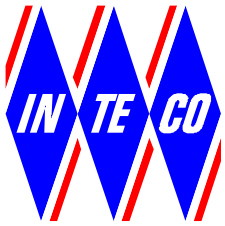
RT-DAC/Zynq

Rev 1.0

User's Manual

Customised FPGA configuration



www.inteco.com.pl

NOTES

SAFETY OF THE EQUIPMENT

The equipment, when used in accordance with the supplied instructions, within the parameter set for its mechanical and electrical performance, should not cause any danger to health or safety if normal engineering applications are observed.

If, in specific cases, circumstances exist in which a potential hazard may be brought about by careless or improper use, these will be pointed out and the necessary precautions emphasised.

Some National Directives require to indicate on our equipment certain warnings that require attention by the user. These have been indicated in the specified way by labels. The meaning of any labels that may be fixed to the equipment instrument are explained in this manual.

Risk of electric shock

PRODUCT IMPROVEMENTS

The Producer reserves a right to improve design and performance of the product without prior notice.

All major changes are incorporated into up-dated editions of manuals and this manual is believed to be correct at the time of printing. However, some product changes which do not affect the capability of the equipment, may not be included until it is necessary to incorporate other significant changes.

ELECTROMAGNETIC COMPATIBILITY

This equipment, when operated in accordance with the supplied documentation, does not cause electromagnetic disturbance outside its immediate electromagnetic environment.

COPYRIGHT NOTICE

ACKNOWLEDGEMENTS

Inteco Sp. z o. o. acknowledges all trademarks.

MICROSOFT, WINDOWS are registered trademarks of Microsoft Corporation.

MATLAB and Simulink are registered trademarks of MathWorks Inc.

CONTENTS

1. INTRODUCTION AND GENERAL DESCRIPTION

1.1 GENERAL INFORMATION

The RT-DAC/Zynq is a multifunction analog and digital I/O board dedicated to real-time data acquisition and control in the Windows , environments. The board contains a Xilinx® FPGA chip. All boards are built as the OMNI version. It means the boards can be reconfigured to introduce a new functionality of all inputs and outputs without any hardware modification.

The default configuration of the FPGA chip accepts signals from incremental encoders and generates PWM outputs, typical for mechatronic control applications and is equipped with the general purpose digital input/outputs (GPIO), A/D and D/A converters, timers, counters, frequency meters and chronometers.

The controller of the RT-DAC/Zynq system uses MicroZed development board as calculation and front-end signal processing unit. The board includes Zynq SoC circuit and consists of two ARM A9 processors and the FPGA fabric (see Fig.1.1). The MicroZed is plugged-into a carrier board, which establishes interfaces to the measurement and actuator signals.

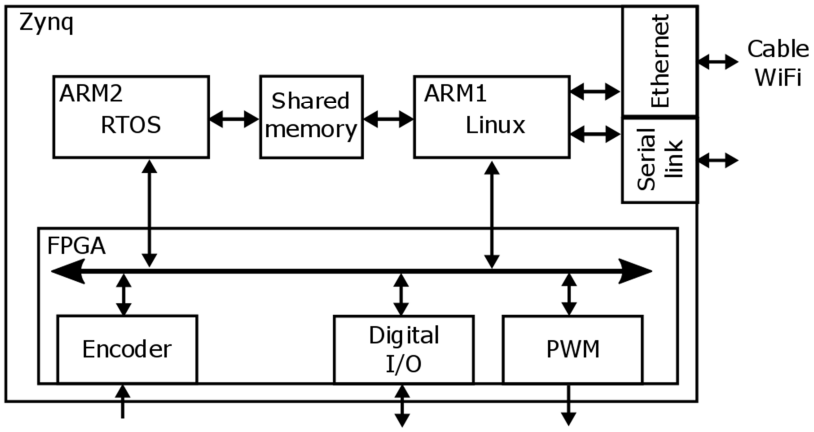


Figure 1.1. The architecture of the heterogeneous RT-DAC/Zynq controller.

The ARM processors operate in Asymmetric Multiprocessing mode. The ARM1 processor runs Linux operating system. It runs all non-real-time tasks like file system support, on-line monitoring and communication with a user. The second processor runs real-time tasks generated automatically from the Simulink diagrams. AMP configuration separates the execution of Linux critical sections from the real-time task, which leads to small real-time jitter level.

Both processors communicate by a shared memory buffer. The communication causes the Linux application to operate as supervisors to the real-time task, which in turn responds for direct processing of RT-DAC/Zynq signals.

The FPGA is responsible for front-end signal processing. It generates PWMs for the DC drives, processes encodes waves and establishes an interface to the general purpose I/Os.

1.1 SPECIFICATION

Analog Inputs

Channels:	16 single-ended, multiplexed
Resolution:	14 bit
Input ranges:	±10V, programmable gain (x1, x2, x4, x8, x16)
Conversion time:	5.4 μs
Trigger:	all the A/D channels are scanned automatically when data are required
Reference voltage:	on-board

Analog Outputs

Channels:	4
Resolution:	14 bit
Output range:	± 10V, ± 5V
Settling time:	10 μs (to 0.01%)
Reference voltage:	on-board

Digital Input/ Output

Channels:	26 bi-directional, direction setting; 8 channels shared with PWM outputs; 8 channels shared with 4 incremental encoder channels
Direction:	bi-directional, individually software programmable
Input voltage:	VIH = 2.0V÷3.6V, VIL = - 0.5V÷0.8V
Output voltage:	VOH = 2.4V (min), VOL = 0.4V (max)
Output current:	2mA per channel
Standard:	LVTTL

PWM Outputs

Channels:	8
Resolution:	32 bits for both High and Low states
Base frequency:	10ns resolution for both High and Low states

Incremental encoders

Channels:	4
Output:	32 bit counter
Index	-

1.2 SYSTEM COMPONENTS

To use the RT-DAC/Zynq system the following software and hardware components are required:

Hardware

- RT-DAC/Zynq unit. SD card plugged into the RT-DAC/Zynq
- Power supply,
- Wiring and/or cables for the CN1 and CN2 connectors. Optionally wiring terminal board.
- Ethernet cable, optional USB mini b cable

Software

- Toolbox for MATLAB/Simulink
- SD cart contents: Petalinux and configuration files

MATLAB cannot be installed in the "Program Files" directory (name of the directory cannot include space).

Manuals:

- *User's Manual*

1.3 SOFTWARE INSTALLATION

Insert the installation CD and follow the displayed commands.

Remember that it is time consuming procedure to be executed (approx. 5 minutes). Simultaneously with the installation of the RT-DAC/Zynq toolbox it is installed the Linaro cross-compiler.

1.4 PETALINUX

The ARM1 processor runs the PetaLinux version 2016.3 operating system. The system boots from SD card and contains at least the following files:

- BOOT.BIN – contains the First Stage Boot Loader (FSBL), bitstream for the FPGA fabric and U-Boot. The Zynq circuit automatically looks for this file at the SD card, downloads bitstream to the FPGA, runs FSBL and finally runs U-Boot.
- image.ub – compressed image of the Linux file system, processed by the U-Boot. The file is uncompressed during booting phase and applied to create the RAM disk Linux file system. **Please remember that the file system is of RAM disk type and any changes to the files will be lost after turning off the power or reloading the system.** The only option of permanent changes in the file system is to change the content of the image.ub file or save files on the SD card.
- RTDACStartup – optional script executed by Linux during boot.

The PetaLinux 2016.3 project applied to the controller of the RT-DAC/Zynq model can be available on request.

1.4.1. RTDACSTARTUP

The script RTDACStartup is located at the SD card and contains commands executed by Linux during the boot phase. It contains the commands described in the table below.

Commands	Description
echo 953 > /sys/class/gpio/export *echo out > /sys/class/gpio/gpio953/direction*	Optional. Configuration of the GPIO of the LED on the MicroZed board. Required to support heartbeat monitoring of the real-time applications.
ifconfig eth0 192.168.1.102	Set IP address of the Ethernet cable port
./ExtModeTunnel.elf &	Run application responsible for the communication in the external communication mode
./uZServer.elf &	Run application responsible for the communication with the real-time task and with the FPGA registers
./elfio.elf uZSimulink.elf -v *./rwmem.elf 0x42800008 4883* *./rwmem.elf 0x42800004 1* *./rwmem.elf 0xFFFFFFF0 0x30000000*	Optional. Commands to start the real-time application immediately after the booting of the Linux.

The RTDACStartup file can be modified by a user on request.

1.4.2. FTP

The file system on the SD card is accessible by FTP protocol. The login is root. The password is not required.

1.4.3. TELNET

A user can connect to the board by Telnet protocol. The login is root and the password is root.

1.4.4. WEB SERVER

An HTTP server is running by Petalinux. The address

http://192.168.1.102/cgi-bin/uZCGI

presents a table that contains diagnostic information.

2. CONNECTOR PIN ASSIGNMENT

The digital version the RT-DAC/USB2D is equipped with one 40-pin I/O connector CN1. The pin assignment of the connector is shown in Table 1.

Table 1. RT-DAC/USB2 I/O CN1 pin assignment

CN1 Pin No	Power supply	Digital I/O	PWM	Encoder
1		DIO 0		ENC 0 A
2	GND	DIO 1		ENC 0 B
3		DIO 1		ENC 1 A
4	GND	DIO 1		ENC 1 B
5		DIO 2		ENC 2 A
6	GND	DIO 1		ENC 2 B
7		DIO 3		ENC 3 A
8	GND	DIO 1		ENC 3 B
9		DIO 4	PWM 0	
10	GND	DIO 1	PWM 1	
11		DIO 5	PWM 2	
12	GND	DIO 1	PWM 3	
13		DIO 6	PWM 4	
14	GND	DIO 1	PWM 5	
15		DIO 7	PWM 6	
16	GND	DIO 1	PWM 7	
17		DIO 8		
18	GND	DIO 1		
19		DIO 9		
20	GND	DIO 1		
21		DIO 10		
22		DIO 11		
23		DIO 12		
24		DIO 13		
25		DIO 14		
26		DIO 15		
27		DIO 16		
28		DIO 17		
29		DIO 18		
30		DIO 19		
31		DIO 20		
32		DIO 21		
33		DIO 22		
34		DIO 23		
35		DIO 24		
36		DIO 25		
37	GND			
38	GND			
39	+5 V			
40	+3.3 V			

Only 26 pins at the CN1 connector are used as general-purpose digital I/O signals (denoted as DIO0 to DIO25). The remaining pins are the ground and power (GND, +5V, +3.3V). The general purpose digital I/O signals can be individually configured to be either the input or output.

The analog and digital version of the RT-DAC/Zynq board is equipped additionally in the CN2 40-pin connector. The pin assignment of the connector is shown in 2.1.

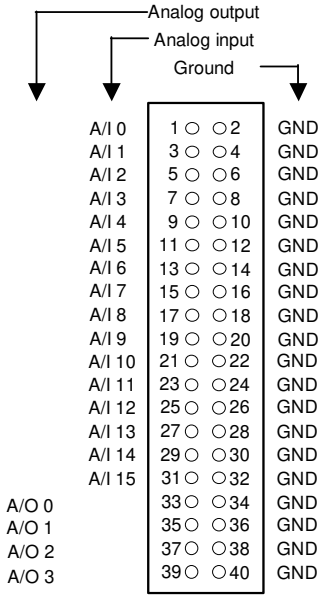


Fig.2.1. RT-DAC/USB2 I/O CN2 connector

The RT-DAC/Zynq is equipped with 16 multiplexed analog inputs located at the CN2 connector. They are denoted as A/I 0 up to A/I 15. The output of the analog multiplexer is connected to the input of the digital programmable analog amplifier. The board is also equipped with four 14-bit D/A converters. These outputs are denoted as A/O 0 up to A/O 3.

As the number of general-purpose digital I/O signals is limited, some of them are shared with the signals of the specialized blocks. The specialized blocks are implemented as the modules in the on-board FPGA structure. It means that the functions of the specialized block are hardware-implemented. The specialized blocks are:

- PWM generators – there are eight PWM blocks. The outputs are denoted: *PWM0*, *PWM1*, *PWM2*, *PWM3*, *PWM4*, *PWM5, PWM6* and *PWM7*,
- incremental encoders – the device contains eight incremental encoder channels. Each channel requires two input signals – wave A (denoted as *ENC0_A* up to *ENC3_A*) and wave B (*ENC0_B* up to *ENC3_B*.

There are two kinds of specialized blocks:

- the first kind of the specialized blocks contains digital output signals (PWM blocks). In this case the appropriate pins of the CN1 connector can operate as the general purpose digital I/O signals or as the output of the specialized block. The operating mode is determined by a mode configuration register (CN1 Pin Mode Register). If they operate as general purpose digital I/Os

their directions and states are determined by the software. If they operate as the outputs of the specialized blocks the state of the output is controlled by the PWM block. The states of the output signals can be read by the software (the software can check the PWM output),

- the second kind of the specialized blocks contains only the digital input signals (the incremental encoders). In this case it is not necessary to select the operating mode of the block signals. If the appropriate general purpose I/O signals are configured to be the inputs then their states can be read by the software and simultaneously the signals excites the specialized block (see Fig.2.2). If the shared general purpose I/O signals are configured to be outputs their states can be set by the software and simultaneously the signals excites the specialized block (see Fig.2.3). This operating mode can be applied for testing of the specialized blocks – for example the encoder can be excited and read in a software manner. This testing strategy is not significant during common device applications. It may be very useful when a user wants to design and test his own FPGA blocks (see *"RT-DAC/Zynq Device XILINX Programming Guide"*)

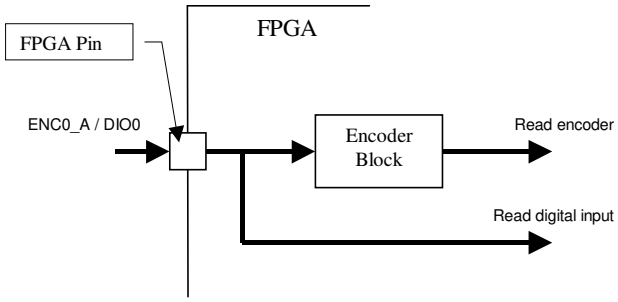


Fig.2.2. The shared FPGA pin configured as the input

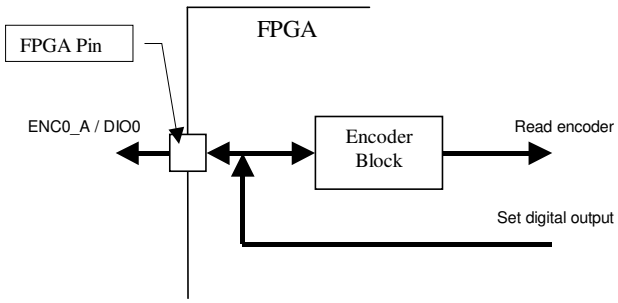


Fig. 2.3. The shared FPGA pin configured as the output

2. STARTING PROCEDURE

2.1 STARTING PROCEDURE

It is assumed that RT-DAC/Zynq toolbox installation was successful.

Invoke MATLAB by double clicking on the MATLAB icon. The MATLAB command window opens.

If the MATLAB R2019 or newer is used run command *rehash toolbox*, close Matlab and open again.

Then type:

RTDACZynq

MATLAB brings up the RT-DAC/Zynq Control Window (see Fig. 2.1).

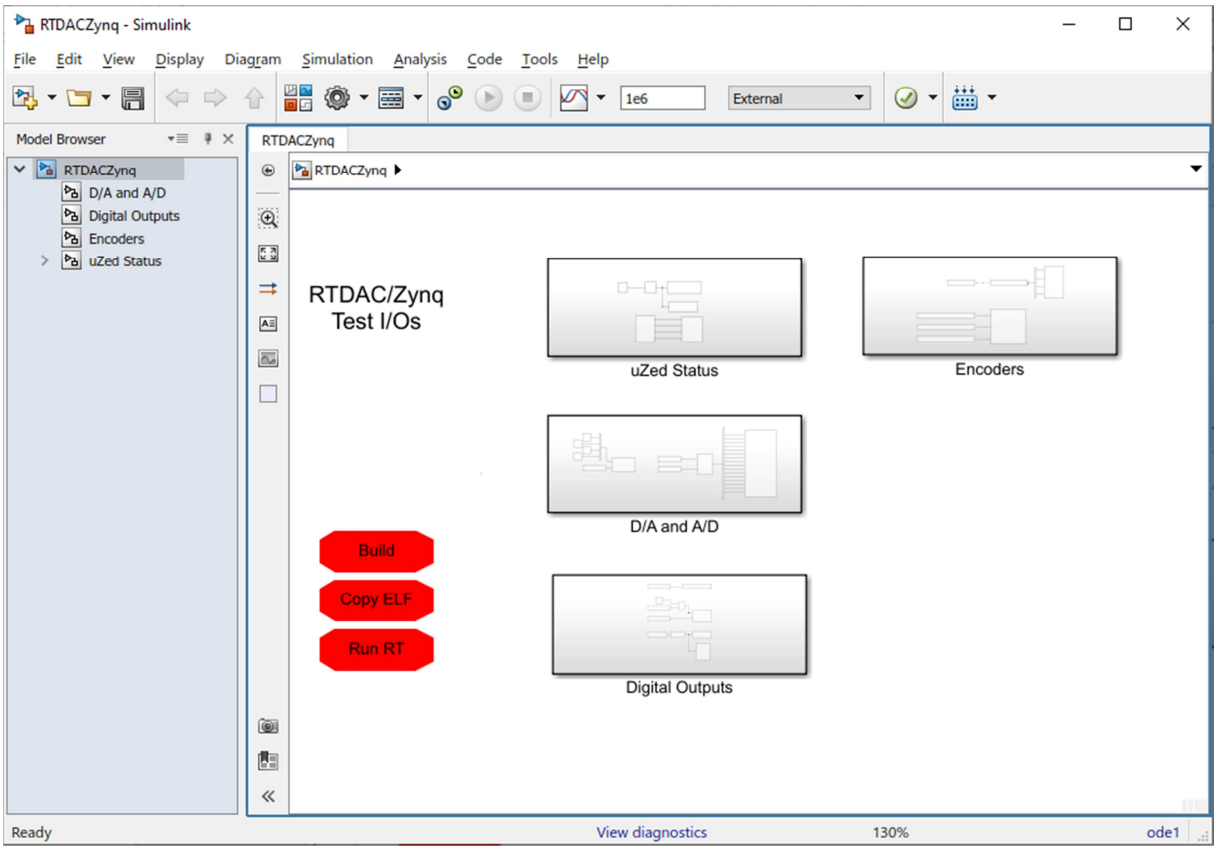


Fig. 2.1 Control Window of the *RT-DAC/Zynq* system

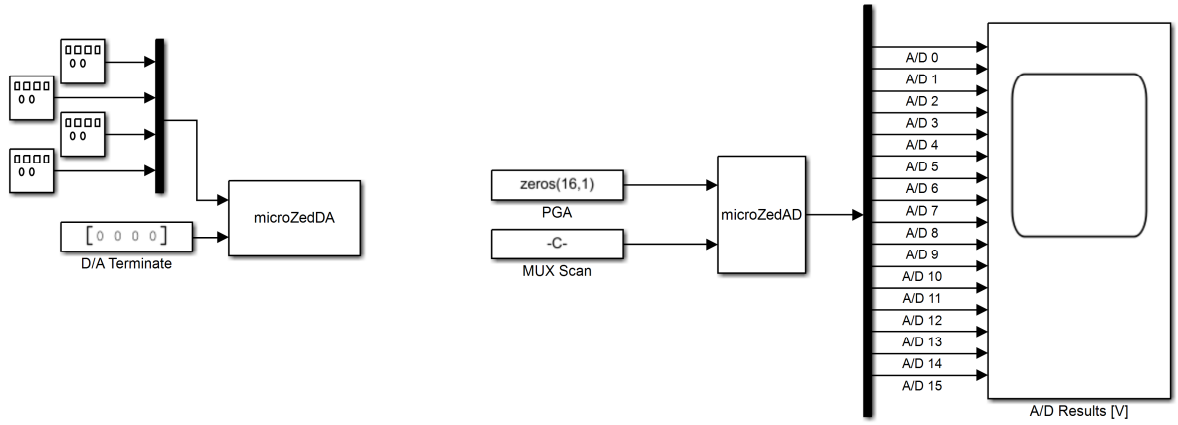
The RT-DAC/Zynq Control Window contains drivers for all I/O types available at the RT-DAC/Zynq board., models and demo applications. See section 3 for the detailed description.

3. RT-DAC/ZYNQ CONTROL WINDOW

The RTDACZynq Simulink diagram contains example application of all board drivers.

3.1 D/A AND A/D

The block contains drivers for the A/D and D/A channels.



3.1.1. MICROZEDDA

The block contains two inputs:

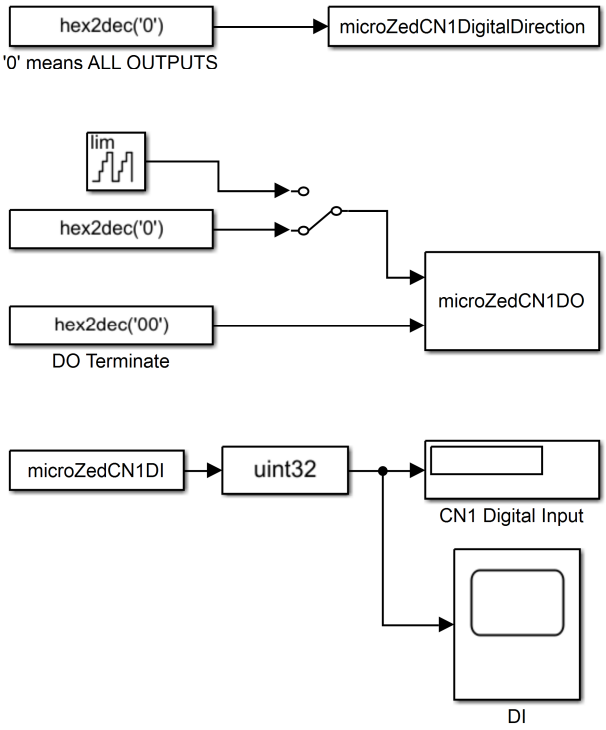
- four-line signal as input to D/A converters. The signals range from -10 to +10. The respective outputs appear at D/A outputs during each sampling period,
- four-line signal which defines the D/A outputs when the real-time application terminates.

3.1.2. MICROZEDAD

The output of the block contains the results of the A/D conversions. The inputs of the block contain default values and shouldn't be changed.

3.2 DIGITAL OUTPUTS

The block contains drivers for the digital inputs and outputs.



3.2.1. MICROZEDCN1DIGITALDIRECTION

The input of the block (26-bit unsigned value) defines the direction of the 26 digital signals. The ‘0’ value defines output and the ‘1’ value defines input at the respective D/IO line.

3.2.2. MICROZEDCN1DO

The block defines the state of the digital outputs. It contains two inputs:

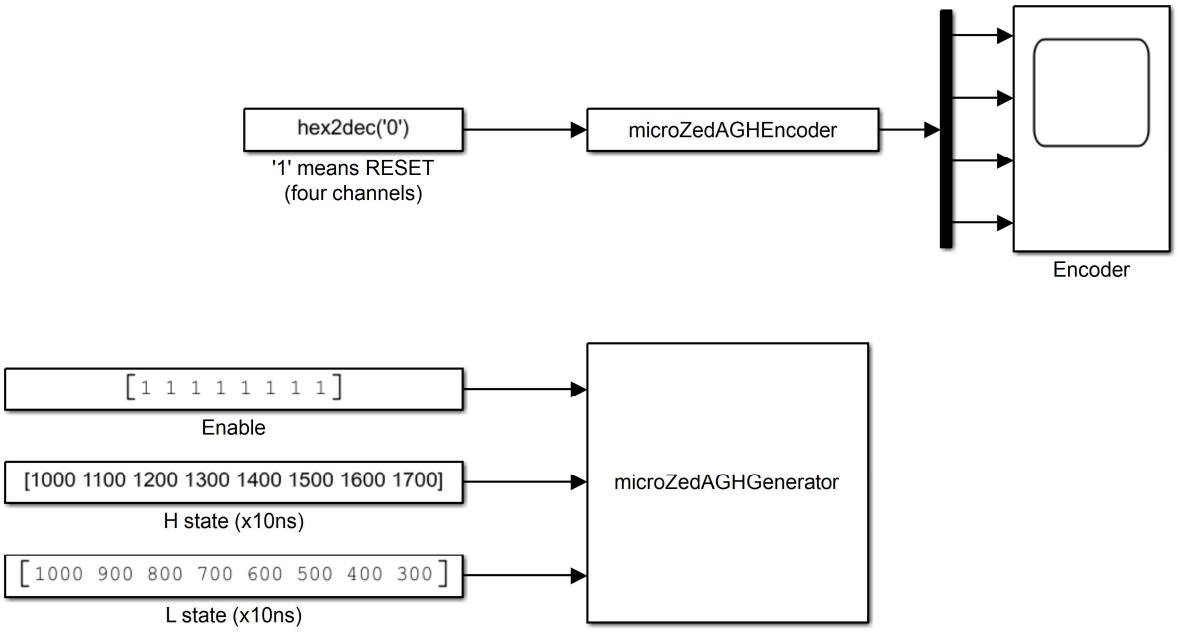
- the 26-bit unsigned value which defines the state of 26 digital outputs. The respective outputs appear at outputs during each sampling period,
- the 26-bit unsigned value which defines the state of the outputs when the real-time application terminates.

3.2.3. MICROZEDCN1DI

The block reads the state of the 26 digital input signals.

3.3 I/Os

The block contains drivers for the functions implemented for the AGH team.



3.3.1. microZedAGHEncoder

The block contains interface to the quadrature incremental encoder channels. The input defines four reset flags for the encoders. The output contains four states of the encoder counters.

3.3.2. microZedAGHGenerator

The block contains interface to the output generator blocks (operate as well as PWM outputs). It contains three inputs:

- Enable – eight signals to enable generators. A generator has to be enabled to observe the wave at the respective output.
- H state (x10ns) - eight signals to define the duration of the ‘1’ state of the generated wave. The resolution of the duration is 10 nanoseconds.
- L state (x10ns) - eight signals to define the duration of the ‘0’ state of the generated wave. The resolution of the duration is 10 nanoseconds.

3.4 REAL-TIME EXPERIMENTS- IMPLEMENTATION

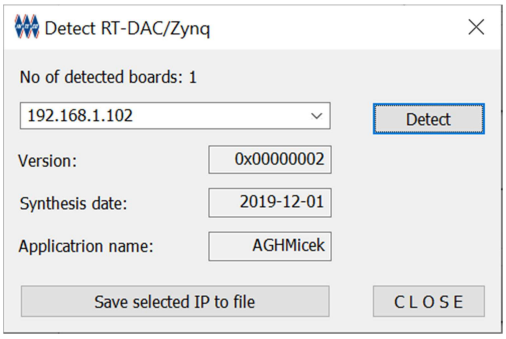
The controller of the RT-DAC/Zynq is running at the MicroZed board. In turn, the development of the controllers is performed in MATLAB/Simulink environment at a PC computer. Therefore there are required some steps to establish connection between the MicroZed and the PC, download the controller code and monitor the execution of the real-time applications. Firstly install the software form provided CD. After successful completion perform steps described below.

1. Establish the connection to the RT-DAC/Zynq

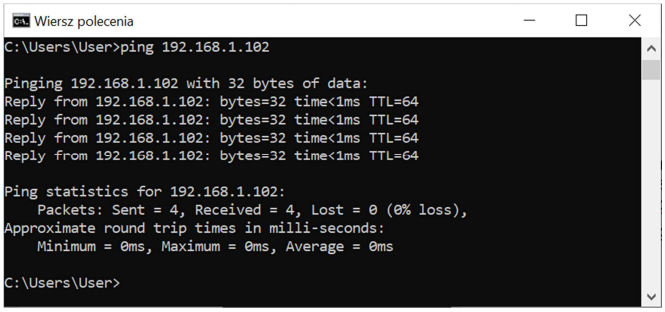
   Connect your PC to the *InTeCo_RT-DAC/Zynq* by Ethernet cable and test the connection. The test can be performed in three ways:

   - Execute the RT-DAC Zynq Detect application. The application is installed in the MATLABROOT\toolbox\RTDACZynq\DetectRTDACZynq folder as the DetectRTDACZynq.exe file. Press the Detect button. The application scans network interfaces to find the controller and at least one controller should be found as given below. The application presents the IP address and the version and synthesis date of the FPGA configuration.

     IP address 192.168.1.102 is the default address of the RT-DAC/Zynq controller.

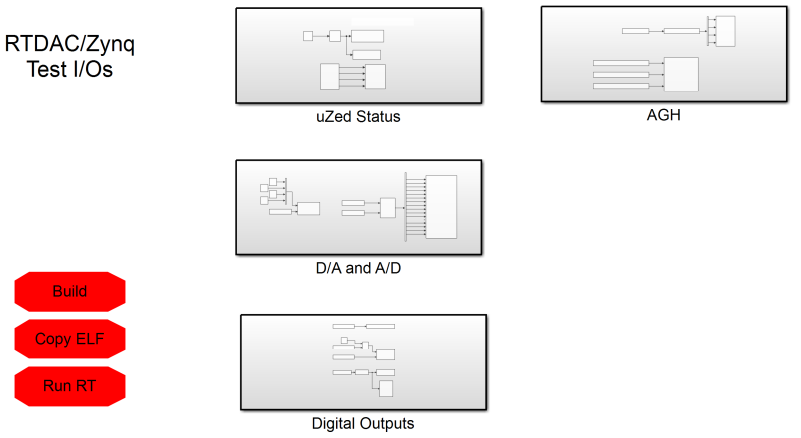


   - It is important to check if the IP of the board is stored in the MATLABROOT\toolbox\RTDACZynq\uZed.ip file. Pressing the *Save selected IP to file* button allows to overwrite the uZed.ip file if needed.
   - At the PC computer, execute the command: *ping 192.168.1.102.* The controller should respond as given below.



   - Use a telnet client software to connect to the address 192.168.1.102. The login is root and the password is root.

2. Build, download, run, monitor and terminate a real-time application. Start MATLAB and type RTDACZynq to start the Simulink diagram. The model contains three red buttons:



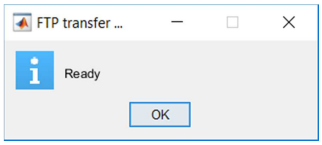
- **Build** – double press the button to build the real-time application. The steps of the build process are presented in the MATLAB command window. The messages:

  `### Created executable  uZSimulink.elf`

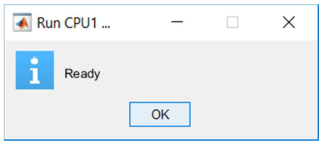
  `### Successful completion of build procedure for model: RTDACZynq`

  indicate success of the build process. The real-time executable file is stored as *`uZSimulink.elf`*.

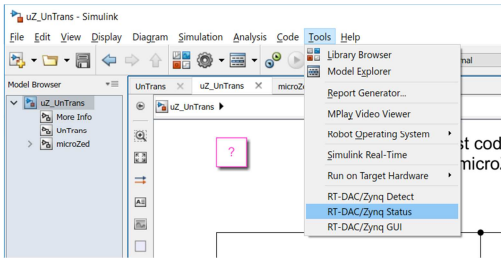
- **Copy ELF** – double press the button to copy the *`uZSimulink.elf`* file to the RT-DAC/Zynq board. When succeeded the following window appears:



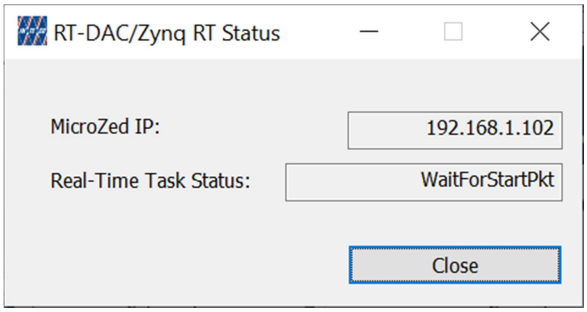
- **Run RT** – double press the button to run the real-time controller. The following window appears:



Run the *Tools/ RT-DAC Zynq Status* Simulink command.



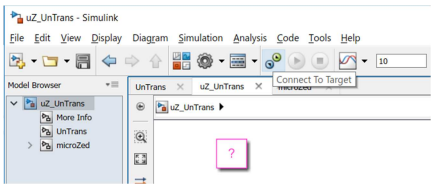
The RT-DAC Zynq Status application displays the status of the real-time application. At the current stage, it is *WaitForStartPkt*. Also the icon of the changes respectively.

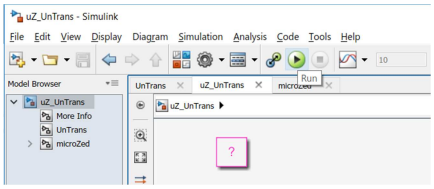


The *WaitForStartPkt* status means that the real-time application performed initializations and is ready for execution but the controller still does not execute in real-time.

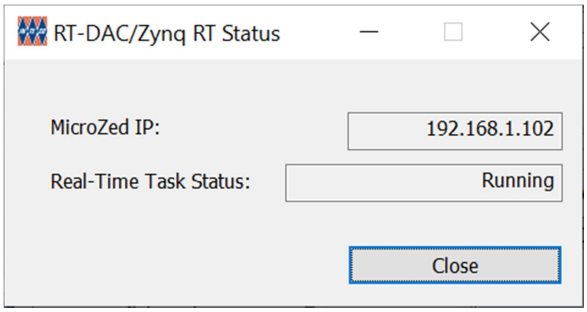
To start the real-time execution a user has to press the *Connect To Target* and *Run* buttons.

It is recommended to place the RT-DAC/Zynq on the floor, motionless, tilted at rest on one of the supports.



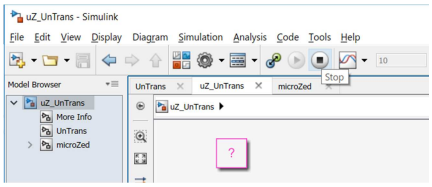


After pressing the *Run* button, the real-time execution begins. The status changes to *Running*.

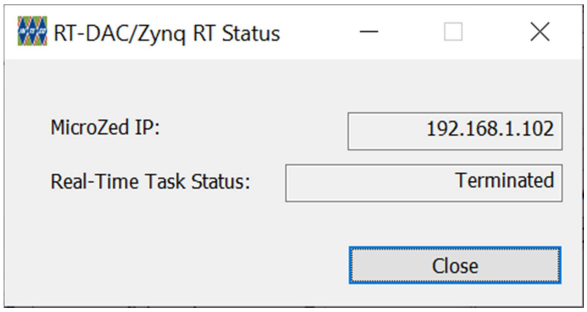


In the *Running* state, a user can monitor the signals using Simulink scopes. In addition, parameters can be changed directly at the Simulink diagram.

When a simulation time elapses or a user presses the *Stop* button, the execution of the real-time controller terminates.



The state changes to *Terminated*.



3.5 TOOLS

There are two tool programs available in the Simulink Tools menu (see Fig.2):

- *RT-DAC Zynq Detect* and
- *RT-DAC Zynq Status*.

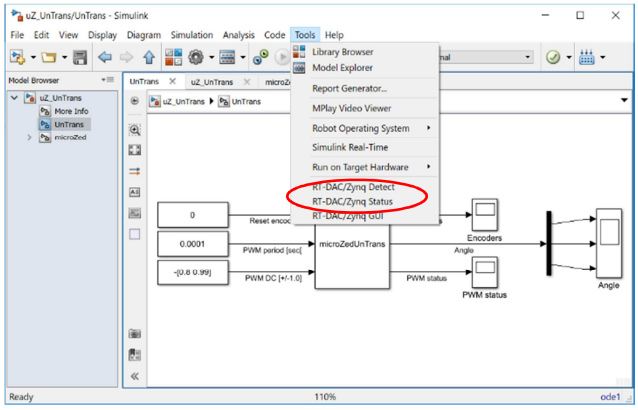
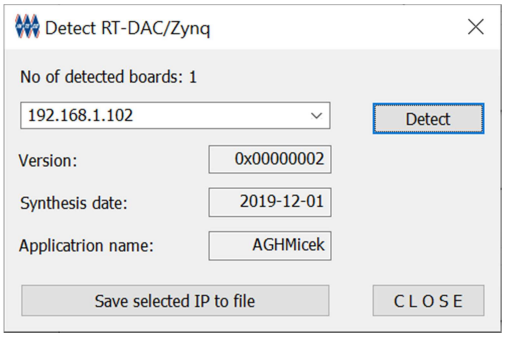


Figure 3.2.Tool programs.

3.5.1. *RT-DAC ZYNQ DETECT*

The application detects all MicroZed boards connected to the computer by cable connection. When the *Detect* button is pressed all network interfaces are scanned. The application presents the number of detected boards and their IP addresses.

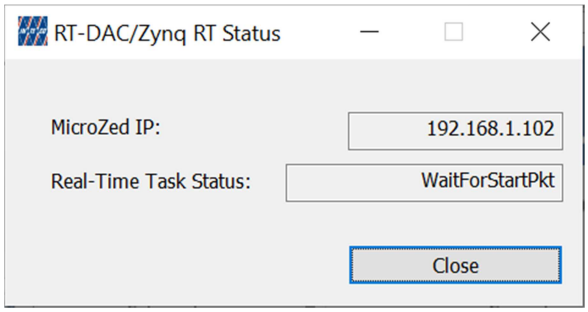


It is important to store the IP of the board we are intend to cooperate with to the MATLABROOT\toolbox\RTDACZynq\uZed.ip file. Pressing the *Save selected IP to file* button allows to overwrite the uZed.ip file. The address from the file is used by all communication functions. **To be sure, that the correct address is stored**

please execute in the MATLAB Command Window the `uZedDefaultIP` command and check if it returns the expected address.

3.5.2. *RT-DAC ZYNQ STATUS*

The application presents the status of the real-time application running at the MicroZed board. The IP address of the board is taken from the MATLABROOT\toolbox\RTDACZynq\uZed.ip file.



The following status messages are presented:

- *Unknown* – the MicroZed does not run any real-time application. Presented immediately after boot of the system or after resetting of the ARM2 processor.
- *Loaded* – real-time application is loaded, but still not initialized. The real-time application performs the initialization procedures in this state, which usually takes a few milliseconds.
- *WaitForStartPkt*– real-time application waits for the Simulink "*Connect To Target*" and "*Run*" commands.
- *Running* – real-time application is running. External communication mode is applied to present the scope signals and to update the parameters.
- *Terminating* – real-time application entered the termination phase. The real-time application executes the *mdlTerminate* functions, disconnect from the host and frees the memory resources. It usually remains a few milliseconds in this state.
- *Terminated* – real-time application is terminated. A new application can be loaded and executed.

The states *Unknown, WaitForStartPkt, Running* and *Terminated* are marked by the different icons of the RT-DAC Zynq Status application. The icons are presented in the table below.

Status	Icon
Unknown	
WaitForStartPkt	
Running	

Terminated	

The icons are presented in the task bar for quick recognition of the status of the real-time application.