



Chell Instruments Ltd
Folgate House
Folgate Road
North Walsham
Norfolk NR28 0AJ
ENGLAND

Tel: 01692 500555
Fax: 01692 500088

CANdaq

Hardware Trigger Functionality & Timing

TECHNICAL PAPER

e-mail:- info@chell.co.uk

Visit the Chell website at:
<http://www.chell.co.uk>

900118-1.0

INDEX

SECTION 1 - INTRODUCTION	1
SECTION 2 - FUNCTIONALITY	2
2.1 Summary	2
2.2 Detailed Breakdown	2
SECTION 3 - TIMING	4
3.1 Description.....	4
3.2 Initial summary	4
3.3 Timer error	5
3.4 Testing.....	5

Section 1 - Introduction

The CANdaq Mk3 and QUADdaq data acquisition systems include a feature enabling acquisition of data on reception of an external hardware trigger pulse. The purpose of this paper is to detail this hardware trigger functionality at the technical level, including timing relationships between the trigger pulse and the data acquisition, to give a better understanding of the determinism and uncertainty of the hardware trigger timing.

Section 2 - Functionality

2.1 Summary

The external hardware trigger input is a typical 5V TTL square wave pulse train. The minimum frequency of this pulse is 2Hz and the maximum is dependant on the number of channels being sampled, as dictated by Table 1.

Channels	16	32	48	64
Max Frequency	1KHz	625Hz	416Hz	312.5Hz

Table 1 – Max Hardware Trigger rate

On the reception of the trigger edge (positive going edge), the system reads the current data of each of the channels being sampled and builds a transmit buffer of the data. When all channels have been read the system then looks to send the buffer out of the appropriate comms channel (RS232, TCP or CAN, depending on where the hardware trigger enable came from). In the case of TCP and CAN, this output will not necessarily occur at the end of every channel cycle. Instead a buffer of a number of cycles is built, the size of which is dependant on the speed of the hardware trigger. This is due to the very high data rates that can be used with TCP & CAN comms. When the appropriate number of cycles has been stored, the whole buffer is transmitted at one go.

The hardware trigger mode is enabled and disabled via a comms command and also drops out of its own accord if no pulse is received within 500ms of the previous pulse. Additionally, with TCP comms, the hardware trigger mode will drop out if it is ready to build a new transmit buffer of data when the previous buffer hasn't been successfully sent yet (eg. due to a large number of retries because of excessive traffic on the TCP network). In this case the user is informed of the drop out by a flashing hardware trigger led and a bit set in the status word.

2.2 Detailed Breakdown

(a) Normal Operation

In 'normal' operation (no hardware trigger, calibration, table rebuild, setup mode, etc.), the data acquisition is free running, off a 50us timer interrupt. At initialisation, the ADC inputs are set to look at channel 0 by setting the scanner index to 0. This index is used to set a port register which actually sets the scanner address lines A5-A0 using standard binary addressing (e.g. for channel 0 set all to 0; for channel 32 set A5 = 1; for channel 63 set all to 1).

When the timer interrupt occurs the data index (used for storage of the read data) is set to the current scanner index and the scanner index is incremented. If the max configured channels is reached, the index simply wraps to 0. An ADC conversion is then initiated and the ADC interrupt is enabled. The ADC inputs are then set to look at the new scanner index (next channel), ready for the next conversion to take place on the next timer interrupt.

When the ADC interrupt occurs, the data associated with the data index is available to be read out of the ADC. The temperature is read first, followed by the pressure. Appropriate filtering and calibration values are applied and the resultant data is then stored in the appropriate array at the data index.

Then the I/O and housekeeping modules are executed. At this point the modules for RS232, TCP & CAN output are invoked if they have been enabled. If streaming mode is disabled then these comms modules will also be disabled, so no transmit buffer will be created and no output will occur.

The whole process then starts again on the next timer interrupt.

If streaming mode is enabled, then whatever data is currently stored for each channel is used to build a transmit buffer which will eventually be sent out of the appropriate comms. Depending on the data rate set for streaming and the comms protocol used for transmission, several cycles of all channels may be acquired and stored before the transmit buffer is sent out of the comms channel. This is to limit network usage and interrogation on fast data rates with TCP & CAN comms.

Once the transmit buffer has been sent, the comms module is disabled but is re-enabled automatically when the correct amount of time has passed as dictated by the rate of streaming set by the user.

(b) Hardware Trigger operational exceptions

When hardware trigger mode is enabled the scanner index is locked at 0, so the free running 50us timer is always forcing acquisition from channel 0. This is so that when the first hardware trigger occurs, we can be sure that the first data acquired is from channel 0.

When a hardware trigger occurs, a flag is set so that the 50us timer becomes free to increment the scanner index as per normal operation. The trigger interrupt also sets the appropriate enable flag for the comms module (e.g. if hardware trigger was enabled using a TCP command then the TCP module is enabled at this point). This means that during the I/O and housekeeping phase of normal operation, the transmit buffer will be built up and transmitted accordingly. Once the transmit buffer has been sent, the enable flag will be cleared until the next trigger interrupt occurs.

The acquisition phase functions as per normal operation, until data from all channels has been acquired. At that point, the scanner index is again locked at 0 so that the system is acquiring data from channel 0 when the next hardware trigger interrupt occurs.

Section 3 - Timing

3.1 Description

This section deals with the timing relationship between the hardware trigger pulse and the actual data acquisition. This will show how synchronised the data is with the hardware trigger pulses and also the uncertainty involved with the hardware trigger mechanism.

In this section we assume an example trigger set up at 10Hz on a system acquiring 64 channels of data. For completeness we also assume we are outputting to the TCP comms, although we are not bothered about the actual output transmission and when it occurs (this would involve understanding of the pre-determined calculation of transmit buffer size based on the number of cycles to perform, based on data rate, which is beyond the scope of this paper). Figure 1 shows a timing diagram to accompany the following information.

As we know from Section 2, the acquisition of data from a channel occurs every 50us. This means it takes a total of $50\text{us} \times 64 = 3200\text{us}$ or 3.2ms to acquire all channels in our example.

From the point at which the hardware trigger is enabled, the acquisition machine is locked to acquiring data from channel 0. This is so that when the actual trigger occurs we can immediately say that the first data available after the trigger is the first channel. To ensure that the ADC is locked at acquiring channel 0, there is a minimum requirement of a 50us wait from the point at which the hardware trigger is enabled to the point at which the first trigger is received. During this time any pulses received cause old data to be acquired.

When the positive edge of the first trigger is detected, the associated interrupt routine clears the 'locking' flag and enables the appropriate comms channel (TCP in this example). The acquisition machine is now free to acquire data from each of the 64 channels, in order from 0 to 63, at 50us intervals. It should be noted that this first interrupt point is the only point at which we have a level of uncertainty. This is because we cannot be certain at which point we are through the previous ADC interrupt when the trigger interrupt occurs. We could be right at the start which would mean channel 0 data would be available virtually immediately, or we could be just after the point at which the channel increment occurs (assuming the locking flag is cleared) which would mean channel 0 data will be available at the next ADC interrupt (i.e. in 50us)

As already mentioned the total acquisition time for 64 channels will be 3.2ms, at which point the acquisition machine is then locked to channel 0 ready for when the next hardware trigger occurs.

So from this we can always determine the time to acquire a particular channel from the point at which the positive edge of the trigger occurs within 50us.

3.2 Initial summary

If hardware trigger positive edge received at $t = 0\text{us}$.

The data from channel n is taken at $t = (n \times 50\text{us}) + 25\text{us}$ with an error of $\pm 25\text{us}$

...where n is a channel number from 0 to 63

This however assumes we have an exact 50us timer.

3.3 Timer error

In reality, the 50us timer has an error of $\pm 0.08\mu\text{s}$.

To give a more accurate formula of uncertainty and error we calculate as follows:

In a worst case situation, taking into account the above error, it takes 50.08us to acquire data for channel 0. Because the timer error is compound it could therefore take $50.08 + (0.08 * 64)\mu\text{s}$ to acquire channel 63. This gives us a worst case time certainty of 55.2us for 64 channels, 52.64us for 32 channels and 51.36 us for 16 channels.

To express this as a time error = $25\mu\text{s} \pm 0.04 \pm (\mathbf{x}/2)\mu\text{s}$

...where \mathbf{x} is the time certainty shown above for 64, 32, or 16 channels.

Applying this to the original formula gives us:

If hardware trigger positive edge received at $t = 0\mu\text{s}$.

The data from channel n is taken at $t = (n * 50\mu\text{s}) + 25\mu\text{s}$ with an error of :

$\pm 27.64\mu\text{s}$ (64ch), $\pm 26.36\mu\text{s}$ (32ch), $\pm 25.72\mu\text{s}$ (16ch)

...where n is a channel number from 0 to 63, 31 or 15 accordingly.

This uncertainty can be reduced further by actually measuring the timer period for any one CANdaq unit. This would remove the compound timer uncertainty because the actual timer period would be known and would be the same no matter what channel is being acquired.

3.4 Testing

To test the timing and synchronicity of the hardware trigger and data acquisition we can perform a few measurements as follows in a worked example:

We can measure the total time taken to acquire 64 channels of data by breaking into the scanner address lines and scoping at line A5. When this address line goes high the scanner is addressing channel 32 and when it goes low again, the scanner is addressing channel 63. At the same time we can scope the hardware trigger and measure the time from the trigger edge to both the rising and falling edges of line A5. This gives us a time for acquiring 32 and 64 channels of data. In our example we get 1.6ms for 32 channels and approx. 3.2ms for 64 channels, which works out at 50us per channel.

As mentioned earlier, there is a slight timing error in the 50us timer. This comes about due to a number of reasons including the instruction execution time, the tolerances in components around the trigger and acquisition circuits and also the tolerance of the crystal clocking the micro controller.

This can be measured by using a high resolution timing counter connected to the A0 scanner address line. This line toggles every time an acquisition occurs, no matter what the channel setup is. By connecting the counter to this it will tell us how long it takes for the A0 line to pulse (that is a low to high to low transition). This gives a time for acquisition of two channels (remember an acquisition occurs when the A0 line goes from low to high and also from high to low), so dividing the result by two gives us a time per data acquisition.

In our example we see that the counter gives us readings of around 99.917us (give or take a couple of nanoseconds). This gives us a 50us timer of 49.959us, which is well within the $\pm 0.08\mu\text{s}$ window given in the timing section above (3.3).

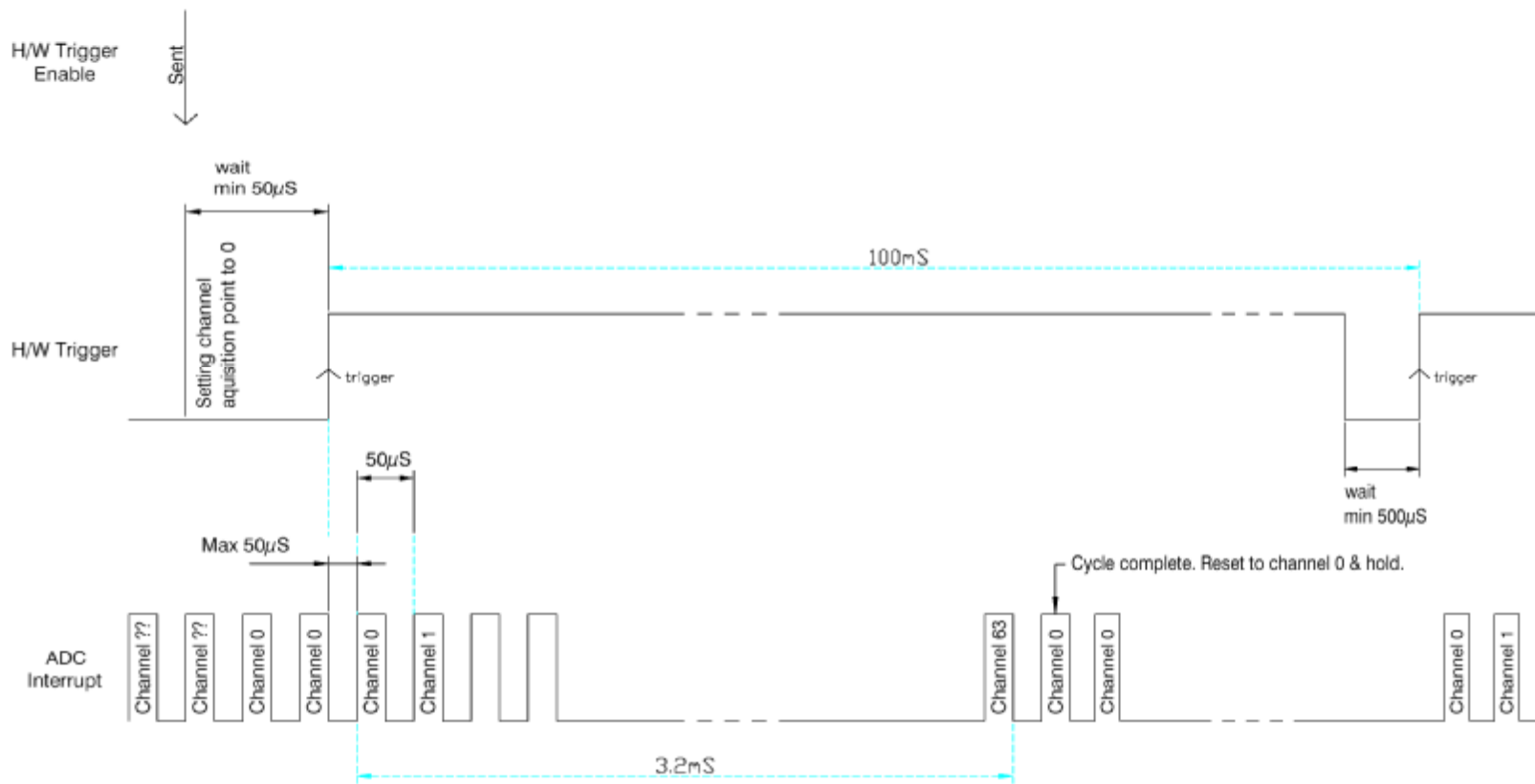


Figure 1 – Hardware Trigger timing diagram