frequency-range, within the uncertainty of the measurement. We see a little tendency that the delay of Ch1 has a slightly higher phase-shift then Ch2. The variations over the frequency-range is due to the noise. Better signal-to-noise can be achieved by longer measurements.

The measurement show that the added signal path through the switch contributes a delay of roughly 320 ps, and we found a difference of about 35 ps between the two channels. The value of the delays found, is within expected range, and can be interpreted as good for frequency-range of DC to 5 kHz.

Stray Capacitance

The stray capacitance of a turned-off PhotoMos was measured for several devices with a HP 4263A LCR bridge. The following table shows the results of series resistance and capacitance. The measurement where perform under four conditions with DC voltage applied with value equal to 0, 1.5 and 2 V. The AC voltage amplitude was 1 V or 0.1 V.

Table 4: measurements results of the stray capacitance for several PhotoMos.

Lot		1 VAC - 0 VDC	0.1 VAC - 0 VDC	0.1 VAC - 1.5 VDC	0.1 VAC - 2 VDC
V94368	pF	138	202.36	123.96	114.47
	kΩ	2.3	1.093	1.6	2.9
V94368	pF	141.4	203.97	124.56	114.67
	kΩ	2.3	1	1.8	2.689
V03768	pF	141.9	209.29	125.98	116.63
	kΩ	10.7	7	2.85	2.9529
V03768	pF	142.22	205.39	125.25	115.69
	kΩ	10.315	8.02	3.89	4.2029
V03768	pF	140.59	204.51	125.10	115.84
	kΩ	11.02	8.13	3.7288	3.367
V03768	pF	133	194.30	122.08	113.40
	kΩ	11.1	8.61	4.028	4.514
V03768	pF	139.2	204.79	124.67	115.62
	kΩ	10.36	8.35	3.235	3.43

The stray capacitance introduces a leakage current with distortion due to the no-lineal characteristics of the measurements capacitance, the next figures shows the estimated model. This model was tested comparing the simulated leakage current with the measurements of the current leakage obtained with an oscilloscope and applying a 100 kHz signal to one device turned-off.

The current introduced by the PhotoMos turned-off can generate an error on the voltage applied on the digitizer by an activated channel. This error depends on the channel on-resistance and the source impedance, and simulations of the complete systems shows that the distortion is negligible for low impedance sources.

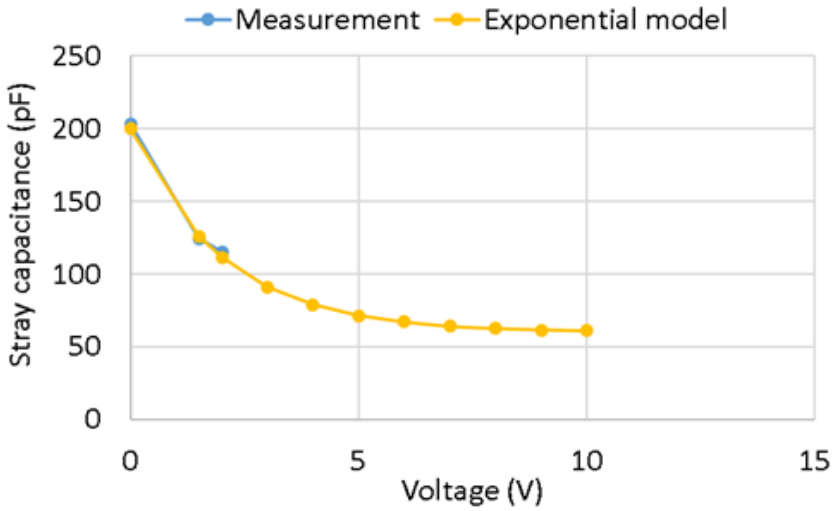


Figure 25: Comparison between measured and modeled stray-capacitance.

Off-isolation and crosstalk

The off-isolation was measured introducing a sinusoidal signal of 1 MHz and 1 V at an input channel turned-off. The output is connected to a digital oscilloscope with the FFT activated to measure the same frequency component. All the set-up is couple at 50 Ω. The crosstalk is measured in the same way but the input channel is activated and the oscilloscope is connected other input channel. The measurement shows a ratio (output-input voltage) lower than –90 dB for the off-isolation and crosstalk. The procedure was repeated for input sinusoidal signal of 1 kHz obtained values lower than –120 dB, limited by the oscilloscope noise floor.
The isolation resistance was also measure at DC applying a fixed voltage to the turned-off channel and measuring the leakage current with a picoammeter. For a voltage of 10 V the output current was lower than 0.1 pA, so the isolation resistance was larger than 100 TΩ.

On-resistance

The channel on-resistance was measured at the input terminal with a Kelvin connection and introducing a short circuit in the output. So, the series resistance of the high and low relay is evaluated, which includes four relay connected in series. The manufacturer specified a typically on-resistance of 0.18 Ω so the expected value was 0.72 Ω. However, the measured value was lower, 0.53 Ω.

Annex 1. Analog circuit description

Board connectors

Table 5: Slave board connectors.

Identification	Connector
Control and supply	Ribbon Cable. PMOD interface.
Way 1, Way 2, Pole	SMA on board
Guard	Unipolar terminal
Ground	Unipolar terminal

Board digital pin-out

A 12-pin double-row connector (PMOD type) is used to connect the controller board and the relay board.

Table 6: Slave board digital lines.

PIN	Name	Comment
1	CH02_GND	Channel 2 ground connection
2	CH02_ENA	Channel 2 enable
3	CH01_ENA	Channel 1 enable
4	CH01_GRD	Channel 1 guard connection
5	CH01_GND	Channel 1 ground connection
6	CH02_GRD	Channel 2 guard connection
7	BD	Digital line tied to GND to indicate that the board is connected.
8	No connected	No connected
9-10	GND	Digital ground
11-12	Vcc	Digital voltage supply

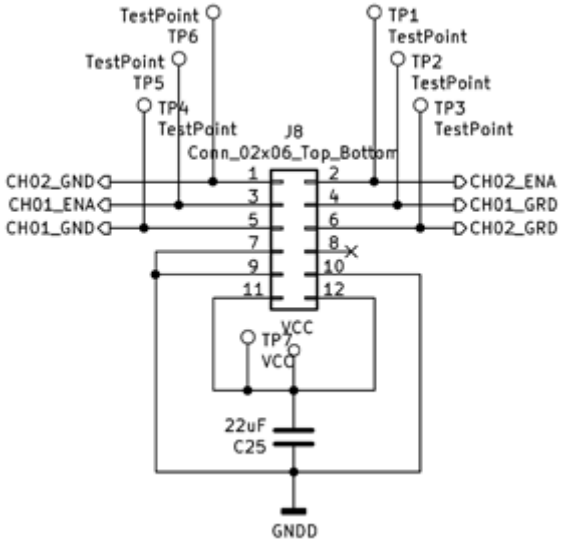


Figure 26: Schematic circuit of the digital lines connector.

Test points list

Table 7: Test points on the SPDT slave board.

Test points	Signal
0	GND Digital
1	CH02_ENA
2	CH01_GRD
3	CH02_GRD
4	CH02_GND
5	CH01_GND
6	CH01_ENA
7	VCC

Annex 2. Digital circuit description

Master board

The master is built with an Arduino board and a shield board, this last one includes the connectors for 12 switches, and a voltage regulator to supply the slave boards. This voltage regulator is connected to an external supply of 12 V using a standard power connector Jack and Plug 2.1 mm x 5 mm. The Arduino-based board and the display are supplied by the USB connector. The used display is a 16x4 liquid crystal display (LCD) with I2C Communication.

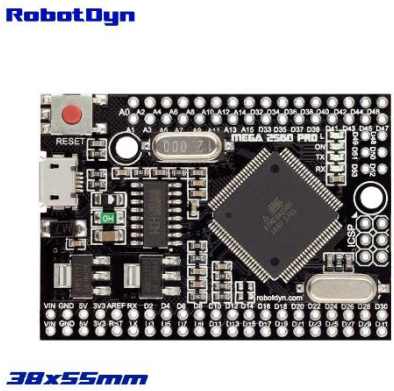


Figure 27: Arduino-based board manufactured by RobotDyn, https://robotdyn.com/mega-2560-pro-embed-ch340g-atmega2560-16au.html.

Table 8: Master board connectors.

Identification	Connector
Control and supply for 6 slave control	Ribbon Cable. PMOD type.
Supply	Screw terminal
Clock IN	Opto-isolated BNC
Communication to PC	USB
Buttons control	6 pinhead connector
Display control	4 pinhead connector

Power-on state

1) All channels OPEN
2) All GRD OPEN
3) Local operation
4) Trigger Internal
5) Internal timer 2 s
6) No valid sequence in memory

Serial configuration

Baud rate: 9600
Data Bits: 8 bits
Parity: None
Stop bit: 1 bit
Termination char: LF
Timeout: 10 s
Enable termination: FALSE
Flow control: none

Driver for CH340 USB-Serial converter

Users of Windows PC must install a driver in order to have connection with the microcontroller by serial COM. The driver is available on the server in the directory: Multiplexer design / QPM design 2020 / Firmware / CH340_WinDRV.
In addition, the driver can be obtained from the board manufactured webpage https://robotdyn.com/mega-2560-pro-embed-ch340g-atmega2560-16au.html.
Or from the chip manufacturer:
http://www.wch-ic.com/products/CH340.html
http://www.wch-ic.com/search?q=CH340&t=downloads

Push Buttons Description

The multiplexer has 4 push buttons:

Table 9: Master buttons.

Identification	Button
RST	Reset the multiplexer as the *RST command does.
LOC/REM	This has two functions. Pressing it for more than 5 seconds change the state from REM to LOCAL operation or viceversa. Pressing it for less than 5 seconds shows on the last line in the display the sequence in memory in the form: SEQ no.: byte 1 in hex | byte two in hex | byte three in hex. For example: 1: 1 4 A => which corresponds to a sequence number 1 as: byte 1 = 1, byte 2 = 4, byte 3 = 10 in decimal
ENA CH	Runs the sequence in memory step-by-step.
µC RESET	This button reset the microcontroller, restarting it from power-on.

Timing diagrams

t_{ENA_ON}: time to close a signal channel after an external trigger pulse
t_{DELAY} = delay time

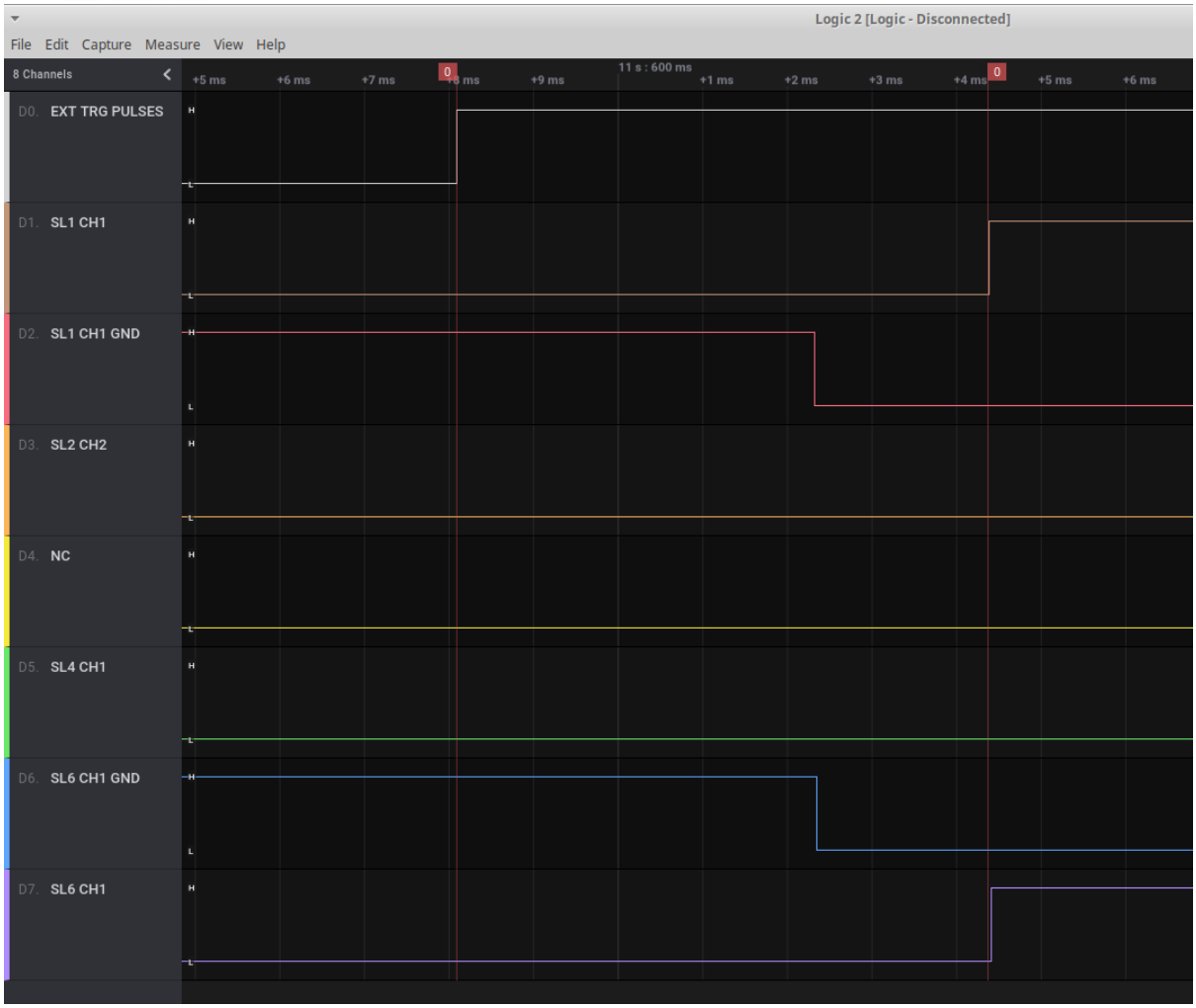


Figure 28: Turn-on timing.

To estimate t_{ENA_ON} time use the following formula: $T_{ena\text{-}on}$ = (3 * *DELAY* + 0.225) ms

DELAY is the current DELAY time setting

t_{ENA_OFF}: Time to wait until the channel is open

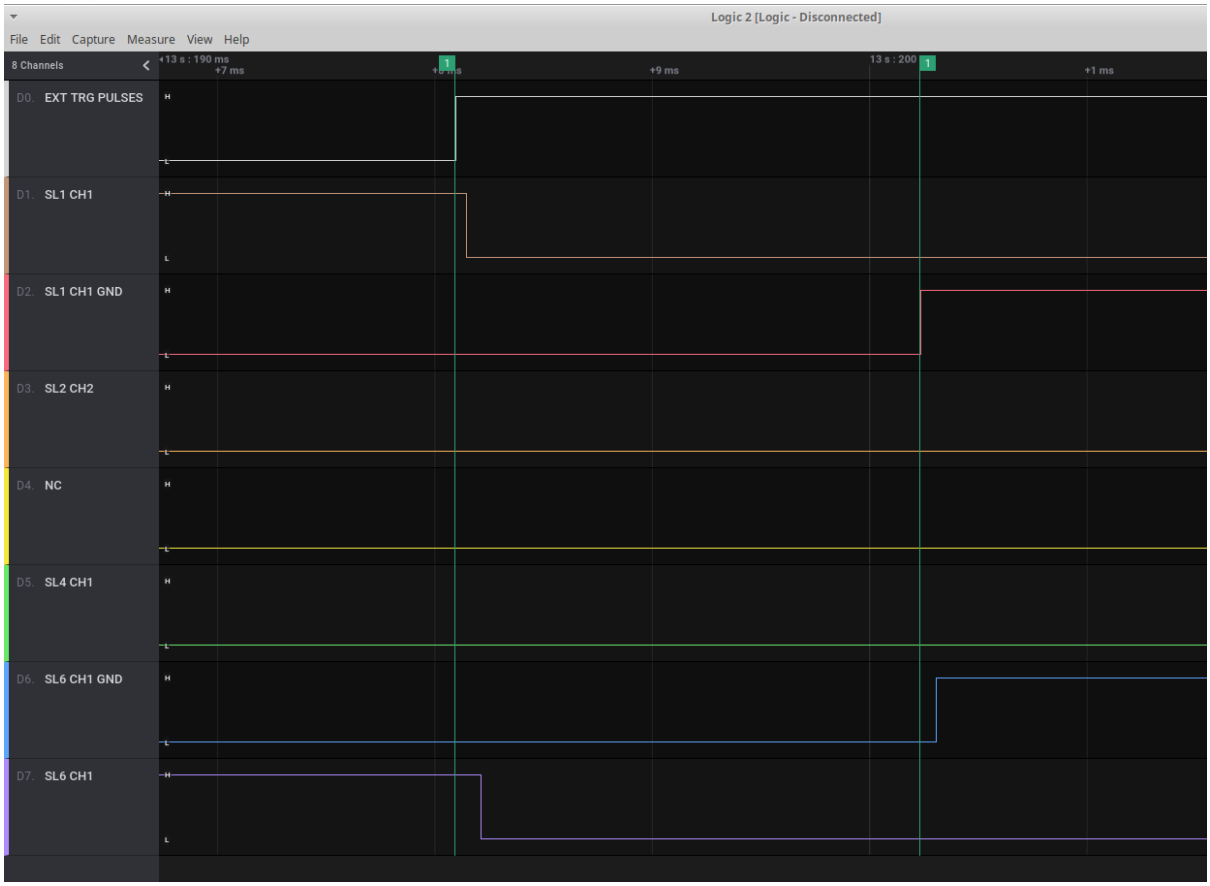


Figure 29: Turn-off timing.

To estimate t_{ENA_OFF} time use the following formula: $t_{ENA\text{-}OFF}$ = (*DELAY* + 0.125) ms

Important: ENABLE ON/OFF times also depends on the source and load impedance

Annex 3. How to upload the firmware

Steps to upload the firmware to the microcontroller

Method one

1) Download and install de Arduino IDE environment from its web page (https://www.arduino.cc/en/software)
2) Open it.
3) Install the following external libraries:
    a. LiquidCrystal_I2C
    b. DebounceEvent
    c. CircularBufferLib
    d. SerialCommands

    To do this step there are two ways install them using the IDE library manager or copy the libraries which are provided in the ZIP file to the IDE library directory
4) Open the project “QuP_Multiplexer_V01” file. To do it go to File → Open menu.
5) Select the Board type under the Tools menu as “Arduino Mega or Mega 2560”.
6) Check that the selected serial COM port is the current port where the board is connected.
7) Finally compile and upload the firmware using the icon → arrow in the tool bar.

Method two

There is another method to upload the firmware without using the Arduino IDE. To do so use the provided python script “upload_.hex.py” and the .hex file “QuP_Multiplexer_V01.ino.mega.hex”.

1) Download and install the Arduino IDE as in method 1, because the application avrdude is needed.
2) Change the avrdude path according to your installation in the script line 10 and 11
3) Set the correct COM port where the board is plugged. Parameter -P line 15
4) run the python script.

If everything was Ok the script should upload the firmware to the microcontroller.

Annex 4. Multiplexer command list

Date: May 2021
Firmware version 2.0
Authors: INTI

Table 10: Command list.

COMMAND	ARGUMENT #1	ARGUMENT #2	DESCRIPTION
ENA	SLx CHx	ON | OFF	Enables channel x
STAT	SLx CHx		Returns channel status
*IDN?			Returns program ID
*RST			Resets the multiplexer. All the slaves are opened (signal relays turned OFF and GND relays turned ON). Guard relays turned OFF. Local mode. Internal trigger. Status byte to default value.
*CLS			Clears the multiplexer. All the slaves are opened (signal relays turned OFF and GND relays turned ON). Errors bits on status byte are cleared.
GTL			Go To Local operation
REM			Remote operation. Disables the front panel push buttons
TRG	INT | EXT		Trigger internal or external
TRGPOL	POS | NEG		Sets TRIGGER polarity
TIMER	< ms >		Sets the internal TIMER
TIMER?			Returns the internal TIMER in ms
START			ARM the multiplexer. Thus, it starts running the programmed sequence for each TRIGGER event.
STOP			Stops running the MUX
PAUSE			Pauses running the MUX
RESUME			Resumes the MUX after pausing it from the last sequence

<table>
<tr><th>COMMAND</th><th>ARGUMENT #1</th><th>ARGUMENT #2</th><th>DESCRIPTION</th></tr>
<tr><td>DELAY</td><td><delay in ms></td><td></td><td>Caution: decreasing this value can damage the multiplexer or the sources. An enable delay time is required to prevent short circuit when the signal relays are opened and the ground relays are closed or viceversa. This time depends on the relays and load.</td></tr>
<tr><td>DELAY?</td><td></td><td></td><td>Returns the current enable delay.</td></tr>
<tr><td>*STB?</td><td></td><td></td><td>Reports for the status byte</td></tr>
<tr><td>GRD</td><td>SLx CHx</td><td></td><td>ENABLE/DISABLE Guard at SLx CHx</td></tr>
<tr><td>ADDSEQ</td><td>SLx CHx W <no. trigger></td><td></td><td>Adds a switching sequence</td></tr>
<tr><td>EDTSEQ</td><td><seq> SLx CHx W <no. trigger></td><td></td><td>Edits sequence number <seq></td></tr>
<tr><td>LDSEQ</td><td><no. of sequences to load></td><td></td><td>Loads a sequence matrix into memory from PC (binary format)</td></tr>
<tr><td>STSEQ(*)</td><td></td><td></td><td>Saves the current sequence into the EEPROM</td></tr>
<tr><td>RLSEQ(*)</td><td></td><td></td><td>Recalls the last stored sequence from EEPROM to internal memory</td></tr>
<tr><td>GTSEQ(*)</td><td></td><td></td><td>Reports the current sequence from Memory to PC (binary format)</td></tr>
<tr><td>DELSEQ</td><td></td><td></td><td>Deletes last entered sequence</td></tr>
<tr><td>SEQ?</td><td>< no. ></td><td></td><td>Returns the sequence no. configuration</td></tr>
<tr><td>NSEQ?</td><td></td><td></td><td>Returns the total number of sequences in memory</td></tr>
<tr><td>NSLAVES?</td><td></td><td></td><td>Returns the number of slaves connected</td></tr>
<tr><td>WSLAVES?</td><td></td><td></td><td>Returns the slaves position in a byte as
<table>
<tr><td></td><td></td><td>SL6</td><td>SL5</td><td>SL4</td><td>SL3</td><td>SL2</td><td>SL1</td></tr>
<tr><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr>
<tr><td>X</td><td>X</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr>
</table>
The Bx is set if the slave is connected at the corresponding position.</td></tr>
</table>

(*) NOT YET IMPLEMENTED!

Important: To start the multiplexer a valid sequence must be in memory. Then the trigger must be set to EXT.

Example 1: Status byte

To read the status byte use *STB?

Table 11: Status byte structure.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LASTERR			RDY	MX_ENA	TRG POL	TRG INT /EXT	REM /LOC
			/RDY=0 RDY=1	ENA=0 DISA=1	POS=0 NEG=1	INT=0 EXT=1	REM=0 LOC=1

Table 12: Status byte bit description.

BIT 0	Is set to 0 when the MUX in on REMOTE operation. Is set to 1 when the MUX is on LOCAL operation.
BIT 1	Is set to 0 when the trigger event is internal. Is set to 1 when the trigger event is external.
BIT 2	Is set to 0 when the trigger polarity is positive. Is set to 1 when the trigger polarity is negative.
BIT 3	Not yet implemented MX_ENA is always equal to 0, the multiplexer is on Enable state.
BIT 4	RDY, is set to 1 (RDY) when the MUX is on IDLE. Otherwise, is set to 0 (/RDY), indicating that a sequence is running.
BIT 5 **BIT 6** **BIT 7**	See error codes table below

Table 13: LASTERR: error codes.

Code number	Description	Condition
0 (000)	No error	There is no error
1 (001)	Command error	Indicates a wrong command
2 (010)	Sequence error	Indicates 0 sequence in memory
3 (011)	Sequence error	Indicates that the maximum number of sequences was reached
4 (110)	ENA command error	Indicates that an error in ENA command

5 (101)	SLV error	Indicates an error in ENA command when setting a SLAVE
6 (110)	GRD error	Indicates an error in GRD command
7 (111)	CH error	Indicates a channel error in the commands.

Example 2: to load a sequence matrix
Use the command LDSEQ <number of states to be load>
The sequence will be loaded using three bytes for each state:

BYTE 1:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
SL4-CH2	SL4-CH1	SL3-CH2	SL3-CH1	SL2-CH2	SL2-CH1	SL1-CH2	SL1-CH1

BYTE 2:

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
X	X	X	X	SL6-CH2	SL6-CH1	SL5-CH2	SL5-CH1

BYTE 3: a number between 1 to 255 indicating the number of trigger pulses to remain on the current sequence state.

Following you can see an example text file which can be uploaded using the VI or python script. The file can be generated by any text editor or by the VI module. The fields should be separated by TABs. Note that the byte 1 and 2 represents the states of the channels in binary format.

```
CH Byte 1	CH Byte 2	Delay Byte 3
1	4	10
10	0	10
4	0	1
8	0	1
16	0	1
32	0	1
64	0	1
128	0	1
0	1	1
0	2	2
0	4	2
0	8	2
0	4	2
0	2	4
0	1	5
1	0	4
64	0	1
10	0	1
```

Figure 30: Example of text file of the sequence matrix to be up load.

Examples 3: configuring the scanner without sequence matrix

ADDSEQ SL1 CH2 SL2 CH1 SL3 CH1 SL4 CH2 SL5 CH1 SL6 CH2 W 3→ Add a sequence to the multiplexer memory. The sequence is: close channel 2 (slave 1), channel 2 (slave 2), channel 1 (slave 3), channel 2 (slave 4), channel 1 (slave 5) and channel 2 (slave 6) and wait 3 trigger pulses to change to the next sequence.

ADDSEQ SL1 CH1 SL2 CH2 SL4 CH1 W 1 → Sequence: close channel 1 (slave 1), channel 2 (slave 2) and channel 1 (slave 4), then wait 1 pulse to the next sequence.

Examples 4: useful commands

To enable a channel:

ENA SL1 CH1 ON → close channel #1 of Slave #1
ENA SL2 CH2 OFF → open channel #2 of Slave #1

To enable the guard relay:

GRD SL1 CH2 ON → close the GRD of Slave 1 channel 2
GRD SL3 CH1 OFF → open GRD of slave 3 channel 1

To set the internal timer:

TIMER 160 → set the internal TIMER to 160 ms (the internal timer is for testing proposes)
To set internal or external trigger
TRG INT → Set the trigger mode to internal triggered
TRG EXT → Set the trigger to external

To start the multiplexer:

START → start running the MUX according to the sequence in memory.

To ask for the number of slaves:

NSLAVES? → Return msg: “TOTAL SLAVES: 2”
WSLAVES? → Return msg: XX000101

Annex 5. Switching Firmware

This annex explains the algorithm used to switch between the multiplexer channels. First, the switching pseudocode and firmware are introduced from an overall point of view. Then, each step of the algorithm is deeply studied.

Switching pseudocode and firmware

1. The PC configures the multiplexer and loads the switching matrix.
2. The PC arm the multiplexer.
3. The switches that connect the source guard are activated and they are not modified during the sequence.

4. The controller detects a trigger signal; the user can select positive or negative slopes.

5. The controller counts the numbers of pulses. When the configured value has reached, the switching event is produced.
6. The controller writes FALSE on the “Ready for Trigger” bit.
7. Switching event (see figure):

```
void runSlaveSequence(void){
uint8_t nSlv = slavesList.Count();
if(TRGCounter == trgWait){

  open_all_slaves(nSlv);

  sequence.openGND(islaveSeq);


  delay(MUX.getdly());


  sequence.closeCH(islaveSeq);


  trgWait = sequence.SeqBuffer[islaveSeq].byteThree;

  islaveSeq++;

  if (islaveSeq == sequence.getSeqSize())
        islaveSeq = 0;

 TRGCounter = 0; }
return;}
```

This condition defines the switching event

a. The signal switches are opened, and the ground switches are

b. The GND switches of the ON channels are turned

c. An enable delay time is implemented to prevent a short circuit. The secure time is 2 by the turn-off time of the relay.

d. The corresponding signal relays are configured based on the corresponding row of the sequence matrix.

e. The switch remains in the actual state during the delay configured.

f. A counter to indicate the following matrix row is incremented.

8. The controller returns to point 4 and writes TRUE on the “Ready for Trigger” bit.
9. If a disarm command is received the controller returns to the configuration state.

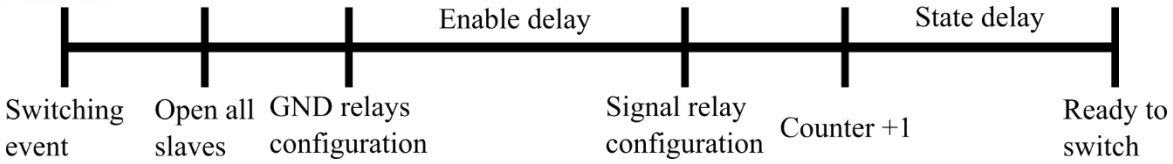


Figure 31: Time diagram of the operations after switching event, items 7 a to 7 f. The Enable delay must be larger than the maximum turn-off time of the relay.

Point 7.a

This function opens all the signal relays and closes the GND relay.
The connected slaves are saved on the list *slavesList[j]*.

```
void open_all_slaves(byte nSlaves)
{
int j=0;
//First OPEN all Signal Relays of all present slaves
for(j=0; j < nSlaves; j++){
   pSlave =  slavesList[j];
   pSlave->CH_ENA(1, OFF);
   pSlave->CH_ENA(2, OFF);
}
```

```
//OPEN/CLOSE the corresponding signal relays
void Slave::CH_ENA(uint8_t iCH, state_t state)
{
uint8_t iPins;
switch (iCH){
   case 1:
     iPins = CH1iPin;
     iState[1] = state;  break;
   case 2:
     iPins = CH2iPin;
     iState[2] = state;  break;
 }
digitalWrite(iPins, state);
return;}
```

```
delay(MUX.getdly());
```

An enable delay time is implemented to prevent a short circuit.

```
//The after the settle waiting time CLOSE ALL GND
for(j=0; j < nSlaves; j++){
  pSlave =  slavesList[j];
  pSlave->CH_GND(1, ON);
  pSlave->CH_GND(2, ON);
}
return;
}
```

```
//OPEN/CLOSE the corresponding GND relay
void Slave::CH_GND(uint8_t iCH, state_t state){
uint8_t gPins;
switch (iCH){
   case 1:
   gPins = CH1gPin; break;
   case 2:
   gPins = CH2gPin; break;
}
digitalWrite(gPins, state);
return;  }
```

Point 7.b

This function opens the GND relays for the channels to be activated.

```
bool Sequence::openGND(uint8_t SeqNo)
{
 Slave *pSlave=NULL;
 uint16_t Activate=0x0000;
 byte Act=0x00;

 //Set Activate variable for all slaves
 Activate = SeqBuffer[SeqNo].byteTwo;
 Activate <<= 8;
 Activate |= SeqBuffer[SeqNo].byteOne;

 //Here I have to parse and config the sequence to be run by the controller
 for(int k=0; k < slavecnt; k++){
  pSlave = (*pslavesList)[k];
  switch(pSlave->ID){
   case 01:
    Act = Activate & MASK_01; break;
   case 02:
    Act = (Activate>>2) & MASK_01; break;
   case 03:
    Act = (Activate>>4) & MASK_01; break;
   case 04:
    Act = (Activate>>6) & MASK_01; break;
   case 05:
    Act = (Activate>>8) & MASK_01; break;
   case 06:
    Act = (Activate>>10) & MASK_01;  break;
   default:
    Serial.println("SEQ ERROR");
    return false; }

  switch (Act){
   case 0b01:
    //CH01
    pSlave->CH_GND(1,OFF); break;
   case 0b10:
    //CH02
    pSlave->CH_GND(2, OFF); break;
   case 0b11:
    //BOTH
    pSlave->CH_GND(1, OFF);
    pSlave->CH_GND(2, OFF); break;
   default:
    break; //NOT Valid }
 }
 return true;}
```

The first and second bytes are included in the same

The corresponding bits are obtained.

If the channel bits indicate that a channel is close, the corresponding GND relay is opened.

If the bits are 00, the GND relays remain close.

Point 7.d

This function closes the signal relays for the activated channels.

```
bool Sequence::closeCH(uint8_t SeqNo)
{
 Slave *pSlave=NULL;
 uint16_t Activate=0x0000;
 byte Act=0x00;

 //Set Activate variable for all slaves
 Activate = SeqBuffer[SeqNo].byteTwo;
 Activate <<= 8;
 Activate |= SeqBuffer[SeqNo].byteOne;

 //Here I have to parse and config the sequence to be run by the controller
 for(int k=0; k < slavecnt; k++){
  pSlave = (*pslavesList)[k];
  //Serial.println(pSlave->ID);
  switch(pSlave->ID){
   case 01:
    Act = Activate & MASK_01; break;
   case 02:
    Act = (Activate>>2) & MASK_01; break;
   case 03:
    Act = (Activate>>4) & MASK_01; break;
   case 04:
    Act = (Activate>>6) & MASK_01; break;
   case 05:
    Act = (Activate>>8) & MASK_01; break;
   case 06:
    Act = (Activate>>10) & MASK_01; break;
   default:
    Serial.println("SEQ ERROR");
    return false;   }

  switch (Act){
   case 0b01:
    //CH01
    pSlave->CH_ENA(1, ON); break;
   case 0b10:
    //CH02
    pSlave->CH_ENA(2, ON); break;
   case 0b11:
    //BOTH
    pSlave->CH_ENA(1, ON);
    pSlave->CH_ENA(2, ON); break;
   default:
    break; //NOT Valid   }
 }
 return true;}
```

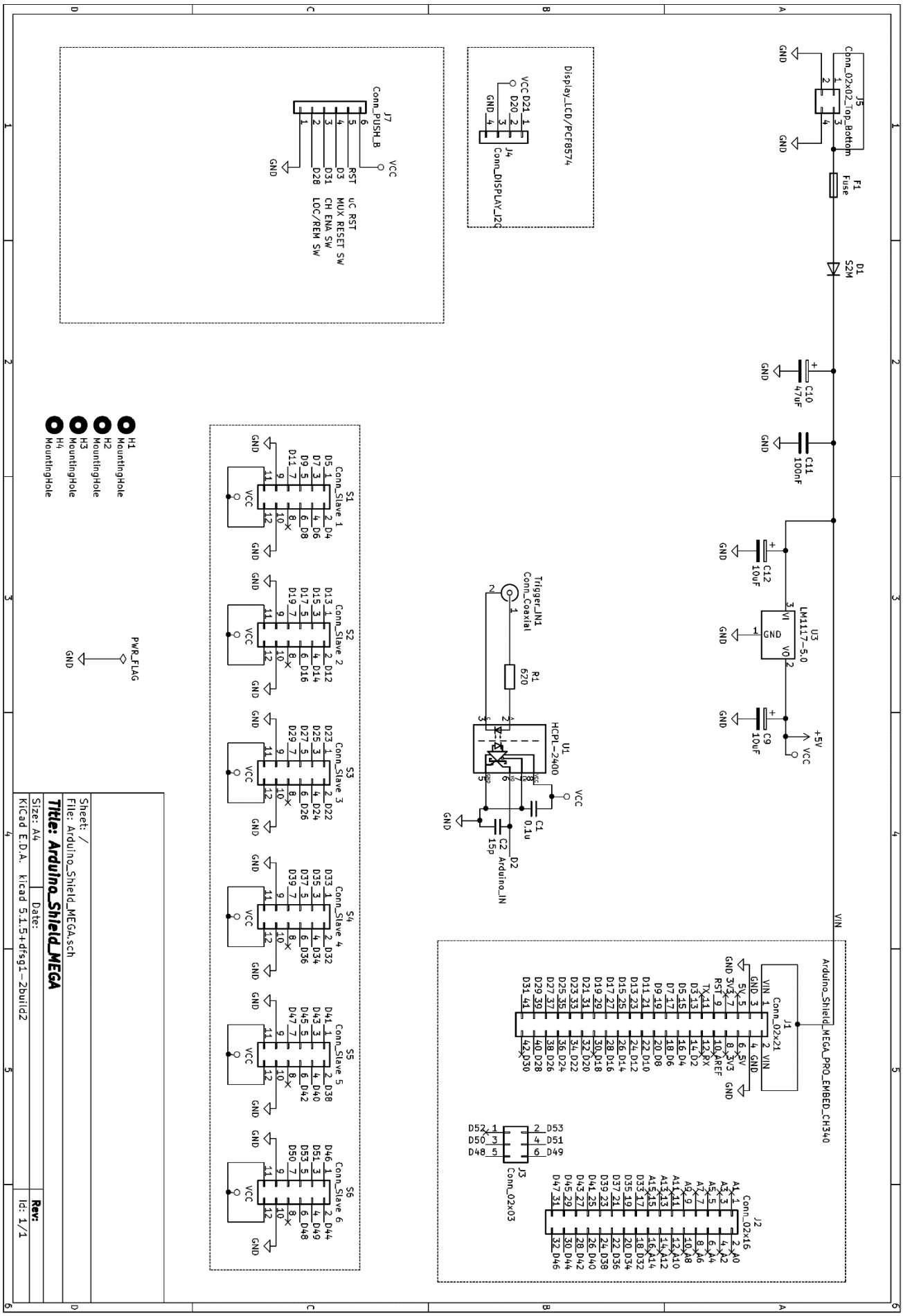
The first and second bytes are included in the same

The corresponding bits are

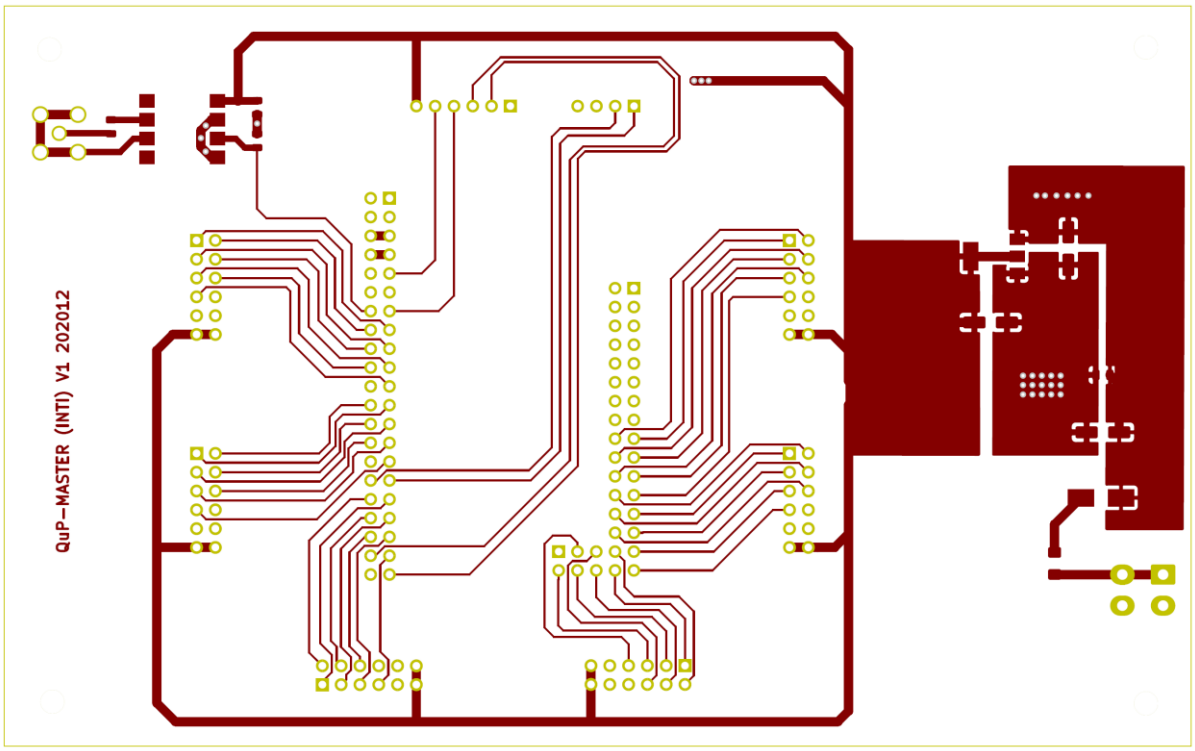
If the channel bits indicate that a channel is close, the corresponding SIGNAL relays are closed.

Annex 6. Master board

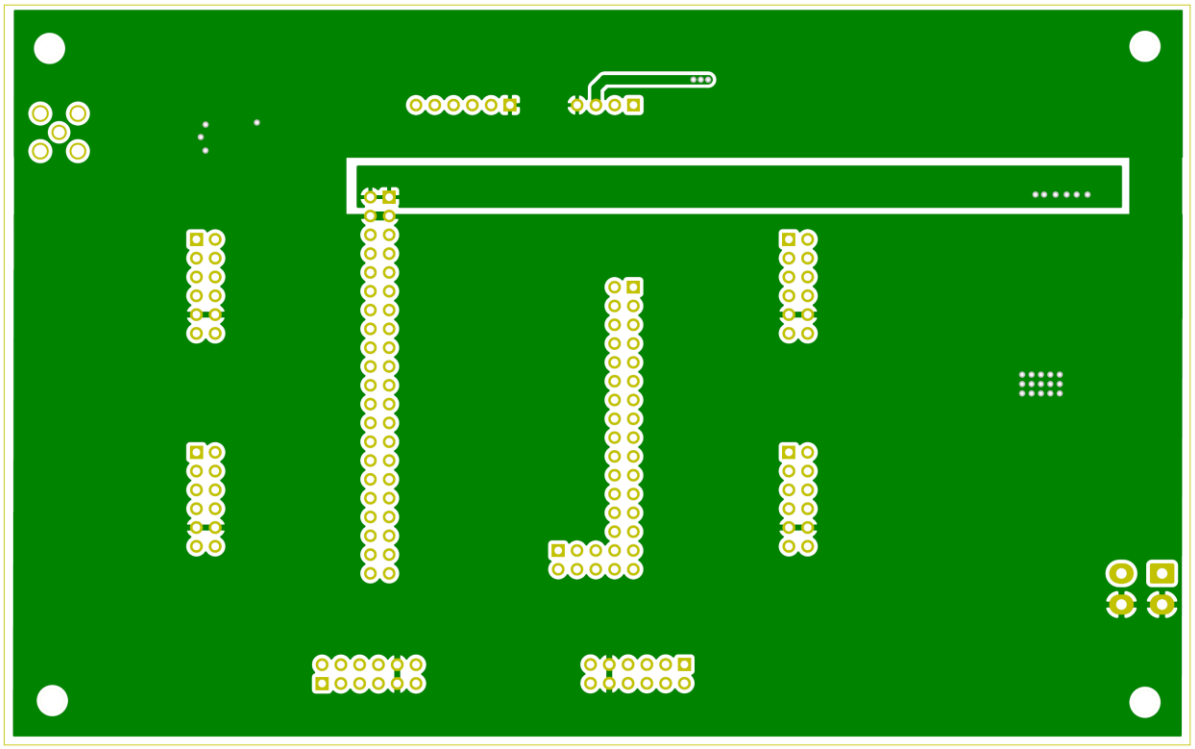
Item	Package	Designation	Vendor
F1	Fuse_Littelfuse-NANO2-451_453	Fuse	Littelfuse Inc. - 015401.5DR
J5	Molex_Mini-Fit_Jr_5566-04A_2x02_P4.20mm_Vertical	Conn_02x02_Top_Bottom	Molex - 39-28-1043
D1	D_SMB_Handsoldering	S2M	ON Semiconductor – S2M
J3	PinSocket_2x03_P2.54mm_Vertical	Conn_02x03	Sullins Connector Solutions – PPTC032LFBN-RC
J7	PinSocket_1x06_P2.54mm_Vertical	Conn_PUSH_B	Würth Elektronik – 61300611121
J7a option R/A	PinSocket_1x06_P2.54mm_Right Angle		Würth Elektronik – 61300611021
Trigger_IN1	SMA_Amphenol_132134_Vertical	Conn_Coaxial	Amphenol RF – 132134
H4,H3,H2,H1	MountingHole_3.2mm_M3	MountingHole	----
U3	SOT-223-3_TabPin2	LM1117-5.0	Texas Instruments – LM1117MP-5.0/NOPB
U1	SMDIP-8_W9.53mm	HCPL-2400	Broadcom - HCPL-2400-300E
S1, S2, S3, S4, S5, S6	IDC-Header_2x06_P2.54mm_Vertical	Conn_Slaves	MOLEX - 90130-1112
R1	R_0805_2012Metric	620 ohm	TE Connectivity – CPF0805B620RE
C12,C9	CP_Elec_5x3	10uF	Panasonic - EEEHB1V100R
C1, C11	C_0805_2012Metric	100nF	KEMET – C0805C104K4RACTU
C10	CP_Elec_5x3	47uF	Panasonic – EEEFT1V470AR
C2	C_0805_2012Metric	15p	KEMET – C0805C150J5GACTU
J1	PinSocket_2x21_P2.54mm_Vertical	Conn_02x21=42 in two parts	
J1A		conn 16pos 2 rows	Sullins Connector Solutions – PPTC082LFBN-RC
J1B		conn 32pos 2 rows	Sullins Connector Solutions - PPTC162LFBN-RC
J2	PinSocket_2x16_P2.54mm_Vertical	Conn_02x16	Sullins Connector Solutions – PPTC162LFBN-RC
J4	PinHeader_1x04_P2.54mm_Vertical	Conn_DISPLAY_I2C	Würth Elektronik – 61300411121
J4 Option R/A	PinHeader_1x04_P2.54mm_Right Angle		Würth Elektronik – 61300411021


Arduino_Shield_MEGA_PRO_EMBED_CH340
Display_LCD/PCF8574
Sheet: /
File: Arduino_Shield_MEGA.sch
Title: Arduino_Shield_MEGA
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2
Id: 1/1

Top:

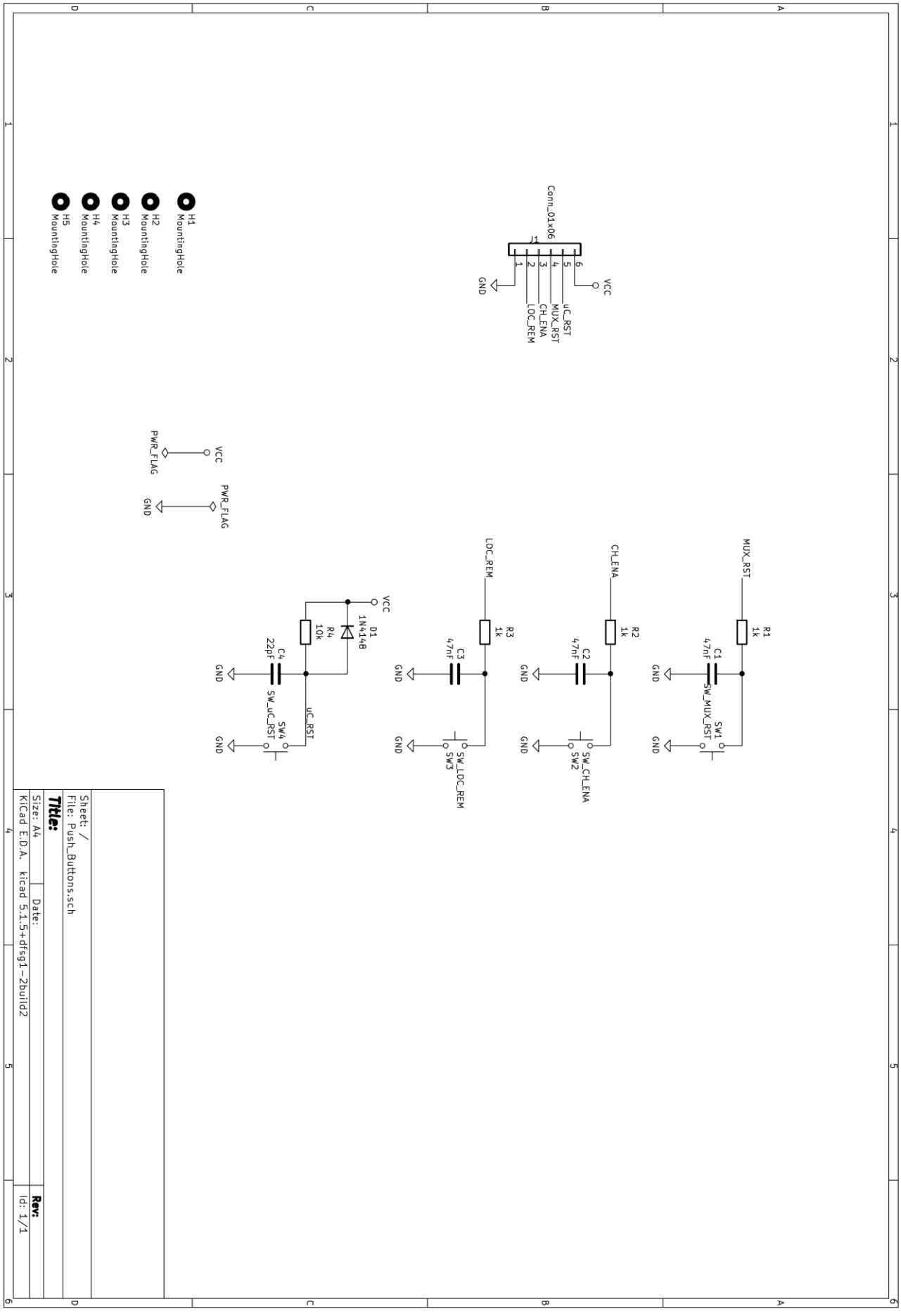


Bottom:

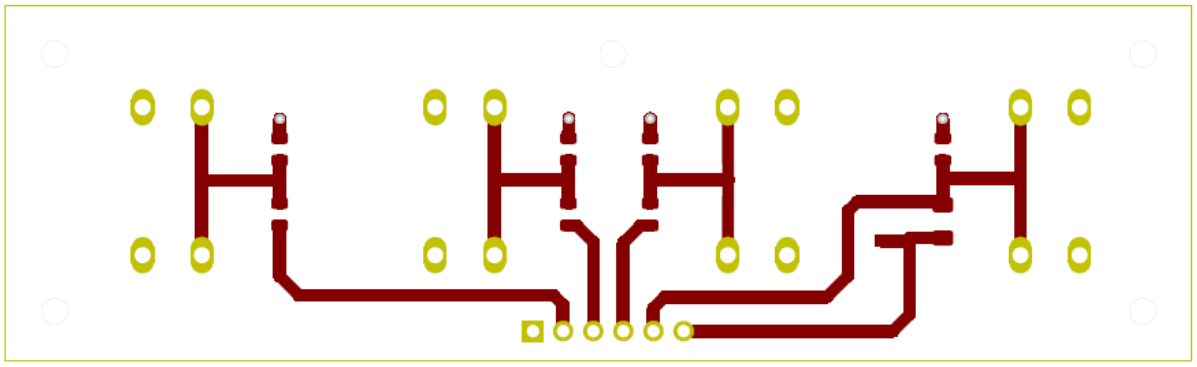


Annex 7. Push buttons board

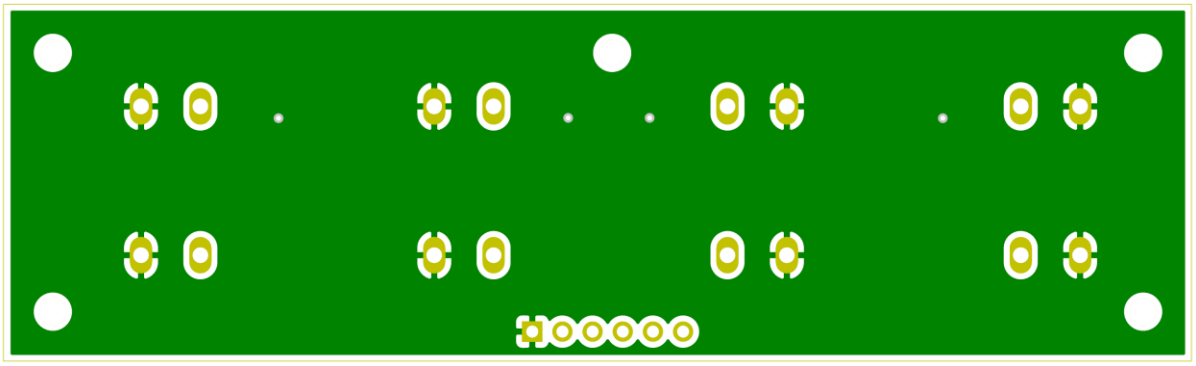
Item	Package	Designation	Vendor
SW4	SW_PUSH-12mm	SW_uC_RST	E-Switch – TL1100F160Q6JBLK – Square
SW3	SW_PUSH-12mm	SW_LOC_REM	E-Switch – TL1100F160Q6JBLK – Square
SW2	SW_PUSH-12mm	SW_CH_ENA	E-Switch – TL1100F160Q6JBLK – Square
SW1	SW_PUSH-12mm	SW_MUX_RST	E-Switch – TL1100F160Q6JBLK – Square
H5,H4,H3,H2,H1	MountingHole_2.2mm_M2	Board Standoff	Würth Elektronik – 971150244
R4	R_1206_3216Metric	10k	Panasonic – ERJP08F1002V
R3,R2,R1	R_0805_2012Metric	1k	Panasonic – ERJP06F1001V
J1	PinHeader_1x06_P2.54mm_Horizontal	Conn_01x06	Sullins Connector Solutions -PEC06SBAN
D1	D_SOD-123	1N4148 or Similar	Vishay Semiconductor Diodes Division – 1N4148W-HE3-08
C4	C_0805_2012Metric	22pF	KEMET - C0805C220J5GACTU
C3,C2,C1	C_0805_2012Metric	47nF	KEMET – C0805C473K5RACTU


H1 MountingHole
H2 MountingHole
H3 MountingHole
H4 MountingHole
H5 MountingHole
Conn_01x06
J1
GND
VCC
uC_RST
MUX_RST
CH_ENA
LOC_REM
PWR_FLAG
VCC
GND
MUX_RST
R1 1k
C1 47nF
SW1 SW_MUX_RST
CH_ENA
R2 1k
C2 47nF
SW2 SW_CH_ENA
LOC_REM
R3 1k
C3 47nF
SW3 SW_LOC_REM
VCC
D1 1N4148
R4 10k
C4 22pF
SW4 SW_uC_RST
uC_RST
Sheet: /
File: Push_Buttons.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2
Id: 1/1

Top:

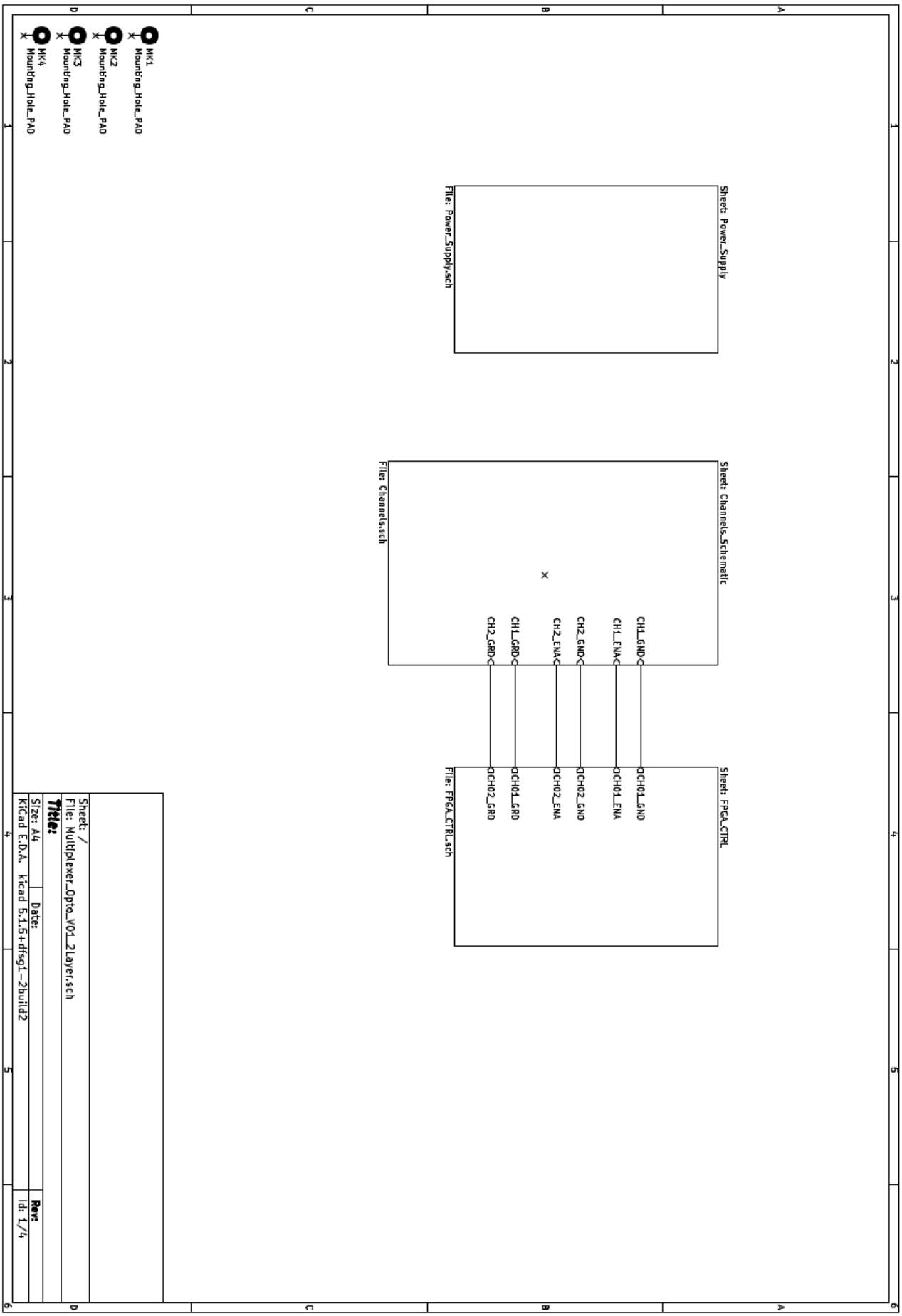


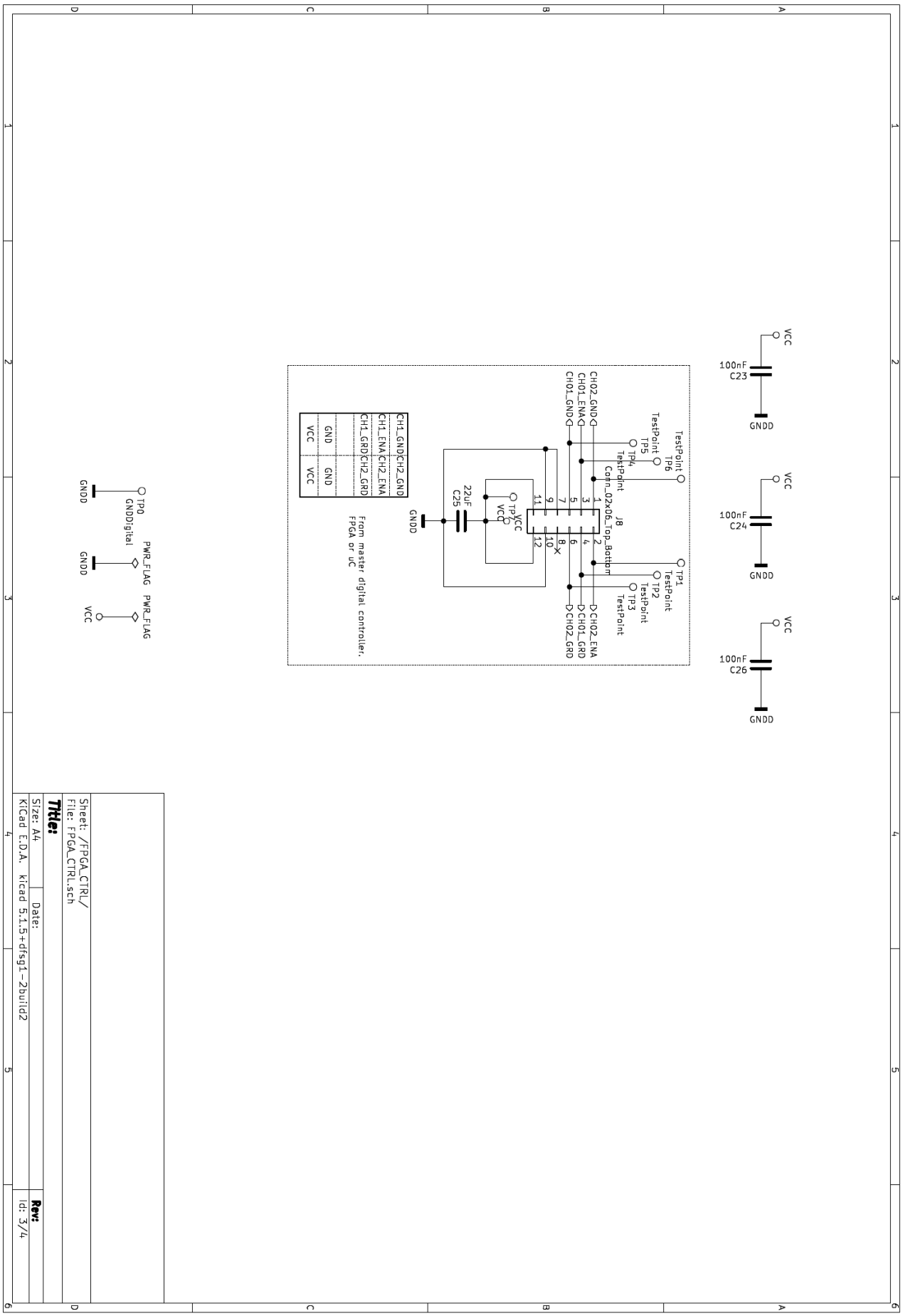
Bottom:

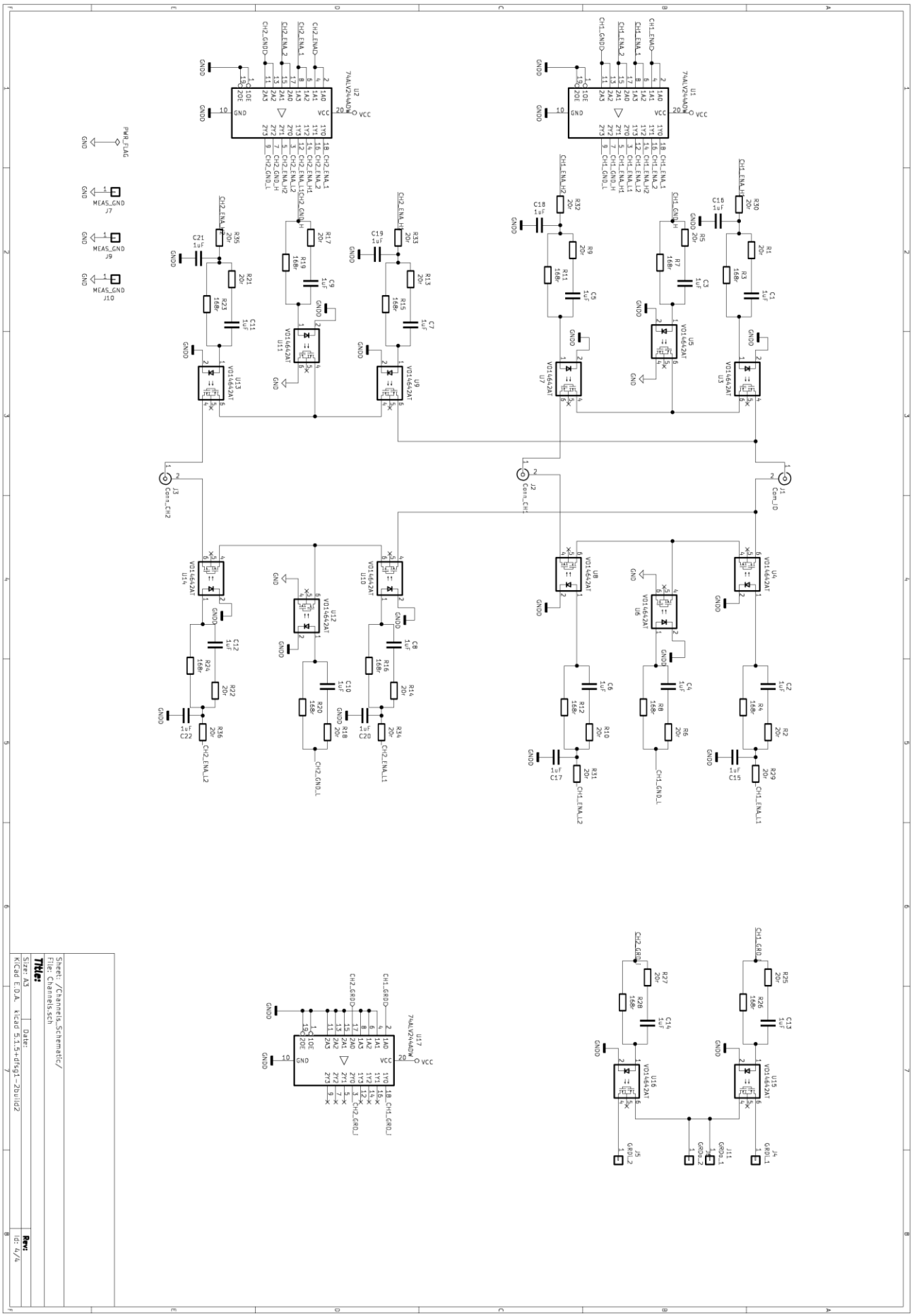


Annex 8. SPDT Slave board

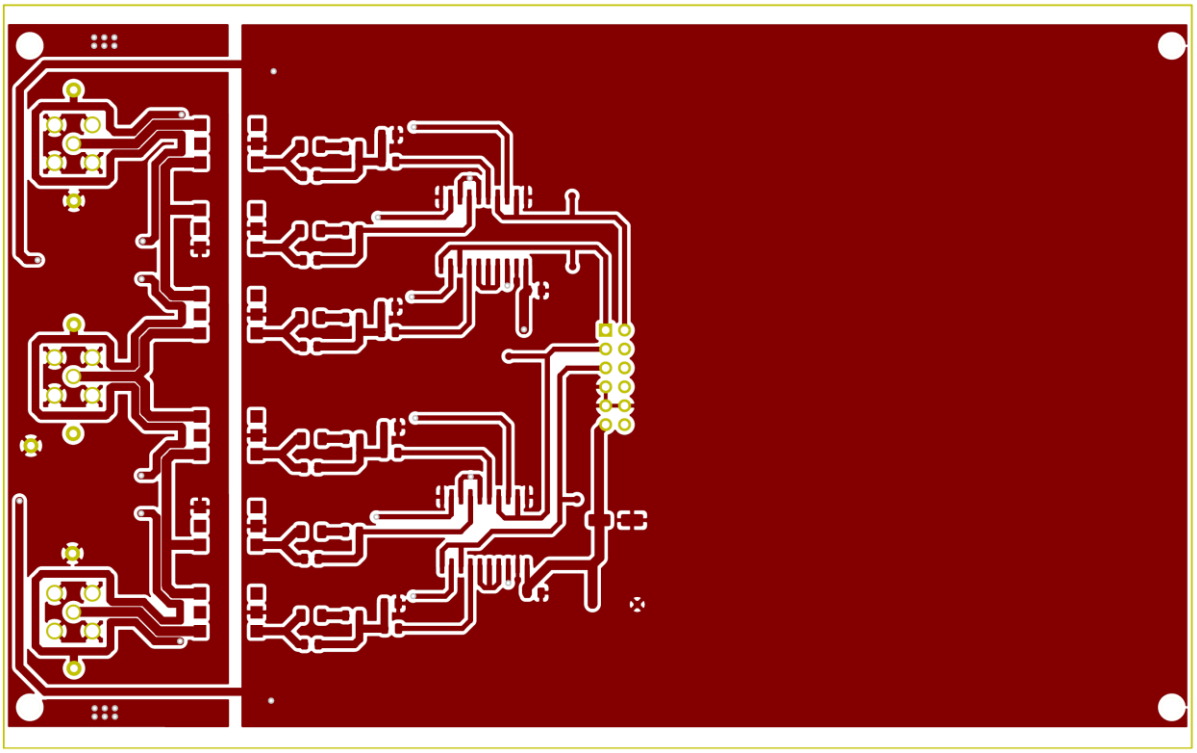
Item	Package	Designation	Vendor
R27,R36,R35, R34,R33,R32, R31,R30,R29, R25,R22,R21, R18,R17,R14, R13,R10,R9, R6,R5,R2,R1	R_0805_2012Metric	20	Panasonic – ERJP06F20R0V
R28,R26,R24, R23,R20,R19, R16,R15,R12, R11,R8,R7,R4, R3	R_0805_2012Metric	390	Panasonic – ERJP06F3900V
C14,C12,C13, C11,C10,C9, C8,C7,C6,C5, C4,C3,C2,C1	C_1206_3216Metric	1uF	KEMET – C1206C105K4RACTU
U17,U2,U1	SOIC-20W_7.5x12.8mm_P1.27mm	*74LS244ADW*	TI – SN74LS244DW
C26,C24,C23	C_0805_2012Metric	100nF	KEMET – C0805C104K4RACTU
C22,C21,C20, C19,C18,C17, C16,C15	C_0805_2012Metric	1uF	KEMET – C0805C105J3RACTU
C25	CP_Elec_5x3	22uF	Panasonic – EEEHB1C220R
U16,U15,U14, U13,U12,U11, U10,U9,U8,U7, U6,U5,U4,U3	SMD-6_W7.62mm_SMDSocket _SmallPads	*VO14642AABTR*	VISHAY – VO14642AABTR
J8	IDC-Header_2x06_P2.54mm_Vertical	Conn_02x06_ Top_Bottom	MOLEX - 90130-1112
J3	*SMA_Amphenol_132136 _Vertical*	Conn_CH2	Amphenol RF – 132136
J2	*SMA_Amphenol_132136 _Vertical*	Conn_CH1	Amphenol RF – 132136
J1	*SMA_Amphenol_132136 _Vertical*	Com_IO	Amphenol RF – 132136
J11	Pin_D1.0mm_L10.0mm	GRDo_1	Amphenol ICC - 68000-401HLF
J10,J9,J7	Pin_D1.0mm_L10.0mm	MEAS_GND	
J6	Pin_D1.0mm_L10.0mm	GRDo_2	
J5	Pin_D1.0mm_L10.0mm	GRDi_2	
J4	Pin_D1.0mm_L10.0mm	GRDi_1	
MK3,MK1, MK4,MK2	MountingHole_2.7mm	Mounting_ Hole_PAD	NA
TP6,TP2,TP5, TP4,TP3,TP1	TestPoint_Pad_D1.0mm	TestPoint	NA
TP7	TestPoint_Pad_D1.0mm	VCC	NA
TP0	TestPoint_Pad_D1.0mm	GNDDigital	NA


MK1
Mounting_Hole_PAD
MK2
Mounting_Hole_PAD
MK3
Mounting_Hole_PAD
MK4
Mounting_Hole_PAD
Sheet: Power_Supply
File: Power_Supply.sch
Sheet: Channels_Schematic
File: Channels.sch
CH1_GND
CH1_ENA
CH2_GND
CH2_ENA
CH1_GRD
CH2_GRD
Sheet: FPGA_CTRL
File: FPGA_CTRL.sch
CH01_GND
CH01_ENA
CH02_GND
CH02_ENA
CH01_GRD
CH02_GRD
Sheet: /
File: Multiplexer_Opto_V01_2Layer.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2
Id: 1/4


VCC
100nF
C23
GNDD
VCC
100nF
C24
GNDD
VCC
100nF
C26
GNDD
CH2_GND
CH1_ENA
CH1_GND
TestPoint
TP6
TestPoint
TP5
TP4
TestPoint
Conn_02x06_Top_Bottom
J8
TP1
TestPoint
TP2
TestPoint
TP3
TestPoint
CH2_ENA
CH1_GND
CH2_GRD
22uF
C25
GNDD
TP
VCC
VCC
From master digital controller,
FPGA or uC
CH1_GND
CH2_GND
CH1_ENA
CH2_ENA
CH1_GRD
CH2_GRD
GND
GND
VCC
VCC
TP0
GNDDigital
GNDD
PWR_FLAG
PWR_FLAG
GNDD
VCC
Sheet: /FPGA_CTRL/
File: FPGA_CTRL.sch
Title:
Size: A4
Date:
Rev:
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2
Id: 3/4


Sheet: /Channels_Schematic/
File: Channels.sch
Title:
Size: A3
Date:
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2
Rev:
Id: 4/4

Top:



Bottom: