MATLAB Embedded Coder

Exploration of code generation using embedded coder for real-time control system

Hong Ting Hui
h.t.hui1@newcastle.ac.uk

Supervisor: Dr. David Atkinson

School of Electrical and Electronic Engineering, Newcastle University, UK

Newcastle University

Introduction

This project explored the programming of microcontroller by using embedded coder of MATLAB. The users can easily configure the code generated from embedded coder to control software interfaces, optimise execution performance, and minimise memory consumption. This project covered the configurations of MATLAB, steps of running and deploying program project into a hardware, and the results of sample projects 'ADC-PWM Synchronization via ADC Interrupt'. Microcontroller Piccolo F28069 ControlSTICK was used for testing the embedded coder.

Background Information

- Embedded Coder is an add-on software of MATLAB by MathWorks. It generates programming codes like C or C++ for embedded system. It is a useful tool that enables engineers to program microcontrollers without having advanced level (or the required area) of C programming skills.
- Microcontroller is a small computer that is used automatically controlled products and devices, such as remote controls, appliances, office machines, toys etc.

Aims & Objectives

Since the online guidelines of the initial setup of embedded coder is not clearly stated, the aim of this project is to guide beginners to use embedded coder.

The followings are the objectives:
1) Set the directories of necessary files
2) Run the example project 'ADC-PWM Synchronization via ADC Interrupt' and download it into the hardware
3) Observe the behaviour of the system

Initial Setup of Embedded Coder

Before installing Embedded Coder, MATLAB requires the user to install Code Composer Studio as well as ControlSUITE. It is suggested to install Code Composer Studio version 5, since there are more examples from MathWorks on using this version.
Embedded Coder needs necessary files from TI to generate codes. The file directories can be set by the command `checkEnvSetup`, while the first parameter is the version of Code Composer Studio and the second is the name of the hardware. This can make sure the hardware to get all necessary files to execute the algorithms.

ADC-PWM Synchronization via ADC Interrupt

This model example is about altering the duty cycle of the PWM output by changing the input voltage at the ADC input pin. Since the page in the above link is not written in detailed, this section is going to show how the example can be run on the F28069 ControlSTICK.
The following window (fig.1) will appear after typing 'c280xx_adcpwmasynctest_ert'. It has two blocks: 'Hardware Interrupt' and 'ADC-PWM Subsystem'.

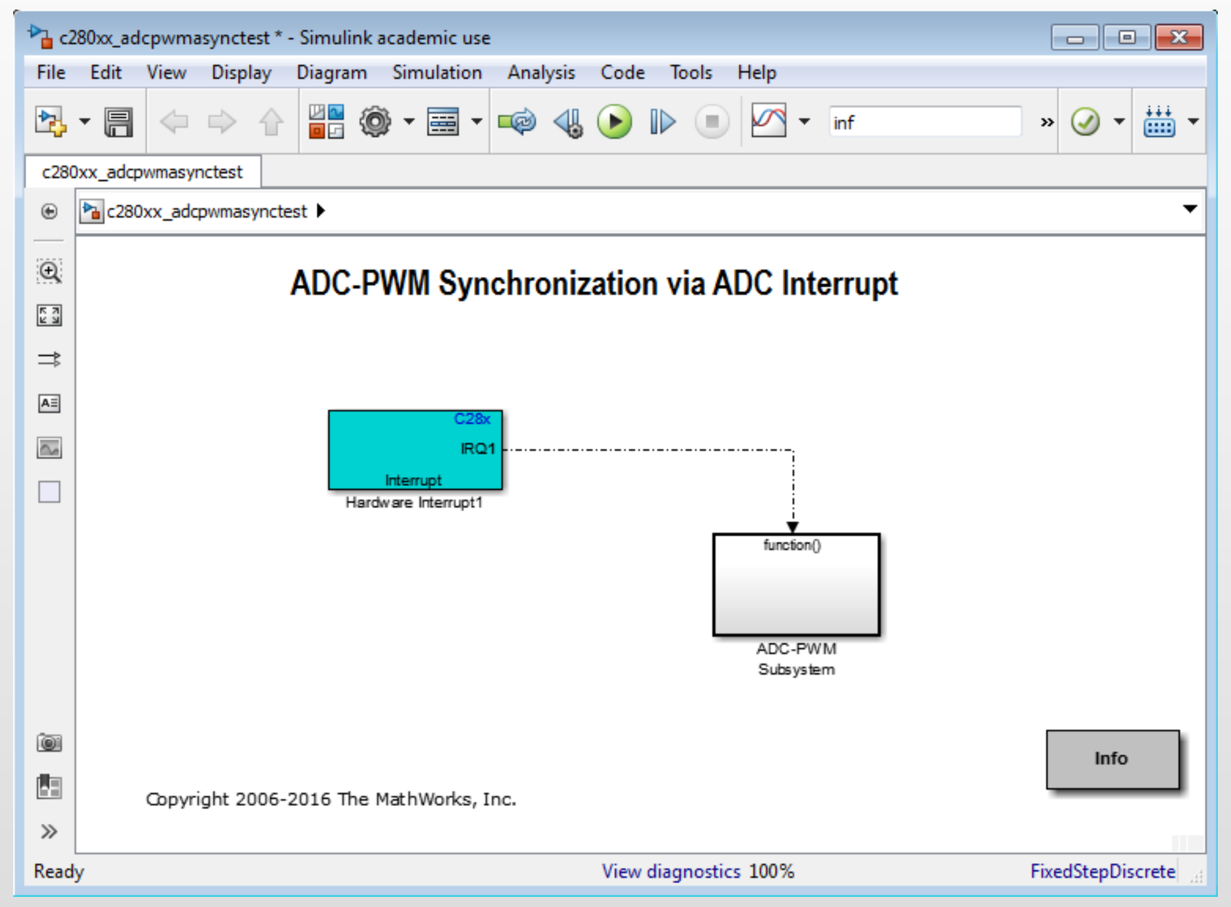
c280xx_adcpwmasynctest * - Simulink academic use
File Edit View Display Diagram Simulation Analysis Code Tools Help
inf
c280xx_adcpwmasynctest
ADC-PWM Synchronization via ADC Interrupt
C28x
IRQ1
Interrupt
Hardware Interrupt1
function()
ADC-PWM Subsystem
Info
Copyright 2006-2016 The MathWorks, Inc.
Ready
View diagnostics 100%
FixedStepDiscrete

Figure 1: Example model of ADC-PWM Synchronisation via Interrupt

- 'Hardware Interrupt' installs an Interrupt Service Routine (ISR) for the model and schedules the execution of the ADC-PWM Subsystem when ADC interrupt (ADCINT) is received.
- 'ADC-PWM Subsystem' consists of an 'ADC' block and an 'ePWM' block. PWM needs to be configured to the start of conversion (SOC) of ADC. The SOC can be configured in the source block parameters by double clicking the ADC block. In default settings, the trigger source is 'ePWM1_ADCSOCA', thus, the PWM output is at pin 17 (i.e. ePWM-1A).

The following steps can generate and deploy code into the embedded system.
Go to 'Simulation', then 'Model Configuration Parameters'.
Go to 'Hardware Implementation', select 'TI Piccolo F2806x' in the drop-down list next to 'Hardware board'. Wait for a few seconds, until 'Hardware board Settings' appear under 'Device details'. Select 'Build, load, and run' next to 'Build action'.
Go to 'Code Generation'. In 'Build process', select 'Texas Instrument Code Composer Studio v5 (C2000)' next to 'Toolchain'. In 'Code Generation Advisor', press 'Set Objectives…' can select priorities for generating codes.
To generate code and deploy to the microcontroller, go to 'Code', then 'C/C++ Code', then click on 'Deploy to Hardware'.
The model report can be generated by clicking on 'Code Generation Report' in 'C/C++ Code', then 'Open Model Report'. The generated code can be viewed.

Results

To test the model, by default settings of the block parameters, the input signal should be connected to pin 3 (ADC-A0) and the output signal can be obtained from pin 17 (EPWM-1A). It is noted that the input voltage of each ADC is ranged from 0V to 3.3V, values exceeding the range may damage the hardware.
Figure 2 and 3 are showing that both the input signals (green traces) and output signals (yellow traces) of the example. It shows that when the input voltage is low, the duty cycle of the output PWM signal is high, vice versa.

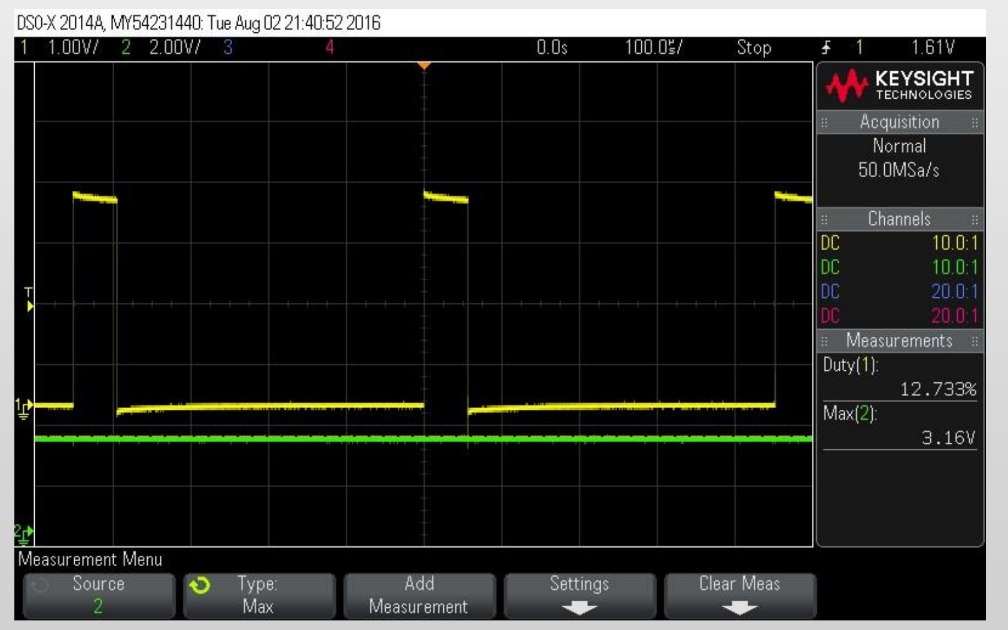

Figure 2: When the input voltage is high (3.16V; green trace), the duty cycle of the PWM output is low (12.7%; yellow trace)

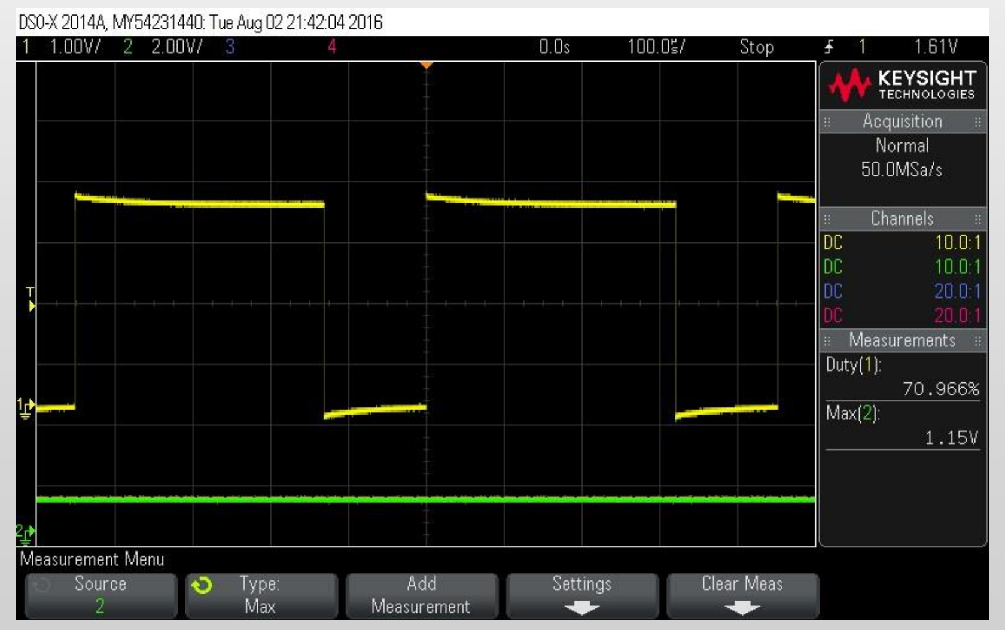

Figure 3: When the input voltage is low (1.15V; green trace), the duty cycle of the PWM output is high (71.0%; yellow trace)

Discussion

This project explored the initial setup and important steps for implementing one example model. The power of embedded coder can be optimised by, for example, investigating more different block diagrams and using various microcontrollers. This project shows that the embedded coder can help users of embedded systems to configure the operations of a hardware by drawing block diagrams and changing its parameters. It is hoped that embedded coder will benefit many of the research projects currently being carried within the Electrical Power Research Group of the School of Electrical and Electronic Engineering.

Reference

1. TI C2000 Support from Embedded Coder: http://uk.mathworks.com/hardware-support/ti-c2000.html
2. F28069 datasheet: http://www.ti.com/product/TMS320F28069/datasheet