121st ASEE Annual Conference & Exposition
Indianapolis, IN
June 15-18, 2014

Paper ID #10078

Programming Embedded Microprocessor Systems: The Autonomous Robotic Car - Dragon Board vs Tower

Dr. Javad Shakib, DeVry University, Pomona
Dr. Mohammad Rafiq Muqri, DeVry University, Pomona

Programming Embedded Microprocessor Systems: The Autonomous Robotic Car - Dragon Board vs Tower

We redesigned the “Programming Embedded Microprocessor Systems” course to help prepare our Engineering Technology students for the Senior Project. They are to construct and demonstrate a project called Autonomous Robotic Car (ARC). The difference between this effort and the Senior Project is that this project will have been done before, whereas the Senior Project will be the students’ own idea.

For this project, the students worked in groups of two or three and they needed to have the robot able to run a course in the shortest amount of time. However the robot had to navigate a one-inch black line without falling off course or without the aid of human intervention. The robot had several key parts to it: a microcontroller, power supply, power filter, dc motors, visual sensors, and computer software to control the robot.

Most of the students had a Tower kit (S12G128). The Tower system is a new development platform from Freescale that allows the user to stack a CPU module and multiple peripheral modules on top of each other. It has room for one CPU module and three peripheral modules. HCS12 CPU modules for Tower use the HCS12Gxxx chips.

Few groups used Dragon12-Plus Kit from Wytec for CodeWarrior, Matlab, and Simulink (DRAGON12P-USB-SM). It is preinstalled with the serial monitor for working with CodeWarrior, Matlab, and Simulink.

With some diligent care and guidance from faculty, all of the engineering technology students were able to complete this project in six weeks while doing all aspects of system engineering. The results, a comparison between Dragon and Tower boards, the type of programming language that was used, the performance of the ARCs and the lessons learned will be explained in this paper in full detail.

1. Introduction

In most undergraduate engineering programs, students are required to complete a capstone design project by building and testing a prototype[1]. Depending on the nature and complexity of the design specifications, the final product may be a composite of analog and digital, hardware and software, discrete components and off-the-shelf parts.

For a majority of the projects in disciplines such as electrical engineering and mechanical engineering, the primary processing component of these projects is a microcontroller unit (MCU).

It used to be that the number of different MCUs and development tools (DT) available to people was pretty limited, but times have changed. Digikey[2] lists over 16,000 different line items under a 'microcontroller' search.

Selecting the ideal MCU and DT for a particular project could be a time-consuming and tedious task for an undergraduate student or the course designer and faculty mentor[3]. This is aggravated by the fact that students are usually only familiar with the MCU, DB and IDE they have used during their undergraduate classroom and lab experience. The students are challenged not just by the design, but by the integration of these various types of technology.

There are other challenges in the curriculum too. A student takes a digital circuits and systems course with its laboratory component. The lab experiences gained may be very suitable for demonstrating the analysis and design of combinational and sequential logic circuits, but they do not teach the students how to interface digital circuits with analog circuits or computer software.

Moreover, there is no gradual path and ramp from those small and compartmentalized lab experiments to a very large open-ended senior project.

2. DeVry University's Senior Project Capstone Course Sequence

The College of Engineering and Information Sciences at DeVry University offers a sequence of senior project courses under its ECET (Electronics Engineering Technology/Computer Engineering Technology) program. The senior project is a four-session course sequences (eight weeks per session) in which students integrate knowledge and skills learnt in the previous courses. In the first course (ECET-390, Product Development) students from Electronics and Computer programs are asked to form teams, and then required to research, plan and develop a project proposal. Then in the next three senior project courses (ECET-492, ECET-493 & ECET-494: 24 Weeks) students implement the project plan by building and testing a prototype. A typical project involves a solution to a software/hardware-based engineering problem. The process of developing and implementing a solution to the problem offers a unique learning opportunity for students to gain new insights and competencies and their team-work, problem-solving and analytical thinking skills.

Senior projects, also known as capstone projects in, the ECET program usually require students to prototype systems that have mechanical, electrical and control components. These capstone projects use a microcontroller unit (MCU) mounted on a development board (DB) as the central processing/control unit. The DB is capable of sensing and controlling various devices with the help of a program written using an integrated development environment (IDE) software. The DB/IDE integration is collectively called a development tool (DT).

3. Important Factors in Selecting Microcontrollers and Development Tools

There is a white paper[4] that addresses the MCU/DT selection dilemma by researching a wide array of popular MCUs and DTs available in the market today, and recommends the most popular, cost-effective, easy-to-use, robust, and highly-functional tools that are best suited for inclusion in undergraduate electrical and mechanical capstone projects.

There have been some efforts to select ideal microcontrollers based on certain criteria[5], but they have been in general and not specifically targeted undergraduate capstone projects. Here is a list of important considerations in MCU, DB, and IDE selection for an undergraduate capstone project. [4]

1. MCU Selection Factors

1. Pin Count
2. CPU Speed
3. Power Consumption
4. Sleep Mode
5. Instruction Set
6. Program memory
7. Data memory
8. Memory Expansion
9. Analog-to-Digital Converter (ADC)
10. Digital-to-Analog Converter (DAC)
11. Interrupts
12. Pulse Width Modulator (PWM)
13. Timers
14. Serial Communication
15. Vehicle Bus
16. Real Time Clock (RTC)

2. DB Selection Factors

1. MCU Support
2. Cost
3. Size
4. Power Supply
5. Power Regulation
6. Embedded IO Devices
7. PC Interface
8. Program Upload

3. IDE Selection Factors

1. DB Compatibility
2. Time
3. Libraries
4. Online Support
5. Ease of Use
6. Languages
7. Simulation
8. Debugging

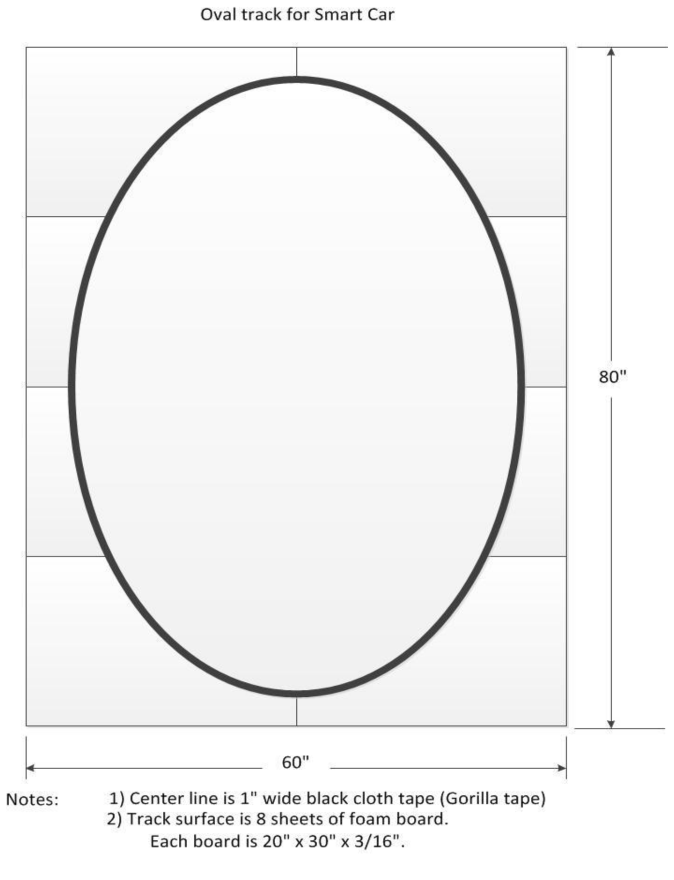
The key to microprocessor selection is identification of the necessities of the design. The aforementioned categories are a good basis to identifying what to look for in terms of a microcontrollers form and function. As mentioned previously, the datasheet for a microcontroller is the best reference to the device capabilities and functions and should be used heavily in the selection process.

However, having a smaller pool of ideal MCUs to choose from can help students move past the selection process and focus their valuable time and effort in designing and building a complex, quality project. Using popular MCUs that are widely available and commonly used in the industry builds real-world engineering design skills as students prepare for engineering practice.

4. Preparing Students Through a Restructured Course

To address these challenges, we redesigned and restructured the "Programming Embedded Microprocessor Systems" (ECET-365) course to help our ECET students for the Senior Project get prepared for their senior project. This course was offered under the new structure in Fall 2013 for 21 students. Students were to construct and demonstrate a project called Autonomous Robotic Car (ARC).

For this project, the students worked in groups of two or three and they needed to have the robot be able to run a course in the shortest amount of time. However, the robot needed to must need to navigate a one-inch black line following the path[7] shown in Figure 1 without falling off course or without the aid of human intervention. The robot had several key parts to it: a microcontroller, power supply, power filter, dc motors, visual sensors, and computer software to control the robot.


Oval track for Smart Car
80"
60"
Notes:
1) Center line is 1" wide black cloth tape (Gorilla tape)
2) Track surface is 8 sheets of foam board.
Each board is 20" x 30" x 3/16".

Most of the students had a Tower kit (S12G128). The Tower system is a new development platform from Freescale, allowing the user to stack a CPU module and multiple peripheral modules on top of each other. It has room for one CPU module and three 3 peripheral modules.

The processor board used for this project was the Axiom TWR-S12G128. It contains the MC9S12G128 Central Processor Unit (CPU). The board has many input/output ports, analog-to-digital (A/D) inputs, and pulse width modulation (PWM) outputs.

Few groups used Dragon12-Plus Kit from Wytec for CodeWarrior, Matlab, and Simulink (DRAGON12P-USB-SM). It is preinstalled with the serial monitor for working with CodeWarrior, Matlab, and Simulink.

5. Programming Embedded Microprocessor Systems: ECET-365 Course Information

In previous courses, the labs consisted of detailed instructions that were designed to lead students through a learning-and-discovery process.   In the Senior Project, however, the student is called upon to develop a project without detailed guidance, produce the project within a definite time schedule, and ensure that the project performs as intended.

To bridge the gap between these two processes, this course has been structured to provide many of the experiences of a Senior Project while removing some of the risks. The difference between this effort and the Senior Project is that this project will have been done before, whereas the senior Project will be the student's your own idea.
For example, the Robotic Car Kit with a Tower unit has been constructed and verified to work and the Construction Notes that contain software which has been tested is provided for students.

6. The new Structure of ECET-365 Course

The course is composed of two parallel efforts[6]. One is to learn more knowledge about designing systems by integrating hardware and software subsystems. Hence, we study devices, communications protocols, and software techniques for servicing devices and handling data.

The other effort is to build the Robotic Car itself. In some versions of this course, the teams may be constructing a robot, but the basic knowledge is very similar for the two projects.

The first two weeks focus on the drive motor subsystem, the steering subsystem, and the visual sensor subsystem. It is a very good idea to assign different team members to "own" these subsystems. Each team will be responsible for the construction and testing of the subsystem to which they have been assigned. Some tasks, such as steering, require cooperation with other teams.
Of course, each person is responsible for reading the text assignments and lectures, and doing the homework and quizzes.

Week 3 introduces system engineering concepts, especially the development of subsystems to meet a System Requirements document.
Students will also need to develop a work schedule. Some of the tasks will have been completed already, but there are other requirements to think about.

Week 4 focuses on the power systems for the project. The biggest challenges here are the interaction of subsystem power supplies on each other, and the interference from the main power on the subsystems.

Weeks 5 considers timing and interrupt techniques. The textbook[8] contains much detailed information on this subject.

Week 6 covers the use of different network protocols. If a student chooses to operate a robot or Smart Car via a wireless channel or network, this part of the course provides a specific example (ZigBee). We also discuss using the Internet to control a project, which allows long-distance manipulation.

Week 7 involves the final assembly and testing of the system. Here, the students will be developing their test plan by incorporating the tests for the subsystems into an overall plan. This plan will be helpful for diagnosing any problems that may occur as the system is tested.

Week 8 is when the project is demonstrated on the race track (for the Robotic Car) or a robotics field. Also, a Project Manual is required. It should be submitted when the project is presented for the demonstration.

7. Class Operation and Results

Six groups (12 students) used Tower kit (S12G128) but five other were allowed to use the Dragon Plus-2 Board with the MC9S12DG256B microcontroller, because they had it already and they were most familiar with it after having worked with it during ECET 330 & 340.

Each person purchased the same kit from the online bookstore, so they all had the same parts. The differences in each person's design started to come with the microcontroller that each person chose.

The TWR-S12G128 CPU is embedded in the tower in the tower system and was programmed using the IDE CodeWarrior Development Studio V5.1. A prototype bread board is used to connect peripherals sensors and actuators without having to solder. The CPU is programmed to control two motors that turn the car while following a black line, while also receiving feedback from two sensors whose ultimate purpose is to sense the black line on the provided race-track. The purpose of the line-following robotic car is to stay on the black line at all times. Any time the car moves off the line, it will readjust back onto the line. This is done with a state-of- the-art algorithm that takes feedback from the IR sensors. In order to efficiently get the reading we had to make our code efficient and fast to ensure we did not skip a beat on the track.

The bill of materials for this system is given in table 1 and the end result is shown in Figure 2. [9]

Bill of Materials		
Item	**Company**	**Price**
TWR-S12G128	Freescale	$79.00
Line Following Motor Robot Kit	Digilent	$159.00

(MRK+Line)		
Erector Set	Schylling	$50.00
Mini Breadboard		$10.00
CodeWarrior	Freescale	free
Total		$298.00

Table 1: Bill of Materials for ECET-365course project with Tower

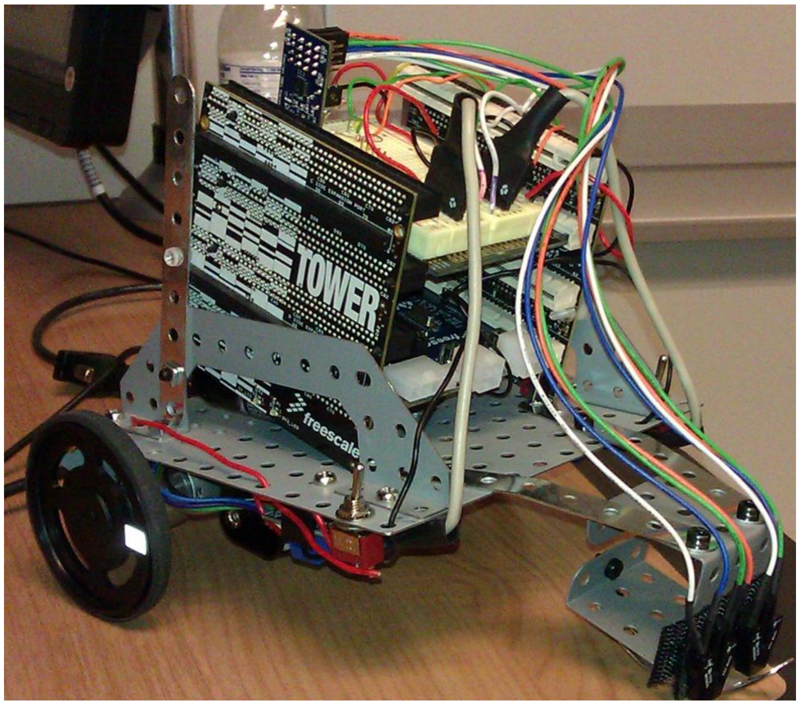


Fig 2. Robotic Car with Tower

The Dragon Plus-2 Board with the MC9S12DG256B microcontroller is different in some respects from the Tower Board (MC9S12G128), and the algorithm, physical design and codes created reflect that. The two groups who used the Dragon board had to make continual changes to each part in order to achieve the best performance from their microcontroller, sensors, and car operation. Their kit came with the H-Bridges, the sensor module and sensors, the chassis, the wheels and any other various parts mentioned which were assembled to create the Black Dragon. Of this system, there are three main subsystems on the Black Dragon: the visual sensors, the motors, and the power supply.

The bill of materials for this system is given in Table 2 and the end result is shown in Figure 3. [10]

Item	Quantity	**Price**
Devry Robotic Car Kit	1	$150.00
Dragon-12 Plus 2 Microcontroller	1	$170.00
8 AA Battery holder	1	$5.00
Breadboard	1	$10.00
Enercell 9v Battery	2	$5.00
24 pack – Duracell AA batteries	1	$10.00

Holder Snap on Connector	2	$1.00
TI LM324 Quad Op-Amp	1	$0.15
Helping Hands	1	$7.00
Weller Solder Iron Tip	1	$6.00
Electrical & Double Sided Tape	2	$1.00 ea
Total		**$372.15**

Table 2: Bill of Materials for ECET-365 Course Project with Dragon

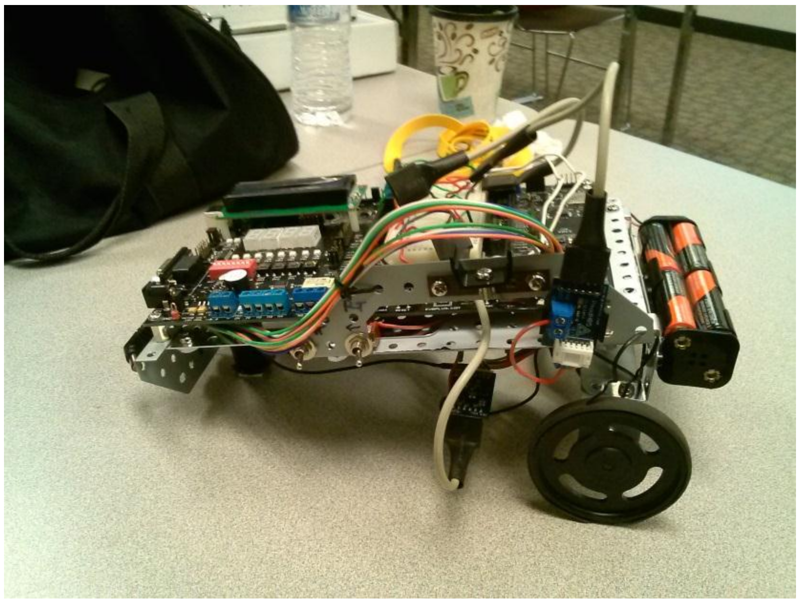


Fig 3. Robotic Car with DragonTower

8. Test Results, Analysis and Project Outcome

The Black Dragon required extensive research and testing to combat the issues that arose when building this project. From the beginning there were challenges building a project that was not meant for the Dragon Board. For those groups using the Tower, their design was much simpler; they had much more wiggle room in regards to the physical space, power supply, and the weight distribution. There are three major issues students had when building the Black Dragon: mounting the Dragon Board onto the chassis, power wiring, and directing travel.

The Dragon 12 plus 2 boards is a great trainer board. It is packed with peripherals and extra embedded devices that can help a student apply their knowledge and develop prototyping projects or just practice. This board offers many different types of options to practice with. For example, it comes with an embedded LCD screen. Additionally, this board comes with a XBEE interface plug and play station, on board keypad, H-bridge, speaker, RS232, capacitive touch sensor, temperature sensor, SD card slot, 8 LED's, and many more.

This board is very useful for those students applying the concepts that they are learning. However, it is very difficult to get started on this board. The first days of using this board can be overwhelming. It does not come with a manual and it refers the user to the website to find the manual. For some, it took about two days to actually find the right link for the correct board. Additionally, it is not plug-and-play and it must be configured to start a connection with the IDE.

It is a tedious to start at the beginning of the road. However, students started to appreciate its peripherals when they got used to it.

For this project and any project that relates to motors, this board can be used directly since it come with a H-Bridge which can control DC motors and Stepper motors. Additionally, a prototyping board is embedded on their for easy and quick circuit implementation running directly to the controller pins. This is a huge plus for those who are trying to develop a quick working prototype.

Fora student this board is great and it can serve many purposes. However, it is not desirable to place it on an actual project. This board is only for training, and the size and extra peripherals make very difficult to develop small electronic devices. Therefore, one must take into consideration that since it has a vast amount of embedded peripherals; it will waste more power drawing up to 500ma. The overall conclusion is, if you want to prototype and learn how to interface, the Dragon 12 plus 2 is a wonderful board to start with. On the contrary if you want to actually create a standalone dedicated application, I do not recommend using this board.

9. Conclusions and Recommendations

With some diligent care and guidance from faculty, all of the engineering technology students were able to complete this project in six weeks, while performing all aspects of system engineering.

The results, comparison between Dragon and Tower boards, the type of programming language that was used, the performance of the ARCs and the lessons learned are going to be explained in this paper in full detail.

The curriculum in any specific area of study tends to narrowly focus students on that area, whereas real-world multifaceted systems tend to incorporate components from multiple disciplines.

We are really just into the beginning stages of getting the tower system rolled out, and ECET-365 is just starting to be taught with it. Although we made a comprehensive comparison between two boards and we used the tower as the primary board, I should emphasize that we are teaching principles. The platform we have is sufficient to teach everything the students need to know at this point in time...

Unless there is a compelling reason to change the platform, we should stick with what we have for a significant amount of time.

References

1. L. J. McKenzie, M. S. Trevisan, D. C. Davis, and S.W. Beyerlein, “Capstone design courses and assessment: A national study,” in *Proceedings 2004 ASEE Annual Conference and Expo.,* Salt Lake City, Utah, *USA, 2004,* pp. 1-14.

2. http://www.digikey.com

3. D. Vyas, "Microcontrollers: options and trends in today's market," in *ACM Proceedings International Conference and Workshop on Emerging Trends in Tech*nology, Mumbai, India, 2010, pp. 1019-1019.

4. D'Souza, J., Reed, A., & Adams, K. (2014). Selecting Microcontrollers and Development Tools for Undergraduate Engineering Capstone Projects. *Computers in Education*, *24*(1), (In press)

5. M. Slade, M. H. Jones, and J. B. Scott, "Choosing the right microcontroller: A comparison of 8-bit Atmel, Microchip and Freescale MCUs," Faculty of Engineering, The University of Waikato, Hamilton, New Zealand, Tech Rep. http://hdl.handle.net/10289/5938 , Nov. 2011.

6. Daniel Cross-Cole, 'ECET 365 Course Overview' document, DeVry University, available on the course shell at http://devryu.net

7. Daniel Cross-Cole, 'Construction Notes for Robotic Car' document, DeVry University, available on the course shell at http://devryu.net

8. *Embedded Microcomputer Systems – Real Time Interfacing*, 3rd ed., Valvano, J. © 2012 Stamford, CT: Cengage Learning

9. Totally Autonomous Binary Yaw (TOBY) car, report by Aaron Wright, Arturo Torres and Juan Rodriguez in their project manual, October 2013

10. The Black Dragon (Line Following Robot), report by Malcolm Q Guidry Submitted in Partial Fulfillment of the Course Requirements for ECET- 365, October 2013