



Intregal
Nonlinearit
Error

Value Written to D/A

Application Note

UEISIM UDP monitoring

Gain/Reference Error

Output Voltage

Ideal Output

Value Written to D/A

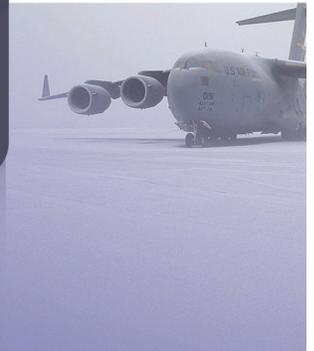
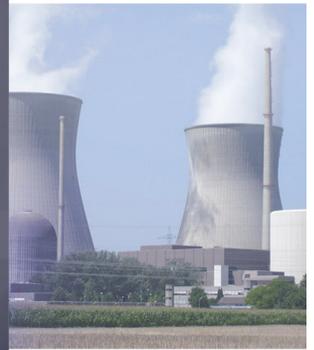
Gain/Reference Error

Output Voltage

Ideal Output

Value Written to D/A

Gain/Reference Error



© Copyright 2014 United Electronic Industries, Inc. All rights reserved.



**United
Electronic
Industries**

Shaping the Future of Computer-Based I/O™

Contents—UEISIM UDP monitoring

- 1 Digital output model..... 1**
 - 1.1 Description1
 - 1.2 Configuration.....2
 - 1.3 Execution.....3
- 2 ARINC-429 model 4**
 - 2.1 Description4
 - 2.2 Configuration.....6
 - 2.3 Execution.....6
- 3 ARINC-429 loopback model 8**
- 4 Synchro/Resolver output model 9**
 - 4.1 Description9
 - 4.2 Configuration.....10
 - 4.3 Execution.....10
- 5 CAN model 11**
 - 5.1 Description11
 - 5.2 Configuration.....12
 - 5.3 Execution.....13

Application Note: UEISIM UDP monitoring

This document details remote monitoring of Simulink models running on UEISIM via UDP.

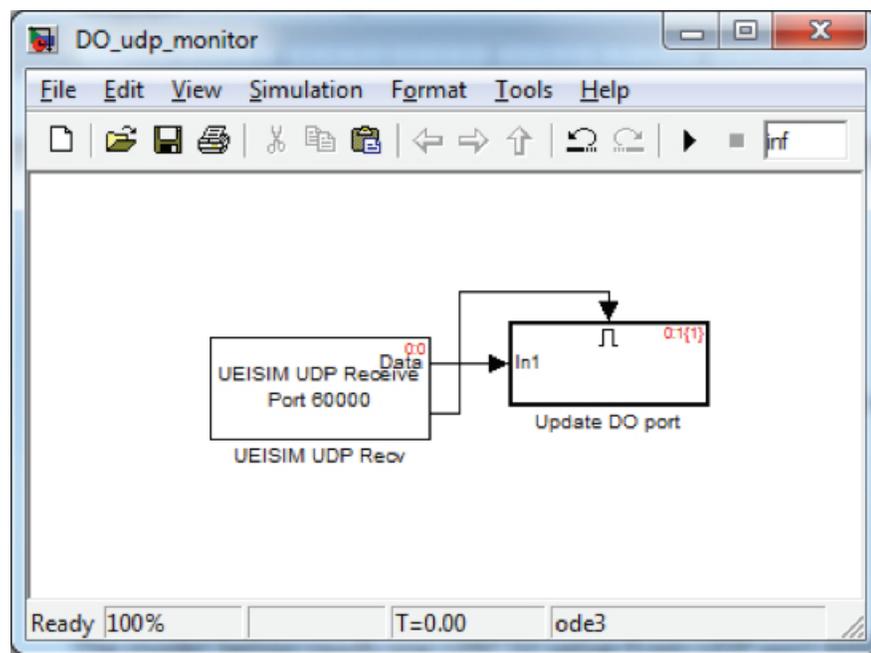
1 Digital output model

1.1 Description

The DO_udp_monitor.mdl model reads one UINT32 value from a UDP port 60000 and updates the DO port

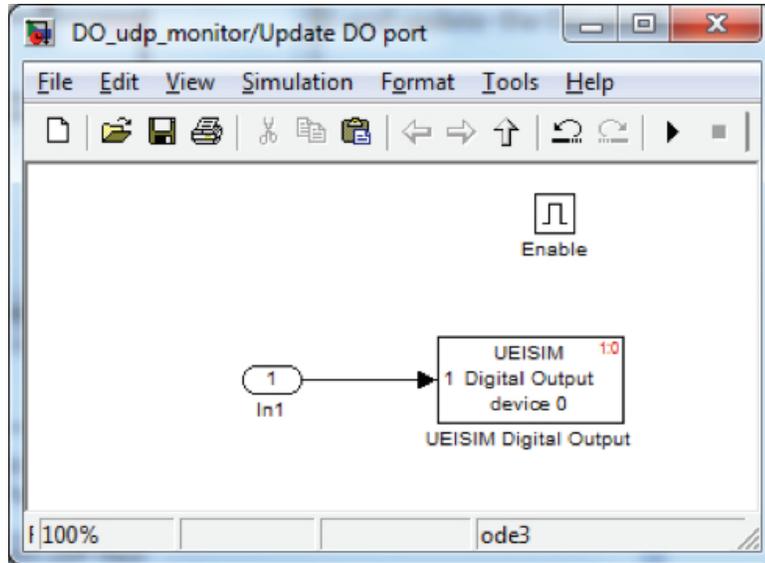
Each of the 32 output lines on the DO port is controlled by the corresponding bit of the UINT32 value.

For example, receiving 0x3 will close DO outputs 0 and 1 and leave outputs 2 to 31 opened.



Note that the status port of “UDP Receive” block is wired to the enable input of an “enabled subsystem” block. This makes sure that DO port is updated only when a new UDP packet is received.

The Enabled subsystem block contains the actual “UEISIM DO Write” block.



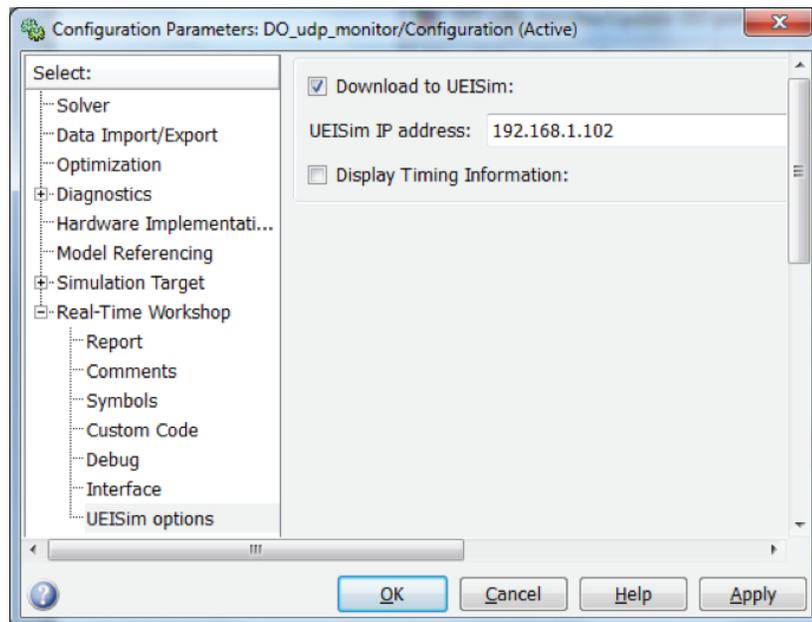
The blocks in this model can be duplicated to control more than one DIO layer.

1.2 Configuration

Double-click on the “UEISIM UDP Receive” block to configure the UDP port you wish to assign to write to the DO port.

Double-click on the “enabled subsystem” block and then double-click again on the “UEISIM DO write” block to configure the layer ID (0 is left-most slot on racks or top layer on cubes).

Select the menu “Simulation/Configuration Parameters...”; click on “Real-Time Workshop/UEISIM options”; select “Download to UEISIM” and enter the UEISIM’s IP address

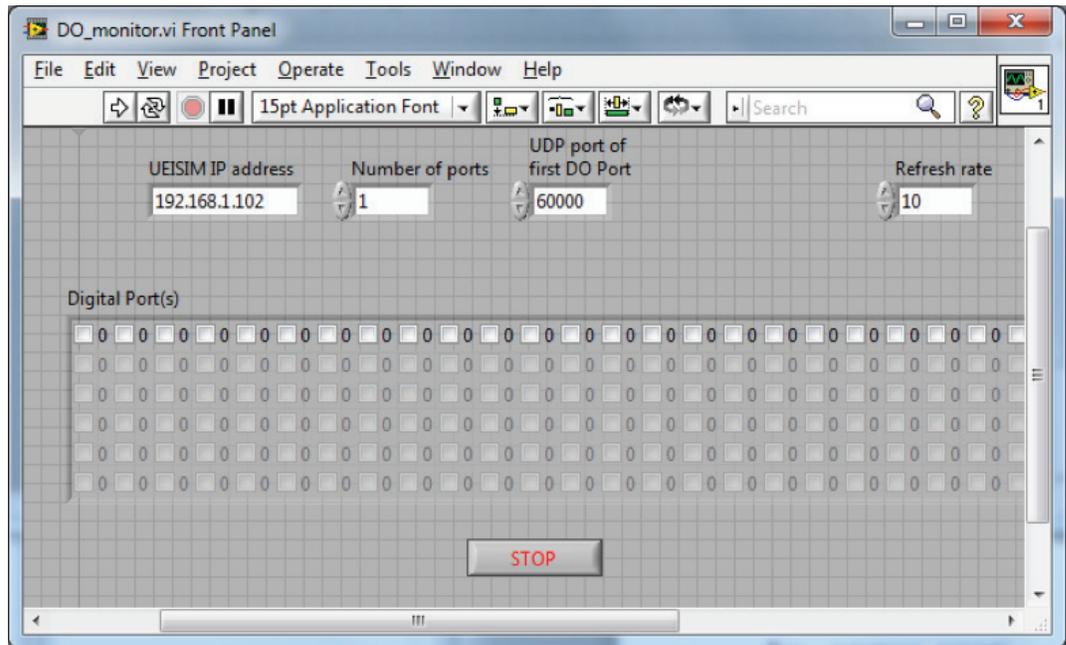


1.3 Execution

Build the model and wait until it is downloaded to the target. Open a telnet or SSH session with UEISIM (user:root, password:root) and run the model: `"/tmp/DO_udp_monitor"`.

You can now send UDP packets from the host PC to set the DO lines state. The UDP packet's payload is one UINT32 coded in big-endian format. Each bit of the UINT32 toggles the corresponding DO line.

The LabVIEW VI "DO_monitor.vi" can be used to send UDP packets from the host PC:



- UEISIM IP address: The IP address of the UEISIM
- Number of ports: The number of DO ports to write to. Each DO port is associated with a different UDP port
- UDP port of first DO port: The UDP port associated with the first DO port. The VI assumes that the model assign contiguous UDP ports to each DO port.
- Refresh rate: The cycle time
- Digital Port(s): bit 31 is controlled by the leftmost check box, bit 0 is control by the rightmost check box.

Run the VI and click on the various check box to toggle the corresponding bit on the DO port.

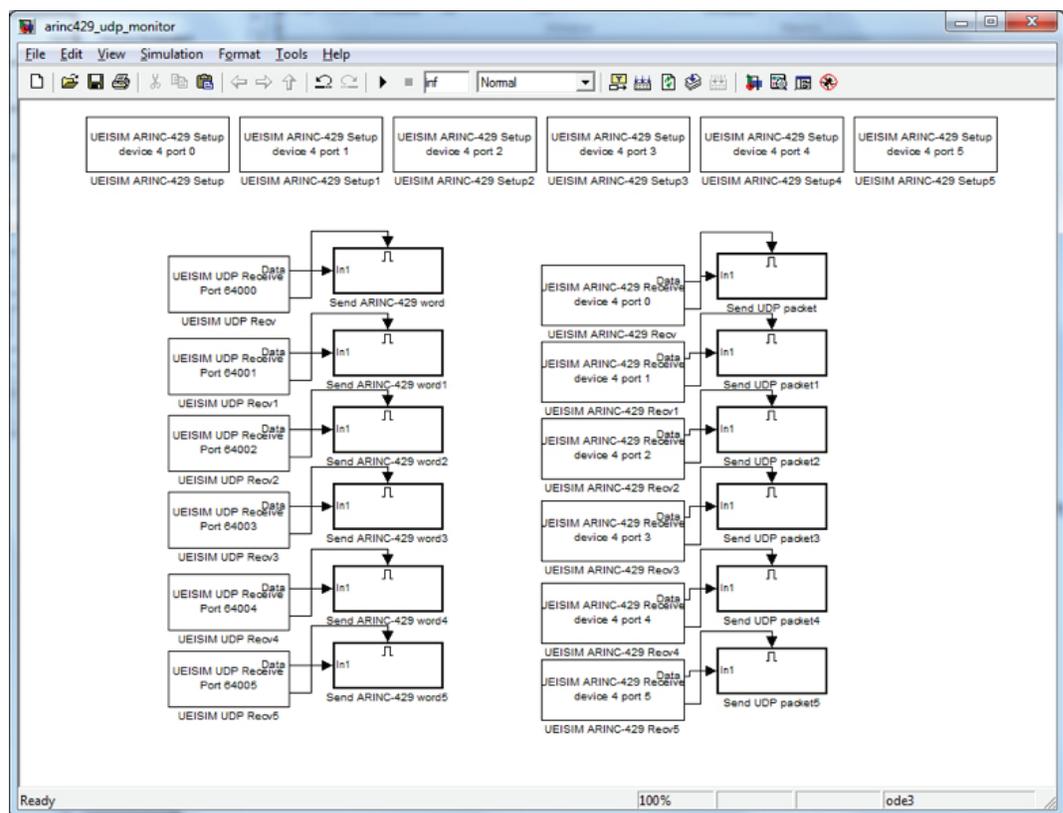
2 ARINC-429 model

2.1 Description

The arinc429_udp_monitor model reads UINT32s from 6 UDP ports and forwards them to 6 ARINC-429 TX ports. It also reads UINT32s from 6 ARINC-429 RX ports and forwards them to 6 UDP ports.

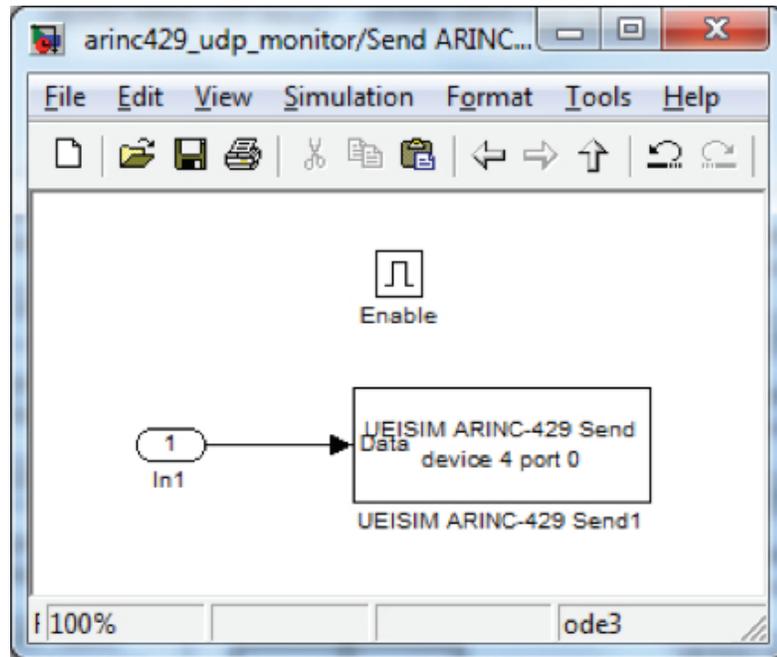
Each UINT32 contains the various ARINC fields according to the table below.

32	31	30	29	11	10	9	8	1
P	SSM		Data			SDI		Label

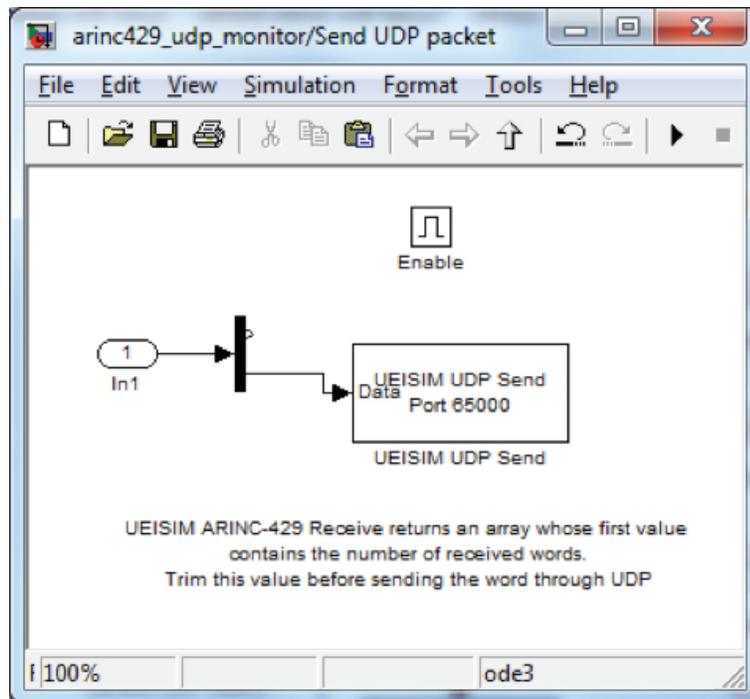


There are 6 “UEISIM ARINC-429 Setup” blocks to configure the 6 TX and RX ports on a DNA-429-566.

Each “UEISIM UDP Receive” block is connected to an “Enabled subsystem” block containing a “UEISIM ARINC-429 block”



Each “UEISIM-ARINC-429-Receive” block is connected to an “Enabled Subsystem” block containing a “UEISIM UDP Send” block.



You can duplicate those subsystems to interface additional ARINC-429 ports located on other ARINC-429 layers.

2.2 Configuration

Double-click on each “UEISIM ARINC-429 Setup” block to configure the layer ID (0 is left-most slot on racks or top layer on cubes), port ID and bits/sec setting.

Double-click on each “UEISIM UDP Receive” block to configure the UDP port you wish to assign to write to each ARINC-429 TX port.

Double-click on each “Send ARINC429 Word” subsystem and then double-click again on the “UEISIM ARINC-429 Send” block to configure the layer ID and port ID.

Double-click on each “UEISIM ARINC-429 Receive” block to configure the layer ID and port ID.

Double-click on each “Send UDP packet” subsystem and then double-click again on the “UEISIM UDP Send” block to configure the UDP port assigned to each ARINC-429 RX port and the IP address of the host PC.

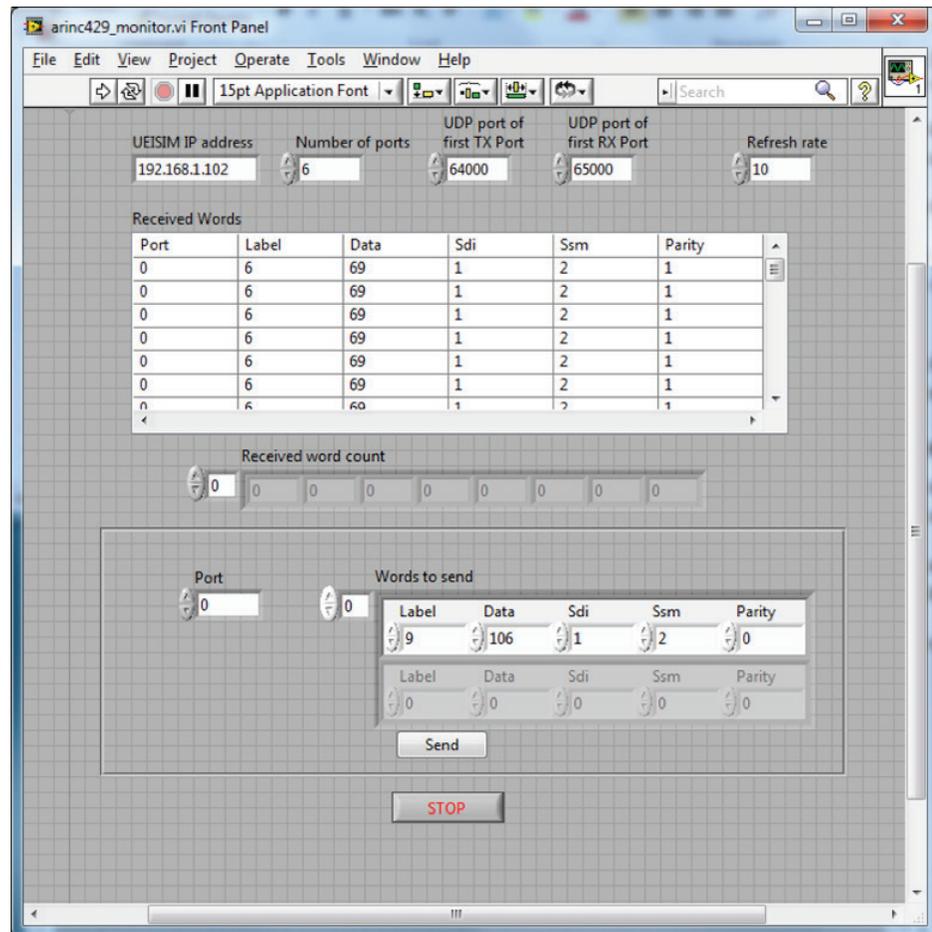
Select the menu “Simulation/Configuration Parameters...”, click on “Real-Time Workshop/UEISIM options”, select “Download to UEISIM” and enter the UEISIM’s IP address

2.3 Execution

Build the model and wait until it is downloaded to the target. Open a telnet or SSH session with UEISIM (user:root, password:root) and run the model: “/tmp/arinc429_udp_monitor”.

You can now send UDP packets from the host PC to write ARINC-429 words. The UDP packet’s payload is one UINT32 coded in big-endian format.

The LabVIEW VI “arinc429_monitor.vi” can be used to send/receive UDP packets:

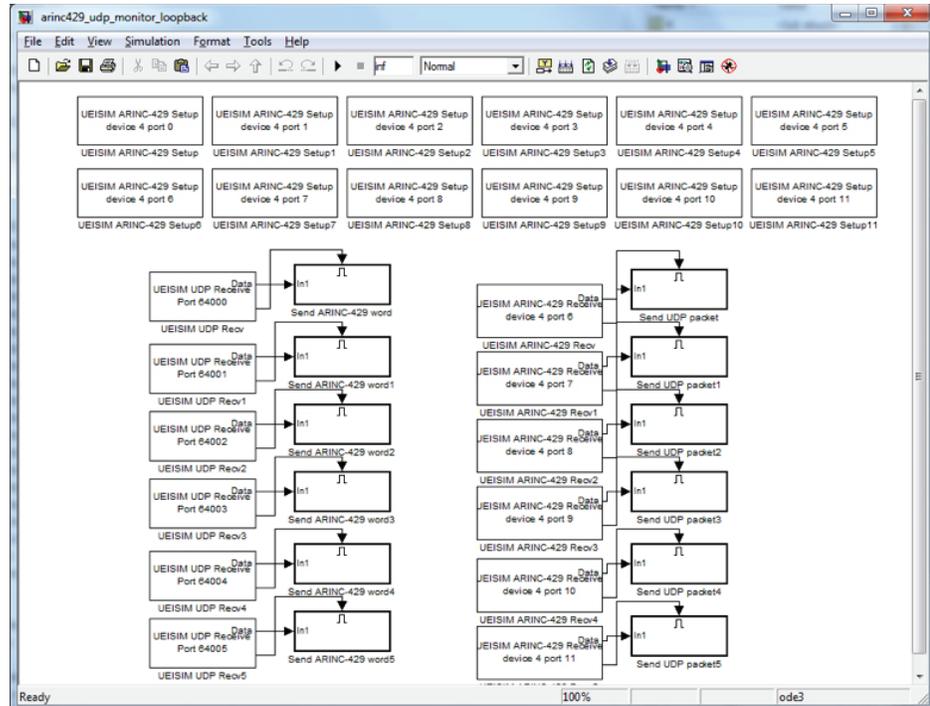


- UEISIM IP address: The IP address of the UEISIM
- Number of ports: The number of ARINC-429 TX ports to write to and number of ARINC-429 RX ports to read from. Each TX and RX port is associated with a different UDP port
- UDP port of first TX port: The UDP port associated with the first ARINC-429 TX port. The VI assumes that the model assign contiguous UDP ports to each TX port.
- UDP port of first RX port: The UDP port associated with the first ARINC-429 RX port. The VI assumes that the model assign contiguous UDP ports to each RX port.
- Refresh rate: Defines how often UDP packets are sent/received, value is in ms.
- Received Words: Displays incoming ARINC-429 words from all ports.
- Received word count: keep track of the total number of words received on each port since VI was started.
- Port: The index of the TX port to write to.
- Words to send: The ARINC-429 word to send
- Send: Each click on this button triggers the transmission of one ARINC-429 word to the selected TX port.

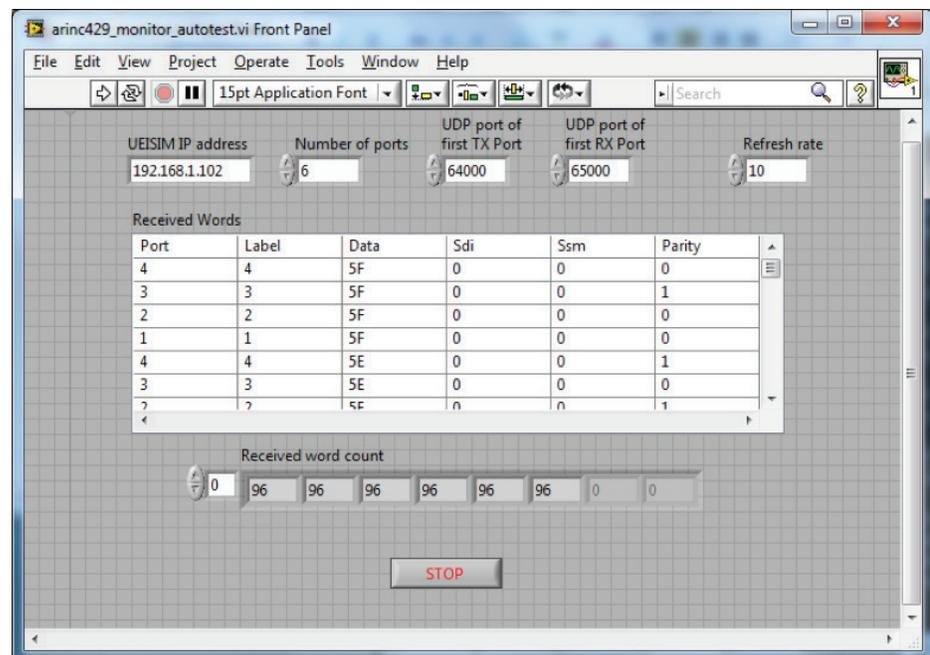
3 ARINC-429 loopback model

The DNA-429-566 layer is equipped with 6 internal RX loopback ports (6 to 11). Every word sent out of TX port N is received on RX port N+6.

The “arinc429_udp_monitor_loopback.mdl” model takes advantage of those loopback ports. It configures 6 TX ports 0 to 5 and 6 RX ports 6 to 11.



Use the VI “arinc429_monitor_autotest.vi” to send data out of the TX ports and receive it back from the RX loopback ports.

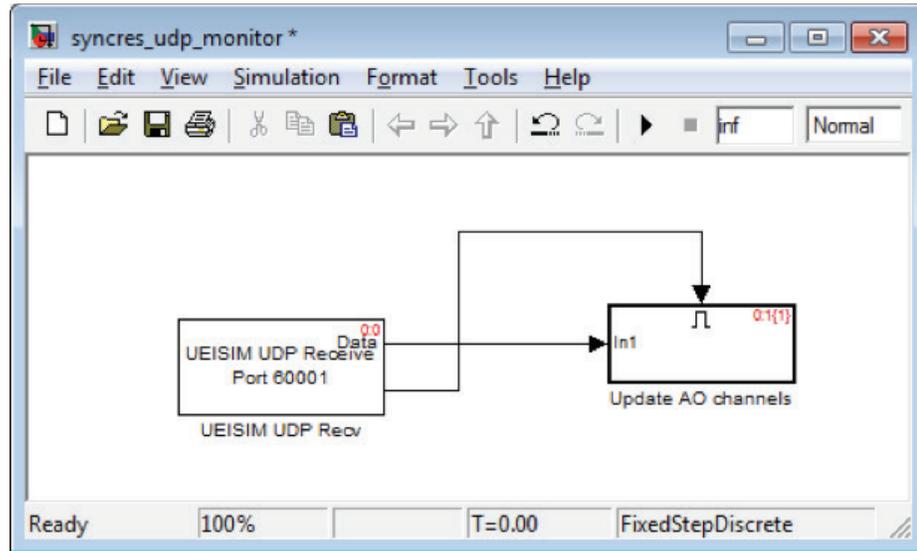


4 Synchro/Resolver output model

4.1 Description

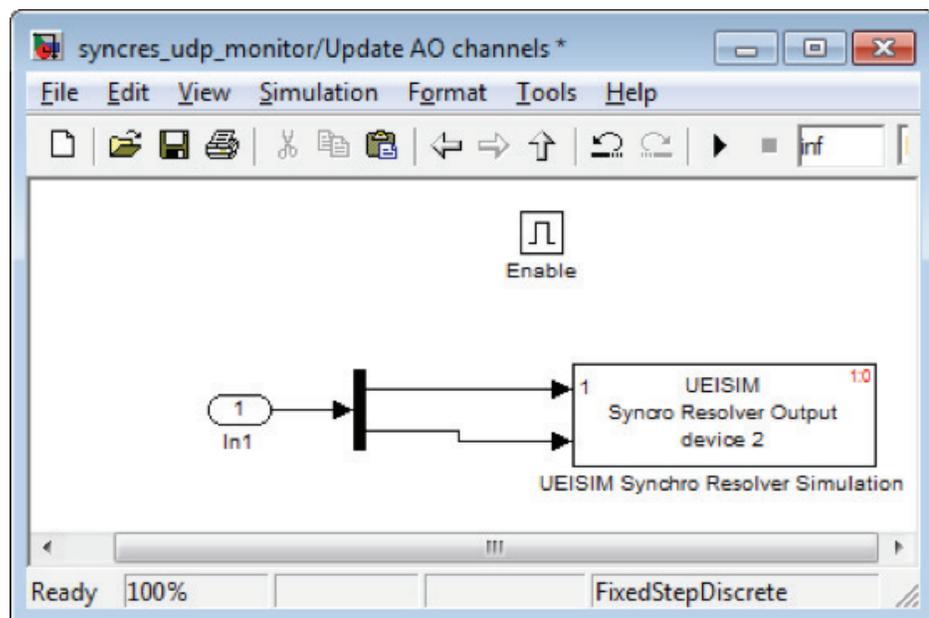
The `synres_udp_monitor.mdl` model reads two DOUBLE values from a UDP port 60001 and updates the two synchro/resolver output channels on AI-255

The two DOUBLE values specify an angle in radians.



Note that the status port of “UDP Receive” block is wired to the enable input of an “enabled subsystem” block. This makes sure that AO channels are updated only when a new UDP packet is received.

The Enabled subsystem block contains the actual “UEISIM Synchro Resolver Output” block



The blocks in this model can be duplicated to control more than one DIO layer.

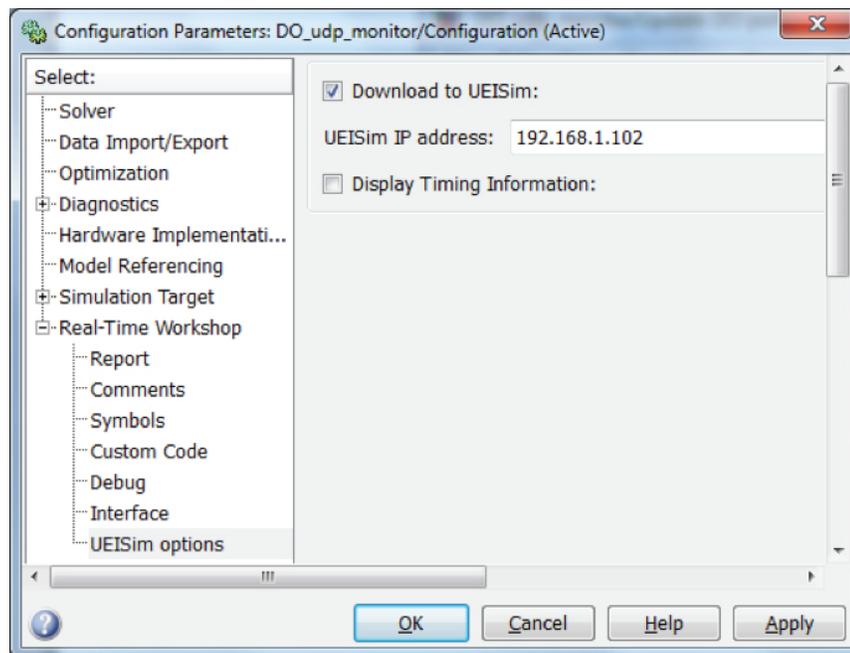
4.2 Configuration

Double-click on the “UEISIM UDP Receive” block to configure the UDP port you wish to assign to write to the synchro resolver channels.

Double-click on the “enabled subsystem” block and then double-click again on the “UEISIM Synchro Resolver Output” block to configure the output settings:

- layer ID (0 is left-most slot on racks or top layer on cubes)
- Mode vector: Specifies whether each channel is simulating a Synchro or a Resolver
- Excitation Voltage vector: The excitation voltage used to power Synchro/Resolver(s) simulated by each channel
- Excitation Frequency vector: The excitation frequency used to power Synchro/Resolver(s) simulated by each channel
- External Excitation vector: Specifies whether channel(s) provide excitation or whether excitation is supplied externally
- Channel vector: Array of channels to simulate from

Select the menu “Simulation/Configuration Parameters...”, click on “Real-Time Workshop/UEISIM options”, select “Download to UEISIM” and enter the UEISIM’s IP address



4.3 Execution

Build the model and wait until it is downloaded to the target. Open a telnet or SSH session with UEISIM (user:root, password:root) and run the model: “/tmp/synres_udp_monitor”.

You can now send UDP packets from the host PC to set the two simulated synchro/resolver angular position.

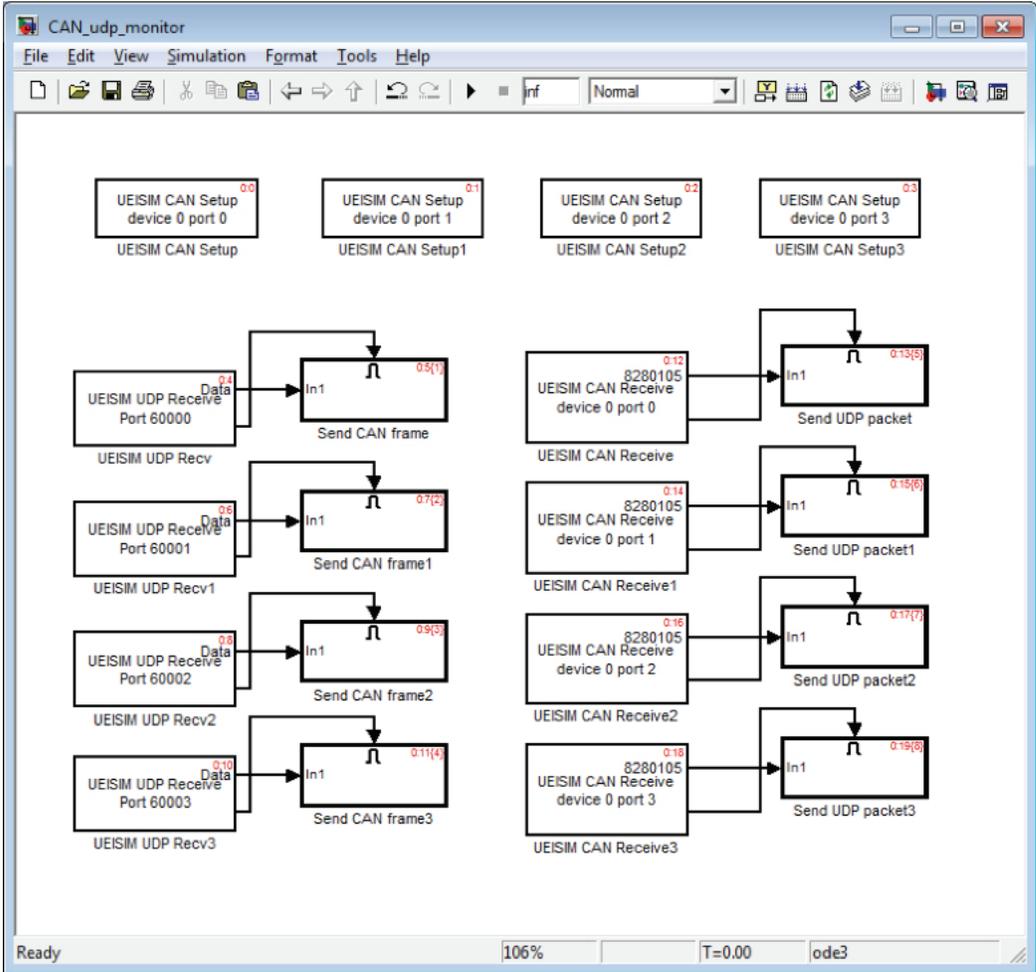
The UDP packet’s payload is two DOUBLE coded in big-endian format

5 CAN model

5.1 Description

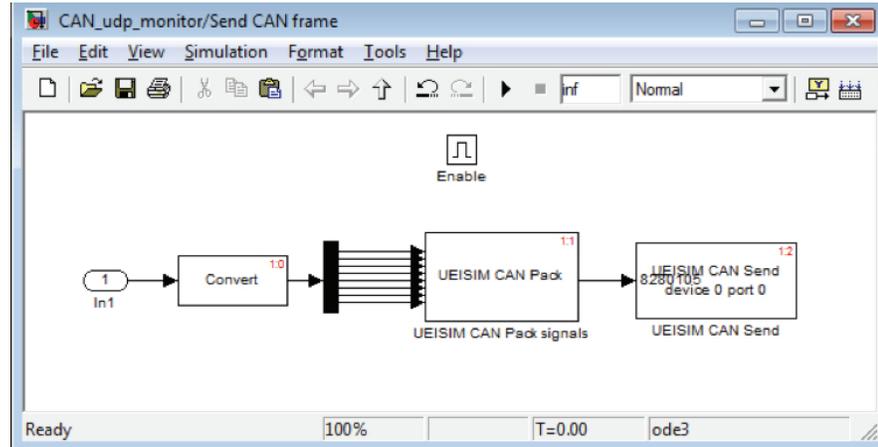
The can_udp_monitor model reads 8 x UINT8s from one UDP port for each CAN arbitration ID and forwards them to the matching CAN port. It also reads CAN frames from all 4 CAN ports and forwards them to UDP port when they match a given arbitration ID.

The 8 UINT8s values are the payload of the CAN frame.

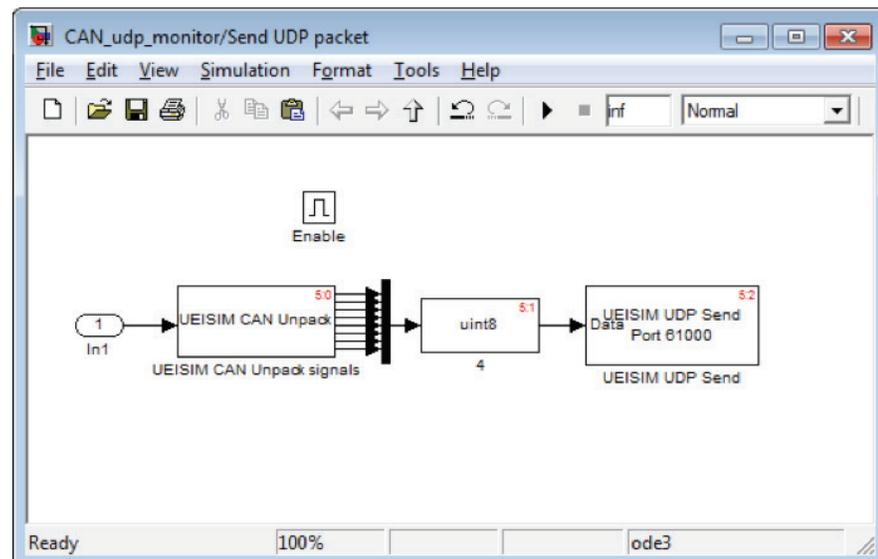


There are 4 “UEISIM CAN Setup” blocks to configure the 4 CAN ports on a DNA-CAN-503. Each “UEISIM UDP Receive” block is connected to an “Enabled subsystem”

The subsystem splits the UINT8 signals into 8 values and packs those values in the CAN frame payload with “UEISIM CAN Pack” block. The CAN frame is sent with the specified arbitration ID using “UEISIM CAN Send” block.



Each “UEISIM-CAN-Receive” block is also connected to an “Enabled Subsystem” block



The subsystem unpacks the CAN frame payload as 8 UINT8 values. Those values are muxed to form a UINT8 vector which is sent using “UEISIM UDP Send” block.

You can duplicate those subsystems to interface additional frame IDs.

5.2 Configuration

Double-click on each “UEISIM CAN Setup” block to configure the layer ID (0 is left-most slot on racks or top layer on cubes), port ID and bits/sec setting.

Double-click on each “UEISIM UDP Receive” block to configure the UDP port you wish to assign to write to each CAN arbitration ID.

Double-click on each “Send CAN Frame” subsystem and then double-click again on the “UEISIM CAN Send” block to configure the layer ID, port ID and arbitration ID.

Double-click on each “UEISIM CAN Receive” block to configure the layer ID, port ID and arbitration ID.

Double-click on each “Send UDP packet” subsystem and then double-click again on the “UEISIM

UDP Send” block to configure the UDP port assigned to each arbitration ID port and the IP address of the host PC.

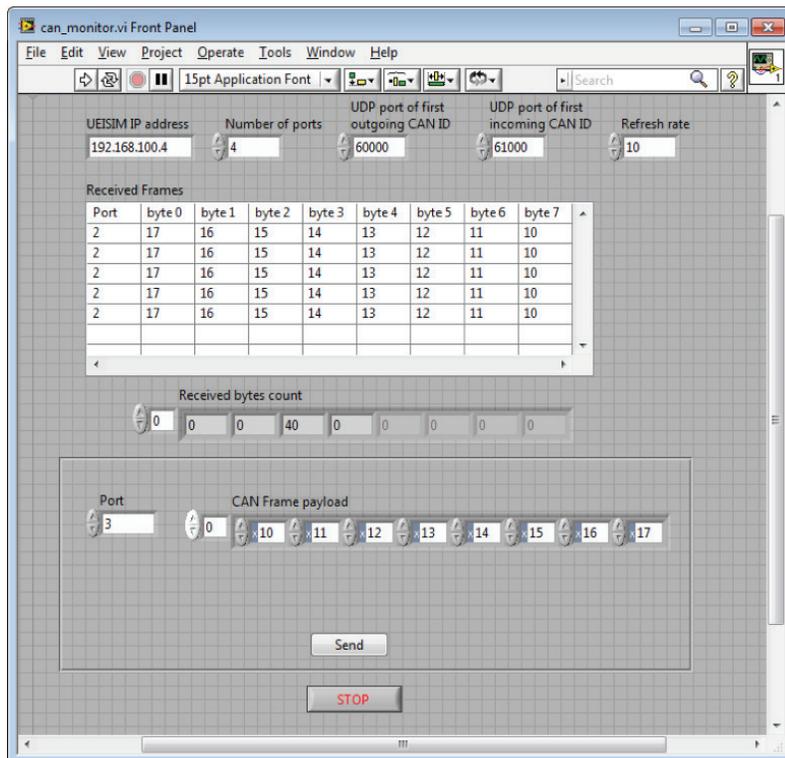
Select the menu “Simulation/Configuration Parameters...”; click on “Real-Time Workshop/UEISIM options”; select “Download to UEISIM” and enter the UEISIM’s IP address

5.3 Execution

Build the model and wait until it is downloaded to the target. Open a telnet or SSH session with UEISIM (user:root, password:root) and run the model: “/tmp/can_udp_monitor”.

You can now send UDP packets from the host PC to write CAN frames. The UDP packet’s payload is 8 UINT8s.

The LabVIEW VI “can_monitor.vi” can be used to send/receive UDP packets



- UEISIM IP address: The IP address of the UEISIM
- Number of ports: The number of arbitration IDs to write and read.
- UDP port of first outgoing frame: The UDP port associated with the first outgoing CAN ID. The VI assumes that the model assigns contiguous UDP ports to each outgoing CAN IDs.
- UDP port of first incoming frame: The UDP port associated with the first incoming CAN ID. The VI assumes that the model assigns contiguous UDP ports to each incoming CAN IDs.
- Refresh rate: Defines how often UDP packets are sent/received, value is in ms.
- Received Frames: Displays incoming CAN frames from all ports.
- Received byte count: keep track of the total number of bytes received on each UDP port since VI was started.
- Port: The index of the CAN ID to write to.
- CAN frame payload: The CAN frame payload to send along the selected CAN ID.
- Send: Each click on this button triggers the transmission of one CAN frame to the selected ID.