Conestoga College

Institute of Technology and Advanced Learning

Document Title	Embedded Controller Design (ECD) Project Charter
Revision	2.7
Date Issued	January 8, 2015
Date Effective	January 8, 2015

Department	School of Engineering and Information Technology
Program	Integrated Telecommunication and Computer Technologies
Audience	ITCT Semester 4 Students
Author	Semester 4 team
Reviewers	Mike Jarabek, Professor Peter Roeser, Professor Bill Stefanuk, Professor

Revision History

Revision	Description of Change	Effective Date
0.1	First Draft - JHC	Jan. 6, 2006
0.2,0.3	Update - WPS	Jan. 7, 2006
1.0	Initial Release	Jan. 9, 2006
1.01	Revised for 2007 - WPS	Jan. 8, 2007
2.0	Revised for 2008 – WPS	Jan. 7, 2008
2.1	Revised for 2009 - WPS	Jan. 5, 2009
2.2	Revised for 2010 - WPS	Jan. 4, 2010
2.3	Revised for 2011 – AJ/WPS	Jan. 10, 2011
2.4	Revised for 2012 – WPS	Jan. 9, 2012
2.5	Revised for 2013 – WPS	Jan. 7, 2013
2.6	Revised for 2014 – WPS	Jan. 6, 2014
2.7	Revised for 2015 – MJ	Jan. 8, 2015

Table of Contents

Embedded Controller Design Project

1. Academic Need

This project provides you with an opportunity to study and apply design principles for the creation of embedded systems.

You will study the theory and application of microcontrollers and related interface circuits and further your expertise in electronic design and fabrication techniques. You will also enhance your understanding and skills in writing low-level software. For this project, you will use common industry tools for software, system and circuit design.

Through a series of project-related learning activities, you will:

- Design an embedded controller system from a specification, integrating such components as: microcontroller, low-voltage power supply, clock and reset logic, motor control outputs, sensor inputs, communication interface, troubleshooting interface.
- Use a schematic capture and board layout package to create printed circuit board manufacturing data and documentation for your embedded controller design and have that board manufactured by an outside vendor
- Design, implement, simulate and test diagnostic software for the embedded controller on your board
- Populate, debug and test your board to prove the design

2. Project Description

For this project, you will be given a set of requirements for your embedded system to meet. From this requirements document, you will design an HCS12-based embedded controller board for the low-level operation of motors, actuators, and motion-related sensors, and communication with a host computer. You will interact with an outside vendor to have your design checked for manufacturability. Once ready, your design data and documentation will be sent to the same outside vendor to have circuit boards fabricated. You will complete the assembly and testing of the board once received from the vendor. To test your board, you will need to write software to exercise all the low-level functions of the hardware.

The final deliverable for this project is a functioning embedded controller board with complete design and manufacturing documents, including diagnostic and control software. Extra credit projects could extend both the hardware design and the software capabilities to include additional functionality.

You will work **individually** for both the hardware and software portions of this project.

The board and software that you develop will be used in project 2 of this semester to control the motion of a robotic camera platform.

The project will meet the following criteria (requirements are identified as **M**ust, **S**hould, **C**ould or **W**on’t):

MUSTs

1. Use HCS12-family processor as specified
2. Include a BDM port for debugging and programming
3. Include RS-232 communication to host
4. Include hardware to drive 2 permanent magnet DC motors
5. Include hardware to read 2 incremental optical encoders for position/speed feedback
6. Include hardware to drive 1 RC servo
7. Include hardware to drive 1 stepper motor with 2 limit switches to support homing of the motor
8. Include hardware to drive a character LCD module
9. Include hardware to accept analog inputs and produce analog outputs
10. Include an on-board accelerometer
11. Include a unique extension to the design functionality (e.g. communication interface, supported I/O device, etc) chosen by the student and approved by faculty
12. Draw DC power from supplied batteries (2 x 12V available) or bench supply
13. Include a monitor program (written by you, using C and assembler) to:
    - 13.1. perform diagnostics on all components
    - 13.2. communicate with a host computer via the RS232 link
    - 13.3. operate all motors and actuators on command
    - 13.4. read and report all sensors on command
    - 13.5. be able to display information on the LCD as required by the application

SHOULDs

14. Include maximum hardware aids for debugging including test points, LEDs, start/stop switches, analog input knob
15. Be hardened against ESD and other electromagnetic interference likely to be encountered in the lab
16. Provide for future reuse through expansion connectors
17. Include a prototyping area to support extensions and design modifications

COULDs

18. Include a CAN bus transceiver (used in Semester 6)
19. Include SPI interface for other modules such as EEPROM, Real-Time Clock, I/O expansion, Zigbee interface, etc

1. Milestones

Project Start: January 6, 2015
Project End: March 28, 2015 (end of week 11)

Deadline	Milestone Deliverable
End of Week 1	• Project Plan: each student will write his/her own project plan for weeks 1-7 • Project Schedule: each student will write his/her own project schedule for weeks 1-7 • Each student will write a hardware functional specification
End of Week 2	• Each student will deliver a schematic hierarchy design • Each student will deliver CPU, power, and RS-232 schematic sheets
End of Week 3	• Each student will deliver stepper motor, DC motor, and RC servo motor schematic sheets
End of Week 4	• Each student will deliver ADC & Accelerometer, DAC (SPI) and LCD schematic sheets • Each student will deliver diagnostic LEDs/GPIO schematic sheets
End of Week 5	• Each student will deliver a preliminary placement • Each student will generate preliminary design files
End of Week 6	• Each student will deliver a set of manufacturing files suitable for submission to the board vendor for design rule check
STUDY WEEK *(No Classes)*	• *Work time to complete final board design and complete successful Design Rule Check (DRC) evaluation at board vendor*
Week 7 **Board Order Week**	• By ***Monday morning***, final board design, design files, and clean DRC report from board vendor to be submitted for evaluation • Each student will provide documentation and design files sufficient to release the design for manufacturing • Each student will deliver a ***properly formatted*** Bill of Materials • The boards will be ordered on **Wednesday**
End of Week 8	*Boards ordered by Wednesday of week 7 should arrive by Wednesday of this week.* *In weeks 8-11 students will assemble and troubleshoot their boards, while creating basic control and diagnostic software. Each week students must demonstrate progress towards the final demonstration of the completed project* • Students should take this time to demonstrate software deliverables as progress towards their final demonstration
End of Week 9	• Students will demonstrate, as a minimum, their board connecting to the debugger and running a simple application to flash LEDs • Students should demonstrate additional hardware and software deliverables as progress towards their final demonstration
End of Week 10	• Students will demonstrate hardware and software progress

End of Week 11	• Each student will demonstrate their embedded controller board in operation, running diagnostic software.

2. Evaluation

- This project counts for 60% of your project course mark.
- You will be evaluated with respect to the milestones and rubrics as provided by faculty.
- The marks will be divided between the software and hardware portions of this project as follows:
  - 60% hardware design, documentation and implementation to the point of demonstrating a working board with flashing LEDs
  - 40% software design and implementation demonstrating the correct operation of the separate hardware functional blocks on your board (For this 40% portion, the marks are divided 2/3 software and 1/3 functional hardware.)

3. References

Altium Designer documentation.

CodeWarrior documentation.

HCS12 documentation from Freescale Semiconductor

Huang,H-W. The HCS12/9S12: An Introduction to Software and Hardware Interfacing. Thomson-Delmar

Specification documents for various selected components.

4. Constraints

- This project shall be completed in eleven weeks (plus study week).
- Each student shall complete this project ***individually***. **Software must be written by each student;** copying of source code is not permitted.
- A date has been set (middle of week 7) when all data will be sent to the vendor for boards to be fabricated. In order to minimize the cost to students, **ALL** students **MUST** have their board design completed by this date. Note that shipping costs increase substantially if a single student needs to send out data. Students are responsible for their own board costs including shipping, handling, and border brokerage fees, including any extra costs incurred should their design not be ready on the board-order date.
- The external dimensions of the board will be the same for all students and will be determined based on the most economical panel size for board fabrication.
- Students are responsible to pay for the fabrication and delivery of their own bare board.
- Students are responsible to procure and pay for the components required to assemble their designs onto their own boards. Some parts may be available from

Dave Sullivan, others may be sampled from vendors, others must be purchased (Digikey, Sayal, etc.)

5. Assumptions

- Each student will dedicate at least one hour of outside study time for each scheduled class hour.
- Each student will attend all scheduled classes.
- Each student will make wise and efficient use of scheduled class time.
- Each student bench is equipped with the required computer and software, capable of running Altium Designer and CodeWarrior.
- Each student will learn to use Altium Designer and CodeWarrior using the online resources and training materials. The use of these tools is not taught directly in any course this semester.
- Board documentation is complete and acceptable to the board manufacturer, and the boards can be turned around in an appropriate time.
- Each student bench is equipped with logic analysers and troubleshooting equipment.