Self Training Guide

DLP Module 02 – Setting up your DLP

Introduction

Now that the basics are understood, there are a few common configuration steps most DLPs require in order to work correctly. Choosing which digital inputs cause a Change-of-State (COS), how to map real IO to telemetry IO and vice versa, and how to debounce digitals that are not just being copied straight across.

This document discusses the steps necessary to achieve these goals

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:

- COSMASK and NCOSMASK
- The CPDIG and CPANL commands.

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

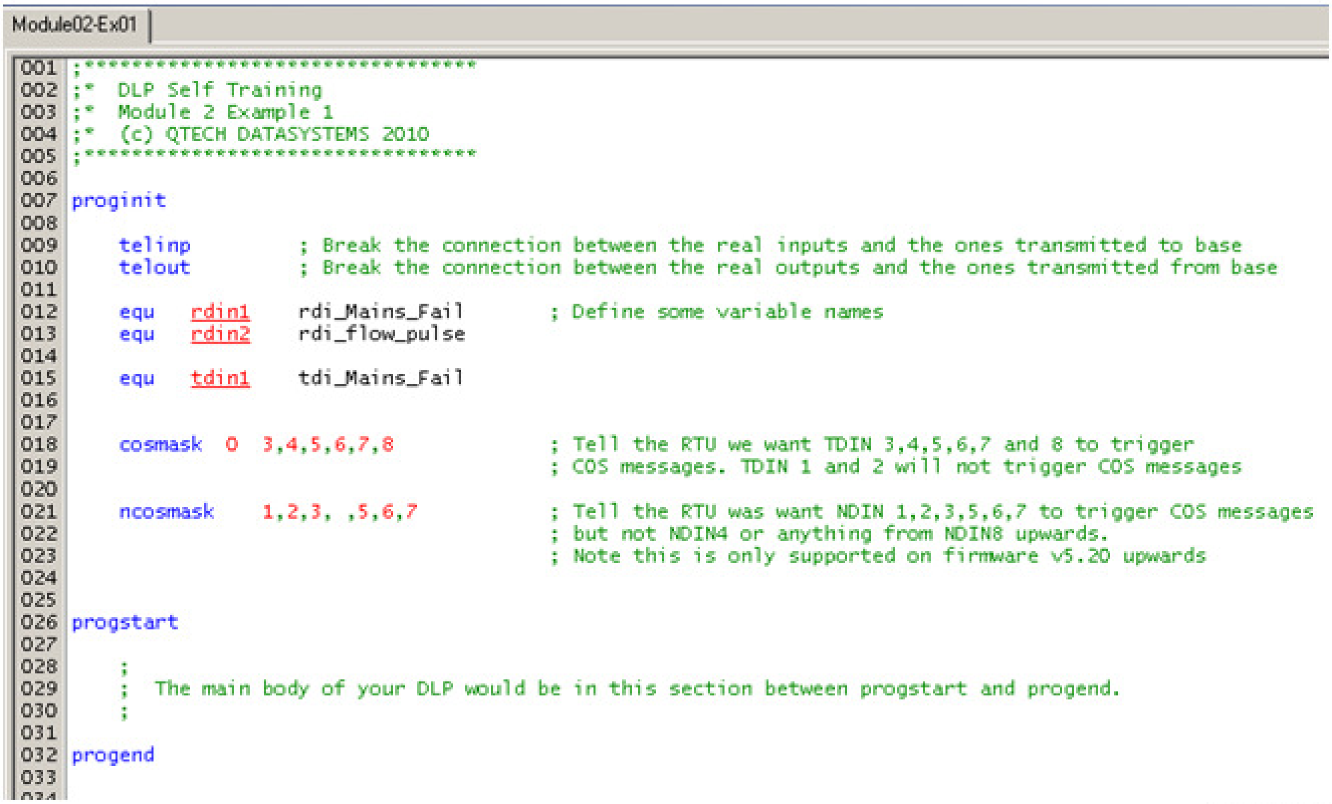
COSMASK and NCOSMASK

When no DLP is loaded into an RTU, digital inputs will cause the RTU to transmit a Change-Of-State message to the base station every time they change. This COS message contains not only the information about the input that changed, but all of the input information for the whole RTU both digital and analogue, notional and telemetry. It also contains several system flags such as the low battery alarm status.

It is not always desirable to allow ALL digital inputs to be able to trigger such behaviour. A flow pulse input into an RTU, for example, may be pulsing often enough to cause the RTU to send a continuous stream of COS messages over the radio network, causing massive overloading and communications problems for all other sites using the same radio channel.

Q90 has a COS configuration section that allows the user to specify which inputs should be allowed to trigger a COS message, but this requires the user to be on site with a laptop to change the configuration. This can be troublesome if changes need to be made at hard to reach sites, or when RTUs are replaced by field staff who may not have, or know how to use Q90.

To overcome these issues the DLP can include a COSMASK statement in the INIT section. This overwrites the COS configuration set up in Q90 and is carried out every time the INIT section of the DLP is executed.



```
;*********************************
;*  DLP Self Training
;*  Module 2 Example 1
;*  (c) QTECH DATASYSTEMS 2010
;*********************************

proginit

    telinp            ; Break the connection between the real inputs and the ones transmitted to base
    telout            ; Break the connection between the real outputs and the ones transmitted from base

    equu  rdin1    rdi_Mains_Fail       ; Define some variable names
    equ   rdin2    rdi_flow_pulse

    equ   tdin1    tdi_Mains_Fail


    cosmask  0  3,4,5,6,7,8             ; Tell the RTU we want TDIN 3,4,5,6,7 and 8 to trigger
                                        ; COS messages. TDIN 1 and 2 will not trigger COS messages

    ncosmask    1,2,3, ,5,6,7           ; Tell the RTU was want NDIN 1,2,3,5,6,7 to trigger COS messages
                                        ; but not NDIN4 or anything from NDIN8 upwards.
                                        ; Note this is only supported on firmware v5.20 upwards


progstart

    ;
    ;  The main body of your DLP would be in this section between progstart and progend.
    ;

progend

```

Example 1

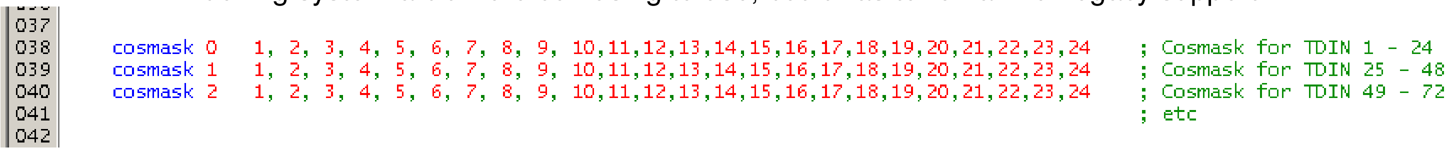
Example 1 above shows a DLP that contains, among other things, a COSMASK statement. An important thing to note about is that it references digital inputs in terms of a base index, and an offset.

Syntax : COSMASK [base index 0…10] [input list 1,2,3…..24]

The base index is a number from 0 to 10 and the input list is a comma separated list of all the inputs that *should* cause a COS message. If the input is not in this list, then it will not cause a COS.

The reason for the indexed addressing is that original DATRAN product range only had digital input cards available in multiples of 24 digital inputs, and the main QRTUs also had 24 digital inputs. Indexed references made things easier as the programmer only had to think of inputs in terms of "Input #6 on the 2nd expansion module".

Nowadays the product range is more comprehensive and modules with digital input counts of 24, 16 and 8 inputs are commonly in use, not to mention the ability to add PLCs or other devices that can have any number of digital inputs. This makes the indexing system a bit more confusing to use, but it has to remain for legacy support.



```
cosmask 0   1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18,19,20,21,22,23,24    ; Cosmask for TDIN 1 - 24
cosmask 1   1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18,19,20,21,22,23,24    ; Cosmask for TDIN 25 - 48
cosmask 2   1, 2, 3, 4, 5, 6, 7, 8, 9, 10,11,12,13,14,15,16,17,18,19,20,21,22,23,24    ; Cosmask for TDIN 49 - 72
                                                                                        ; etc
```

COSMASK applies to TELEMETRY digital inputs only. In the past, if the user wanted other I/O types such as analogues or notional digital inputs to trigger a COS, then specific pieces of code had to be written for each one to detect a change and then force a COS using the SENDCOS command. From firmware v5.20 onwards a new NCOSMASK command has been introduced so that notional digital inputs can also automatically trigger COS messages. The syntax for the NCOSMASK statement is very similar to COSMASK except there is no index, and the comma separated list goes all the way up to 255.

The NCOSMASK command is only supported in firmware version v5.20 and above. If a DLP containing the NCOSMASK command is transmitted into an RTU with a lower version of firmware, the DLP will be rejected by the RTU. This may mean that the base continually tries to transmit the DLP until the user intervenes.

CPDIG and CPANL

While we frequently need to disconnect the automatic link between the "Reals" and the "Telemetries" so we can delay or otherwise modify some I/O, most IO needs to be copied back to the base station without being modified.

TELINP and TELOUT are indiscriminate and will sever the link to all IO. There is no way of just specifying a few points, so we need a way of reconnecting the points we don't want to modify. That's where CPDIG (Copy Digitals) and CPANL (Copy Analogs) come in.

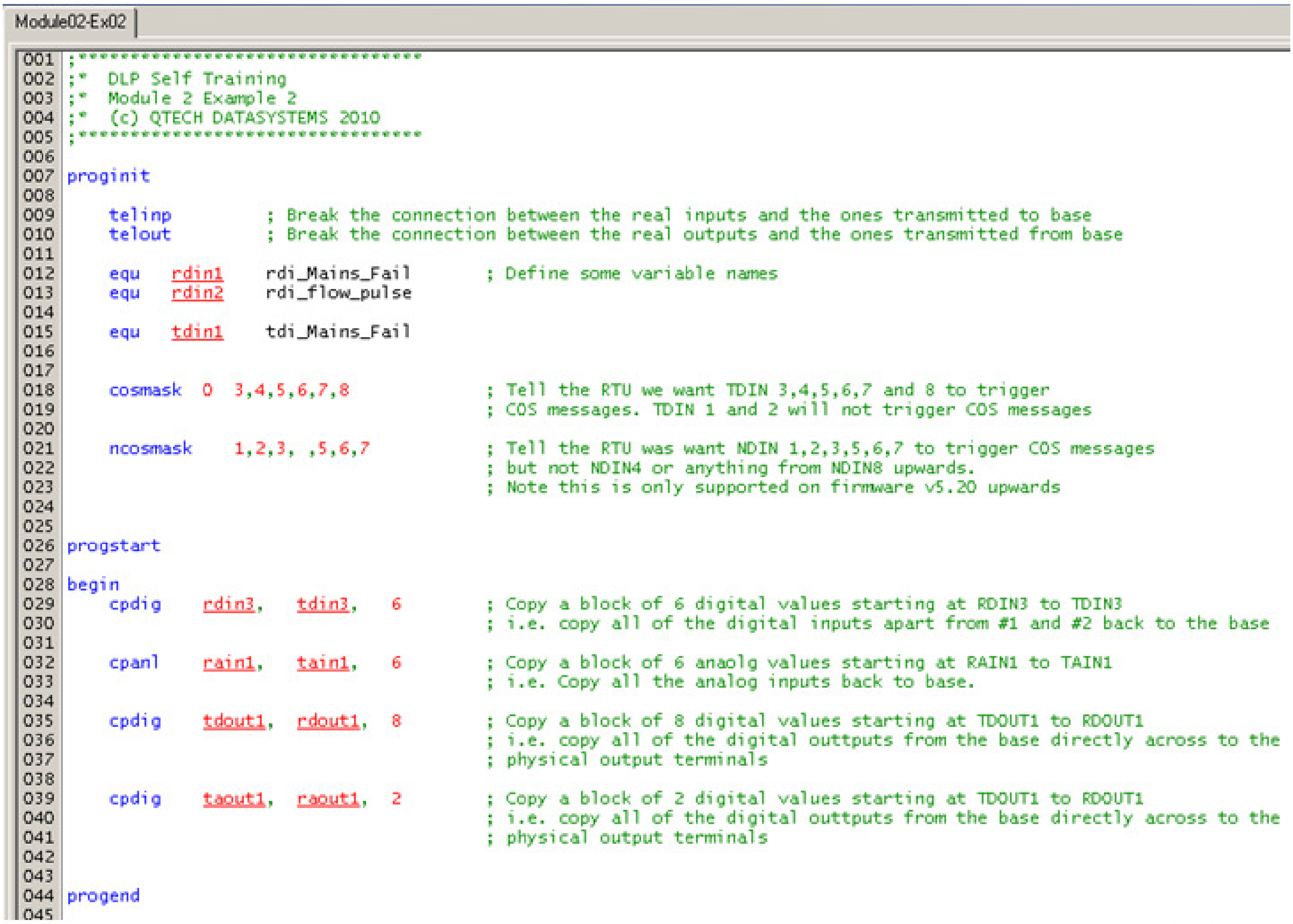
Both CPDIG and CPANL use the same arguments:

CPDIG   <SOURCE> <DESTINATION>   <Size of Block to Copy>

Where <SOURCE> is the starting register of the block of IO to copy, <DESTINATION> is the starting register of the block of IO to copy TO, and <Size of Block to Copy> specifies how many sequential pieces of IO should be copied.

e.g.: CPDIG   RDIN1   TDIN1   2

Will copy a block of two digitals starting at RDIN1 and will write them to a block starting at TDIN1. i.e. The value from RDIN1 will be written to TDIN1and the value from RDIN2 will be written to TDIIN2.



```
;************************************
;*  DLP Self Training
;*  Module 2 Example 2
;*  (c) QTECH DATASYSTEMS 2010
;************************************

proginit

    telinp          ; Break the connection between the real inputs and the ones transmitted to base
    telout          ; Break the connection between the real outputs and the ones transmitted from base

    equu   rdin1    rdi_Mains_Fail        ; Define some variable names
    equ   rdin2    rdi_flow_pulse

    equ   tdin1    tdi_Mains_Fail


    cosmask  0  3,4,5,6,7,8                ; Tell the RTU we want TDIN 3,4,5,6,7 and 8 to trigger
                                          ; COS messages. TDIN 1 and 2 will not trigger COS messages

    ncosmask    1,2,3, ,5,6,7              ; Tell the RTU was want NDIN 1,2,3,5,6,7 to trigger COS messages
                                          ; but not NDIN4 or anything from NDIN8 upwards.
                                          ; Note this is only supported on firmware v5.20 upwards


progstart

begin
    cpdig    rdin3,   tdin3,   6          ; Copy a block of 6 digital values starting at RDIN3 to TDIN3
                                          ; i.e. copy all of the digital inputs apart from #1 and #2 back to the base

    cpanl    rain1,   tain1,   6          ; Copy a block of 6 anaolg values starting at RAIN1 to TAIN1
                                          ; i.e. Copy all the analog inputs back to base.

    cpdig    tdout1,  rdout1,  8          ; Copy a block of 8 digital values starting at TDOUT1 to RDOUT1
                                          ; i.e. copy all of the digital outtputs from the base directly across to the
                                          ; physical output terminals

    cpdig    taout1,  raout1,  2          ; Copy a block of 2 digital values starting at TDOUT1 to RDOUT1
                                          ; i.e. copy all of the digital outtputs from the base directly across to the
                                          ; physical output terminals


progend
```

Example 2

Example 2 shows the beginnings of out first DLP. Here we can see that the TELINP and TELOUT commands have been used to break the Real/Telemetry link, some of the IO has been equated, a COSMASK and an NCOSMASK have been defined and the CPDIG and CPANL commands have been used to reunite all the Reals and Telemetries except for RDIN / TDIN numbers 1 and 2.

Things to take note of:

- Clear commenting has been used throughout. There is no need to go to quite the level of detail contained in these training manuals, but accurate and adequate commenting in your DLP is important. It is particularly important for the next person who comes along after you and wonders what it was you were trying to achieve.
- The equated variable names have been prefixed with a 3 letter code to indicate what type of IO register they are. This is not mandatory, but you will find that debugging a complex DLP becomes a lot easier when you can tell what IO type a variable is just by looking at it.
- A gap has been left in the number list in the NCOSMASK statement. Again, this is not mandatory and the list could have been entered as 1,2,3,5,6,7. Leaving a placeholder as shown makes it easier to tell at a glance which points have been intentionally left out of a NCOSMASK or COSMASK statement.