



HUNT ENGINEERING
Chestnut Court, Burton Row,
Brent Knoll, Somerset, TA9 4BP, UK
Tel: (+44) (0)1278 760188,
Fax: (+44) (0)1278 760199,
Email: sales@hunteng.co.uk
<http://www.hunteng.co.uk>
<http://www.hunt-dsp.com>



Configuring HEART, statically or dynamically.

Rev 1.0 P.Warnes 24-10-03

HEART is a real time communications architecture used on HERON module carriers. The connections are software programmable, and after they are programmed they guarantee performance. It's like making a phone call, first you establish the connection and then you can use that connection when you want. If your system does not send data, the connection remains there with the bandwidth available for your use.

Normal use is to make the connections before your system starts to run, but in fact individual connections can be changed at any time.

This document discusses the various ways that HEART can be controlled – either as a one off operation or as a re-configurable architecture.

History

Rev 1.0 First written

HEART details

Normal users of HEART do not need to understand the details of the architecture. However there is a technology document called “Technical Description of HEART” that discusses the details if you want to see them. Parts of this document assume that you have at least cursory knowledge of the architecture, and dynamically re-configuring the system will require you to read that document too.

HEART control registers

The HEART architecture is configured by setting registers both at the sending end and the receiving end of each connection. It is the combination of setting a FIFO to transmit on a particular timeslot, and having another FIFO configured to receive data from that same timeslot that makes the connection. Setting further FIFOs to receive data from the same timeslot makes the connection a multi-cast connection. Setting a downstream FIFO to transmit into the same timeslot breaks that connection and starts a new one.

So any system configuration is simply a set of register settings which are made at each node of the network.

Heron Serial Bus

HERON systems have a multi-mastered serial bus connected to all of the nodes. This is called HERON Serial Bus (HSB) details of which can be found in another technology document.

The FPGA devices that are used to implement HEART have HSB connected to them. This is how the registers that make the connections are programmed.

Because HSB is multi-mastered, these messages can come from the Host PC, or any module that has an HSB connection.

Host PC

Most HERON systems are plugged into a PC, where the system software will be developed. Even systems that will ultimately be embedded can start out this way due to the way that the HERON module carriers have been designed to operate in a stand alone mode as well as a peripheral to a host computer.

Hence the normal way to configure your HEART system is from the Host PC.

In this method, the HUNT ENGINEERING network file can be used to define the connections you need. The format of that file is documented separately, and is used as a common format for several of the HUNT ENGINEERING system software tools.

The HEARTConf tool takes the connections requested in that file and translates them into the required HEART register settings. Finally that tool builds HSB messages to set those registers and sends them to the hardware for you.

Again this tool is documented separately, but it has several options. It can be run as many times as you like, and **MUST** be run after each system reset because that reset disconnects the previous HEART connections.

FPGA module

An FPGA module has a connection to HSB, made via the control FPGA. The Hardware Interface Layer VHDL provided for the module includes an interface to allow you to control the HSB from your FPGA design. There is a separate document that discusses the details of how to use that, but combined with an option of HEARTConf that will generate you some VHDL that contains the right messages to connect the system as per your network file, it is very easy to use.

In this way you can use an FPGA module to configure your HEART connections. This will be necessary in a stand alone system where there is no Host PC.

C6000 module

A C6000 module has a connection to HSB, made via some memory mapped peripheral. The HERON-API software provided for the module includes functions that allow you to send and receive HSB messages from your C6000 program. There is a separate document that discusses the details of how to use that, but combined with an option of HEARTConf that will generate a C function that contains the right messages to connect the system as per your network file, it is very easy to use.

In this way you can use a C6000 module to configure your HEART connections. This will be necessary in a stand alone system where there is no Host PC.

Don't use the connections before they are made!

When you use the conventional method of configuring your HEART network, the Software tools ensure that C6000 modules do not start to access FIFO connections until after the configuration stage.

There is no such automatic method employed for FPGA modules. These may start to access the FIFO connections as soon as the FPGA is configured. During the connection of the HEART FIFOs some of that data may be lost. If your application is a data streaming one, this probably won't concern you, but if the loss of that data at the beginning of your system running is a problem to you then you need to design your system to prevent it.

There are several methods you could use to do that, i.e an HSB message must be received from the node that makes the connections, before any data is sent, or the use of a UMI line to indicate when the HEART connections are made, or perhaps the Config line.

Anyway, this could result in data being lost at the initialisation stage of your system, and can be prevented if this is a problem to you.

Of course embedded systems need to care about the same issue.

Dynamic re-configuration

There is nothing in the HEART architecture that prevents you from changing the HEART connections at any time while your system is running.

The only thing that may prevent you from doing so, is that if you re-configure a connection while it is active, the content of the data on both the old and new connections cannot be defined. Data may be lost or gained in this process. However it would be a fairly simple exercise to design your system to stop using the connection before re-configuring it.

Because of this concern it is probably best not to re-send all of the connection messages, but merely the HSB messages that are necessary to disconnect the old connection and connect the new one. In most cases you only need to change the register at one end of the connection.

It is up to you to decide which node in the system sends the messages to perform the re-configuration, as they can originate from the Host, an FPGA module or a C6000 module. It is not even necessary to use the same node that performed the original configuration, i.e. it would be possible to start a system running using a connectivity programmed using HEARTConf running on the host and have individual modules modify those connections during system operation.

One thing to be careful of in dynamically re-configuring your system, is that normal use of HEARTConf allows that tool to choose the timeslots to use for each connection. The timeslot chosen may vary when you change the network file, or even the version of the HEARTConf tool. When you re-configure your connection you will obviously need to know which timeslot(s) are used for the original connection. In these cases it is best to use the option in the original network file that allows you to specify the timeslot(s) used for the connection that will be re-configured. Then your FPGA or C6000 module can be certain to change the right connection and not interfere with any of the others.

Inter-board connections

If your system is spread over multiple boards, you can use inter-board connection modules to extend your Real Time FIFO connections between the modules on the two boards.

HEARTConf and the network file syntax allows you to define connections between the modules on the two boards directly, and the tool generates the messages to connect from the first module to the inter-board connection module, and then from the second inter-connection module to the second module.

Reconfiguring this connection could be complicated, and it is probably best in this case to specify the two connections separately in the network file, and then to dynamically re-configure just one of them.

This will not change how the connection works but may make it easier to work out how to make the re-configuration.

Support

The HEART architecture is a powerful tool, and it is possible to do many things with it. Reconfiguring is entirely possible but sometimes a difficult process to get right if you don't intimately understand the architectural details. Of course the HUNT ENGINEERING support team will always be happy to assist you to define the details of your re-configuration.