**Experimenting with HyperRAM/Analog Functions on the P2-EVAL board**

*Richard J. S. Morrison, March 21st 2019*

This document describes a HyperRAM interface for the Parallax P2-ES evaluation board. The interface uses an ISSI 66-67WVH8M8ALL-BLL, providing an additional 8MB of memory that is accessed as 65536 blocks of 128 bytes. In the current implementation the HR chip is mounted on a small carrier board that plugs into headers J203/201 on the P2-ES, as seen in Fig. 1.

Fig. 1 A HyperRAM carrier board for the P2-EVAL

To evaluate HyperRAM perfomance, a LabVIEW host interacts with the P2-ES (Fig. 2). This LabVIEW vi was developed to exercise the analog functions of the P2, incorporating HyperRAM access for long record data acquisition. From the front panel the user can set the number of points in the data acquisition and the ADC sampling period. For each “point”, 3 samples are recorded – ground (GIO), signal (PIO) and Vcc (VIO). The vi displays each of these measurements, as well as various histograms and the corrected analog signal. These graphical displays have proven very useful in looking at ADC performance and in seeing correlations between the different measurements. During the course of this work some quite significant differences in analog performance depending on which of the P2-EVAL’s pins is being used has been noted; these findings will be reported separately. It would be very useful to repeat these tests on other P2-EVAL’s here in Melbourne at some stage (Tubular, OzPropDev).

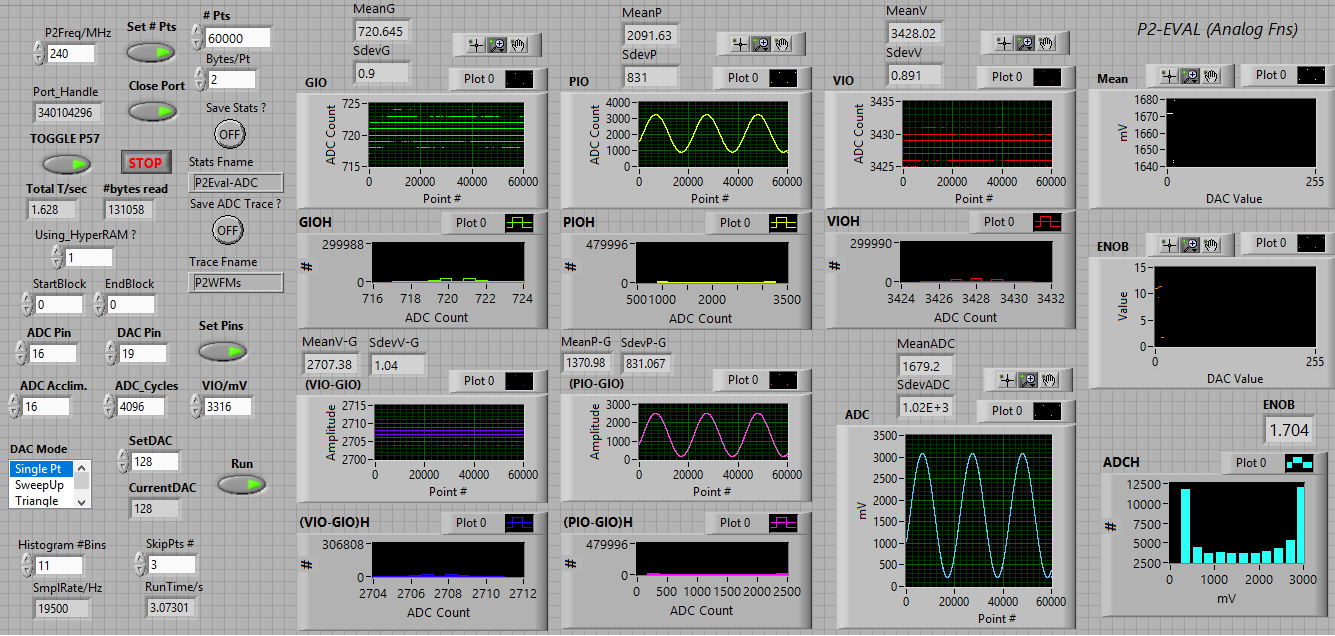


Fig. 2 The LabVIEW vi for evaluating analog/HyperRAM performance

Initially, my ADC sampling routine stored each data sample as two consecutive bytes into HUB RAM. Given the P2's HUB RAM space this limited the length of data records to around 80000 points, or 240000 samples. The code has subsequently been modified to make use of HyperRAM in an “on-the-fly” fashion as will now be described.

For analog input the P2 chip allows any one of its I/O pins to become a “smart pin” that can be configured as an ADC. After a pre-determined sampling period an ADC sample complete interrupt occurs and each ADC sample is then stored into HUB RAM. After every 64 such samples (128 bytes) the resulting data buffer is copied into a HyperRAM write buffer (see below for details) and written to HR. After this the HR block number is incremented by 1 and the HUB RAM buffer pointer is reset. This all takes place without disturbing the ADC reading sequence and allows up to 4M ADC samples to be acquired by the P2-ES.

Reads and writes from/to the HyperRAM are currently performed in blocks of 128 bytes. To simplify HyperRAM interactions, data is moved to/from HyperRAM via dedicated WR and RD buffers that are each located in HUB RAM at fixed addresses. Before any HyperRAM transaction, an address is calculated from the block number and this is placed into HUB RAM ahead of the actual data in the buffer space prior to actual performing the RD/WR operation.

In order to make effective use of both HyperRAM and HUB RAM a suite of commands in a monitor COG has been implemented to allow movement of data between these areas as well as to/from the LabVIEW host vi. Here is a brief description of these commands :

*LabVIEW <---> HUB RAM transfers*

A W command, followed by 128 bytes of data is used to fill the HR write buffer from the LabVIEW host.

An X command will cause an upload of 128 bytes from the HR read buffer back to the LV host.

A Cn, command is used prior to moving n bytes from LabVIEW to the main HUB RAM buffer - whose start address is $0B00.

A Un, command sets up a transfer of n bytes from the main HUB RAM buffer back to the LabVIEW host (maximum n=65536).

*HUB RAM to HyperRAM WR buffer transfer (128 bytes = 1 block)*

To prime the HyperRAM write buffer with 128 bytes of data, a fast routine called HUB\_LUT\_HUB is invoked. Prior to calling this routine PTRA and PTRB are set to the start addresses of the source block in HUB RAM and the start address of the HR WR buffer (also in HUB RAM). This provides a means to quickly move data located anywhere in HUB RAM into the HyperRAM write buffer.

HUB\_LUT\_HUB first copies the data block as 32 longs = 128 bytes into the COG's LUT area and then from the LUT into the HR WR buffer.

*HyperRAM to HUB RAM transfers (64k bytes)*

A un, command moves a 64k byte region of HyperRAM (as 512 contiguous 128 byte blocks) commencing at block number n from HyperRAM to the main HUB RAM buffer space starting at $0B00. Once this command has completed, a Un command allows that same data to be transferred back to the LabVIEW host. By issuing a sequence of un, Un, commands the entire contents of HyperRAM can be uploaded to LabVIEW.

**Timing information**

The HR\_write and HR\_read routines developed here each take ~1500 nsec to transfer a 128 byte buffer. This corresponds to a data transfer rate (once the data is in the RD/WR buffer) of 85.3 MB/sec. The 128 byte write buffer is located at address HRbuf\_WR2 while the corresponding read buffer commences at address HRbuf\_RD3. Read/write transactions move data between these buffer spaces and a HyperRAM block # which can be 0...65535. This block number must be specified prior to calling the HR transfer routines.

Here are some more precise timings determined for the calls to the various routines :

HR\_write 1.47 usec

HR\_read 1.54 usec

hub\_lut\_hub 400 nsec

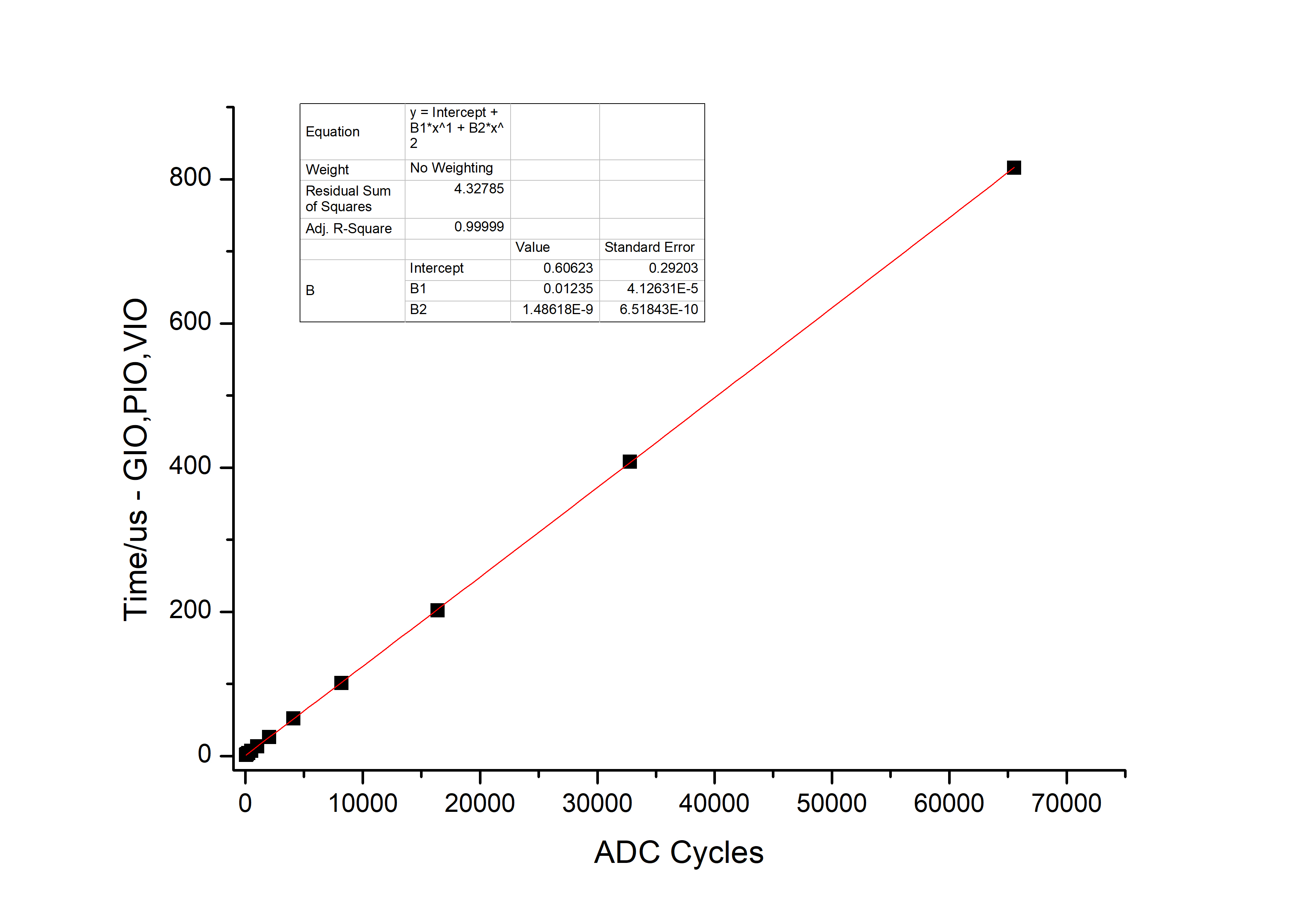
HR\_hub\_64k 1.16 msec

hubuploadn 218 msec for 64k bytes

Using this data one sees that to move a 64k block of data from HR to HUB (which requires use of both HR\_read and hub\_lut\_hub routines) a more realistic estimate of actual performance is 56.5 MB/sec.

The HyperRAM code (which includes the initialization, read and write routines) run in a dedicated cog. A separate monitor COG handles all of the afore-mentioned letter commands, while a mailbox facility implemented using a smart pin (long repository mode) passes a block number from the latter to the former. HR reads and write operations are then triggered by issuing a COGATN command from the monitor COG.

With the significant additional memory space afforded by HyperRAM, one can record ADC data over quite long periods of time. Fig. 3 shows a plot of sampling time vs the number of ADC cycles taken per sample measured using the P2-EVAL/HyperRAM combination. For example, for 14 bit sampling (cycles = 16384; actual ENOB would of course be less than 14 bits) the sampling rate is ~ 5KHz, generating 10 kilobytes per second of data. An 8MB memory capacity in this example allows for up to 13 minutes of data acquisition.

Fig. 3 ADC sample time vs # of ADC Cycles for a P2 running at 240MHz

I am currently working on using the P2-EVAL for ECG and pulse oximetry measurements and hope to be able to describe progress on these fronts in the coming months. This project is also involving some wireless telemetry using an ESP32.