Self Training Guide

DLP Module 05 – Pulse Counting

Introduction

A common task that will be encountered by an intrepid DLP programmer will be that of accumulating pulses that are presented to a digital input on an RTU. Usually these pulses represent a given volume of water flowing through a flow meter, but they can also be used to indicate energy consumption, rainfall, wind speed and various other quantities. The same techniques used to count physical pulses can also be used to count other slower events such as the number of times per day a given pump starts up.

This document discusses the techniques used to accumulate pulses on an RTU and shows examples of the two main types and when and where they should be used.

Before reading this document, you should have read the previous module(s) and be comfortable with the concepts discussed within. This document also assumes that you be familiar with the Q90 configuration software, and have successfully installed the DLP IDE software.

Additional details on the syntax of all DLP commands can be found in the online help.

In this document any DLP commands are presented in BLUE TYPEFACE while all DLP system variables and IO registers are in RED TYPEFACE.

The .ASM file for any DLP shown in this document is available separately.

This module contains help on:

- Conventional edge-detect pulse counting.
- Pulse Analog (or PANL) and PDIG registers.
- Daily resets of pulse counters. CKSEC, CKMIN, CKHOUR, CKDOW, CKDATE, CKMTH and CKYEAR.

QTech Data Systems Ltd
Unit 6 Amuri Park
404 Barbadoes Street
Christchurch
New Zealand

Phone: +64 3 3663713
Fax: +64 3 3652815
E-mail: sales@qtech.co.nz
Web www.qtech.co.nz

Conventional Pulse Counting (Edge-Detect)

'Edge Detect Pulse Counting', or 'Conventional Pulse Counting' is a simple technique in which all of the work to count pulses is done in the DLP. In older model RTUs, where the DLP cycle time was once every 100 ms, this worked quite well as it meant pulses could be counted as long as the pulse duration was at least 100 ms. In later model RTUs where the amount of functionality has vastly increased, the DLP cycle time has been slowed down to once every 500 ms to make room. This means that the conventional method of pulse counting is no longer as widely used as it once was, but it is still useful for counting events such as pumps running that have slow cycle times.

The theory is that at the end of each DLP cycle we use a spare digital register to "remember" what the current state of the digital input we are monitoring is. The next time the DLP executes it will compare this spare digital "memory" with the current state of the actual input. If the two values are different, that means there has been a change of state in the physical input, this may be a rising edge *or* a falling edge. Counting both will mean we end up with double the number of pulses that we should have, so we need to be careful only to count either the rising edge or the falling edge.

Module05-Ex01

```
001 ;*********************************
002 ;*  DLP Self Training
003 ;*  Module 5 Example 1
004 ;*  (c) QTECH DATASYSTEMS 2010
005 ;*********************************
006
007 proginit
008
009 ;   Real Analogue Inputs
010     equ   rdin1     rdi_Flow_Pulse
011
012 ;   Notional Analog Inputs
013     equ   nain1     nai_flow_total
014
015 ;   Spare Digitals
016     equ   spd1      spd_Flow_Pulse
017
018
019     cosmask  0    2,3,4,5,6,7,8         ; Tell the RTU we dont want RDI 1 causing a COS message
020
021 progstart
022
023 begin
024     tce       rdi_Flow_Pulse                    ; If the flow pulse input has gone high
025     tcd       spd_Flow_Pulse                    ; But the spare digital has not gone high yet
026     incanl    nai_flow_total                    ; There must have been a rising edge on RDI 1,
027                                                 ; so increment the flow pulse
028
029 begin
030     tce       rdi_Flow_Pulse                    ; Copy the value of the real digital input and
031     ec        spd_Flow_Pulse                    ; store it in the spare register for next time around.
032
033 progend
```

Example 1

Example 1 shows a DLP that counts pulses on RDIN1. It compares RDIN1 with SPD1 and if finds that RDIN1 is in the high state while SPD1 is in the low state, the pulse count is incremented. Immediately after this check is made, the DLP makes SPD1 equal RDIN1.

If we wanted to count pulses on a negative edge instead, we would check for RDIN1 to be in the low state while SPD1 was high.

This same technique can be used to count other events that would not conventionally be thought of as 'pulses' such as pump runs etc.

Pulse Counting using the built-in Pulse Analog (PANL) registers.

To overcome the difficulties in counting pulses in RTUs where the DLP only runs twice per second, the Q03 firmware has been enhanced so that the DLP can ask the firmware to do the hard work for it.

When this feature is used, the firmware will maintain a pulse count for each of the digital inputs on an RTU so that the DLP can read the total at any time. The firmware will automatically set the total count back to zero after each time the register is read. Effectively this register will therefore contain the number of pulses that have occurred since the last time the register was read.

These registers are called Pulse Analog (or PANL) registers. By default the firmware will not be counting pulses and using a PANL register for each digital input, so there is a separate digital register associated with each input called the Pulse Digital (or PDIG) register. To enable PANL counting on a particular digital input, the programmer must set the associated PDIG register to True during the PROGINIT section of the DLP.

Module05-Ex02

```
001 ;*********************************
002 ;*  DLP Self Training
003 ;*  Module 5 Example 2
004 ;*  (c) QTECH DATASYSTEMS 2010
005 ;*********************************
006
007 proginit
008
009 ;   Real Analogue Inputs
010     equ   rdin1     rdi_Flow_Pulse       ; The physical flow pulse input
011
012 ;   Notional Analog Inputs
013     equ   nain1     nai_flow_total       ; The flow total transmitted back to the base station
014
015 ;   Pulse Analogs
016     equ   panl1     pan_Flow_Count_Raw   ; The pulse analog register for the flow pulse wired into RDIN 1
017
018 ;   Spare Analogs
019     equ   spa1      spa_flow_temp        ; A temporary register to use when reading the Pulse Analog.
020
021
022     initdig     pdig1     true           ; Turn on pulse counting on RDIN 1
023
024
025     cosmask  0    2,3,4,5,6,7,8          ; Tell the RTU we dont want RDIN 1 causing a COS message
026
027 progstart
028
029 begin
030     cpanl     pan_Flow_Count_Raw    spa_flow_temp  1        ; Copy the Pulse Analog value to the temporary spare register.
031                                                               ; (This has the effect of resetting the PANL back to zero)
032     addr      nai_flow_total        spa_flow_temp            ; Add the value we just got from the PANL to the running total
033     cpanl     anlacc1               nai_flow_total 1         ; Copy the result into the register that holds the running
034                                                               ; total.
035
036
037 progend
```

Example 2

Daily Resets

Once you have created your pulse counters and have them working properly you will find that you will need to reset them periodically, if not the total count will most likely exceed the maximum number the DLP can deal with (65535) in a matter of days.

Most often the reset of pulse counters occurs at midnight each day, or at midnight of a certain day every week, or at midnight on the first day of a month. These different reset times will give daily, weekly, or monthly totals at the base station.

Historically the way this was accomplished was to transmit a signal from the base station on an NDOUT and have the DLP respond accordingly. However, this causes a traffic jam

on the comms channel as all sites that need to do a reset tend to want to do them simultaneously.

Recent techniques use resets that are triggered internally in the DLP using the RTU real time clock registers. This way several sites can all carry out a midnight reset simultaneously without cluttering up the radio comms.

There are several real time clock registers that may be referenced inside a DLP, these are:

CKSEC – The current value of the seconds register (0-59)
CKMIN – The current value of the minutes register (0-59)
CKHOUR – The current value of the hour register (0-23)
CKDOW – The current day of the week (1-7 = Sunday to Saturday)
CKDATE – The current date. (e.g. if it was the 5th of December, CKDATE = 5)
CKMONTH – The current value of the month of the year register (1-12)
CKYEAR – The last two digits of the current year. (e.g. in 2012, CKYEAR = 12)

By testing one of more of these registers to see if they equal a certain value, or by waiting for one or more to change state, we can cause an event (like a pulse count reset) to occur at any time we choose.

Module05-Ex03

```
001 ;**********************************
002 ;*  DLP Self Training
003 ;*  Module 5 Example 2
004 ;*  (c) QTECH DATASYSTEMS 2010
005 ;**********************************
006
007 proginit
008
009 ;   Real Analogue Inputs
010     equ   rdin1     rdi_Flow_Pulse        ; The physical flow pulse input
011
012 ;   Notional Analog Inputs
013     equ   nain1     nai_flow_total        ; The flow total transmitted back to the base station
014     equ   nain2     nai_yesterday_flow_total
015
016 ;   Pulse Analogs
017     equ   panl1     pan_Flow_Count_Raw    ; The pulse analog register for the flow pulse wired into RDIN 1
018
019 ;   Spare Analogs
020     equ   spa1      spa_flow_temp         ; A temporary register to use when reading the Pulse Analog.
021     equ   spa2      spa_ckdow_last        ; A temporary register to detect a change in CKDOW between executions.
022
023
024     initdig     pdig1     true            ; Turn on pulse counting on RDIN 1
025
026
027     cosmask  0    2,3,4,5,6,7,8           ; Tell the RTU we dont want RDIN 1 causing a COS message
028
029 progstart
030
031 ; Pulse Counting Code
032
033 begin
034     cpanl     pan_Flow_Count_Raw   spa_flow_temp  1        ; Copy the Pulse Analog value to the temporary spare register.
035                                                             ; (This has the effect of resetting the PANL back to zero)
036     addr      nai_flow_total       spa_flow_temp           ; Add the value we just got from the PANL to the running total
037     cpanl     anlacc1              nai_flow_total 1        ; Copy the result into the register that holds the running
038                                                             ; total.
039 ; Midnight Reset Code
040
041 begin
042     ner       ckdow     spa_ckdow_last                      ; If the day of the week is not the same as last time
043     jumpf     no_reset                                      ; then it must be midnight. JumpF jumps the next section
044                                                             ; of code if it is NOT midnight
045     cpanl     nai_flow_total     nai_yesterday_flow_total   1  ; Copy todays total to yesterdays register
046     initanl   nai_flow_total     0                          ; and then reset todays total back to zero.
047                                                             ; Initanl ALWAYS executes regardless of the state of the
048                                                             ; accumulator. Thats why we had to jump around this sectior
049 no_reset                                                    ; if ckdow had not changed.
050
051 begin
052     cpanl     ckdow     spa_ckdow_last     1               ; Copy the CKDOW register to remember what it was next time
053
054 progend
```

Example 3

Example 3 above builds on the code from example 2 and adds a midnight reset function. Note that because the midnight reset happens without any need to communicate with the base station, it is possible to miss the last few pulses of each day if only the “Total Today” is being reported. To get around this it is quite common to also include a “Total Yesterday” value, as shown.