MIMORPH - User Guide

List of Content

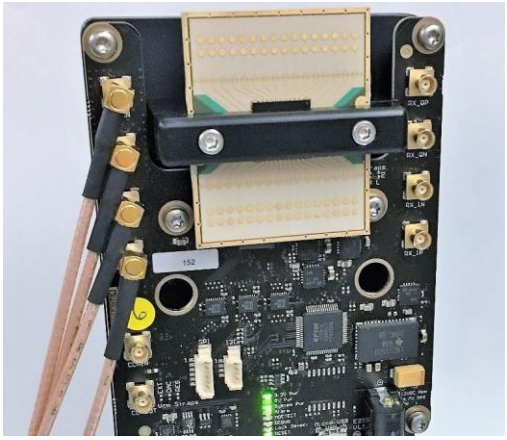
RFSOC PREPARATIONS

1. Download the BOOT.bin file and save it at the SD card.
2. Set Switch SW6 pins to 1110(OFF, OFF, OFF, ON) to configure the RFSoC in SD boot mode.

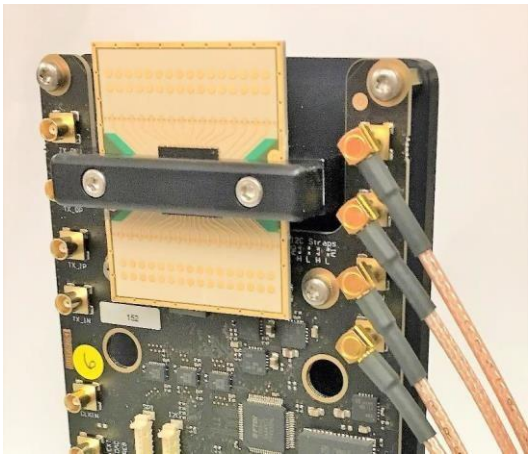
SW6 Configuration

3. Assign a static IP address in the host machine.
   IP address: 192.168.1.3
   Subnet mask: 255.255.255.0
   Default gateway: 192.168.1.1

4. Plug an ethernet cable from the FPGA to the PC Host.
5. Plug the 4 MMCX to SMA-F cables into I_N, I_P, Q_N and Q_P for the receiver and transmitter antenna respectively.

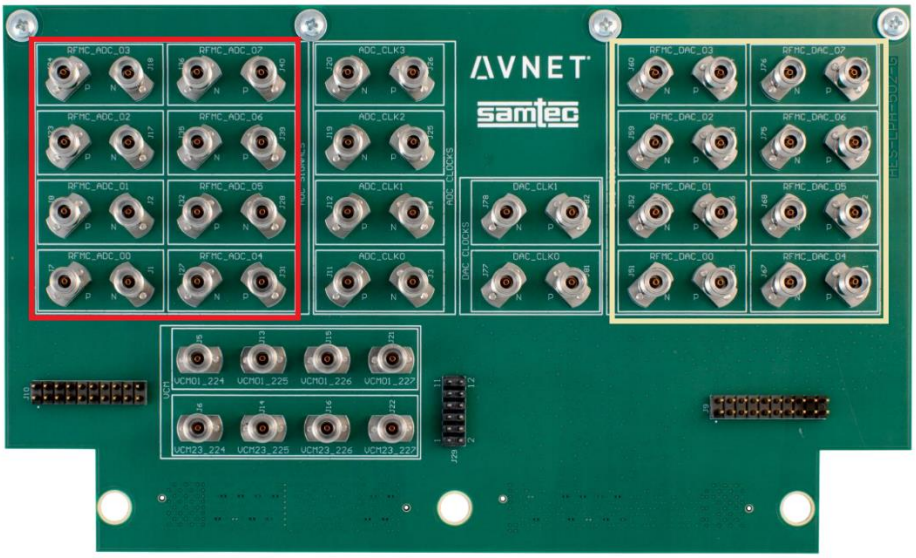


Connection for TX antenna



Connection for RX antenna

6. Use the SMA-M/SMA-M cables to connect between the FPGA converters to the antennas. Check for you daughterboard model and see the figure to match the connections.



AVNET LPA-502-G Daughterboard

XILINX XM500	AVNET LPA-502-G	IQ SIGNAL	XILINX XM500	AVNET LPA-502-G	IQ SIGNAL
ADC_224_CH_0	ADC0	RX_I_0	DAC_228_CH_0	DAC0	TX_I_0
ADC_224_CH_1	ADC1	RX_Q_0	DAC_228_CH_1	DAC1	TX_Q_0
ADC_225_CH_0	ADC2	RX_I_1	DAC_228_CH_2	DAC2	TX_I_1
ADC_225_CH_1	ADC3	RX_Q_1	DAC_228_CH_3	DAC3	TX_Q_1
ADC_226_CH_0	ADC4	RX_I_2	DAC_229_CH_0	DAC4	TX_I_2
ADC_226_CH_1	ADC5	RX_Q_2	DAC_229_CH_1	DAC5	TX_Q_2
ADC_227_CH_0	ADC6	RX_I_3	DAC_229_CH_2	DAC6	TX_I_3
ADC_227_CH_1	ADC7	RX_Q_3	DAC_229_CH_3	DAC7	TX_Q_3

7. Plug the two GPIO-SPI cable (check section about how to build the cable and information for the connection) into the indicated connectors (see figure). Make sure you connect it in the right direction.

GPIO-SPI cable connectors at the daughterboard

8. Turn on the FPGA

SIVERS SOFTWARE ENVIROMENT

This section was directly extracted from the *User Manual EVK06002/0.* You can find this

document within the installation files from the stfp repository.

System requirements

• A computer with USB interface and Linux Ubuntu 16.04 or later. Equivalent Linux distribution might also work.
• Python 2.7 or later

Installation instructions

1. Connect the USB to Micro USB cable to the computer

2. Connect the Power adapter to EVK06002/00 and a Wall Wart.

3. Download software package from sftp.

    IP Address: 193.104.100.110
    Use an stfp-ssh client:
    - Host: 193.104.100.110
    - Port: 22
    - Protocol: Choose ”SFTP – SSH File Transfer Protocol”
    - User: Will be sent after NDA is signed
    - Password: Will be sent after NDA is signed

4. Open Linux command window.

5. From root create a new directory.

    > mkdir ~/evk06002/

6. Copy ederenv_<date>_<time stamp>.tar.gz to the newly created evk06002 directory.
    > cp ~/media/your-usb/ ederenv_<date>_<time stamp>.tar.gz ~/evk06002/

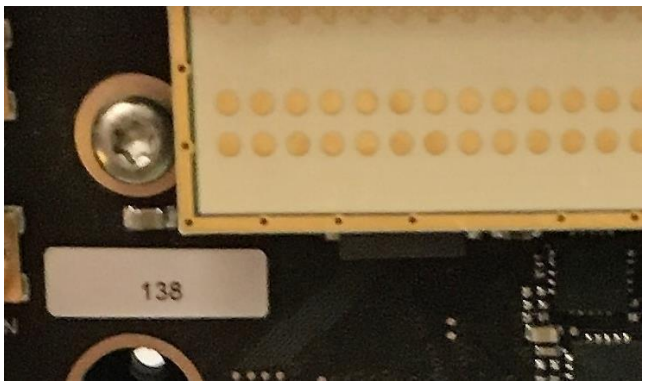
7. Unpack the Eder software environment.
    > tar -zxvf ederenv_<date>_<time stamp>.tar.gz

8. Run Eder install script.
    > ./install_mb1.sh

9. At this point it’s possible to choose to run either Eder command line shell or Eder GUI

10. For Eder command line shell run start script mb1. Script shall contain serial number located at the motherboard PCB Sivers SN motherboard ID sticker,
    > ./start_mb1_b.sh SN<device serial number>

institute imdea networks



EVK Serial Number

11. Copy 'CONFIG_TX.PY' and 'CONFIG_RX.PY' from the IMDEA repository to the folder *./Eder_B* inside the framework. These two scripts will prepare the RX and TX antennas respectively to send/receive data and change BPs.

> execfile('CONFIG_TX.PY')

> execfile('CONFIG_RX.PY')

12. In order to enable/disable the antennas, use these functions. Take into account that there are different commands for RX and for TX.

> eder.rx.enable()

> eder.rx.disable()

> eder.tx.enable()

> eder.tx.disable()

14. In order to enable external GPIO pulses to change through different BP, use the following function:

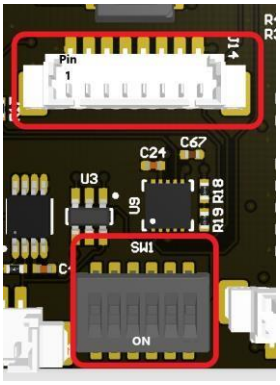
> eder.evkplatform.drv.bfoff ()

15. Finally, send this command to disable commands from USB andto manage the antennas via external SPI signals.

> eder.evkplatform.drv.spioff ()

Connect GPIO control cable

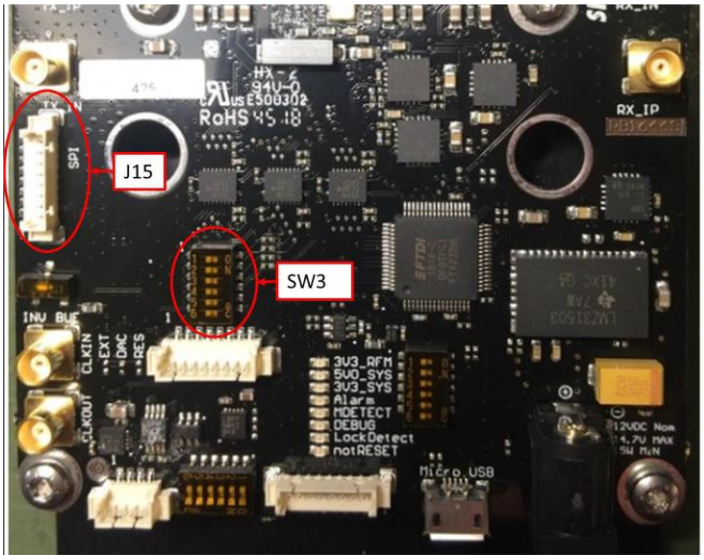
Move pin 3,4 and 5 switches to ON (low) to allow external BP changing. Plug the BF control cable to J14.



GPIO Switch and connector

Connect SPI control cable

Move pin 1,3 and 4 switches to ON (right) to allow external SPI control commands from the FPGA to the RF kit. Plug the SPI control cable to the J15 connector.



SPI Switch and connector

MATLAB CONTROL FUNCTIONS

- **<ts> = system_init (ip)**: configures the TCP/IP stack in the host PC and opens a connection with platform.
    - **ip** -> platform IP address.
    - **ts** -> tcp session identificator.

- **<pl> = transmit_XCH (ts, path , idle_time, SD):** sends the set of I/Q samples the userwants to transmit using the number of macro channels X, loads them and starts the transmission synchronously in all enabled channels.
    - **ts** -> Tcp session identificator.
    - **path** -> location of the samples to transmit**.**
    - **idle_time** -> samples between packets.

- **SD** -> (*true*: reads data from SD; *false*: reads data from PC Host via TCP/IP).
- **pl** -> packet length.

- **capture_samples_XCH (ts,rx,nSamples,path,filename,SD):** enables the continuous capture of I/Q samples. The captured samples can be sent via TCP/IP or saved on an SD card (if present).
  - **ts** -> Tcp session identificator.
  - **rx ->** receiver bitmask.
    - '0001' – stream 1
    - '0010' – stream 2
    - '0100' – stream 3
    - '1000' – stream 4
  - **nSamples ->** number of samples to store.
  - **path** -> location of the samples to store.
  - **filename** -> name of the file.
  - **SD** -> (*true*: reads data from SD; *false*: stores data at PC Host via TCP/IP).

- **capture_pkt_XCH (ts,tx,nPackets, pl,path,filename,SD)**: it captures a certain number of packets of a specific length. The captured packets are again sent to the host PC or saved on an SD card.
  - **ts** -> Tcp session identificator.
  - **path** -> location of the samples to store.
  - **nPackets ->** number of packets stored by packed detector.
  - **pl ->** packet length.
  - **filename** -> name of the file.
  - **SD** -> (*true*: reads data to SD; *false*: stores data at PC Host via TCP/IP).

- **configure_PD (ts, highTime)**: configures the settings for packet detector. Set this function before starting to capture packets.
  - **ts** -> Tcp session identificator.
  - **pl** -> length of the packet to be detected.

- **configure_AWV_control(ts, type, P, M, N, L, T_INIT, T_HIGH, enable):** manages the configuration of the GPIO pulses. Enable and disables the block.
  - **ts** -> Tcp session identificator.
  - **P**-> Number of cycles or the first TRN field (without BP change.
  - **M** -> Number of BP changes per P beam patterns.
  - **N** -> Number of clock cycles per each of the M BPs.
  - **L** -> Number of repetitions of P+M BPs.
  - **T_INIT** -> Number of delay cycles after the trigger before start with the BP P.
  - **T_HIGH** -> Number of clock cycles the INC, RTN and RST pulses stay high at the output.
  - **enable** -> (1: enable pulses; 0: disable pulses).

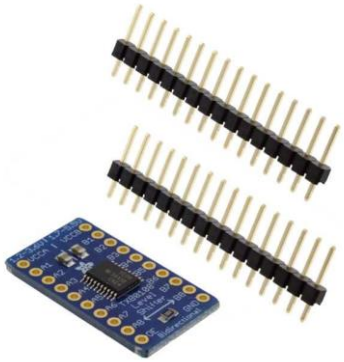
- **write_SPI (ts, addr, val, interface, ant)**: writes a value to a SPI register
  - **ts** -> Tcp session identificator.
  - **addr** -> address of the SPI slave registers.
  - **val** -> byte to write.
  - **interface** -> select between SPI drivers. For SISO designs use '0'.
  - **ant** -> select between SPI slaves. For SISO designs use '0'.

BUILD SPI-GPIO CABLE

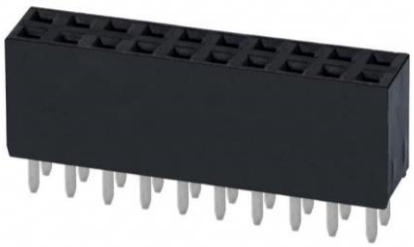
This is the list of components you will need:

- Pre-Crimped cables

- Level shifter 1.8 to 3.3V

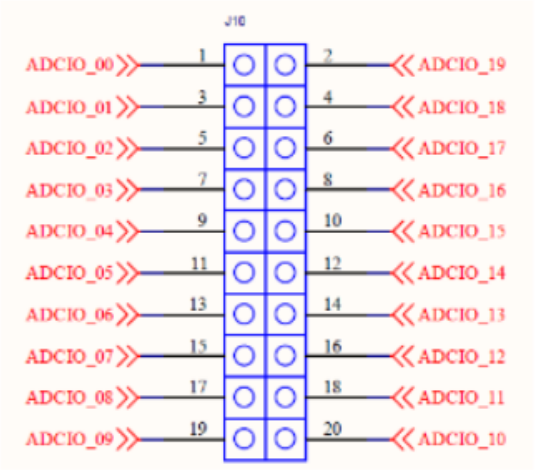


- Female Header 0.1” 2x10

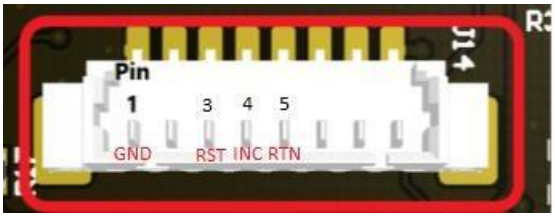


- Molex 8 pin cable

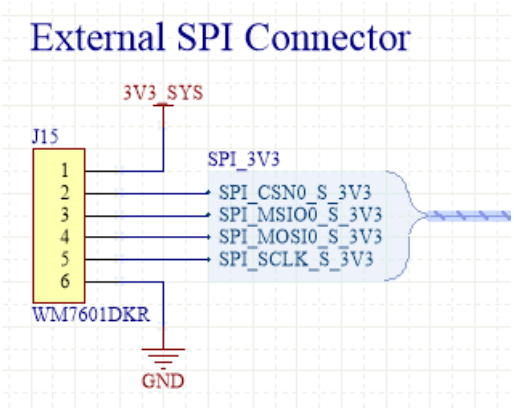
1. Each antenna will need an individual level shifter. You can connect to two level shifter per daughterboard connector (there are two per daughterboard). This means you will need 2 cables with 2 level shifter each to do MIMO 4x4.
2. First solder the level shifter to the row of pins
3. Connect the pre-crimped cables to the female header. You will need 18 cables for one cable. We will be using from ADCIO_00 to ADCIO_17. See picture below.



4. Identify the matching connections from the header to the level shifter by following the table at the end of the section. Focus on column BREAKOUT BOARD and LEVEL SHIFTER.
5. Remove the connector of one side the molex cable. This side will be pluged into the level shifter. Follow again the table to build the GPIO cable to the SIVERS. Focus on columm LEVEL SHIFTER and GPIO



6. Do the same for SPI cable. Focus on columm LEVEL SHIFTER and SPI.



7. The cable should look something similar to this:

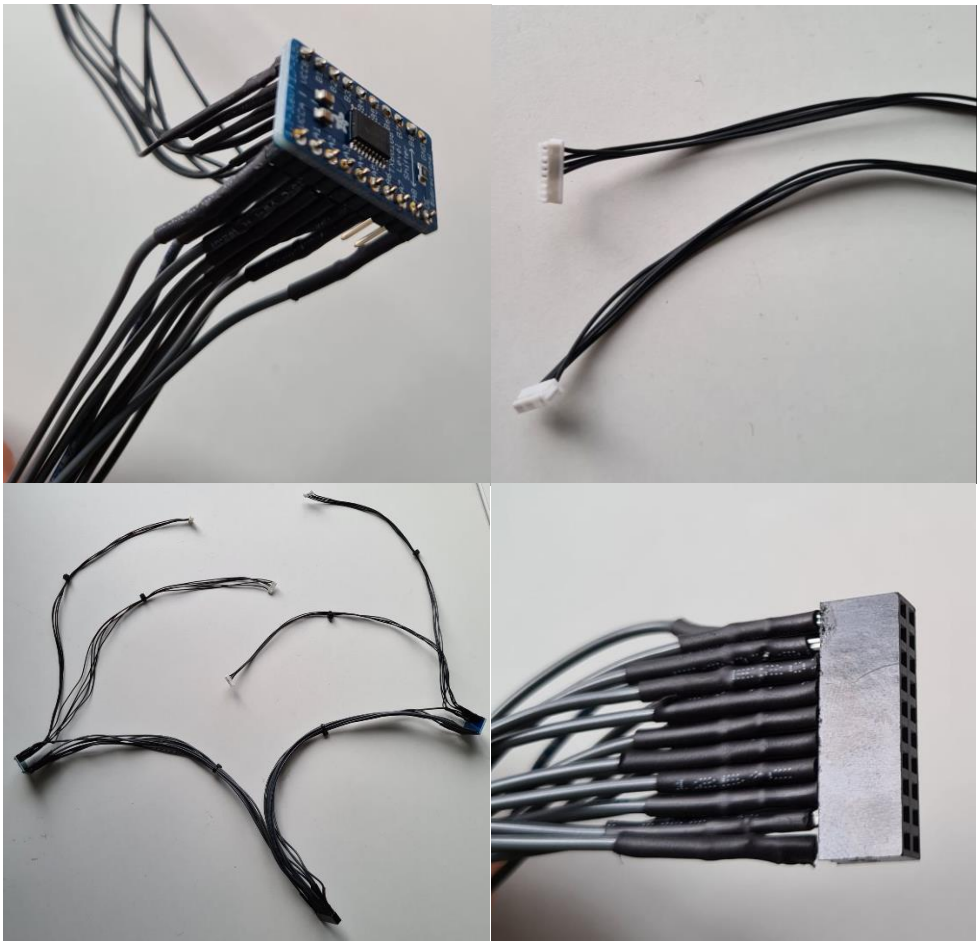


Table: List of corresponding pins

	BREAKOUT BOARD	RFSoC PIN	Level Shifter	GPIO	SPI
INC0	ADC00	AP5	1	4	XX
RST0	ADC01	AP6	2	5	XX
RTN0	ADC02	AR6	3	3	XX
SS0	ADC03	AR7	4	XX	7
SCLK0	ADC04	AV7	5	XX	4
MO0	ADC05	AU7	6	XX	5
INC1	ADC06	AV8	1	4	XX
RST1	ADC07	AU8	2	5	XX
RTN1	ADC08	AT6	3	3	XX
SS1	ADC09	AT7	4	XX	7
SCLK1	ADC10	AU5	5	XX	4
MO0	ADC11	AT5	6	XX	5
INC2	DAC00	A9	1	4	XX
RST2	DAC01	A10	2	5	XX
RTN2	DAC02	A6	3	3	XX
SS2	DAC03	A7	4	XX	7
SCLK2	DAC04	A5	5	XX	4
MO2	DAC05	B5	6	XX	5
INC3	DAC06	C5	1	4	XX
RST3	DAC07	C6	2	5	XX
RTN3	DAC08	B9	3	3	XX
SS3	DAC09	B10	4	XX	7
SCLK3	DAC10	B7	5	XX	4
MO3	DAC11	B8	6	XX	5
3v3	XX	XX	VCCB	XX	8
GND	ADC12 - DAC12	XX	GND	1	1
GND	ADC13 - DAC13	XX	GND	1	1
OE1	ADC14 - DAC14	XX	OE	XX	XX
1v8	ADC15 - DAC15	XX	VCCA	XX	XX
OE2	ADC16 - DAC16	XX	OE	XX	XX
1v8	ADC17 - DAC17	XX	VCCA	XX	XX

institute imdea networks