Control Network Emulation Platform Software Package Description

The software package developed by Sandia National Laboratories is intended to allow the integration of Simulink models into emulations of control networks. To accomplish this, three programs are included:

- Simulink S-Function
- Data Broker
- End Point

Requirements

The Data Broker and S-Function can only be used on Linux systems due to their reliance on the inter-process communications (IPC). The Data Broker and Simulink model must be co-located on the same virtual machine. The End Point is Python based and can be run on any operating system that can run Python. The Data Broker and each End Point must have clear TCP/IP and UDP lines of communication and be able to ping each other. The End Points will need 1 CPU core with 2 threads, and approximately 100mb RAM. The Data Broker and S-Function's computational requirements will be dependent on the simulation they are attached to. Each program's dependencies are listed below.

Data Broker

- Linux OS (Ubuntu recommended)
- ZMQ C code library

Simulink S-Function

- Linux OS (Ubuntu recommended)
- Matlab Simulink (for compilation)

End Point

- Python 3.x
- ZMQ python library

Configuration

The Simulink S-Function is configured with the tags as an input parameter which can be set under the S-Function parameters. The tags are input following the convention:

'Input_Tag_1;Input_Tag_2','Output_Tag_1;Output_Tag_2'

Input refers to data going into the S-Function to be distributed across the network, and Output refers to data retrieved from the network and reported out of the S-Function. The S-Function will use these tags to automatically configure the number of inputs and outputs, and it will

report these to the Data Broker to configure the network communication handling. These tags will be used to call and report data from the End Point to the Data Broker.

The Data Broker reads an input .JSON file named “input.json” that must be co-located with the Data Broker. This must specify the IP of the machine each End Point is located on, the IP of the PLC the End Point will communicate with, the scan time of the PLC, the tag names of data reported (SensorNames) and retrieved (ActuatorNames) and their associated PLC memory register. This file will also set the End Points to be either be sensors, actuators, or both. This JSON file will be used by the Data Broker to configure the End Points on the network. An example file will be included in the distribution.

The Data Broker is also responsible for controlling the Simulink simulation which must be compiled using Simulink coder. The compiled file must be located with the Data Broker. The name of the compiled Simulink model must match that in the Sim_Control.c file, line 44. The Data broker can be compiled by running the included “compile.sh” script.

The S-Function may be copied or drag and dropped into the model intended for use. The GOTO names must match the inputs and outputs desired. Included is a switch function on each output, this allows internal controllers to be used if external ones for that function are not connected. These are optional but can allow the user greater configurability without recompiling the simulation. If it is necessary to recompile the S-Function, Simulink mex should be used as “mex sfun_connector.c -lrt”. To compile the Simulink model with the S-Function included use the below settings for the Simulink Coder:

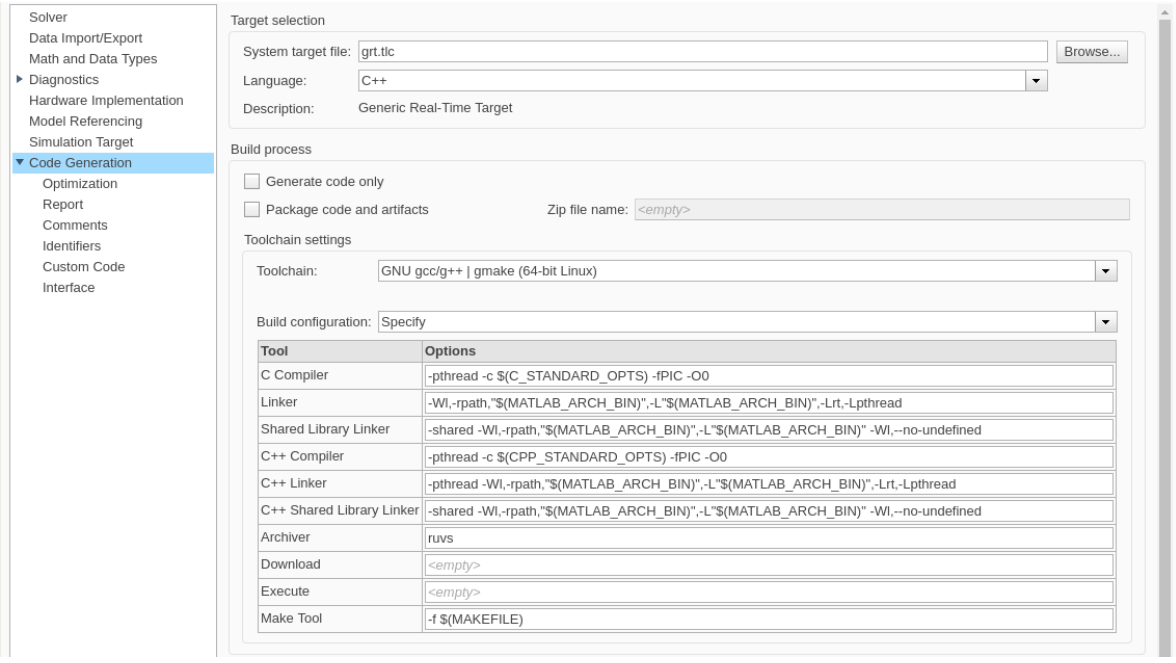

Figure 1: Simulink coder settings.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

When setting up the control network, the End Points and PLC's should be started first, then the Data Broker should start. This will ensure that the system starts up correctly. It is recommended to use two emulated networks. One for normal plant communications, and a second for only the data communications of the Data Broker. Though this system will work across a single network, so long as UDP and TCP/IP communications can be established between the Data Broker and End Points.

Functional description

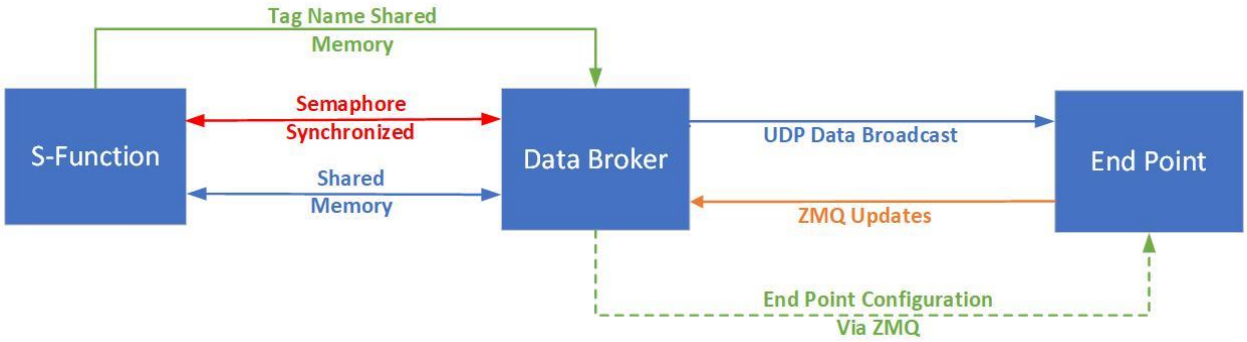

Figure 2: Functional diagram of system

Initialization of the system runs through the following automatic procedures:

1. The Data Broker starts the compiled simulation
2. The S-Function communicates Tag names to the Data Broker
3. The Data Broker holds the simulation and begins End Point configuration
4. The Data Broker reads the JSON input file
5. The Data Broker establishes communication with each End Point and transfers its configuration.
6. The End Point reads configuration data and responds as ready to the Data Broker
7. The Data Broker confirms ready response from all End Points listed in JSON input file and begins real time synchronization of simulation.

The communication path functions as follows:

1. The simulation S-Function reports data values to the Data Broker via shared memory
2. The Data Broker broadcasts this new data via UDP to the End Points
3. End Points read the UDP broadcast and send the correct data to the PLC they are connected with.
4. The End Point monitors the PLC for an updated actuation value and retrieve the value when updated.
5. The End Point send the updated value to the Data Broker via ZMQ messaging.
6. The Data Broker reads this message and reports it to the S-Function on its next timestep
7. The S-Function reads in actuation values from the shared memory and exports these to the model.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.