

5.4 Example 4: PI Controller Running on CLA

5.4.1 Example Overview

This example demonstrates one method of running a PI controller on the CLA. The CPU program contains an ISR which is triggered by an ADC end-of-conversion in the same way as example 3. Feedback data is read from the ADC in an ISR on the C28x CPU and passed to the CLA using a variable (yk) which is located in CPU-to-CLA message RAM, together with the servo reference (rk). The ISR converts the ADC result into signed floating-point format, then triggers task 3 on the CLA and waits for it to complete.

CLA task 3 calls the function `DCL_runPI_L1()` which computes the PI controller in an assembly function. The result is stored in the variable `uk`, which is located in CLA-to-CPU message RAM.

The PI controller result is read by the ISR, converted into a scaled un-signed 16-bit integer, and written to the PWM duty cycle register.

5.4.2 Code Description

The following lines in the file "Example_F28069_PI.c" are important.

Lines 19-24: create instances of the control variables and assign them to the appropriate message RAM blocks

Lines 26-27: create an instance of the PI controller structure and place it in CPU-to-CLA message RAM. This allows controller parameters to be modified from code running on the C28x CPU.

Lines 44-49: initialize the PI controller parameters

Line 168: reads the feedback data from the ADC and converts it into floating-point format

Line 172: starts CLA task 3 and waits for it to complete

Lines 175-176: convert the controller result to 16-bit unsigned integer and write it to the PWM duty cycle register

The following line in the file "F2806x_PI_CLA.cla" is important.

Line 24: calls the PI controller function `DCL_runPI_L1()`

Note that in this example, initialization of the PI controller is performed on the C28x CPU, so there is no need to allocate a separate CLA task for that purpose.

5.4.3 Running the Example

Build and load the project onto the C28x, then load the symbols onto the CLA. Place a break-point at the last instruction in the ISR (line 178), and run the program. Open an Expressions Window in CCS, and inspect the control variables and PI controller structure.