DCmp_s_Demo

1.0.0

Features

- Demonstrates the CONSUL*TRON* **Digital Comparator component** (DCmp**)** features.
- Single character driven menu commands.
- Terminal device access to a Host computer.
- Demo project to run on the PSoC5

General description

The DCmp_s_Demo demonstrates the features of the DCmp component by using a Terminal communication to a Host computer. It uses the MenuCmds Component to create a menu command structure to exercise the DCmp API functions and features.

Features of the DCmp_s_Demo:

- There are menu commands to exercise API calls to set or get:
  - Increase or decrease data loading frequency
  - Change the comparison thresholds (1%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% and 99%)
- The DCmp operational parameters status.
- Multiple TopDesign variations* to demo.
  - Stimulus_ISR_ramp. The stimuli are two ramp counters loading into DCmp using the CPU.
  - Stimulus_DMA_ramp. The stimuli are two ramp counters loading into DCmp using DMA.
  - Stimulus_ADC. The stimuli are two ramp VDACs connected to two ADC_SARs loading the converted data into DCmp using DMA.

*Only one variation can be used at a time. To use a variation, Disable the schematic page of the current variation **FIRST** then enable the desired schematic page of the variation to be tested. Note: Leave the "Diag" schematic page enabled.
Each variation can be configured to demo as unsigned or signed comparisons.

DCmp_s_Demo Requirements

- PsoC Creator 4.2 or higher.
- The MenuCmds Component is tied to the standard UART in the PSoC design. **Device**

Families Supported

- *PSoC5LP*

Target Boards

CY8CKIT-059 board

The **DCmp_s_Demo** project was built for the CY8CKIT-059 board. If this board is used, NO ADDITIONAL external circuits are needed! The configuration uses an unaltered stock board.

The default pin assignments are:

- UART communications:
  - [UART type] Tx => P12[7]; Rx => P12[6]
- Output1 Comparison outputs
  - EQ1 = P2[0]
  - NEQ1 = P2[1]
  - LT1 = P2[2]
  - LTE1 = P2[3]
  - GT1 = P2[4]
  - GTE1 = P2[5]
- Stim1 cycle toggle = P2[6]
- Stim1 VDAC1 out = P2[7]
- Output2 Comparison outputs
  - EQ2 = P3[0]
  - NEQ2 = P3[1]
  - LT2 = P3[3]
  - LTE2 = P3[4]
  - GT2 = P3[5]
  - GTE2 = P3[6]
- Stim2 cycle toggle = P3[7]
- Stim1 VDAC2 out = P15[0]

Other PSoC5 boards can be used however you may have to select a different PSoC5 device and you may have to assign the Rx and Tx pins to other ports. (If the UART Type selected to be “USBUART”, the PSoC5 hard-codes the USB pins to 15[7] and 15[6].)

Menu Commands

Once the PSoC is successfully programmed, connect to a Terminal program on the Host computer using the assigned COM port. (If using the UART type, the connection spec is 115.2Kbps 8N1.

Here is a list of the currently supported menu commands.

```
?=> display FULL help list
h=> display minimized help list
b=> display Build info
s=> display current Status info
M=> Input mode Menu
     i=> One input (2 thresholds)
```

```
          I=> Two inputs (1 threshold each)
     f=> Stimulus 1 Frequency Select Menu
          ==> Increase Stimulus 1 frequency
          -=> Decrease Stimulus 1 frequency
     F=> Stimulus 2 Frequency Select Menu
          +=> Increase Stimulus 2 frequency
          _=> Decrease Stimulus 2 frequency
     t=> Threshold 1 Select Menu
          `=> Set Threshold 1 to 1%
          1=> Set Threshold 1 to 10%
          2=> Set Threshold 1 to 20%
          3=> Set Threshold 1 to 30%
          4=> Set Threshold 1 to 40%
          5=> Set Threshold 1 to 50%
          6=> Set Threshold 1 to 60%
          7=> Set Threshold 1 to 70%
          8=> Set Threshold 1 to 80%
          9=> Set Threshold 1 to 90%
          0=> Set Threshold 1 to 99%
     T=> Threshold 2 Select Menu
          ~=> Set Threshold 2 to 1%
          !=> Set Threshold 2 to 10%
          @=> Set Threshold 2 to 20%
          #=> Set Threshold 2 to 30%
          $=> Set Threshold 2 to 40%
          %=> Set Threshold 2 to 50%
          ^=> Set Threshold 2 to 60%
          &=> Set Threshold 2 to 70%
          *=> Set Threshold 2 to 80%
          (=> Set Threshold 2 to 90%
          )=> Set Threshold 2 to 99%
```

The single character menu command is to the left of the “=>” (case sensitive).

?=> display FULL help list

h=> display minimized help list

Displays the menu list.

b=> display Build info

Displays the App name, build date and time and Silicon ID.

s=> DCmp status

Displays the current operational parameters of the DCmp component and stimulus 1 and 2.

- The Stimulus type (schematic page) currently active.
- Signed or Unsigned comparisons.
- Bit size of the comparison.
- Number of inputs available and the number of Inputs in use.

- Stimulus 1 frequency.
- Stimulus 2 frequency. (If two input mode is selected.)
- Threshold 1 value.
- Comparison Output1 outputs enabled.
- Threshold 2 value. (If not signed compare and Output2 enabled.)
- Comparison Output2 outputs enabled. (If not signed compare and Output2 enabled.)

i=> One input (2 thresholds)
Set the input mode to one input. If using unsigned comparisons, two thresholds are available. Thresh1 ⇔ Data1 ⇔ Thresh2

I=> Two inputs (1 threshold each)
Set the input mode to two inputs. If using unsigned comparisons, one threshold is assigned to each input. Data1 ⇔ Thresh1, Data1 ⇔ Thresh2.

= => Increase Stimulus 1 frequency
+ => Increase Stimulus 2 frequency
Increases the Stimulus frequency. This is useful to find the upper limit of performance.

- => Decrease Stimulus 1 frequency
_ => Decrease Stimulus 2 frequency
Decreases the Stimulus frequency.

`,1,2,3,4,5,6,7,8,9,0=> Set Threshold 1 to xx%
~,!,@,#,$,%,^,&,,(,)=> Set Threshold 2 to xx%*
Set the Threshold to xx% listed.

Demo Variations

I've provided three examples of using the DCmp component to aid in your understanding on how to use it. In all variations, you can select one or two input stimuli, change threshold settings and the frequency of the stimuli. To use unsigned or signed comparisions, you need to make configuration changes in the component parameters then re-build.

Stimulus_ISR_ramp

This example uses the CPU through Interrupt Service Routines and DCmp component API calls to load the data into the component for comparison. This is the slowest and least effective method of comparison. Therefore it is not generally recommended. This example is present to more thoroughly test more of the API calls for this component.

Stimulus_DMA_ramp

This example uses the DMA to load the data into the DCmp component for comparison. This is the most effective method of comparison and therefore the recommended method.

Figure 1 – Stimulus DMA ramp @ Unsigned 32-bit 1 input 2 thresholds (Thresh1 = 10%, Thresh2 = 90%)

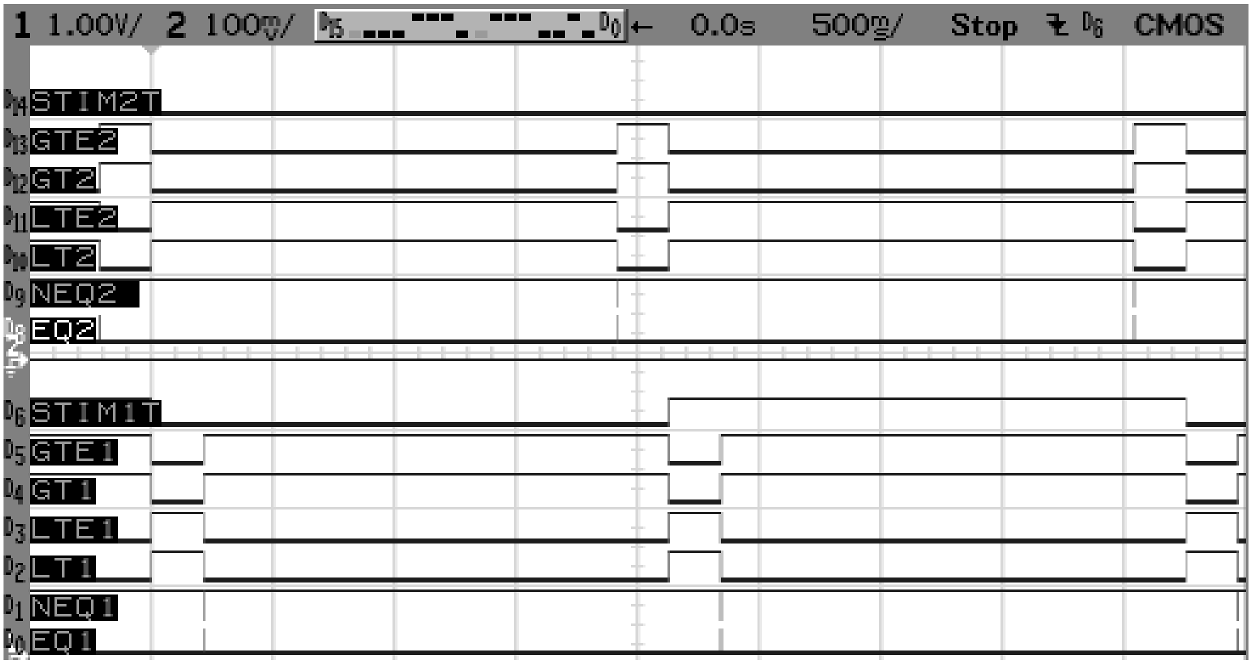


Figure 2 - Stimulus DMA ramp @ Unsigned 32-bit 2 input 2 thresholds (Thresh1 = 10%, Thresh2 = 90%)

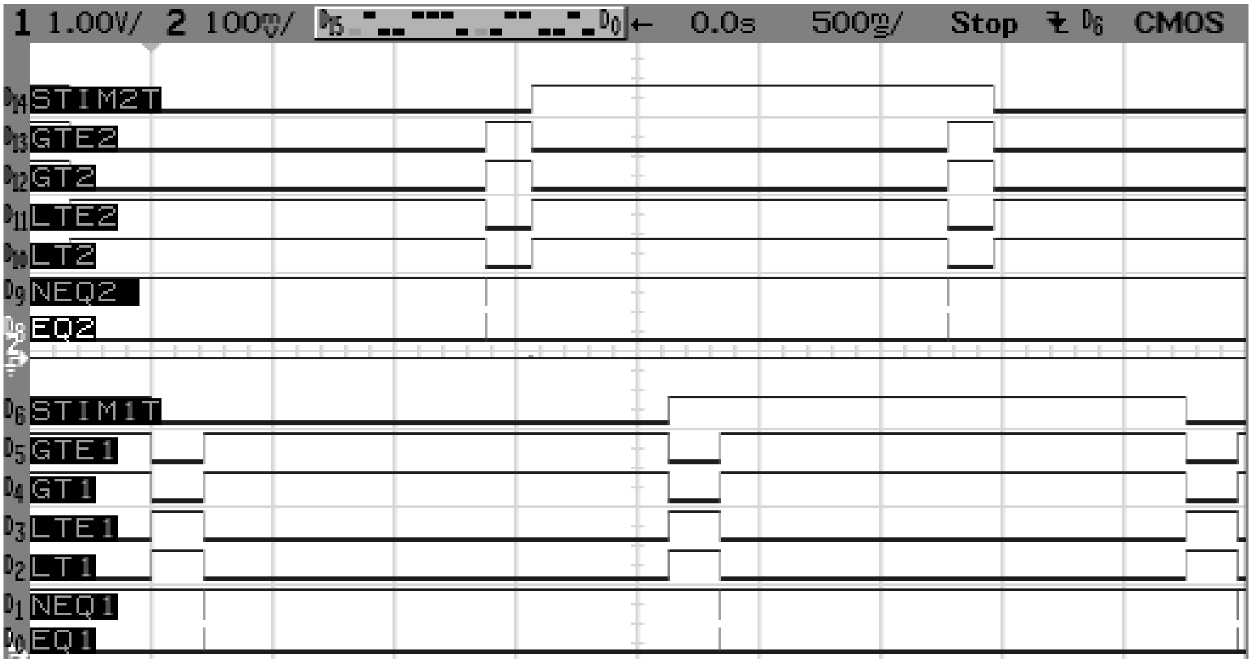
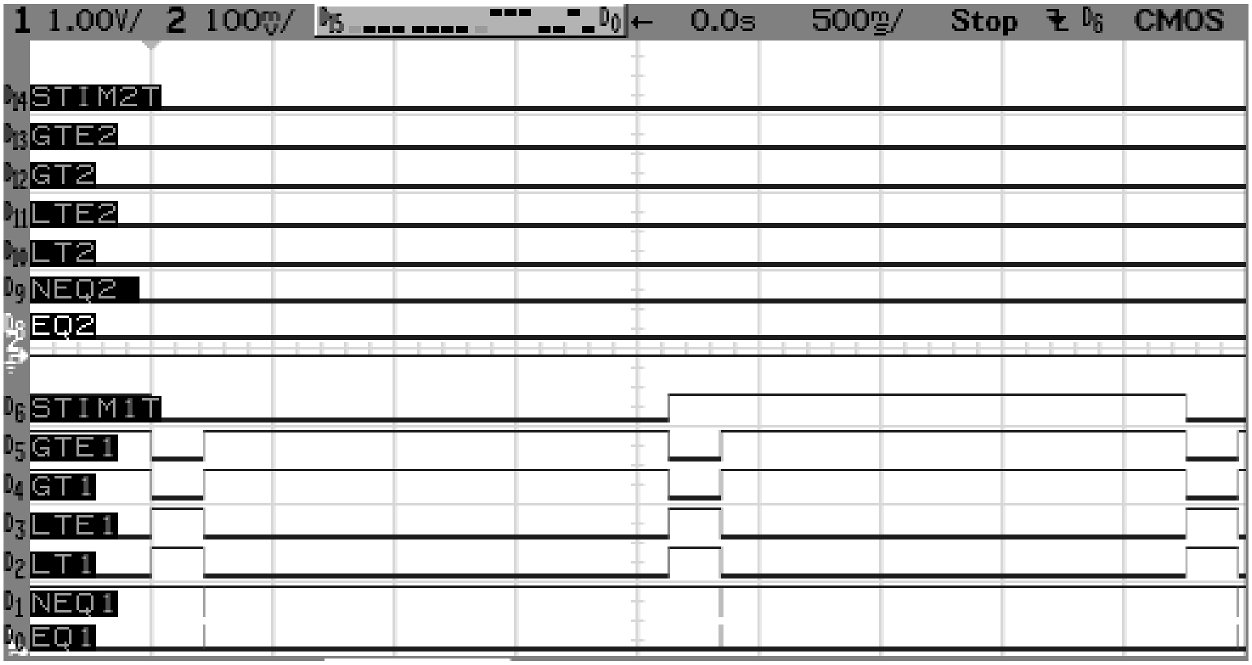


Figure 3 - Stimulus DMA ramp @ Signed 32-bit 1 input 1 thresholds (Thresh1 = 10%)



Stimulus_ADC

This example uses VDACs to create an analog ramp and ADC_SARs to convert the analog signal into ADC counts. The data is then loaded into the DCmp component using DMA for comparison. This example can illustrate how the component can be used to set thresholds to signal downstream circuits (or SW) to make changes.

Figure 4 - Stimulus ADC @ Unsigned 12-bit 1 input 2 thresholds (Thresh1 = 20%, Thresh2 = 60%)

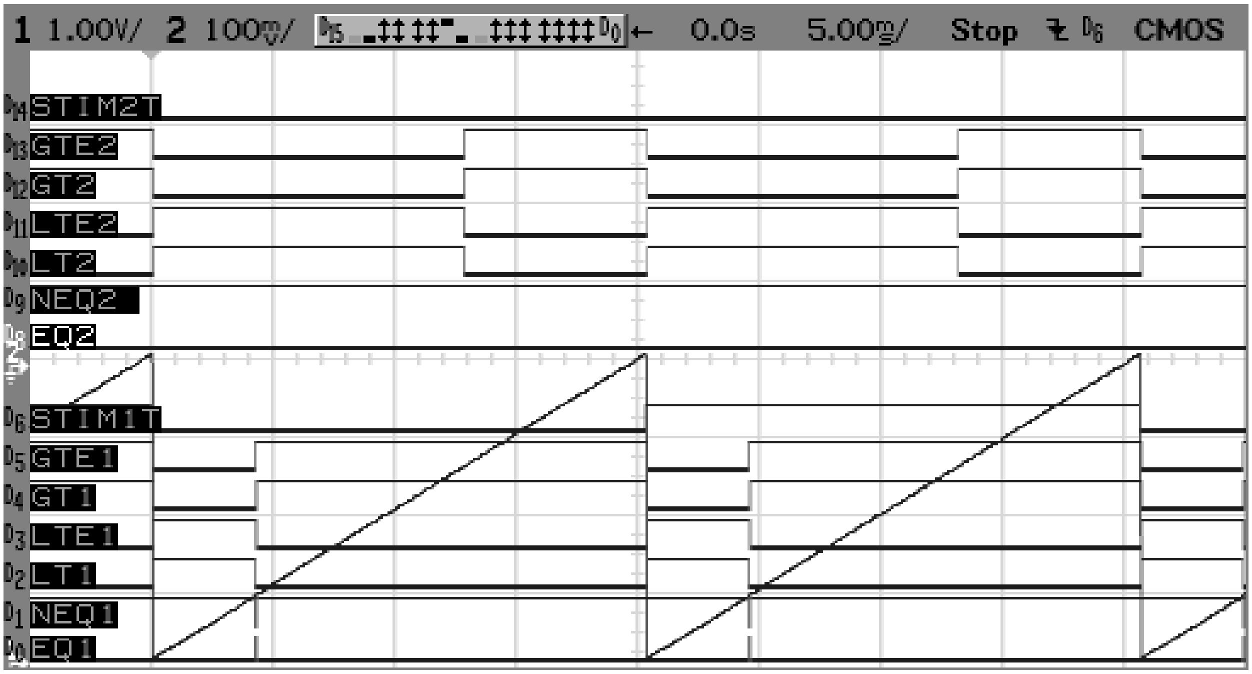
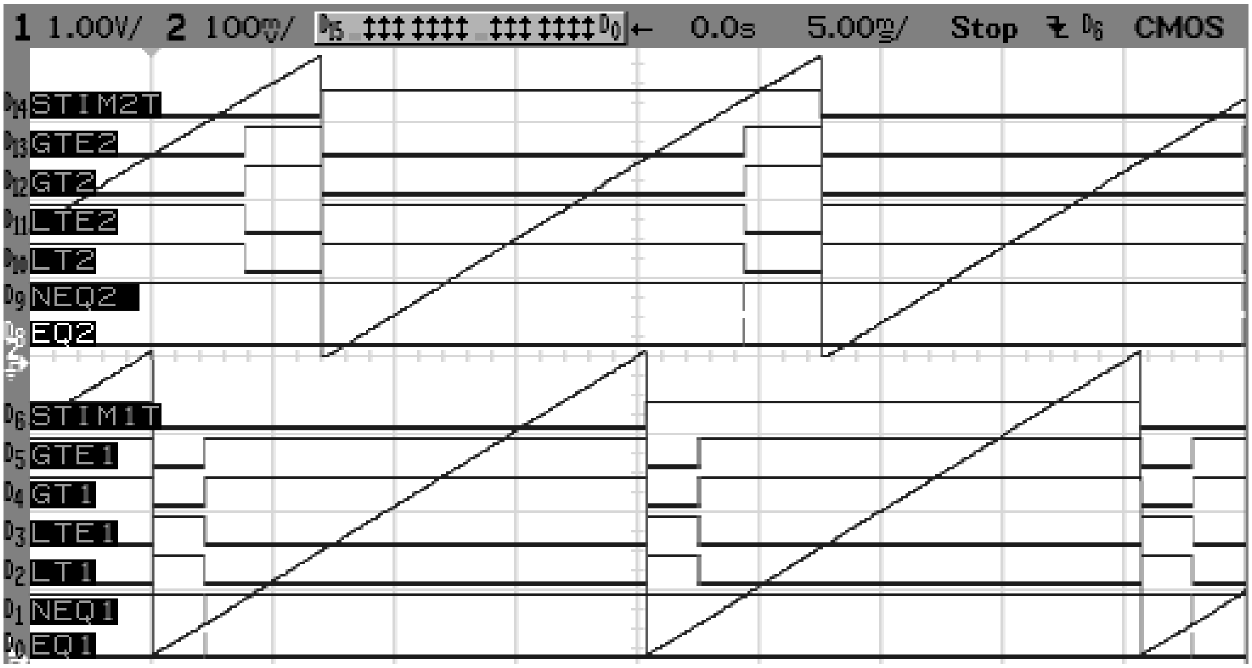


Figure 5 - Stimulus ADC @ Unsigned 12-bit 2 input 2 thresholds (Thresh1 = 10%, Thresh2 = 80%)



Project Changes

Version	Description of changes	Reason for changes/impact
1.0.0	Initial Demo released version	initial version

References

DCmp component: DCmp_v1_1.pdf	MenuCmds Component: MenuCmds_v3_5.pdf

DISCLAIMER: CONSUL*TRON* provides NO WARRANTY expressed or implied.
This product is intended for non-critical or non-safety use and is intended for educational purposes.