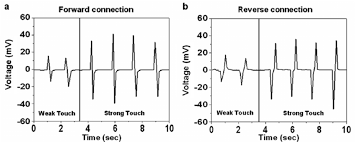
## Polarity measurement of piezo transducer

Many methods are found in www, most common is to poke the transducer cone and measure excitation signal with scope. Standard 10:1 scope probe is ok for that.

The signal looks like that



It is easy explained. If positive going pulse is first followed by negative pulse it means that scope probe is at the positive terminal. You can easy back check by just revert polarity oft he scope probe.

The Amplitude may differ according to piezo, applied force and measurement equipment – will be lower if 1:1 probe is used.

With 10:1 it may go up some 100mV up to some volts.

Well this method is quite OK, but may damage the transducer if to much force is applied.

The polarity is measured as it is not reliable that marking is correct. In the vdatp it is crucial to have correct polarity on all of the transducers.

Picture – bottom side of the transducers used with + marking on the positive terminal.



## Alternate polarity measurement of piezo transducer

Well we can do some different kind of setup to measure polarity, using transmitter/receiver method.

One transducer is put into transmitter mode, and the other is used as receiver. (speaker / microphone setup)

We transmit acoustic wave that will cause an excitation on the receiver transducer.

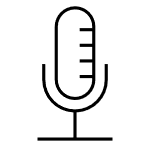
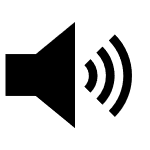
On the scope we have to measure the orignal excitation signal and the received signal.

Simplified test setup:

(speaker and microphone piktogram represent the transducer in the way it is operating)

**NF generator**

* Set to transducer resonantfrequency
* Max amplitude
* Sinus excitation



Scope

10:1 probe on DUT

CH1 (gen out) CH2

Based on the physical distance between TX and RX there is a natural propogation time that is based on the speed of sound in air (approx 343m/sec) – that will show up on the scope as a phase shift between CH1 and CH2. Variation of distance between RX and TX within the wavelength will give a full 360° phase-shift. Please note there is minimum distance for that behaviour.

If you keep this distance a constant value and revert polarity on one transducer then the phase shift will have additional 180°.

In this example with 44kHz it will be 0.78cm.

Use the resonant frequency to have good efficincy on the transmitter and therefore get higer amplitude on the receiver side.

**Real Measurement:**

NF Generator Setup:

* frequency = 44kHz
* amplitude = +-10V (symmetrical)
* waveform: sinus-excitation

**Scope Setup:**

* Timebase: 10µsec / div
* Trigger: CH1 – edge triggered (generator signal)
* CH1: 10V / div
* CH2: 500mV / div

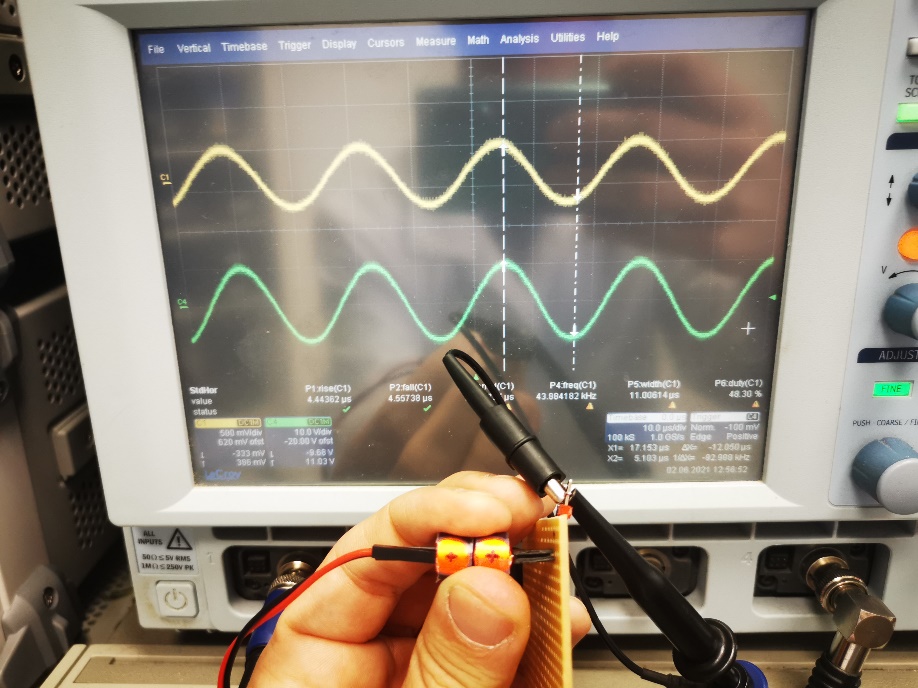
2 transducers RX/TX are marked with (+) on the side. Polarity was tested with first method described.

On the left side is transmitter, on the right side is receiver transducer.

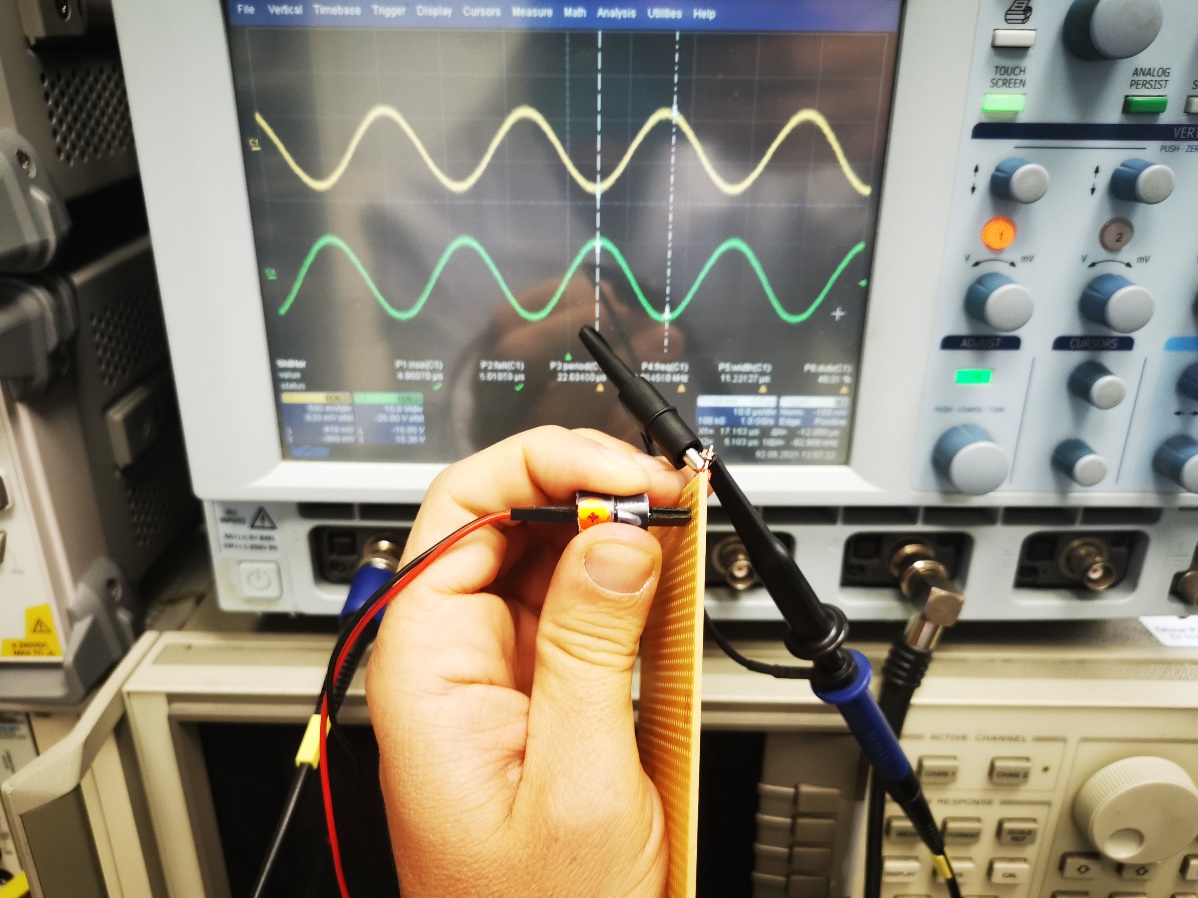
The easiest way to have constant distance is to just hold them together.

In the pictures the yellow signal (upper) is the received signal.

Result with identical polarity on rx and tx transducer



Result with reverted polarity on receiver transducer (180° phase-shift on yellow signal)



# Assembly Notes

Some notes:

- SPI goes to all 4 FPGAs simultaneously. MISO is not used

- there are jumpers in place to allow the board to operate with only one FPGA installed (and only 50 transducers) but to run both FPGAs, the following jumpers must be configured:

-- R122, R123, R124, R126, R127, R128 <= populated

-- R125, R129 <= not populated

-- R105, R111, R113 <= not populated

-- R112 <= populated

- Raspberry Pi W installed on top board only. Top board needs 5V DC/DC converter populated, leave unpopulated on bottom board

-- Rasberry Pi header soldered on backside of Pi

- U101 and U102 are soldered on top board (note errata on U101). Only U101 populated on bottom board

-- only simple twisted pair required for each RS485 connection

- LEDs only connect on top board (populate LED drivers and connectors)

- each board requires a max of 2A at 24V

- to allow the config device to program the FPGAs

-- JP201/JP202 must be left unpopulated

-- JP11001/JP11002 must be cut and msel0/1 must be connected to ground with jumper

- each FPGA must have jumpers configured to tell it which transducers it is responsible for

-- top board: populate R11004

-- bottom board: populate R11003, R203, R204

- U101 and U102 are soldered on top board (note errata on U101). Only U101 populated on bottom board

-- only simple twisted pair required for each RS485 connection

- LEDs only connect on top board (populate LED drivers and connectors)

- each board requires a max of 2A at 24V

- each board requires a max of 2A at 24V. No power sequencing required

- to allow the config device to program the FPGAs

-- JP201/JP202 must be left unpopulated

-- JP11001/JP11002 must be cut and msel0/1 must be connected to ground with jumper

- each FPGA must have jumpers configured to tell it which transducers it is responsible for

-- top board: populate R11004

-- bottom board: populate R11003, R203, R204

- original ultrasonic transducer is no available on aliexpress but https://manorshi.en.alibaba.com/product/60248714908-801018150/Manorshi\_10mm\_40khz\_piezo\_ultrasonic\_Transmitter\_Receiver\_sensor.html should work but I've not tested (make sure to order the transmitter version)

- distance between boards is not super critical - mine are set up 145mm apart but other distance work just as well

- FPGAs can be programmed with a USB blaster type device. I use https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=46

-- all FPGAs get the same program

- distance between boards is not super critical - mine are set up 145mm apart but other distance work just as well

- FPGAs can be programmed with a USB blaster type device. I use https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=46

-- all FPGAs get the same program

- the transducers have a polarity marked on them - but don't trust the markings as they are wrong much of the time. As well, I've been informed that there are phase discrepancies inherent in most of these cheap transducers. I corrected for polarity errors by testing each transducer individually but did not correct for phase errors - this may be the cause of some of the distortions I saw. I did check for these phase discrepancies and found a maximum of about 3% phase error across the transducers. I don't know if this is enough to cause serious distortions but it may be worthwhile for someone to investigate. If this is of concern, a simple LUT can be implemented to compensate for these errors on a transducer by transducer basis

- running these boards at 15V causes the drivers to get really hot - I suggest running them at 10V.

## 100 transducer configuration

- there are jumpers in place to allow the board to operate with only one FPGA installed (and only 50 transducers) but to run both FPGAs, the following jumpers must be configured:

-- R122, R123, R124, R126, R127, R128 <= populated

-- R125, R129 <= not populated

-- R105, R111, R113 <= not populated

-- R112 <= populated

## Top Board

- Raspberry Pi W installed on top board only.

- Top board needs 5V DC/DC converter populated, leave unpopulated on bottom board

- Rasberry Pi header soldered on backside of Pi

- U101 and U102 are soldered on top board (note errata on U101).

- LEDs only connect on top board (populate LED drivers and connectors)

-- top board: populate R11004

## Bottom Board

-- bottom board: populate R11003, R203, R204

---Only U101 populated on bottom board

### Top/Bottom differences summary

- U101 and U102 are soldered on top board (note errata on U101).

-Only U101 populated on bottom board

- each FPGA must have jumpers configured to tell it which transducers it is responsible for

-- top board: populate R11004

-- bottom board: populate R11003, R203, R204

## Programming

- to allow the config device to program the FPGAs

-- JP201/JP202 must be left unpopulated

-- JP11001/JP11002 must be cut and msel0/1 must be connected to ground with jumper