# Self Training Guide

# DLP Module 04 – Analogue Setpoints and Alarms

## Introduction

DLPs will frequently need to respond to changes in analogue values, whether they be physical analogue inputs, derived analogue values or system analogue values such as the time of day.

This document discusses various commands available for comparing analog values with setpoints or constants, and builds on the digital controls discussed in module 3 to set an alarm, or turn digital outputs off and on as might be needed for pump controls.

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:

- GES and GER
- GTS and GTR
- LTS, LTR, LES and LER
- EQS, EQR, NES and NER

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

## GES and GER

All of the commands used for analogue comparison in the DLP are abbreviations of their actual function, which makes them easy to remember after the programmer has used them a few times.

Analogue registers can be compared with either hard-coded constants, or with other analog registers. In each case, the programmer must select the appropriate command as they are not interchangeable.

GES stands for "Greater than or equal to setpoint". This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is greater than, or equal to, the setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

GER stands for "Greater than or equal to register". This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is greater than, or equal to, the second register then the logic accumulator will be set to true, otherwise it will be set to false.

It can be seen that while these commands are very similar, it is the 'S' or the 'R' at the end of the command word that indicates whether the command should be used to compare to a constant or another register.

Example 1 on the next page shows a DLP where a reservoir level is being monitored and two things will happen when the reservoir gets full enough

- When the reservoir level is greater than, or equal to, the hard-coded level of 95% an indicator lamp connected to RDOUT1 will turn on to show the operator that the reservoir level is too high.
- When the reservoir level is greater than, or equal to, a 'Pump Stop Setpoint' which can be changed from the base station, then the DLP will make sure the control that runs the pump is turned off on RDOUT2

## GTS and GTR

GTS and GTR are very similar to GES and GER except that :

GTS stands for "Greater than setpoint". There is no "or equal to". This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is greater than the setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

GTR stands for "Greater than register". This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is greater than the second register then the logic accumulator will be set to true, otherwise it will be set to false.

Module04-Ex01

```
;**********************************
;*  DLP Self Training
;*  Module 4 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

;   Real Analogue Inputs
    equu   rain1    rai_Reservoir_Level              ; 4-20 mA = 0% to 100% full

;   Real Digital Outputs

    equ    rdout1   rdo_Reservoir_High_Indicator_Lamp
    equ    rdout2   rdo_Pump_Run_Control

;   Notional Analog Outputs
    equ    naout1   nao_Pump_Stop_Setpoint

initanl   nao_Pump_Stop_Setpoint  58981              ; Define a default pump stop level at 90% full
                                                     ; The DLP will use this value after a reset until
                                                     ; the base station transmits the updated value.

    cosmask  0  1,2,3,4,5,6,7,8          ; Tell the RTU we want all 8 TDINs to trigger a COS message

progstart

begin
    cpdig    rdin1,   tdin1,   8         ; Copy all 8 digital inputs back to base with no modification


; Create a high level alarm and turn on a visual indicator lamp if the reservoir level gets above 95% full
; 95% in terms of a raw 16 bit number is 65535 x 0.95 = 62258.25. The DLP does not know what to do with
; decimal points, so just round it off to 62258.

begin
    ges      rai_Reservoir_Level, 62258                 ; If the reservoir level is >= 95 %
    tmr      1,150                                      ;
    tce      timerd1                                    ; continuously for 15 seconds or more (debounce)
    ec       rdo_Reservoir_High_Indicator_Lamp          ; turn on the indicator lamp, else turn it off.

; By the end of module 4 this DLP will also fully control a pump, turning it off and on when the reservoir needs water.
; This part of the DLP is responsible for turning the pump off when the reservoir gets full enough.

begin
    ger      rai_Reservoir_Level, nao_Pump_Stop_Setpoint  ; If the reservoir level is >= the setpoint.
    resdig   rdo_Pump_Run_Control                         ; Then stop the pump from running.


progend
```

*Example 1*

## LTS, LTR, LES and LER

As we have a set of  commands to find out if a certain analog value is greater than another, or greater than a static value, we also have a set of commands to test for an analog register being less than another register or setpoint. These are named and work in much the same way as the “Greater than…..” commands we have looked at so far:

LES stands for “Less than or equal to setpoint”. This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is lower than, or equal to, the setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

LER stands for “Less than or equal to register”. This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is lower than, or equal to, the second register then the logic accumulator will be set to true, otherwise it will be set to false.

LTS stands for “Less than setpoint”. There is no “or equal to”. This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is lower than the setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

LTR stands for "Less than register". This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is lower than the second register then the logic accumulator will be set to true, otherwise it will be set to false.

Module04-Ex02

```
;*********************************
;*  DLP Self Training
;*  Module 4 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

;   Real Analogue Inputs
    equu   rain1     rai_Reservoir_Level               ; 4-20 mA = 0% to 100% full

;   Real Digital Outputs

    equ    rdout1    rdo_Reservoir_High_Indicator_Lamp
    equ    rdout2    rdo_Pump_Run_Control
    equ    rdout3    rdo_Reservoir_Low_Indicator_Lamp

;   Notional Analog Outputs
    equ    naout1    nao_Pump_Stop_Setpoint
    equ    naout2    nao_Pump_Start_Setpoint

initanl    nao_Pump_Stop_Setpoint   58981              ; Define a default pump stop level at 90% full
                                                       ; The DLP will use this value after a reset until
                                                       ; the base station transmits the updated value.

initanl    nao_Pump_Start_Setpoint   32767             ; Define a default pump stop level at 50% full
                                                       ; The DLP will use this value after a reset until
                                                       ; the base station transmits the updated value.

    cosmask  0  1,2,3,4,5,6,7,8          ; Tell the RTU we want all 8 TDINs to trigger a COS message

progstart

begin
    cpdig    rdin1,    tdin1,    8       ; Copy all 8 digital inputs back to base with no modification
    cpanl    rain1,    tain1,    6       ; Copy all 6 analogue inputs back to base with no modification


; Create a high level alarm and turn on a visual indicator lamp if the reservoir level gets above 95% full
; 95% in terms of a raw 16 bit number is 65535 x 0.95 = 62258.25. The DLP does not know what to do with
; decimal points, so just round it off to 62258.

begin
    ges      rai_Reservoir_Level, 62258                ; If the reservoir level is >= 95 %
    tmr      1,150                                     ;
    tce      timerd1                                   ; continuously for 15 seconds or more (debounce)
    ec       rdo_Reservoir_High_Indicator_Lamp         ; turn on the indicator lamp, else turn it off.

; Create a low level alarm and turn on a visual indicator lamp if the reservoir level gets below 30% full
; 30% in terms of a raw 16 bit number is 65535 x 0.3 = 19660.5. The DLP does not know what to do with
; decimal points, so just round it off to 19661.

begin
    les      rai_Reservoir_Level, 19661                ; If the reservoir level is <= 30 %
    tmr      2,150                                     ;
    tce      timerd2                                   ; continuously for 15 seconds or more (debounce)
    ec       rdo_Reservoir_Low_Indicator_Lamp          ; turn on the indicator lamp, else turn it off.

; Control the pump as according to the needs of the reservoir, based on its current level.

begin
    ger      rai_Reservoir_Level, nao_Pump_Stop_Setpoint  ; If the reservoir level is >= the setpoint.
    resdig   rdo_Pump_Run_Control                         ; Then stop the pump from running.

begin
    ler      rai_Reservoir_Level, nao_Pump_Stop_Setpoint  ; If the reservoir level is <= the setpoint.
    setdig   rdo_Pump_Run_Control                         ; Then stop the pump from running.


progend
```

*Example 2*

Example 2 shows the same DLP as Example 1, this time functionality has been added to turn this into a complete simple reservoir level controller.

The normal operation will be that when the reservoir level reaches the pump stop level, the pump will be stopped. Then, as the reservoir level falls, when it reaches the pump start level it will start the pump again. The start and stop levels will be able to be changed at any time from the base station via NAOUT1 and NAOUT2.

If during these operations something abnormal happens such as the pump failing to run or failing to stop, the reservoir level will reach either the low or the high level alarm levels. When this happens, the RTU will activate outputs that are connected to warning indicators on the reservoir. These could also be connected to sirens or strobe lights.

In addition, some NAIN registers could also be used to transmit the status of the high and low level alarms back to the base station to alert operators via the DATRAN SMS alarm system.

## EQS, EQR, NES and NER

Now that we can test if certain values are higher or lower than each other, we need a few more tools in our repertoire to complete the set.

EQS stands for "Equal to Setpoint". This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is exactly the same as the setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

EQR stands for "Equal to Register". This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is exactly the same as the second register then the logic accumulator will be set to true, otherwise it will be set to false.

NES stands for "Not Equal to Setpoint". This will take an analogue register and compare its value to a specific hard-coded setpoint. If the analogue register is different to setpoint then the logic accumulator will be set to true, otherwise it will be set to false.

NER stands for "Not Equal to register". This will take an analogue register and compare its value to another register as specified by the programmer. If the first analogue register is different from the second register then the logic accumulator will be set to true, otherwise it will be set to false.