Seattle
Making Value for Society

122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Paper ID #11649

Teaching Digital Designs by Building Small Autonomous Robotic Vehicles Using an FPGA Platform

Dr. Cheng Chih Liu, University of Wisconsin Stout

Cheng Liu is an Associate Professor in the Computer Engineering Program at University of Wisconsin Stout. He taught courses in electrical, computer engineering, and engineering technology. His teaching and research interests are embedded systems, systems on programmable chips, FPGAs, controls and instrumentation.

Teaching Digital Designs by Building Small Autonomous Robotic Vehicles Using an FPGA Platform

This article discusses the experiences of implementing a new model in teaching and learning digital designs using Verilog in an embedded systems design course. This paper discusses the course structure, laboratory exercises, student projects and project evaluation process, and finally the student evaluation outcomes. Students' course assessment and student learning outcomes were very positive. In many existing digital designs curriculum, students learn how to create structural and behavioral models in Verilog Hardware Description Language (HDL) to design simple combinational and sequential logic, and then use the computer architecture theories as the guidance of system design and development for student projects using Field Programmable Gate Arrays (FPGAs). Several textbooks [1] – [4] and papers [8] – [12] discuss those techniques in detail. Different from many existing approaches, our embedded systems course focuses on digital designs of FPGA-based systems with an emphasis on small-scale autonomous vehicles. To facilitate student robotics projects, the course was restructured with a series of FPGA-based laboratory exercises before the students began to build their autonomous vehicles. The goal is to allow the students to learn mechatronics and apply knowledge of FPGAs as they construct autonomous vehicles to sense and react to their surroundings. Artificial intelligence such as left-turn and right-turn algorithms has been implemented by the students to allow their robots to follow a line made of black electrical tape and solve line maze problems. Throughout the semester, students participate in lab exercises and a final project, and must provide documentation of their designs with lab reports. The low cost robot allows each student to have their own robot for a semester, so they can work on assignments, labs, and projects outside the classroom. In addition, the FPGA robot has the capability of adding any sensors and communication system modules that support problem-based learning curriculum. In particular, the FPGA robot can support advanced topics in system-on-chip (SoC), networks-on-the-chip, real-time Bluetooth communication, and Android OS curricula, which encourage students' exploration into FPGAs and robotics in new ways. Finally, the student project reports indicated a great interest in FPGA robots which helped them learn FPGAs and Verilog HDL.

Course Background

FPGAs are an ideal fit for many industrial applications due to their programmable nature [13]. However, to this day, using FPGAs for rapid prototyping is still a difficult subject to learn and teach, and this challenge is increasing as FPGAs become more complex [14, 15]. In order to lower the barrier in learning FPGAs, efforts to establish a FPGAs ecosystem are needed for students to benefit from the collective wisdom of the whole, such as the Android OS, Arduino, development communities. Since such open-source support for FPGAs does not exist today, and therefore, teaching practical FPGA-based design requires significant effort for course development. When this course was created, it was developed specifically for computer engineering students who have prior hardware experiences (e.g., circuit analysis, digital logic, and signals and systems) and software experiences (e.g., C/C++, Assembly, and Java). With those backgrounds, students are prepared to learn more advanced topics in digital designs using Verilog HDL through hands-on FPGAs implementation. The benefit of FPGAs is the ability to reconfigure digital systems on-the-fly which is ideal for rapid prototyping embedded solutions. The ability to re-define the connections in the FPGAs fabric makes it possible for designers to future-proof designs without any substantial modifications to hardware. These FPGAs features are not available to embedded designers who use embedded processors in their designs. The run time reconfiguration of FPGAs is the basis for the flexibility and rapid growth of embedded design solutions with FPGAs. Verilog HDL was relatively easy to learn for our computer engineering students with their prior software and hardware backgrounds. In the new course structure, weekly lab exercises and a final project were now focused on sequential logic, sensors and servos interfacing, state machine techniques and artificial intelligence algorithms to build small autonomous vehicles. Teams often compete with each others to see what their robots can do, e.g., a robot navigates north using a magnometer, or a robot navigates with infrared sensors to detect obstacles and avoid collisions. Building robotic vehicles and software development using Verilog HDL becomes the central focus of the course. Competition of the student robots seemed to create lots of fun among students and students saw their hard work come to fruition in their final project presentations. The downside of using FPGAs to build a robotic vehicle was its cost of $189 if the Nexys-3 boards [5] were used in the student projects. However, the cost of a DE0-NANO board [6] was $59 after an academic discount and was offset by the student course fees, so no extra cost was increased for our students to build their robots.

Altera’s Quartus II and Xilinx’s ISE are Programmable Logic Device (PLD) design software which is suitable for high-density FPGAs designs and Complex Programmable Logic Devices (CPLD) designs. FPGAs design suite is an essential tool for synthesis and analysis of HDL designs, enabling the students to compile Verilog code and perform timing simulations, simulate a design’s reactions to different stimuli, and configure the target FPGA device with the programmer. The ISE and Quartus II design suites are free to download, so software cost is zero unless certain tool preferences are needed. Quartus II and Xilinx ISE have gone through a number of releases. The version known as Quartus II 13.1 web pack edition and Xilinx ISE, 14.0 were used in this course. FPGAs by Xilinx and Altera were both introduced to the students because both software tools can synthesize code written in Verilog, so students can choose either FPGAs (i.e. Spartan-6 or Cyclone IV) to implement their designs. However, the DE0-NANO boards were used in student final projects because the size of the board is compact (7cm by 5cm) and has very rich embedded peripherals for both analog and digital interfaces, and is very useful for battery-powered robotics applications.

The main focus of this paper is the new approach in teaching FPGAs by using robots which inspire students to pursue careers in digital and embedded design areas. The laboratory exercises enable students to accelerate their learning curve in creating a functional robotic vehicle using an efficient Verilog HDL. The course structure was created upon standard FPGAs which allow a seamless integration of hardware and software pertaining to embedded FPGA systems used by engineers in industry.

Previous Work

One of the earliest papers with a multidisplinary emphasis for the design of a complete embedded system was by Wolf *et al.* [7]. Recently, there has been a growing interest in discussing various teaching approaches for digital designs and embedded systems education as shown by several papers published in the *IEEE Transactions on Education* [8]-[12] and other publications such as *ACM Transactions on Embedded Computing Systems*. Akash Kumar, *et al*.

[11], for example, taught students how to use the Xilinx Spartan-3E board to develop a five-a-side soccer game system. Newman *et al* [16] used the Altera UP-1 CPLD board to design robots in their digital logic design class. In fact, programmable logic has been used in computer architecture classes [9, 10] and has been beneficial to students in such classes. Overall, previous research and experiences in using programmable logic or gate arrays (CPLD or FPGAs) for computer design classes have been positive, although difficulties for students still exist. The feedback from the students suggests that they felt the use of programmable logic boards was rewarding, despite the increased effort required [9]. It was noted that almost all of the previous work emphasized that good laboratory practices are essential to student success in the class [9, 10].

In this paper, teaching methodologies are compared with digital and embedded system courses using HDLs and FPGAs, and a comparison is summarized in Table 1. The comparison examines whether the use of FPGAs in labs accelerates student learning curve, or whether FPGAs are used in student projects, or whether advanced designs are addressed (i.e., FPGAs with wireless connectivity using Bluetooth and/or WiFi for mobile computing and Internet of Things applications).

Table 1 Comparison of Features in teaching digital/embedded systems design course

	Use of FPGAs in lab assignments and/or lab exercises to accelerate student learning curve	Use of FPGAs to inspire students for their projects	Use of FPGAs for advanced designs: wireless network communications	Use of HDL as programming language to solve design problems
Tyson S. Hall, *et al* [8]	Maybe	No	No	No
José N. A. [9]	Maybe	No	No	Yes
Hiroyasu, *et al.*, [10]	Maybe	No	No	No
Akash K., *et al.*, [11]	Yes	Maybe	No	Yes
Christos T., *et al.*, [12]	Yes	No	No	Yes
This paper	Yes	Yes	Yes	Yes

New Course Structure

Teaching digital designs by building small FPGA-based vehicles is a new teaching method to spark student's interest in FPGA technology and is the primary focus of the course. The 3-credit hour course meets two days per week and two hours each time, so the course has one more hour than traditional 3 credit hour courses, and is organized into a 40-minute lecture and a one hour and twenty minutes lab exercise for each class. On the first day of the class, the instructor reviewed the syllabus of the course, course structure, and described the expectations of the course and student expectations from the instructor, and explained how a course grade and lab work were determined and evaluated. The project grading rubric was also introduced to the students in the first day of the class and is shown in Table 6. A 40-minute lecture covers topics about the FPGAs and Verilog syntax. Outlines of the key lectures for robotics are:

1) Getting started with the FPGA device
    a. Outline of the FPGA architectures (e.g. Xilinx Sparten-6 and Altera Cyclone IV)
    b. Embedded hardware (e.g. memories, data path, and controller) in the FPGAs
    c. State Machine Design in Verilog HDL for robotics

2) Robotic Programming using the FPGAs
    a. FPGA design process
    b. Verilog HDL syntax, and pin assignment of the FPGAs
    c. How to use Verilog HDL on both FPGA platforms (Xilinx & Altera)
    d. Robot Locomotion – servo motors and calibrations

3) FPGAs interfacing with digital sensors, cameras, and wireless radios
    a. Demonstrate the use of FPGA GPIO pins
    b. Introduce analog sensors and analog-to-digital converter
    c. Introduce I2C/SPI-based digital sensors and the communication protocols
    d. Introduce wireless radio (e.g. Bluetooth, WiFi or ZigBee if time permits)

Each lab exercise requires students to complete one cycle of the design process as illustrated in Figure 1. Within Quartus II, students started with a new project in the Quartus II software, created Verilog HDL model of their design, wrote a test bench program to verify if a Verilog HDL model performed correct functions and met timing requirements, configured simulation environment, and then compiled an entire project for simulation. After simulation was successful, Verilog code was programmed into FPGAs with a bit stream file.

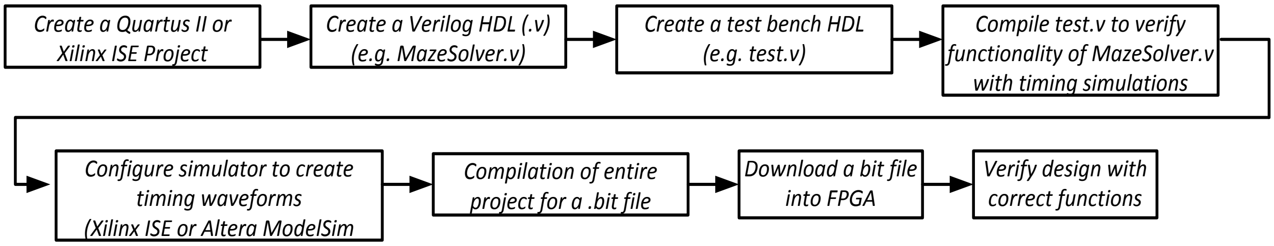

Create a Quartus II or Xilinx ISE Project
Create a Verilog HDL (.v) (e.g. MazeSolver.v)
Create a test bench HDL (e.g. test.v)
Compile test.v to verify functionality of MazeSolver.v with timing simulations
Configure simulator to create timing waveforms (Xilinx ISE or Altera ModelSim
Compilation of entire project for a .bit file
Download a bit file into FPGA
Verify design with correct functions

Fig. 1 FPGA Design Process

The weekly lab exercises were designed for students to gain sufficient experience to allow them to start building their robot vehicle with FPGAs, although programming in Verilog HDL and code debugging has been the most challenging part of robotic design for students. Most students were able to get acquainted with Verilog syntax and debugging techniques through the lab exercises before working on their robotic projects. Those lab exercises included basic Verilog programming to create circuits, writing a test bench program to verify the functionality of the circuit, code debugging, analog and digital sensors interfacing, servos and DC motors control techniques to achieve a specific accuracy for position or velocity. In some lab exercises, students simply copied and pasted a sample Verilog code (e.g. I2C/SPI lab) to simulate behaviors for sending data between FPGAs to a FPGAs peripheral (e.g. LEDs) to understand how serial communication works. Labs like this were not challenging for students until they encounter similar problems when working on an actual sensor. By the time students need to know digital interfaces for SPI or I2C, they already have a reference design to review. Students may choose to work individually or in a group of two on lab exercises or the final project. Each team can be self-formed by the students or assigned by the instructor. Students who are not good programmers often ask if she/he can work with a student who has better programming skills. In this case, the instructor helps the students to form a team.

Course Resources

The course objective is to teach digital designs with Verilog by building FPGA robots. Robotics education requires a low cost robot to support a pedagogical model of one robot per student and to attract significant students' interest [17]. This interest is expected to grow in engineering education. One robot per student also allows students to build robotics outside the classroom. Fig. 2 shows the FPGA robot hardware and is a low cost robotic design which supports advanced designs with sensors, network connectivity, and communications with Bluetooth and WiFi networks. The DE0-Nano board features a powerful Altera Cyclone IV FPGA (with 22,320 logic elements), 32 MB of SDRAM, 2 KB EEPROM, a 64 MB serial configuration memory, a built-in USB Blaster for FPGA programming, inputs and outputs (I/O) including 2 pushbuttons, 8 user LEDs and a set of 4 dip-switches. To facilitate using external sensors with the FPGAs, the DE0-Nano has 72 I/O pins and three 8-channel 12-bit A/D converters, a 13-bit 3-axis accelerometer. The small breadboard area on the circuit board allows for solder-less prototyping of FPGAs with external sensors to achieve any specific sensing and navigation needs.

A bag of sensors and wireless modules were provided to each student to carry out the required lab exercises. These sensors were I2C/SPI/UART based digital sensors and include the following parts: infrared proximity sensor, reflectance sensor array, magnetometer with six-degree-of-freedom, speaker, and Bluetooth and WiFi radios. Android tablets with built-in Bluetooth and WiFi were available in the lab for students to connect to the Bluetooth module on the robots.

Figure 2 shows a robot chassis and an assembled FPGA robot. The total cost of the assembled parts was $160 per robot (aluminum robot chassis $25, two continuous rotation servos $26, two plastic wheels and one wheel ball $10, a Parallax board of education shield PCB $35, a DE0-NANO FPGA board $59, 5-cell AA battery pack, nuts, screws, and standoffs $5). This low cost robot made it possible to have one robot per student. The university provides each student a laptop for four years so the students can use the required design software on their laptop for FPGA designs. These resources facilitated the effective use of self-paced learning for students and encouraged exploration with the FPGA robotics hardware.

Our embedded system lab has a 22 student capacity and had 18 students enrolled in the Fall 2014. To reduce the chaos with the students in the lab, the instructor distributed discrete components required for each lab during lab time.

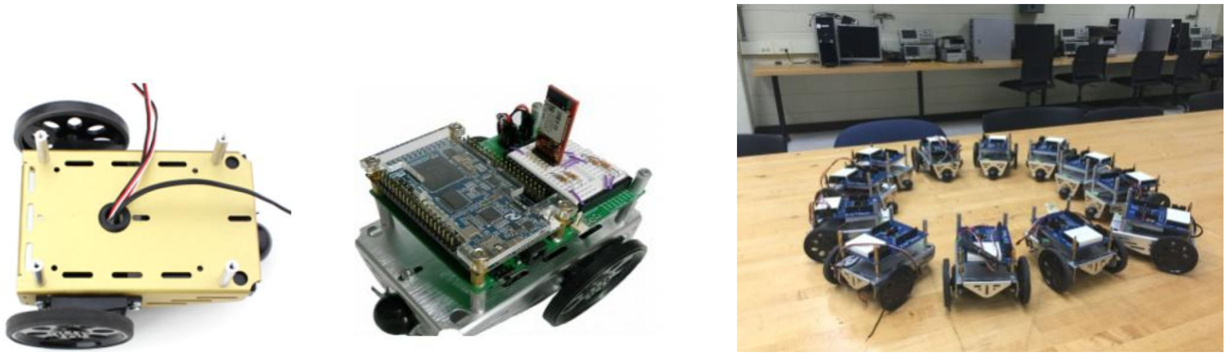


Figure 2 Robot Chassis and Assembled FPGA Robots [18]

The objective in each lab exercise was to orient the class more toward achieving some robotic objectives. The lab focuses include servo calibrations, servo position controls with pulse width modulations, obstacle detections, and a line following function using a reflectance sensor array, etc.. These topics cover some of the common issues which students are likely to encounter while integrating FPGAs into their robotic projects. The new labs for robotics address some common obstacles in building robotic vehicles which are included in the following lab exercises.

Building Robotics in Labs and Projects

Educators have developed several FPGA-based courses [8]-[12], but few courses were created to accelerate student learning curve for FPGAs and digital sensors from the practitioner's point of view. Lab exercises with a focus on digital sensors interfacing to FPGAs are an extremely important component of digital designs. Our lab exercises focus on how to use FPGAs with its rich communication interfaces on the DE0-NANO board. These labs include accelerometer (G-sensor) with I^{2}C interface, SPI based ADC, and Bluetooth with UART interface. Each lab exercise was conducted as follows: first, a bill of materials to use in a lab exercise was reviewed with the students. Second, the concepts or lab procedure related to each lab work, which has been introduced in lectures, were reviewed with lab handouts. Third, as programming for FPGA is the main task in each lab, a set of keywords, clocking strategies, ports declaration, always blocks, looping statements, blocking or non-blocking statements were reviewed with the students for efficient programming during the lab time. Each lab exercise has a lab packet which includes detail lab procedure and a code template. The code template contains most source code written for the lab and the students only need to figure out the missing source code and test it to complete a lab. This ensures that students can complete each lab on schedule with a good learning experience of using FPGAs. The demonstration robot for each lab exercise which incorporated all of the discrete components on a breadboard, Verilog source code, was provided to students after each lab was completed. Students can use the lab exercises as references to construct application specific projects for larger project development.

Laboratory Exercises with a focus on FPGA Robots

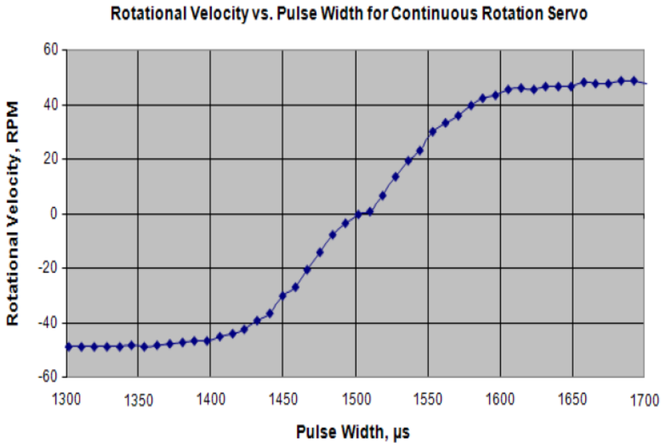
The students were required to complete the following lab exercises to prepare themselves for their final projects.

1) **Servo Calibration Lab** – Center the servo
The students learned pulse width modulation (PWM) technique to calibrate servos with predefined PWM signals in Verilog HDL.

Table 2 Servo Calibration Parameters

Pulse width (ms)	**Servo position**
1.5	Center and hold position
1.3	Clockwise rotation
1.7	Counter Clockwise rotation

2) **Servos speed control Lab**
Students created a transfer curve graph during the motor calibration process as illustrated in Fig. 3. Each servo motor may have different performance characteristics. With this graph, students can get a good idea of what to expect from their servos for a certain pulse width. This particular servo motor ranges from −48rpm to 48 rpm over the test pulse widths from 1300µs to 1700µs.



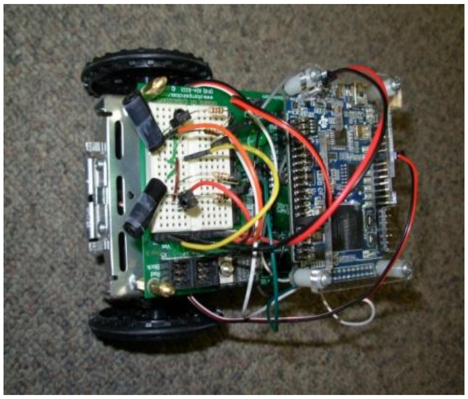


Fig. 3 Servo Transfer Curve Graph [18]

Fig. 4 Robot Navigation with IR sensors [18]

3) **Robot locomotion Lab** – basic robot actions
This lab allowed students to learn how each robotic vehicle navigate in forward, backward, left turn and right turn directions according to a pulse width it received from the FPGAs as shown in table 3.

Pulse width (left servo)	**Pulse width (right servo)**	**Robot Locomotion**
170	130	Forward
130	170	Turn back
170	170	Turn right
130	130	Left Turn
170	150	Run a circle in clockwise direction
150	130	Run a circle in counter clockwise direction
150	150	Stop all motions

Table 3 Pulse Width Parameters (in milliseconds) for Robot Navigation

4) **Lab for Obstacle Avoidance for Mobile Robots**
This lab was to demonstrate how to create a robotic vehicle to detect unknown obstacles in its path simultaneously with the steering of the robot to avoid collision and advance forward. Students learned a navigation algorithm and how a TV remote control works using the infrared LEDs (IR) and receivers to navigate a robot. The robot setup is shown in Figure 4.

5) **Digital sensors with I^{2}C/SPI interfaces**: this lab introduced students how to use I^{2}C/SPI-based sensors and memories. A 2K-bit I2C serial EEPROM and ADXL345 3-axis digital accelerometer were embedded components on the DE0 FPGA board and introduced to the students. This lab focused on serial communication protocols (I^{2}C and SPI) and how to use Verilog HDL to capture the output data through their I^{2}C and SPI interfaces.

6) **ADC Interfacing with FPGAs**
Figure 5 shows how the students used the Analog to Digital Converter (ADC128S022) to interface hardware for a position sensor to the FPGAs. This lab used an 8 channel 12-bit ADC available on the DE0 FPGA board and demonstrated how to use a SPI-based ADC device works to acquire sensor data to the FPGAs.

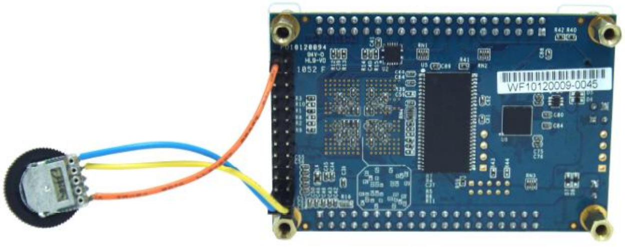
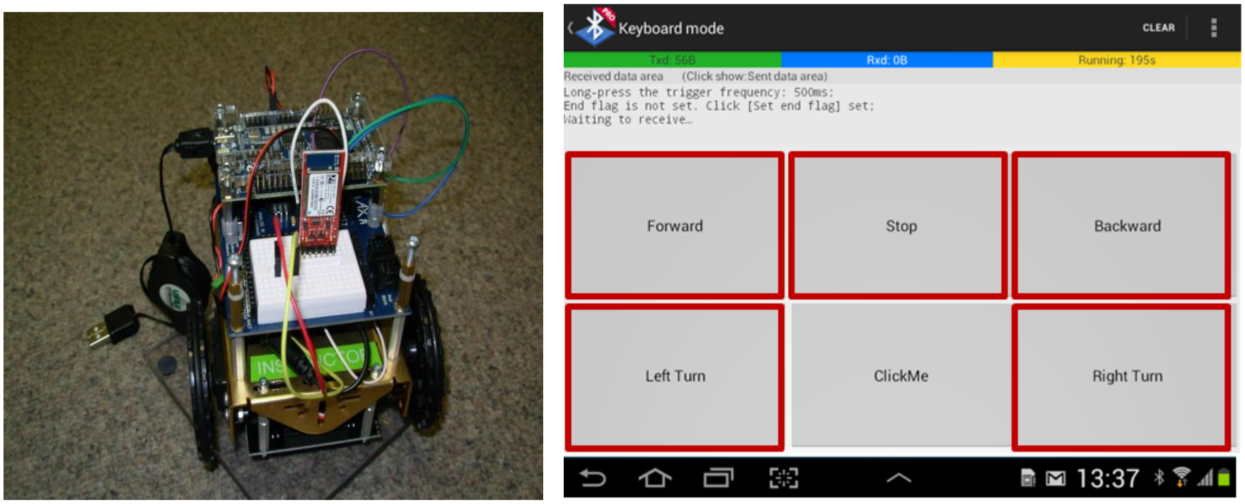


Figure 5 Hardware Setup for the ADC Lab [6]

7) **Bluetooth Remote Control (R/C) Robots**
The last lab exercise, a Bluetooth RC robot, was the culmination of all the lab exercises and was used to create a radio control robotic vehicle with the RN-42 Bluetooth radio. This lab allowed the students to learn a system-on-chip using FPGAs that supports running applications in a mobile operating system. This lab introduced the students to the design of an UART interface in Verilog HDL for Bluetooth communications on FPGA based systems. The students worked on the Android application file (Bluetooth SPP.apk) provided in the class and had it run on their Android devices. This app allowed students to transmit user commands and navigated their robots. Fig. 6a shows the Android application, known as Bluetooth SPP Pro, used in this lab, which is available for download in Google’s Play Store.



Txd: 56B Rxd: 0B Running: 195s

Received data area (Click show:Sent data area)

```
Long-press the trigger frequency: 500ms;
End flag is not set. Click [Set end flag] set;
Waiting to receive…
```

Figure 6a Bluetooth Connected Robot and Remote Control Using an Android Phone [18] [19]

Figure 6b shows the combined lab exercises (1 to 7). These lab exercises were developed for students to easily expand any robotics labs and to reuse the code to build the “brain” and “eyes” of an autonomous vehicle. The functional architecture in Fig. 6b illustrates how the students use the lab exercises to build a complete autonomous robot with wireless control. All of the lab exercises combining lab 1 to 7 help the students choose their final projects and make their projects successful. Additionally, code work from the lab exercises can be reused to create different robotics projects, and thus accelerating student learning curve in design designs with Verilog HDL.

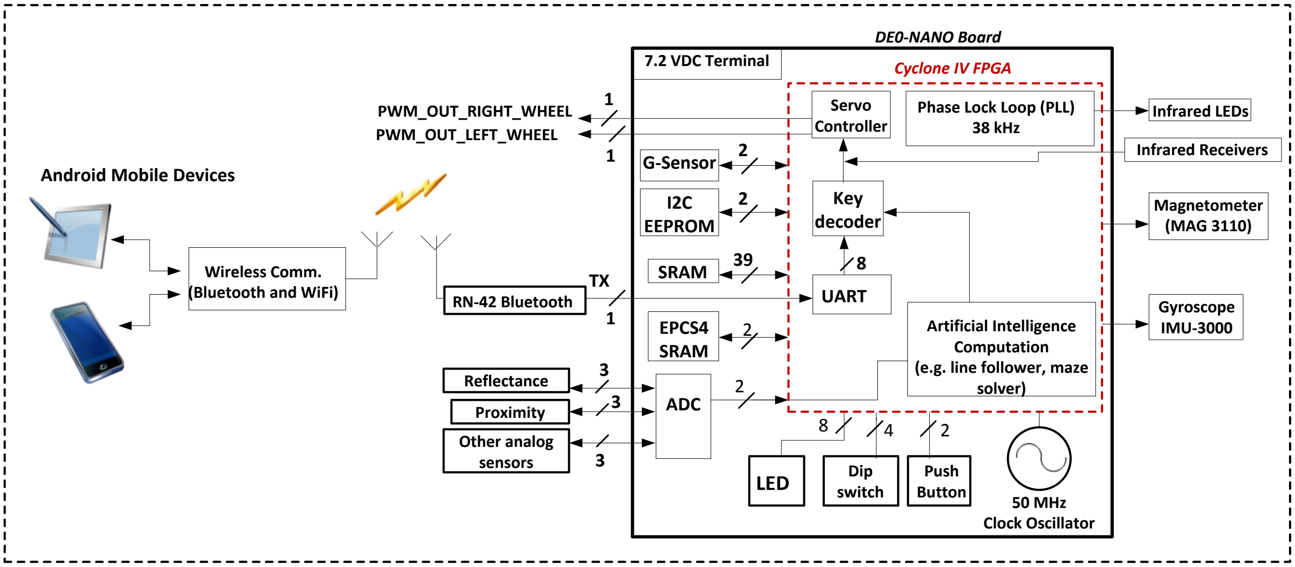


Figure 6b Functional Architecture of the Mobile Robot

Course Structure/Schedule

A detailed class schedule is shown in Table 4. Notice that a final project proposal is due on week 10 and all of the lab exercises and assignments were completed by week 10. Each final project proposal was reviewed by the instructor and the instructor provided feedback to the students in terms of their project complexity, time frame, and possible design issues. Students were allowed to change their projects after receiving feedback from the instructor. Once students and instructor agreed on the project proposals, students began to work on them. Individual team meetings were then held and each team selected a team leader who managed the progress of a project and assigned work to each team member. Each team may seek the instructor's help anytime during the class period. Such an arrangement allows students to develop communication and teamwork skills and helps them succeed in their senior design projects and also to simulate a working environment in industry.

Table 4 Detailed Class Schedule

	Monday	Wednesday
Week 1	FPGA architectures	**Introduction**, syllabus, lab exercises and final project. Xilinx ISE/Altera Quartus II Digital Design Process
Week 2	**Combinational logic** Structural vs. behavioral Verilog models; Verilog programming; test bench Xilinx Lab 1 (Nexys-3)	**Combinational logic** Structural vs. behavioral Verilog; Verilog Programming Xilinx Lab 2: multiplexers decoders/1-bit adder (Nexys-3)
Week 3	**Sequential logic** Verilog model: latch and flip flop; multi-bit adders, ripple carry adder, look ahead carry adder; counters	**Sequential logic** Verilog model: latch and flip flop; multi-bit adders, ripple carry adder, look ahead carry adder; counters
Week 4	Clock divider design Switch debounce techniques LEDs **Finite State machine (FSM)** Xilinx Lab 3 FSM (Nexys-3 & DE0) Traffic light controller design	Catch up Day
Week 5	Clock divider design Switch debounce techniques LEDs **Finite State machine (FSM)** Xilinx Lab 3 FSM (Nexys-3 & DE0) Traffic light controller design	4x4 Keypad scanner and encoder interface with FPGAs **Lab#4 Knight Rider LED lights design** (Nexys-3 & DE0)

Week 6	Data path and controllers; 32-bit **Arithmetic Logic Unit (ALU)** (Nexys-3)	Motors (DC/Stepper/Servos) **Lab#5 pulse width modulation technique** (Nexys-3 & DE0, and H-Bridge board)
Week 7	Motors (DC/Stepper/Servos) **Lab#5 pulse width modulation technique** (Nexys-3 & DE0, HB boards)	Catch up Day
Week 8	**Intro. to Robotics**; parts and supplies; sensors for projects	**FPGA Memories, data path, and controller.** Robotics Lab#1 servo calibration
Week 9	Robotics Lab#1 **servo calibration** Robotics Lab#2 **servo speed control**	Robotics Lab#1 **servo calibration** Robotics Lab#2 **servo speed control**
Week 10	Robotics Lab#3 **ADC** ***Final Project Proposal Due***	Robotics Lab#3 **ADC**
Week 11	Robotics Lab#4 **I2C/SPI-based sensors**	Robotics Lab#4 **I2C/SPI-based sensors**
Week 12	Robotics Lab#5 **Obstacle Avoiding robot basics**	Robotics Lab# **Bluetooth remote control robot**
Week 13	Robotics Lab#6 **Bluetooth remote control robot**	Robotics Lab#6 **Bluetooth remote control robot**
Week 14	**Final Project Work**	**Final Project Work**
Week 15	**Final Project Work**	**Final Project Work**
Week 16	Project Presentation & demonstration	Project Presentation & demonstration

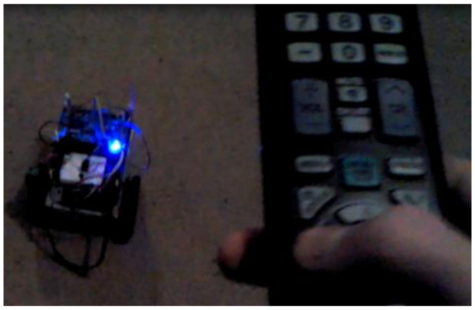
Student Final Robotic Projects

There were 18 students enrolled into the class in the fall 2014. 13 teams were formed and each of the 13 teams was expected to use different sensors to accomplish different navigation features. Each final project may have a team of two students or an individual student depending on the students' preferences. Among the 13 teams, 12 teams presented their projects successfully and demonstrated functional robotic prototypes except one team. A successful project was defined as the following:

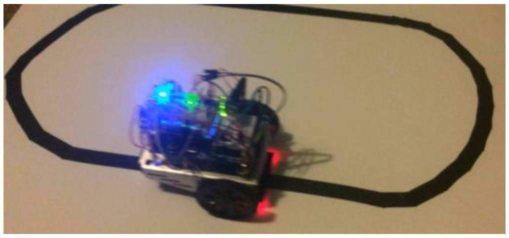
Type 1: The project was a success if it delivers all or most of graded items on the rubric (Table 6) and the rubric score is 90 and above.

Type 2: The project was a success if it delivers on all agreed project objectives, scope, quality and outcomes independent of the graded items on the rubric.

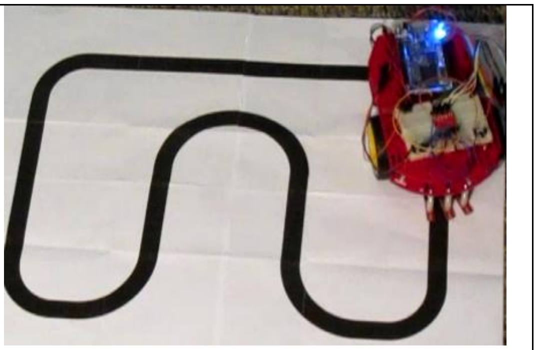
Table 5 List of Representative Final Robotic Projects

Team	Project Title	Project Description	Robot Prototypes
Team 1	Infrared Remote Controlled FPGA Robot	Use a television remote that communicates using an infrared led, and the receiver outputs pulses that represent the signal sent by the remote. The FPGA receives the signals and determines a direction each of the servo motors should turn. ***Sensors used in this project:*** IR transmitter -Samsung TV remote; IR Receiver –TSOP382	

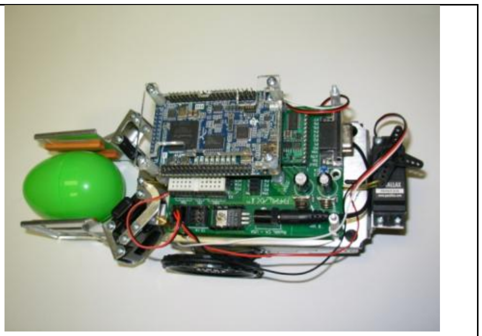
The student project summary from team #1: "the goal of this project (Infrared Remote Controlled FPGA Robot) was to implement wireless controls on the robot used in class. The project works as expected and has a simple design so it can be applied to a variety of real life applications if desired."

Team 2	Line Follower FPGA Robot (motors are servo)	Create a mobile robot that can traverse along a curved line using input from a reflectance sensor array. ***Sensors used in this project:*** Zumo reflectance sensor array from pololu.com	

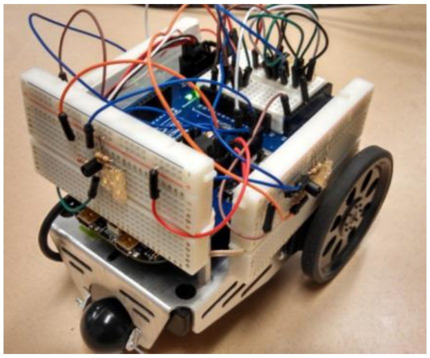
The student project summary from team #2: "Overall this project (Line Follower FPGA Robot with servo motors) was fairly challenging and it actually forced me to sit down and spend some hard hours creating and debugging code and I now feel much more comfortable with Verilog than I did before. I also learned that Bluetooth communication isn't as horrible as it sounded when I first contemplated the idea of this project"

Team 3	Line Follower FPGA Robot (motors are DC geared motors)	Create a self-navigating robot to detect a line. An Analog-Digital converter was used to read the values from the sensors. A 1-amp motor driver was used to supply power to the motors, and was controlled through the GPIO pins on the DE0 NANO. ***Sensors used in this project:*** – Sparkfun part# 11769 Reflectance sensor	

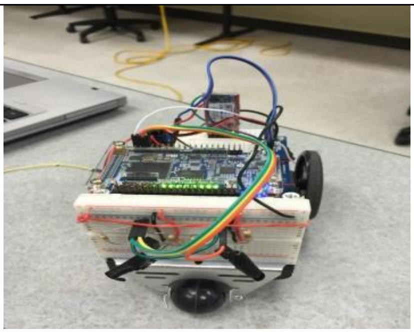
The student project summary from team #3: "Overall this project (Bluetooth line follower with DC motors) taught me a great deal on Bluetooth connectivity works. It also taught me how to handle multiple different functions of the robot with the use of case statements. I felt this project demonstrated a great deal of what I learned this semester and I enjoyed the challenge."

Team 4	A FPGA Robot with Gripper	Create a robot to follow a line with a reflectance sensor bar and use its gripper to pick up an object at its start point and release the object at the end point of a line maze.	

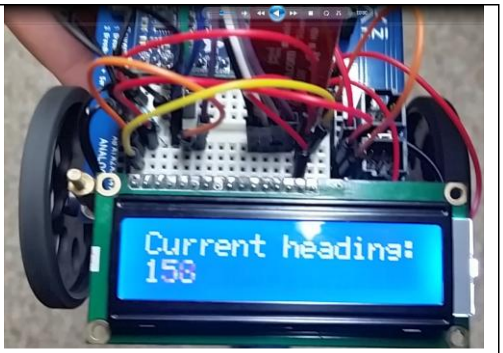
The student project summary from team #4: "The goal of this project (A FPGA Robot) was met and we now have a fully functional robot that is capable of navigating and solving a maze. This was accomplished with an Altera Cyclone 4 FPGA board controlling two servo motors on a robotic car. The car used an array of six IR sensors to detect the location by looking for black or white under the robot. The robot was designed to solve the maze with a simple left-turn algorithm which was limited to solving any tree-structure mazes."

Team 5	MouseBot	Create a mouse-like robot. MouseBot has 4 Infrared sensors, used as proximity sensors, and will sense when something is close to it. When something is close to it, it will sense where that object is, and then will "run" away from the object. ***Sensors used in this project:*** Infrared receiver: OP245A; IR transmitter: LTR-4206E	

The student project summary from team #5: “My project (MouseBot2.0) had an added Bluetooth controller that didn’t have too many issues at all.”

Team 6	Invisible Wall	Create a robot to navigate within a confined area. Using the technique of distinguishing between light and dark surfaces, the robot can be controlled anywhere within the range of the black line. Once the robot attempts to move past the black line, the sensor will be triggered and the robot will not be allowed to move in the forward direction.	

The student project summary from team #6: “Though the objective of this project (Invisible Wall) was successful, due to time constraints some additional constraints could have been included. For instance the original goal was to use two sensors from the sensor array (i.e. sensor position 4 &5). The ADC on the DE0-nano board can only check one channel at a time and the original goal was to toggle between both sensors with a state clock. I was not able to implement this into my code so I decide to choose A0 as the sensor to take readings, since it is in the center position on the sensor array. Overall the project did complete the goal.”

Team 7	Magnetically Aware Robot	Create a robot to measure the strength of a magnetic field. The magnetometer on the robot can detect irregularities in the earth’s magnetic field. It is possible to use the magnetometer to indicate the location of magnetic iron ore. The robot shows “Current heading 158 (degree)” on a 2 by 16 LCD display ***Sensors used in this project:*** Magnetometer --- LSM303DLHC	

The student project summary from team #7: “Overall this project (Magnetically Aware Robot) was a successful implementation of a basic robot using a magnetometer. There are many more ways on which to expand this project, from increasing the accuracy of the robot (to decrease the variance needed to find its heading), to allowing user to select custom degree headings. From this project I was able to get a firm grasp on many concepts, like I2C and serial communication.”

Grading

A grading rubric was used to evaluate the outcomes of the student projects (Table 6). Each team was allowed to present a project work in 10 to 15 minutes and 5 minutes for questions from the class. Criteria for a presentation were evaluated as follows:

Project Description	Bill of Materials
Design Process: circuit schematic, flow charts	Construction and testing
Project meets specifications	Live demonstration

Table 6 Grading Rubric - Final Project

Presenters:	Project Being Evaluated:

Instruction: To help provide feedback for the presenters and the instructor, please provide a score (via a checkmark) between 1 and 4 with 1=poor and 4=excellent

Criteria	1(10%)	2(50%)	3(75%)	4(100%)	Total	Poss. Pts
Project proposal was submitted on time. Project goal, bill of materials, cost, and design process are described and specified						
Project has demonstrated at least one type of sensor to enhance robot's navigation capability						
Circuit schematic was shown						
Construction and testing was clearly discussed						
Complexity						
Presentation was meaningful and successful						
Demonstration of an ability to develop a prototype using FPGAs						

Table 7 Assessment of Laboratory Experience: Percentage of Student Responses on End-of-semester Survey (Q1: N = 18 from the students' project reports), (Q2, Q3, Q4, and Q5 from the students' on-line survey response: N=7)

Question	**SA**	**A**	**N**	**D**	**SD**
Q1: My final project for the FPGA mobile robot was successful.	**0**	**16**	**0**	**2**	**0**
Q2: This class was presented in an interesting and stimulating manner.	**0**	**3**	**3**	**1**	**0**
Q3: The class presentations were clear, logical, and easily understood.	**0**	**6**	**0**	**1**	**0**
Q4: Course objectives were clearly defined and reflected in the course evaluation	**2**	**3**	**1**	**1**	**0**
Q5: The instructor's use of lab equipment and materials was valuable	**4**	**2**	**1**	**0**	**0**

SA = Strongly Agree, A = Agree, N = Neither Agree nor Disagree, D=Disagree,
SD = Strongly Disagree

Conclusion

Overall, introducing digital designs with a hands-on FPGAs implementation has been very instrumental in the development of successful student projects. Students enjoyed building autonomous robots and often carried their robots in the campus and worked together on their lab assignments and projects. The use of FPGA robots sparked their interest and helped them learn FPGAs. Using reconfigurable FPGAs for robotic projects was a very helpful teaching tool which allows a higher level of abstraction in student projects than traditional non-reconfigurable hardware, which is especially helpful in simplifying designs for Bluetooth communications with an UART interface to FPGAs. The use of FPGAs allowed the complexity of student projects to increase as they developed artificial intelligence algorithms or any kind of compute-intensive programs. From the student evaluation survey in Table 7, some students also commented that they felt it was too difficult to contact the instructor, or could have more planning in advance for delays in the schedule. The delays in the schedule can happen in this class especially with back-to-back scheduling of two lab activities. The delays in schedule can be remedied by adding a lab assistant to the lab. Some students felt that the instructor should have provided directions if they chose a bad project. Teams or individuals are expected to self-discover a solution for their final projects. This can be improved by a better communication with the students. Some students preferred to use Apple iOS over Android operating system to transmit user commands to their robots. The problem was that the Android system supports the Bluetooth Serial Protocol Profile (SPP) for data connections whereas Apple's iOS doesn't. Establishing Bluetooth data connections with Apple devices requires a unique discovery and pairing sequence with the Apple authentication co-processor. Therefore, a new Bluetooth module (e.g., Bluetooth Low Energy) which supports Apple's iOS is needed. The improvement items will be remedied and are summarized as follows:

- Recruit a lab assistant to help organize the lab activities better and make use of the class time more efficiently.
- Extend office hours when students are working on their robotic projects.
- Add a Bluetooth module that supports Apple's iOS.

The best way to determine if the new course structure improves its effectiveness is to evaluate the quality of student projects. Based on the students' final projects, most of them were able to accomplish all of their originally proposed objectives but with one exception. Therefore, a general conclusion is that the new course structure helped to engage students in learning FPGAs and have helped them to succeed in their projects. However, it is necessary to collect statistical data for a few years to ensure that this course model is sustainable. One of the disadvantages of this course is that a significant number of sensors and mechanical parts are necessary since the lab activities and student projects require extensive use of hardware. As the class size grows, more instruction resource will be needed.

In conclusion, the course presented a systematic approach to FPGA implementation using Verilog HDL. The lectures and lab exercises provided students with essential design methodologies from basic coding principles to hardware implementation on real FPGAs hardware. The course provided valuable skills to students who will pursue career in FPGA design and embedded software development.

References

1. "Advanced Digital Design with the Verilog HDL", by Michael D. Ciletti, 2011, Prentice Hall, ISBN 9780136019282
2. "Fundamentals of Digital and Computer Design with VHDL", by Richard S. Sandige, Michael L. Sandige, McGraw Hill, 2012
3. "Digital Design, An Embedded Systems Approach using VHDL", by Peter Ashenden, ISBN 97801203695284
4. "Digital Systems Design Using Verilog", by Charles H. Roth Jr., Lizy Kurian John, and Byeong Kil Lee, Cengage Learning, 2016, ISBN 9781285051079
5. Digilent, Inc., Nexys-3 board, available on: http://www.digilentinc.com
6. Altera Inc., DE0-nano board, available on: http://www.altera.com/education/univ/materials/boards/de0-nano/unv-de0-nano-board.html
7. W. Wolf and J. Madsen, "Embedded systems education for the future," *Proc. IEEE*, vol. 88, no. 1, pp. 23-30, Jan. 2000.
8. Tyson S. Hall, *et al.*, "System-on-a-Programmable-Chip Development Platforms in the Classroom," *IEEE Transactions on Education*, Vol. 47, no. 4, August 2004.
9. José Nelson Amaral, "Teaching Digital Design to Computing Science Students in a Single Academic Term," *IEEE Transactions on Education*, Vol. 48, no. 1, August 2005.
10. Hiroyasu, *et al.*, "Use of Student Experiments for Teaching Embedded Software Including HW/SW Co-Design," *IEEE Transactions on Education*, Vol. 52, no. 3, August 2009.

11. Akash Kumar, *et al*., "Project-Based Learning in Embedded Systems Education Using an FPGA Platform," *IEEE Transactions on Education*, Vol. 56, no. 4, August 2013.
12. Christos Ttofis, *et al*., "FPGA-Based Laboratory Assignments for NoC-Based Many core Systems, "*IEEE Transactions on Education*, Vol. 55, no. 2, August 2012.
13. "What is an FPGA and FPGA applications", Xilinx Inc. [online]: http://www.xilinx.com/training/fpga/fpga-field-programmable-gate-array.htm
14. Patrick P. Fasang, "Prototyping for industrial applications,"*IEEE Industrial Electronics Magazine*, March 2009.
15. Juan J. Rodriguez-Andina, *et al*., "Features, Design Tools, and Application Domains of FPGAs," *IEEE Transactions on Industrial Electronics*, Vol. 54, no. 4, August 2007.
16. K.E. Newman, J.O. Hamblen, and T. S. Hall, "An introductory digital design course using a low-cost autonomous robot," *IEEE Transactions on Education*, Vol. 45, no. 3, August 2002.
17. James McLurkin, *et al*., "Using Multi-Robot Systems for Engineering Education: Teaching and Outreach with Large Numbers of an Advanced, Low-Cost Robot," *IEEE Transactions on Education*, Vol. 56, no. 1, Feb. 2003.
18. Board of education shield product guide, Parallax Inc., available on: http://www.parallax.com
19. Bluetooth SPP app., available on Google play store.