



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>



HUNT ENGINEERING

HERON-BASE2-FIFO

***Hi-Speed USB (2.0) based, Two-
slot, HERON Module Carrier***

USER MANUAL

***Hardware Rev B
Document Rev B
M.Siggins 31/1/06***

COPYRIGHT

This documentation and the product it is supplied with are Copyright HUNT ENGINEERING 2005. All rights reserved. HUNT ENGINEERING maintains a policy of continual product development and hence reserves the right to change product specification without prior warning.

WARRANTIES LIABILITY and INDEMNITIES

HUNT ENGINEERING warrants the hardware to be free from defects in the material and workmanship for 12 months from the date of purchase. Product returned under the terms of the warranty must be returned carriage paid to the main offices of HUNT ENGINEERING situated at BRENT KNOLL Somerset UK, the product will be repaired or replaced at the discretion of HUNT ENGINEERING.

Exclusions - If HUNT ENGINEERING decides that there is any evidence of electrical or mechanical abuse to the hardware, then the customer shall have no recourse to HUNT ENGINEERING or its agents. In such circumstances HUNT ENGINEERING may at its discretion offer to repair the hardware and charge for that repair.

Limitations of Liability - HUNT ENGINEERING makes no warranty as to the fitness of the product for any particular purpose. In no event shall HUNT ENGINEERING'S liability related to the product exceed the purchase fee actually paid by you for the product. Neither HUNT ENGINEERING nor its suppliers shall in any event be liable for any indirect, consequential or financial damages caused by the delivery, use or performance of this product.

Because some states do not allow the exclusion or limitation of incidental or consequential damages or limitation on how long an implied warranty lasts, the above limitations may not apply to you.

TECHNICAL SUPPORT

Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section <http://www.hunteng.co.uk/support/index.htm> on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to <http://www.hunteng.co.uk> for the list of current re-sellers.

HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.

TABLE OF CONTENTS

INTRODUCTION.....	4
GETTING STARTED	6
INSTALLATION.....	6
LEARN HOW TO USE YOUR SYSTEM	7
USING YOUR HERON-BASE2-FIFO	8
<i>Features</i>	9
HERON SERIAL BUS (HSB)	11
HARDWARE RESET	12
<i>Software Reset (Code Composer Studio)</i>	12
PROCESSOR JTAG CONNECTOR	12
CONFIG.....	13
I/Os.....	13
EMBEDDED SYSTEMS WITH THE HERON-BASE2-FIFO.....	14
RESET.....	14
C6000 MODULES.....	14
MODULES WITH FPGA	14
GDIO MODULES.....	14
JTAG	14
MOUNTING THE HERON-BASE2-FIFO	15
HARDWARE DETAILS	17
POWER SUPPLIES	17
BOARD NUMBER SWITCH.....	17
MODULE IDS.....	18
<i>FIFO CLOCKS</i>	18
<i>Calculating Latency</i>	18
<i>FIFO flushing</i>	20
LEDS	21
DEFAULT ROUTING JUMPERS	21
JTAG HEADER	22
PHYSICAL DIMENSIONS OF THE HERON-BASE2-FIFO	23
POWER REQUIREMENTS OF THE HERON-BASE2-FIFO	24
UNCOMMITTED MODULE INTERCONNECTS	25
FITTING MODULES TO YOUR CARRIER	26
ACHIEVABLE SYSTEM THROUGHPUT	27
MODULE TO MODULE COMMUNICATIONS	27
USB COMMUNICATIONS	27
TROUBLESHOOTING	28
HARDWARE	28
SOFTWARE	28
CE MARKING	29
TECHNICAL SUPPORT	30
APPENDIX 1 – HERON SERIAL BUS COMMANDS	31

The HERON module is a module defined by HUNT ENGINEERING to address the needs of our customers for real-time DSP systems. The HERON module is defined both mechanically and electrically by a separate HERON module specification that is available from the HUNT ENGINEERING CD, via the “Technology Documents” section from the CD browser, or online from <http://www.hunteng.co.uk> and going to the user area.

The HERON module specification also defines the features that a HERON module carrier, such as the HERON-BASE2-FIFO, must provide. HERON stands for Hunt Engineering ResOurce Node, which tries to make it clear that the module is not for a particular processor, or I/O task, but is intended to be a module definition that allows “nodes” in a system to be interconnected and controlled whatever their function. In this respect it is not like the TIM-40 specification which was specific to the ‘C4x DSP.

As the HERON-BASE2-FIFO was developed, HUNT ENGINEERING have developed HERON processor modules that carry various Xilinx FPGAs and also members of the TMS320C6000 family of DSP processors from TI. In addition to these modules, the HERON specification is a super-set of the pre-existing HUNT ENGINEERING GDIO module, so the GDIO modules from our C4x product range can also be used in HERON systems.

The HERON module connects to the carrier board through several standard interfaces.

- The first is a FIFO input interface, and a FIFO output interface. This is to be used for the main inter-node communications. (It is usually also used for connection to the HOST computer).
- The second is an asynchronous interface that allows registers etc to be configured from a HERON module. This is intended for configuring communication systems, or perhaps to control some function specific peripherals on the carrier board. (This interface is not used on the HERON-BASE2-FIFO).
- The third is a JTAG (IEEE 1149.1) interface for running processor debug tools.
- The fourth is the HERON Serial Bus, used for configuration messages.
- The last is the general control such as reset, power etc. (of course these are provided by the HERON-BASE2-FIFO).

HUNT ENGINEERING defined the HERON modules in conjunction with HEART – the Hunt Engineering Architecture using Ring Technology. This is a common architecture that we adopt for some of our HERON carriers, that provides good real time features such as low latency and high bandwidth.

However, it is not a requirement of a HERON module carrier that it implements such features. The HERON-BASE2-FIFO is an example of a module carrier that does not implement the HEART architecture. In fact our customers could develop their own module carrier and add our HERON modules to it. Conversely our customers could develop application specific HERON modules themselves and add them to our systems.

The HERON-BASE2-FIFO is a HERON module carrier that has 2 heron module slots, that have FIFO input and output interfaces between each other, and between each module

slot and the Hi-Speed USB connection to the HOST computer. The HERON-BASE2-FIFO is a 'self-powered' USB peripheral, i.e. it does not take its power supply from the USB cable, but instead has its own external +5V DC power source.

The HERON-BASE2-FIFO is a HERON module carrier that connects to a HOST computer with a Hi-Speed USB connection.

Any modules that your system has should be fitted to the HERON-BASE2-FIFO and their retaining nuts fitted, before the HERON-BASE2-FIFO is powered up and connected to your host machine (see a later section of this manual for details).

Any default routing jumpers on modules have no impact on the connections between the nodes in your system, as the configuration is fixed.

The USB is by design a "plug 'n play" type bus, so the board is not assigned a base address by the user, but is accessed through the USB drivers of the operating system.

The HERON-BASE2-FIFO, however, provides a switch with which to select the "board number" in the system. This is the number that the API software uses to access the board. Make a note of the setting of this switch, and ensure that if you have more than one HERON-BASE2-FIFO in your system, that they all have unique switch settings.

Installation

The HERON-BASE2-FIFO should be powered up with the +5V power supply provided, and connected to your host computer with the USB cable provided (see a later section for advice).

You must use Windows 2000, Windows XP or Linux. A Windows operating system will detect that new hardware has been installed and will ask you for the device drivers. Choose the option that lets you continue without installing the drivers, and let the HUNT ENGINEERING software installation install the drivers.

You can follow the movie provided on how to install your system, found on the HUNT ENGINEERING CD under the Getting Started section. What this shows you how to do is:-

If you have purchased C6000 modules for your system, you will need to install Code Composer Studio at this point.

Next you should run the install option found under getting started on the HUNT ENGINEERING CD. First this program will install the HUNT ENGINEERING Application Programming Interface (API) and run the confidence checks provided with it. (See the API user manual for details).

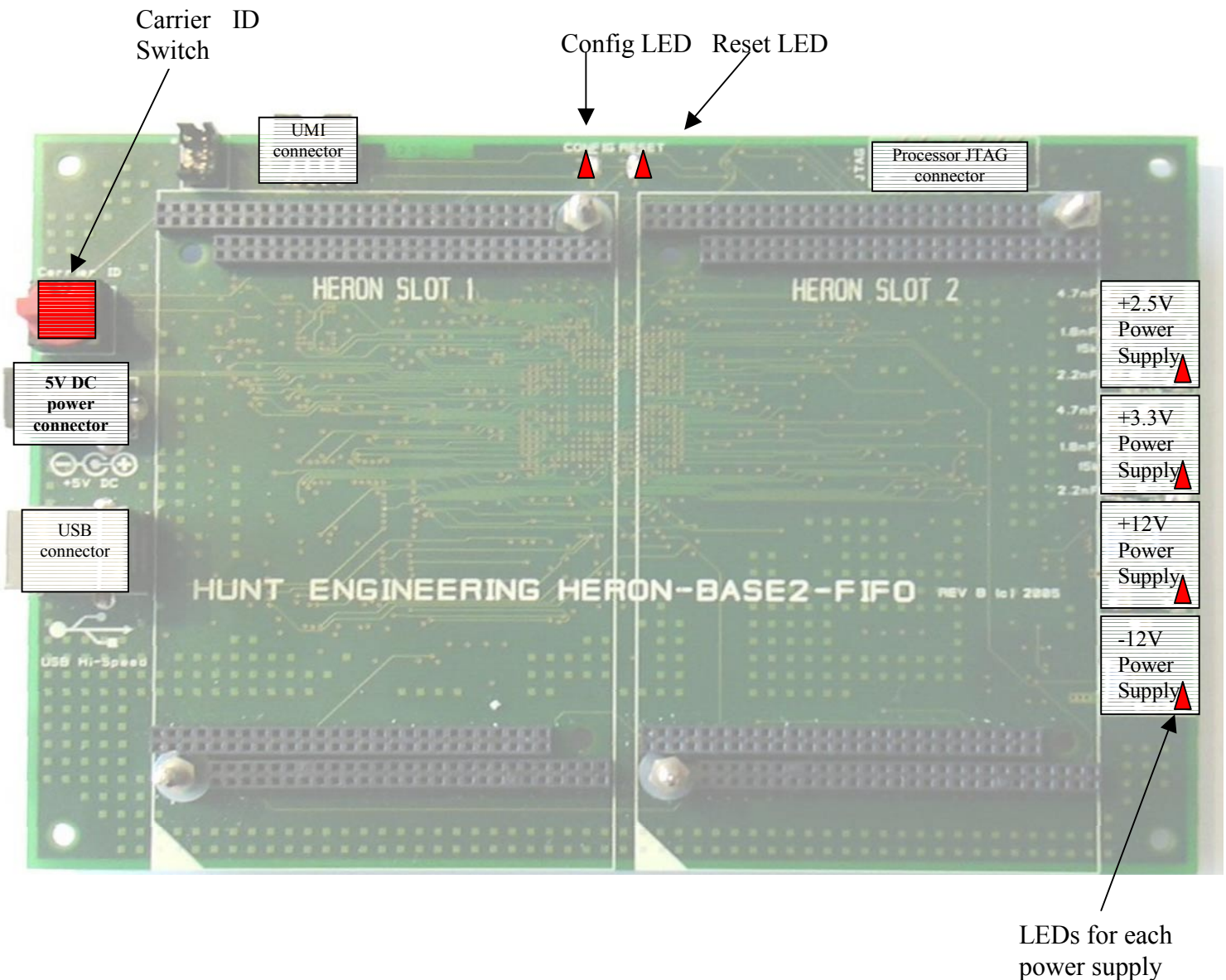
Next it will install the HUNT ENGINEERING Software developers pack which is required only if you have C6000 modules in your system.

Learn how to use your system

There are 'Getting Started' examples for FPGA and C6000 modules, and tutorials about using the software tools. You should follow the tutorials that are relevant to your system to learn how to use it, and get the best use of it.

There are many examples provided on the HUNT ENGINEERING CD, and this is the best place to start developing your application program.

Using your HERON-BASE2-FIFO

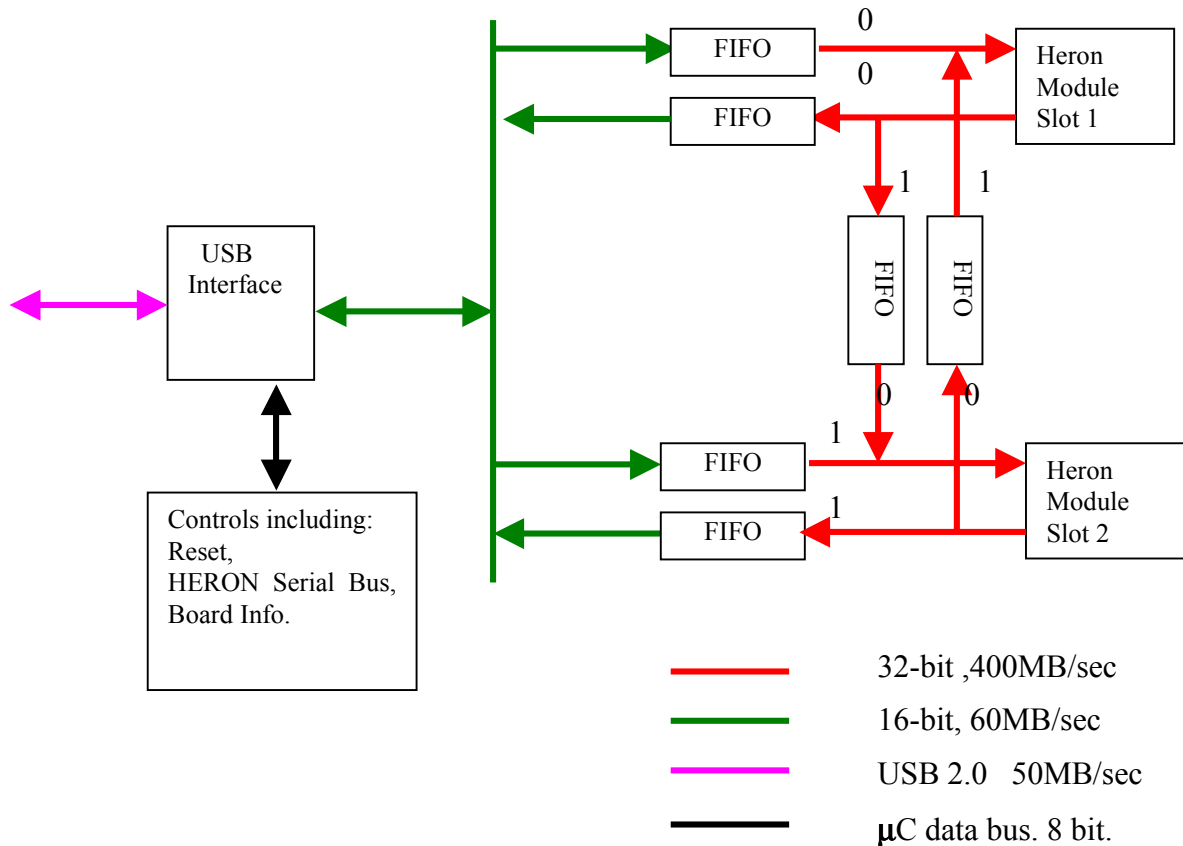


LEDs for each power supply

The HERON-BASE2-FIFO provides two HERON module sockets, each of which can be populated or left empty. Each module site can accept any 32-bit HERON module, or a subgroup of the 16-bit GDIO modules. The HERON-BASE2-FIFO will automatically configure for the correct module type without the need for setting any jumpers or software.

The connectivity of the HERON-BASE2-FIFO is fixed. It has point to point connections between the two module sites and from each module to the USB interface, provided by FIFOs. Each FIFO is 32-bits wide, but can switch into a 16 bit wide mode if a 16-bit module is detected by the hardware (see the HERON specification for details of how this is done). Each FIFO can accept a clock rate of between 60 and 100MHz from the module sites, allowing a maximum transfer speed into and out of those FIFOs of 4x100Million bytes/second. The USB interface has a maximum transfer speed of 60MillionBytes / second. This is higher than the maximum theoretical data rate of USB 2.0.

Features



HERON modules are designed to be interfaced to carrier boards that provide synchronous FIFOs.

The HERON-BASE2-FIFO provides these FIFOs in a fixed configuration, as shown in the diagram above. There is also a FIFO flush feature that may be useful in a real time Digital Signal Processing system. This is discussed more fully later in this manual.

FPGA modules

A HERON module that has a user configurable FPGA, has the pins of the HERON module connected to the pins of the FPGA. So the “program” for the user FPGA needs to correctly interpret the FIFO signals to access the FIFOs.

This will be done using the Hardware Interface Layer that HUNT ENGINEERING provides for these modules.

The FPGA has a separate 32 bit data path for Input and Output FIFOs, so can read one FIFO and Write one FIFO at the same time. If the “Almost” flag shows that there are many things that can be done, i.e. an Input FIFO is not Almost Empty, or an Output FIFO

is not Almost Full, the FPGA can transfer data at the rate of one 32bit word per FIFO clock. The FIFO clock of the HERON-BASE2-FIFO must be between 60 and 100MHz, so the transfer can take place at between 240 and 400 Million bytes per second.

There will be conditions though, where such bandwidth cannot be sustained. The most obvious case is where data is being transferred to/from a FIFO that connects to the USB interface, as the USB bandwidth is much less. It is also possible when transferring data to/from a FIFO that connects to the other module slot, that the other module cannot supply or accept data that quickly. Under these conditions, eventually the Almost flag will be asserted. Then the transfers must check the “limit” flag before each transfer. The data is then transferred more slowly than one data item per clock depending on the logic used.

The Hardware Interface Layer provided for the FPGA by HUNT ENGINEERING takes care of these things automatically, while presenting to the user a simple friendly interface.

DSP modules

A HERON module with a C6000 processor will have registers where the flags of the FIFOs can be read, and separate addresses in its memory map where each FIFO can be read/written. The hardware will have features that allow the DMA engines of the processor to access these FIFOs. The users program for the processor must include the software that accesses the FIFOs using DMAs or direct processor access. The software needed will be different depending on the Module hardware design.

The HERON-API software will be used to access these FIFOs, linked into the user program for the DSP.

C6000 processors usually have only one memory bus connected to the FIFOs, so it is not possible to read a FIFO and write a FIFO at the same time. That is any bus cycle can be either a read or a write but not both. Transfers can occur in both directions at once, by alternating the direction of the cycles, but this means the 240 to 400Million Bytes/second of the HERON interface is shared between reads and writes.

Usually to get the most efficient access of the FIFOs, the processor will use the “Block” flags to indicate that it is possible to transfer a block of data. On the HERON-BASE2-FIFO, the Block flags indicate that it is possible to transfer 64 words (32 bits wide). With the right hardware on the module it is possible to DMA data on consecutive FIFO clocks during this burst, but at the end of a burst the processor will need to take some actions to enable the next burst. When the transfer size is below the block size, the processor must transfer the data one word at a time, testing the limit flag between each transfer. This means it will take at least one cycle to read the flags, and another to access the FIFO data. In reality the C6000 architecture does not allow these processor driven accesses to be achieved in a single cycle and each one may take a few tens of cycles. This makes the transfer of small blocks of data quite inefficient. To help alleviate this problem the HERON-BASE2-FIFO provides “Almost” flags. These “Almost” flags indicate that it is possible to transfer 4 words, enabling the processor to perform a DMA burst of 4 consecutive accesses, so it is only when these flags become asserted, that the processor has to resort to transferring one word at a time.

Functions in HERON-API handle these issues for you, allowing you to use read and write calls to start transfers and then offering you a choice of ways that you can be informed when the transfer has completed. HERON-API offers you a consistent interface regardless of the hardware design of the particular module you are using.

GDIO modules

GDIO modules have a subset of the HERON module pins only. They are 16 bit modules with no user programmability. They can only access FIFO #0 of the module slot, and this access is made by the hardware of the module.

Typically GDIO module transfers cannot be blocked, and data will be lost if the FIFO reaches it's limit (full or empty), but this is not always the case.

The HERON-BASE2-FIFO automatically detects that the module is a 16 bit version, and packs two of these 16 bit entities into each 32 bit word. The first 16 bit item is the lowest 16 bits of the 32 bit word. This is only implemented on FIFO #0 as this is the only one connected to a GDIO interface.

USB (Universal Serial Bus)

The USB has access to the FIFOs that connect to each module slot. This means that there are two input and two output FIFOs that can be accessed by the Host PC via the USB.

The USB interface of HERON-BASE2-FIFO supports the 'Hi-Speed' mode of operation, (as defined in UBS 2.0), that is a bit rate of 480Mbits/sec. This bit rate however has to include all phases of the protocol, as well as the data itself. The maximum THEORETICAL data rate for 'Hi-Speed' bulk transfers, (as mentioned in the USB 2.0 specification), is 53.248 Mbytes/sec. It should be noted however that the data rate achieved is dependant on the how the controller in the Host machine is configured, and how many USB devices are connected.

The HERON-BASE2-FIFO also supports the 'Full-Speed' mode of operation, that is a bit rate of 12Mbits/sec. The maximum THEORETICAL data rate for 'Full-Speed' bulk transfers is 1.216Mbytes/sec, and, as for 'Hi-Speed' mode, the data rate actually achieved is dependant on the host machine.

USB is a 'host-centric' bus, meaning that it is the host that initiates all data transfers to and from peripherals like the HERON-BASE2-FIFO. The underlying protocols within USB handle the handshaking and flow control of data transfers, transparently.

The HUNT ENGINEERING Host API is used to access the HERON-BASE2-FIFO. The Host API provides an identical interface to the FIFOs, HSB and other features, across different Operating Systems and different module carriers.

Heron Serial Bus (HSB)

The HERON-BASE2-FIFO provides a Heron Serial Bus that runs to both module slots. This provides a non-real time interface for configuration type messages. The HSB is also used internally to the HERON-BASE2-FIFO to program the FIFO flush features.

A C6000 module would access the HSB using functions provided in the HERON-API, and an FPGA module would access HSB using the HE_USER interface in the Hardware Interface layer provided.

HSB can be used for other system configuration or control messages, i.e. it is used by the `hrn_fpga` utility to download the user configuration from the PC to an FPGA module.

It must be remembered though that the HSB is not only slow, it is also arbitrated, so it cannot be relied upon for real time operation unless your system is carefully defined so that arbitration failures will never occur.

Hardware Reset

Before the HERON-BASE2-FIFO can be used, it must receive a Hardware reset. Actually, the HERON-BASE2-FIFO generates such a reset on power up. This is generated using a large counter chain so that the reset remains asserted for some time after the power supplies are stable and the fpga that performs many of the system functions on board, is configured

The reset is also asserted and de-asserted via the HOST-API function `HE_RESET`.

This reset initialises all of the HERON-BASE2-FIFO circuitry and the modules on it, (via the Module Reset), into a known state. This includes flushing all FIFOs.

Module Reset however does not “clear” the configuration of an FPGA based module.

When the reset is asserted the LED at the top edge of the HERON-BASE2-FIFO labelled “RESET” will light up. When the reset is removed it will no longer be lit.

When running a tool like the HUNT ENGINEERING server/loader, the reset LED will flash for a short time when the system is reset prior to booting.

Software Reset (Code Composer Studio)

Code Composer Studio has a menu that allows “DSP reset”. This must never be confused with the hardware reset controlled via the USB – it is not the same thing. The Code Composer “DSP reset” simply resets some of the internal registers of the DSP but will NOT empty the hardware FIFOs. It will also only affect the DSP where the menu is selected, and cannot affect any I/O boards in the system.

The reset LED will not flash when a DSP reset is made from Code Composer Studio.

HUNT ENGINEERING provides a Reset Plug-in for Code Composer Studio that allows you to use the Hardware Reset from within Code Composer. This does affect all of the modules, and clear the hardware FIFOs.

Processor JTAG connector

The processor JTAG connector is provided to allow access to the JTAG connections of the HERON modules, from an external emulator, in order to run Code Composer Studio, (the development environment for the ‘C6000 processors. These connections are not used by FPGA modules, but are connected to a C6000 processor on a C6000 HERON module.

The design of the HERON module means that the HERON-BASE2-FIFO can determine if a HERON slot does not contain a module, or contains a module that does not implement

JTAG (such as a GDIO module).

Code Composer Studio needs to be configured to understand how many processors there are in the chain. When this is done please note that the hardware JTAG chain is connected to slot 1, then slot 2. However, different versions of Code Composer Studio list these in a different order during setup, so be careful when setting CCS up manually.

The connector is a standard 14 way JTAG connector as used by TI on their emulators such as the XDS510, XDS560, and also by compatible emulators from other companies. This includes the HEPC9 from HUNT ENGINEERING.

The connector on the HERON-BASE2 is a simple 0.1 inch header in a 7x2 configuration. It accepts the standard connector on most emulators, but beware that the C6000 JTAG chain is a 3.3V one. For this reason 5V JTAG emulators should not be used.

Config

There is a system wide Config signal that is an open collector output from each module and hence requires a pull up to be provided by the carrier board. The HERON-BASE2-FIFO provides this pull up. Each processor-based module will drive this signal low after a Hardware reset. It can then be released under software control after booting.

The Config signal is provided for any hardware that needs to be disabled until the entire system has booted. If there is any hardware that uses this feature it will not function until all processors have removed their config signal.

The Config is buffered and used to control an LED labelled "CONFIG" at the top edge of the HERON-BASE2-FIFO. This LED will illuminate when the config signal is asserted (low).

I/Os

There are some "Uncommitted Module Interconnect" (UMI) signals defined by the HERON specification, which are simply connected to all modules.

These are intended to connect control signals between modules, for example a processor module can (via software) drive one of these signals with one of its timer outputs. Then if an I/O or FPGA module is configured to accept its clock input from one of these signals, it is possible to implement a system with a programmable clock. There will be other uses for these signals that are module design dependent.

The HERON-BASE2-FIFO pulls these signals high with 10K resistors.

The HERON-BASE2-FIFO also connects these four signals to the UMI connector near the top edge of the board.

Embedded systems with the HERON-BASE2-FIFO

The HERON-BASE2-FIFO can be used as an embedded HERON module carrier. In this case there is no USB connection to a HOST machine.

Reset

The HERON-BASE2-FIFO automatically generates a reset signal following power up. This means that the system will be properly initialised if you simply “switch on” the power.

C6000 modules

A C6000 HERON module boots its processor from some on board FLASH ROM. Normally this is a pre-boot that initialises the hardware and starts to accept a boot stream from a FIFO.

In an embedded system there is no way to boot your application program onto the module using the FIFOs. In this case you must program your application into the on board FLASH ROM. Utilities and instructions that help you do this can be found on the HUNT ENGINEERING CD.

Modules with FPGA

Modules that have FPGAs can have PROMS fitted to them that initialise the FPGA with your application program. You need to refer to the user manual of the particular module that you are using for details of the options provided and how to use them.

GDIO modules

GDIO modules do not require any programming but are simply hardware that starts to run after reset.

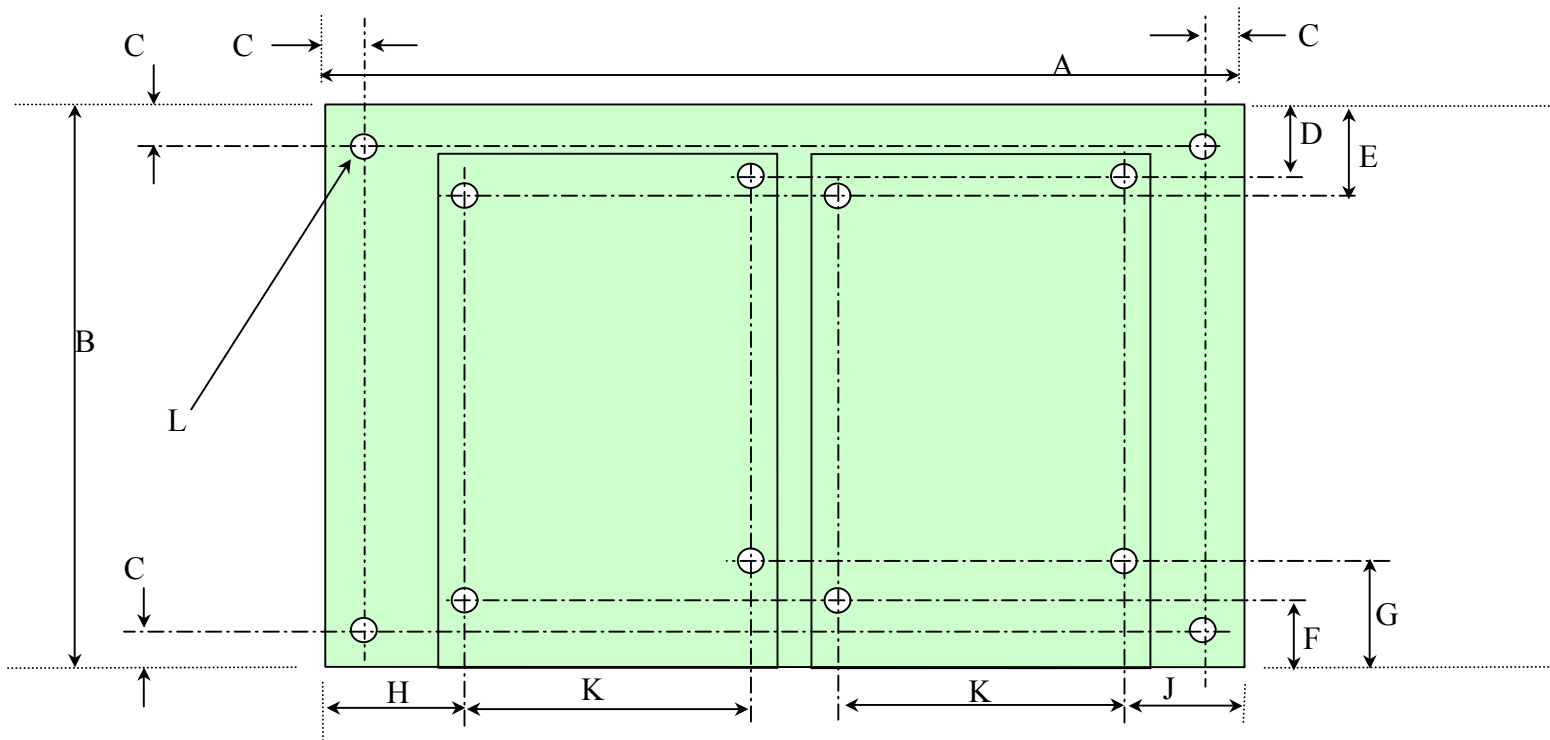
JTAG

The HERON-BASE2-FIFO is always a JTAG slave, so the JTAG header is an input for connecting to an external JTAG master. Hence an embedded HERON-BASE2-FIFO can accept a JTAG input from a Hunt Engineering HEPC9 or HECPCI9 or a TI XDS510 or XDS560 without requiring any further set up.

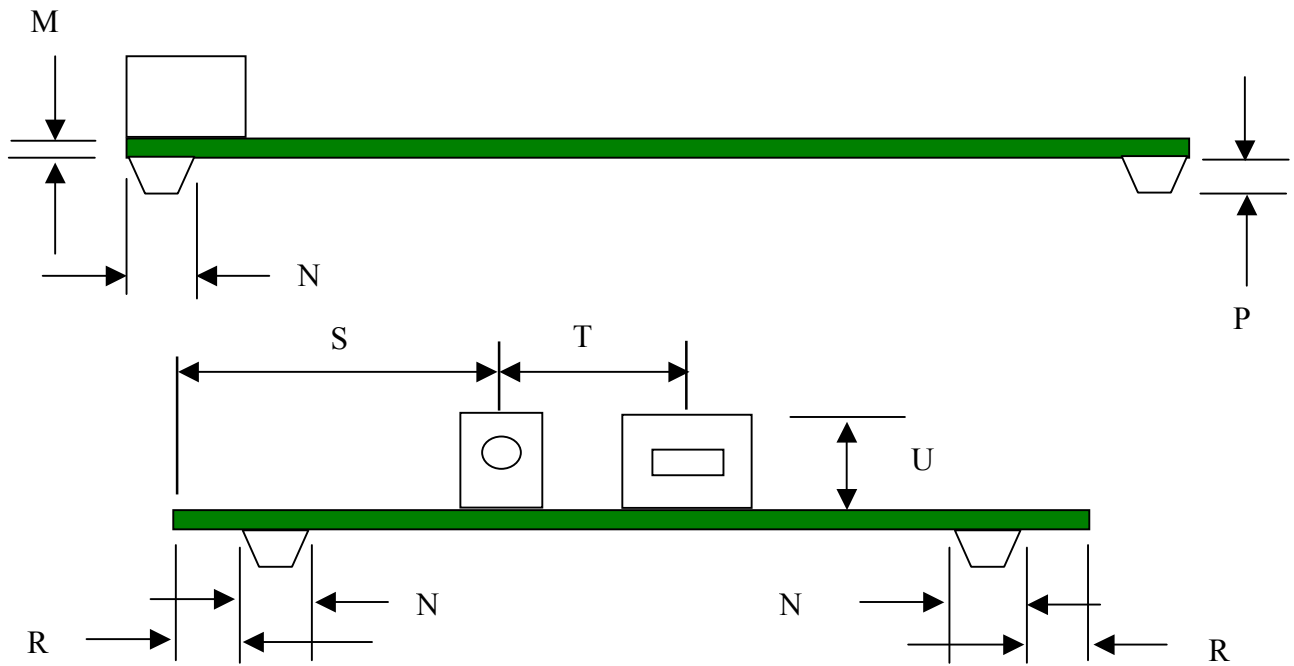
Mounting the HERON-BASE2-FIFO

The HERON-BASE2-FIFO is supplied with four polyurethane ‘feet’ on the underside of the board, so that it may be used freestanding.

If required however, the four fixing holes can be used to mount the HERON-BASE2-FIFO. Because of the length of the HERON-BASE2-FIFO, it is also recommended that one or more fixings are used in the centre of the board. The Module Fixing positions can be used for this purpose.



Reference	Description	Dimension
A	Board length	160 mm
B	Board height	110 mm
C	Board edge to fixing hole	5 mm
D	Top edge to module primary fixing hole	13.5 mm
E	Top edge to module secondary fixing hole	18.5 mm
F	Bottom edge to module primary fixing hole	14 mm
G	Bottom edge to module secondary fixing hole	19 mm
H	Left edge to module fixing holes	21.9 mm
J	Right edge to module fixing holes	72.1 mm
K	Left module fixing holes to right module fixing holes	54 mm
L	Diameter for all fixing holes	3.175mm (0.125")



Reference	Description	Dimension
M	Board Thickness	1.6mm
N	Width of support feet	12.7mm
P	Height of support feet	5.8mm
R	Distance of support feet from top and bottom edges.	10mm
S	Top edge of board to centre of power jack	42mm
T	Centre of power jack to centre of USB connector	21.75mm
U	Connector height above board	11mm

The following sub-sections provide details on the hardware such as connector pinouts and signal levels etc, but they are placed here, as the "system" configurer does not normally need this information. It is, however, necessary for a user who needs to develop compatible hardware or for use in system troubleshooting.

Power supplies

The HERON-BASE2-FIFO is a stand-alone board that requires an external +5V DC power supply. A connector is provided at the left edge of the board for this purpose; it is a socket with a 2.1mm centre pin. The centre pin is the positive connection to the +5V power plane of the PCB. An example of a suitable mating plug is one available from Farnell Electronics, order code 224-923. It is also available from Cliff Electronic Components, part number DCPP1.

The table below indicates the current required by the HERON-BASE2-FIFO from the external power supply.

Voltage	Typical (& measured)	Maximum
+5V	1.1A	1.5A

Remember when calculating the total system power that the power requirements of each module must be added to these.

Board Number Switch

The only user configurable option on the HERON-BASE2-FIFO is the board number switch. This switch can be turned by hand and a hexadecimal number between 0 and F is displayed to indicate its current setting.

The number selected by this switch is the "board number" that must be used when accessing this board using the HUNT ENGINEERING API software. This is achieved by the driver layer of the API reading the switch value, and using this to identify this board to the upper layers of the API.

The value of this switch is also used by the HERON processing modules when they are booted. This is to make sure that they do not boot from data intended for another processor. Refer to the user manual of the relevant HERON processing module for details.

The HSB also uses the board number setting to address the nodes.

There is no restriction on the setting of this switch other than there cannot be two boards with the same setting of this switch.

Module IDs

Each “node” of the HERON-BASE2-FIFO has a slot number assigned to it, as defined in the HERON specification. The combination with the Board number switch and this slot ID allows the module to identify itself uniquely in the system.

The boot prom of the HERON processing modules, use this for boot purposes as does the HSB for addressing particular nodes.

Node	Slot ID assigned
HERON slot 1	1
HERON slot 2	2
Host node	5

FIFO CLOCKS

The FIFO clocks on the HERON-BASE2-FIFO are according to the 100MHz FIFO timings shown in the HERON module specification. The clocks have a minimum frequency of 60MHz and a maximum frequency of 100MHz.

The clocks from each module, and the read and write clocks can all have different frequencies with any phase relationship.

Calculating Latency

A feature of the fixed data paths of the HERON-BASE2-FIFO is that the latency of communication can be calculated.

The use of FIFOs means that the latency varies depending on how you use them. For example if you use a block flag to determine when to write, that write may be delayed until there is space for a block of data, and this can obviously take longer than the delay incurred of waiting for space for just a single word, i.e. when you would use the full flag.

From Heron Module to USB

1. Writing a data item to the write FIFO will take two FIFO clocks.
2. That data will be available at the USB end of the FIFO after a further 10 cycles of the FIFO write clock, and 1 or 2 cycles of USB chip’s 30MHz clock.
3. There is a common 16-bit bus for the USB end of the four FIFOs between the USB interface and the module slots, and only one transfer can take place at a time. Ie. A transfer from the Slot 1 write FIFO, a transfer to the Slot 1 read FIFO, a transfer from the Slot 2 write FIFO, or a transfer to the Slot 2 read FIFO. These transfers take place in a rotating priority, so at best, no other transfers are taking place, and the on-board logic is sampling the FIFOs status just one cycle after data has become available. At the other extreme, the on-board logic may just miss the change of this FIFO’s status, and the other three FIFOs all have transfers pending and they are all of maximum packet size, (512 Bytes), so there will be $3 \times 256 = 768$ cycles at 30MHz, plus 12 cycles of ‘endpoint switching’, making 780 cycles, until the logic

starts to service the FIFO in consideration.

4. The data is clocked into an IN 'endpoint' buffer of the USB device, at a rate of one 16-bit word per 30MHz cycle. It must be pointed out though that these buffer contents are not 'seen' by the serial engine of the USB device, until the buffer is committed. This happens in one of two ways. (a) the buffer is filled to its maximum packet size, or (b) the buffer is only partially filled, but the logic determines that there is no more data to send and commits the short packet. So, the shortest time for a packet to be committed is when there is a single 32-bit word in the FIFO. In this case only five 30MHz cycles are need to write the data, and then the logic commits the short packet taking another two cycles. The longest time for a packet to be committed is when there are 127 32-bit words in the FIFO, (one less than a maximum packet), as that will take 254 cycles to transfer the data into the endpoint buffer, plus the extra cycles to commit the short packet.
5. At this point there is a committed packet in the IN endpoint buffer, available to the serial engine of the device. Exactly when the data is transferred to the host, is not under the control of the HERON-BASE2-FIFO, it is dependent only on the host. This is a feature of USB – the host initiates all transfers on the bus.

In the best case, items 1, 2, 3 and 4 make a total of 12 clock cycles of the FIFO write clock and 10 cycles of the 30MHz. Assuming a FIFO write clock of 100MHz, this equals $120\text{ nS} + 333\text{ nS} = 453\text{ nS}$. This is the time between the HERON module writing a single 32-bit word, and the 4-byte packet being committed.

In the case where the HERON module is sending a lot of data, (more than one packets-worth), and sending at least as fast as the USB interface can accept data, (60Mbytes/sec), step 4 simply becomes 256 cycles of the 30MHz clock, as the complete 512-byte packet is automatically committed. The time now, from the HERON module starting to write its data, and the 512 byte being committed, is $120\text{ nS} + 260\text{ cycles of the }30\text{MHz clock}$, which is equal to $8.78\text{ }\mu\text{S}$

In the worst case, step 3 is 780 cycles instead of just one, so the latency for a single 32-bit word is $26.42\text{ }\mu\text{S}$, and for a complete packet of 512 bytes, (128 x 32-bit words), the time from the HERON module starting to write its data, and the 512 byte being committed $34.74\text{ }\mu\text{S}$.

From Heron Module to USB

1. The arrival of data packets from the host, is signalled to the same on-board logic mentioned in step 3 of the above section, so again, the best scenario is that the data packet is ready just before the status is sampled, so this step takes one cycle. At worst, it could take 780 cycles. I.e. from 33 nS to $26\text{ }\mu\text{S}$.
2. The data will be available to the Heron Module after a further 4 cycles of the 30MHz clock, and 14 cycles of the Module's receive FIFO clock. Assuming the Heron Module is driving its receive clock at 100MHz, this step will take 273 nS

In the best case, when a single 32-bit word is being transferred, the time taken from the packet arriving from the host, and the empty flag being de-asserted at the Heron Module connector, is 306 nS . When the host is sending a lot of data, the same latency will be observed for the first word of the packet, with subsequent words of the packet becoming available to the Heron Module at a rate of approximately 15Mwords/sec. The module is at

liberty to read incoming data words as they becomes available according to the empty flag, or it may wait for a block of data to become available, (using the block ready flag), and perform one large burst read access.

In the worst case, when a single 32-bit word is being transferred, the time taken from the packet arriving from the host, and the empty flag being de-asserted at the Heron Module connector, is 26.27 μ S. When the host is sending a lot of data, the same latency will be observed for the first word of the packet, and, as above, subsequent words of the packet become available to the Heron Module at 15Mwords/sec.

Between Heron Modules

1. Writing a data item to the write FIFO will take two write FIFO clocks.
2. It then takes between 9 and 11 cycles of the 100MHz clock for the data to traverse the main inter-slot FIFO. The variation is due to synchronisation as clock domains are crossed.
3. The data is available to the other Heron Module, (empty flag de-asserted), after two cycles of the receiving modules Read FIFO Clock.

Assuming the sending modules Write FIFO clock, and the receiving modules Read FIFO clock are both running at 100MHz, steps 1,2 and 3 add up between 130nS and 150nS.

The calculations above show that there is a wide range of possible latencies for FIFO transfers involving the USB interface, but for transfers between the two modules, the range of latencies is small.

FIFO flushing

A feature of the HERON-BASE2-FIFO is that the FIFOs in a connection can be cleared by a module asserting one of its UMI lines. In that case the clearing event is programmed for a FIFO using the HSB configuration registers. Multiple FIFOs can be programmed to “flush” on the same UMI line, so sending and receiving halves of a bidirectional ‘pipe’ can be cleared, or even, multiple pipes cleared without asserting the hardware reset.

This feature will be selected by the HERON-BASE2 configuration tool. Once this setting is selected for a particular FIFO, the flushing can take place by writing the UMI as an output from a C6000 module or simply driving the UMI line from an FPGA.

For definitions of those registers see the appendix of this manual.

LEDS

There are several LEDS on the HERON-BASE2-FIFO.

One of them, on the reverse side of the board and labelled 'DONE', is used to indicate if the FPGA that provides much of the functionality of the board, is properly configured from its PROM. Sited close to the FPGA, this LED should flash briefly on power up, but then remain off. This shows that the FPGA has configured. If at any time this LED is illuminated, try powering your system off (that is switch off, not re-boot) and back on again. If the LED is still illuminated then there is a hardware fault and you should contact Technical support at your supplier.

On the front side of the board, along the right-hand edge, there are four more LEDS: These are labelled 2.5V, 3.3V, +12V and -12V. These are normally illuminated to indicate that their respective power supply circuits are functioning. If any of these is not illuminated then the Power supply is not working. This could be because you short-circuited a connection and the safety feature has activated to prevent damage to your hardware. If at any time one of these LEDS is **not** illuminated, try powering your system off (that is switch off, not re-boot) and back on again. If the LED is still **not** illuminated then there is a hardware fault and you should contact Technical support at your supplier.

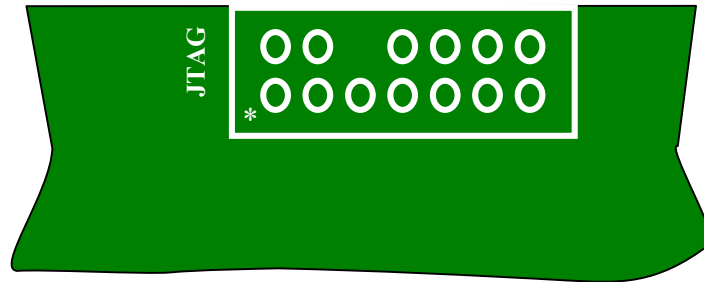
The remaining two LEDS along the top edge of the board, are indicators of the state of the Reset and Config lines. They are labelled "RESET" and "CONFIG" and illuminate when the respective signal is asserted (low).

Default Routing Jumpers

The default routing feature is not used on the HERON-BASE2-FIFO, as the FIFO connections between the host and the Heron Modules, are fixed.

JTAG header

There is a 14 way JTAG header on the HERON-BASE2-FIFO, along the top edge of the board, above the Slot2 Heron connectors.



This connector is the standard pinning for a 14 way JTAG header so it accepts the cable supplied with, for example, a TI XDS510 or 560. The HERON-BASE2-FIFO is only ever a JTAG slave.

TMS	* O	O	TRST
TDI	O	O	GND
PD	O		Polarisation
TDO_RET	O	O	GND
TCK_RET	O	O	GND
TCK	O	O	GND
EMO0	O	O	EMU1

Physical Dimensions of the Heron-Base2-Fifo

The HERON-BASE2-FIFO is 160mm by 110 mm overall.

There are no components inside the footprint of the Heron Modules on the top side of the board. This maintains conformance with the Heron Specification.

The maximum height of the HERON-BASE2-FIFO above the PCB is 11mm which will be less than the height of the components on the modules when fitted.

Power Requirements of the Heron-Base2-Fifo

The HERON-BASE2-FIFO is powered by a single +5V DC source. This is supplied via the centre-positive socket at the left hand edge of the board.

The +12V, -12V, +3.3V, and +2.5V required by the Heron-Base2-Fifo, are all generated on-board by efficient switch-mode circuits, from the +5V.

Uncommitted Module Interconnects

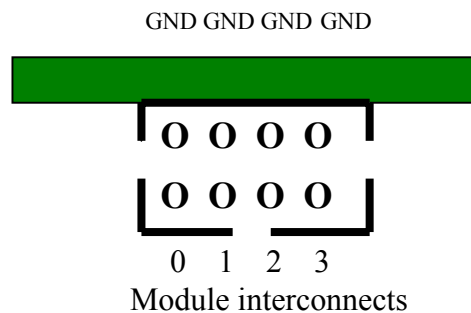
As per the HERON module specification there are four signals that are connected between the HERON modules. They are uncommitted in that they serve no fixed purpose, but HERON modules can provide functions that use these signals.

For example a HERON processor module might drive one of its timers onto one of these signals. A HERON-IO module might then be configured to use this as its sample clock.

In such an example it is easy to see that the same timer could be used to provide the sample clock of a HERON-IO module on another board.

For these reasons the four uncommitted Module interconnect signals are provided unbuffered on a connector on the HERON-BASE2-FIFO. This is situated on the top edge of the board. It is a 2mm pitch connector that is angled to accept its cable “flush” with the pcb. This connector is a MOLEX type 877333-0820, which requires MOLEX type 50394 crimp terminals together with MOLEX type 51110-0860 crimp housing for the cable termination.

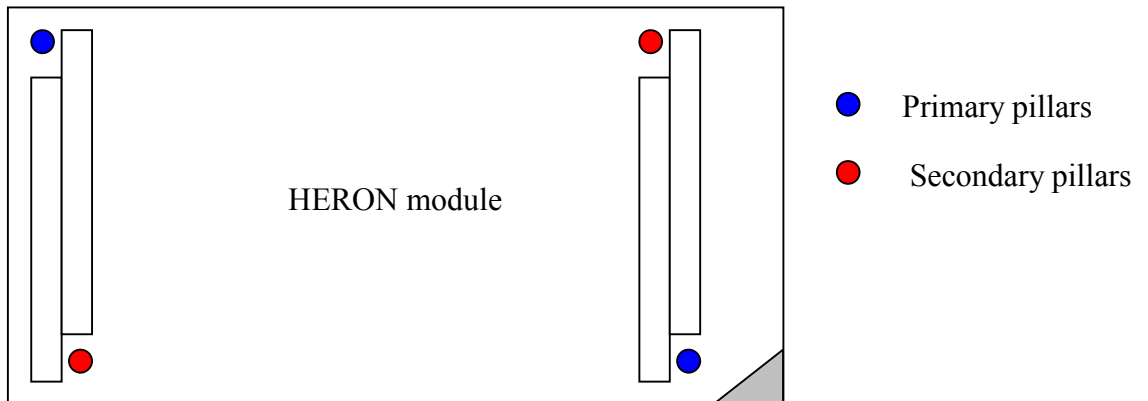
The pinout, as seen from above, is as follows:



Fitting Modules to your Carrier

Fitting HERON modules to your carrier is very simple. Ensure that the module carrier does NOT have power applied when fitting modules, and normal anti-static precautions should be followed at all times.

Each HERON slot has four positions for fixing pillars

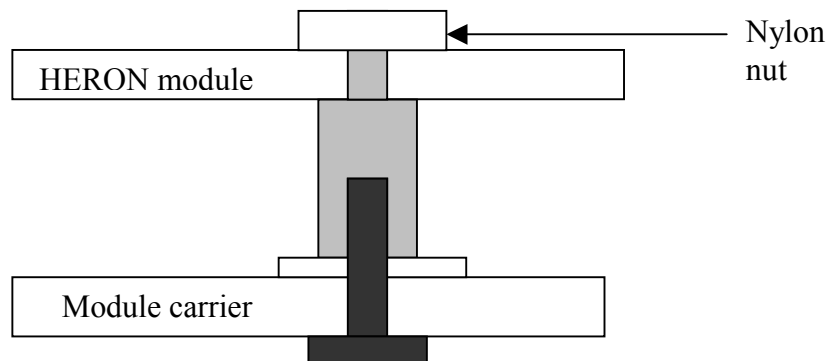


The Carrier card will probably only have spacing pillars fitted to the primary location for each HERON slot. The pillars for the secondary locations will be supplied as an accessory. The reason for this is that the legacy GDIO modules cannot be fitted if the secondary pillars are in place.

The HERON modules are asymmetric about their connectors, so if a module is fitted entirely the wrong way round, the module does not line up with the markings on the carrier card. In particular, notice the triangles on the silk screen of the HERON modules and the HERON slots of the carrier card. These should be overlaid when the module is fitted.

The HERON connectors are polarised, preventing incorrect insertion. So if more than a gentle force is needed to push the module home, check to make sure that it is correctly aligned. Take care not to apply excessive pressure to the centre of the module as this could stress the module's PCB unnecessarily.

Normally the primary fixings will be enough to retain the modules, simply fit the nylon bolts supplied in the accessory kit to the top thread of each mounting pillar.



If the environment demands, the secondary fixing pillars can be fitted to modules that allow their use.

Achievable System Throughput

In a HERON system there are many factors that can affect the achievable system throughput. It must be remembered at all times that the part of the system that has the lowest limit on bandwidth will govern the throughput of the system.

Heron modules can access the carrier board's FIFOs in 32-bit mode. FPGA modules can, (with the right program), transfer one 32bit word in and another out in the same clock cycle. It is possible therefore, that an FPGA module running it's read and write FIFO clocks at the maximum rate of 100MHz, giving sustained data rates of 400Mbytes/sec, for both reads and writes.

Module to Module Communications

The clock for the FIFO is generated by the HERON or GDIO module, so in fact this 400Million bytes/second will not be achieved unless the module provides the maximum clock frequency of 100MHz, AND the module performs accesses in consecutive clock cycles.

For the bandwidth that will be achieved between the modules in your system, you must check the bandwidth limits of the modules at each end of the FIFO as well as the limit of the FIFO.

For example two FPGA based modules will be able to send data at 400Mbytes/second in both directions at ones, but a DSP module will achieve lower rates due to the overheads of configuring DMAs, receiving interrupts etc

USB Communications

The USB communications throughput is dependent on the Host PC performance.

All parts of the HERON-BASE2-FIFO have been designed to be capable of 60Mbytes/second peak on the USB bus.

The theoretical maximum data rate over a 'Hi-Speed USB' connection is 53.248 Mbytes/sec, while for 'Full-Speed' it is 1.216Mbytes/sec. However, while data rates close to these maximums can be achieved, it should be noted that the data rate actually achieved is very dependant on the controller in the Host machine, how it is configured, data block sizes, and how many USB devices are connected. Large variations in performance are not unusual between different PCs. For example, one PC with on-board USB 2.0 hardware might give 40Mbytes/sec for reads and for writes. In contrast, a PC using a PCI add-in USB port that *claims* USB 2.0 performance, but only give 4Mbytes/sec for writes, and 8Mbytes/sec for reads.

For more detailed discussions of this topic please refer to the HUNT ENGINEERING web site.

As a guideline on-motherboard USB 2.0 devices can normally achieve over 40Mbytes/second if large enough transfers are used.

The following sections attempt to cover all likely problems. Please check through this section before contacting technical support.

Hardware

If the Hardware has been installed according to the Instructions there is very little that can be wrong.

- If the API returns an error on opening a newly installed board check the setting on the “Board Number” switch is set to the same board number as you are using in software.
- If the open call to the API is successful, but the booting of the system fails there is probably a mistake in the network file.

Software

As long as the software has been installed using the installation program supplied on the HUNT ENGINEERING CD, there should be little problem with the software installation.

If you have problems then return to one of the example programs supplied with the system.

HUNT ENGINEERING have performed testing on its products to ensure that it is possible to comply with the European CE marking directives. The HERON-BASE2-FIFO cannot be CE marked as it is a component in a system, but as long as the following recommendations are followed, a system containing the HERON-BASE2-FIFO could be CE marked.

The immense flexibility of the HUNT ENGINEERING product range means that individual systems should be marked in accordance with the directives after assembly.

1. The host computer or housing in which the HERON-BASE2-FIFO is installed is properly assembled with EMC and LVD in mind and ideally should itself carry the CE mark.

2. Any cabling between boards or peripherals is either entirely inside the case of the host computer, or has been assembled and tested in accordance with the directives.

The HERON-BASE2-FIFO I/Os ARE protected against Static discharge, so if the cabling does exit the case, there is suitable protection already fitted.

HUNT ENGINEERING are able to perform system integration in accordance with these directives if you are unsure of how to achieve compliance yourself.

Technical Support

Technical support for HUNT ENGINEERING products should first be obtained from the comprehensive Support section <http://www.hunteng.co.uk/support/index.htm> on the HUNT ENGINEERING web site. This includes FAQs, latest product, software and documentation updates etc. Or contact your local supplier - if you are unsure of details please refer to <http://www.hunteng.co.uk> for the list of current re-sellers.

HUNT ENGINEERING technical support can be contacted by emailing support@hunteng.co.uk, calling the direct support telephone number +44 (0)1278 760775, or by calling the general number +44 (0)1278 760188 and choosing the technical support option.

Appendix 1 – HERON Serial Bus Commands

The HERON-BASE2-FIFO does not implement HEART, so there are no HSB messages needed to make FIFO connections. However some HSB commands are implemented by the HERON-BASE2-FIFO to offer some of the advanced features.

Normally these features are enabled and configured using the HERON-BASE2 configuration tool, but the HSB commands are listed here for information and for use in the case that these features are needed in an embedded system. In that case these messages can be sent from a module.

(1) UMI reset

Some systems may reach a stage where they would like to clear data from a particular FIFO connection without making a complete system reset. The HERON-BASE2-FIFO offers a way to achieve that using the UMI signals from the modules.

First a FIFO must be enabled to accept a reset from a particular UMI line. This is done with the following HSB command.

The message is sent to device 7 on the board you are using i.e. board id in the top

4 bits 7,6,5,4, and 0x7 in the bottom 4 bits.

Where board id is the red switch setting.

The message data is 4 bytes and built up as follows:

byte 0: device id : board id in the top 4 bits 7,6,5,4,
and 0xE in the bottom 4 bits.

Where board id is the red switch setting.

byte 1: command byte: 0x7.

byte 2: fifo number : defined as :-

FIFO 6 : From USB to Module slot 1

FIFO 7 : From Module slot 2 to Module slot 1

FIFO 8 : From USB to Module slot 2

FIFO 9 : From Module slot 1 to Module slot 2

FIFO 10 : From Module slot 1 to USB

FIFO 11 : From Module slot 2 to USB

byte 3: UMI reset : 4 bit register, bit 0 = UMI 0, etc.

If bit set, UMI connected, if bit clear

UMI is not connected.

Following this set up, a Module can clear the FIFO(s) by asserting the nominated UMI lines.

A hardware reset will clear these registers.

(2) Slot 2 FIFO swap

Normally the FIFO numbers at the Module slot 2 are FIFO0 is connected to Module 1, and FIFO 1 is connected to the USB connection.

However if a DSP module is to be booted in Slot2, without a DSP module in slot1, this needs to be changed for the booting process. This is because DSP modules boot from FIFO 0 always.

So there is an HSB message that temporarily swaps the fifo numbers at slot 2, making FIFO 1 be connected to Module slot 1, and FIFO 0 be connected to the USB.

The message is sent to device 7 on the board you are using i.e. board id in the top

4 bits 7,6,5,4, and 0x7 in the bottom 4 bits.

Where board id is the red switch setting.

The message data is 4 bytes and built up as follows:

byte 0: device id : 0.

byte 1: command byte: 0x7

byte 2: fifo number : 0x10

byte 3: UMI reset : 0 = no fifo swap
1 = swap fifos

Server/Loader and confchk tools use this, but always return the setting to not swapped after running.

A hardware reset will set the FIFOs to non-swapped.