Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous)

Report of the Event

Title of the Event: Workshop on TI C2000 MCU for Real Time Control Applications

Organized Date : 11th August 2015 to 12th August 2015

Summary: To get started with C2000 real-time microcontrollers we recommend you first thoroughly review the available hardware, software, and toolchains. After selecting a series from the Piccolo or Delfino family that meets your needs, find the right supporting hardware and software for device or application development. Then find specific device or application training and support.

controlSUITE had been the content delivery tool for all software and hardware, offering support for Delfino and Piccolo series (except F28004x). All new device software updates will be through C2000Ware only. controlSUITE will continue to be updated into 2018 only for Application Package releases. C2000Ware is a cohesive set of development software and documentation created to minimize development time. It includes device-specific drivers, libraries, and peripheral examples. C2000Ware is the recommended content delivery tool vs. controlSUITE, but is required for the Piccolo F28004x series.

Software interfacing

The programming model for C2000 MCUs offers flexibility with various layers of abstraction. While C is the prevalent language, software methods are provided that allow for assembly instructions with direct access to the registers, a #define header file system with easy to interpret bit-fields (ex: EPwm1Regs.CMPA.half.CMPA), or a classical API driver (ex: PWM_setDutyA (PWM_MODULE_2, duty);)

Level 1 – Registers and addresses

- Baseline assembly communication to all hardware registers and addresses

Level 2 – Bit fields

- Bit fields can be manipulated without masking
- Flexibility to access a register as a whole or by bits
- Advanced CCS™ IDE features for ease-of-use

Level 3 – API drivers

- C functions that automatically set register bit fields
- Further reduces learning curve for new embedded programmers
- Common tasks and peripheral modes supported

Level 4 – Framework

- State Machine / ISR Based OS
- Function-based device initialization

- Built-in task management