EECS113: Processor Hardware-Software Interface
Final Project Report:
Application Controlled Car

"Squad 3"
William Hu - 10482351
Edgar Prieto - 39185940
Tam Nguyen - 12276989

June 9, 2017

1 Project Idea and Outcome Changes

In the project proposal, we said that our main goal is to have a car that will be controlled by a phone application through bluetooth. We said that we will try to incorporate the gyroscope on our phone and make is to that when we tilt forward, the car goes forward, and when we tilt left, the car goes left etc. What we ended up having in the end, was a car that was controlled by a phone application through bluetooth but instead of the gyroscope implementation, we had buttons that would send data to the bluetooth and eventually the rotors will function accordingly.
Also, we talked about how we would try to implement a sensor that would stop the car when it's about to fall off an edge but what we did instead was have a sonar that would just auto-brake the car if something is less than a specified distance that we chose from the sonar, which would be at the front of the car. Aside from having to buy some extra parts or using extra materials, those are changes that we have made throughout the couple weeks that we worked on the project.

2 A Rough Timeline of Our Highlights

*A description on highlights and major things we did during certain weeks.
*Also, connections will be explained in more detail in the section where we talk about peripherals we used and how we connect them.
*This may be a little more clear if the Project Proposal was read first to have an idea of our project.

Week 5:
We thought about materials and placed orders for the initial parts that we thought were needed to make the project work. In this case it was the Chassis kit, battery box, bluetooth PMOD.

Week 6:
We received all our parts by the end of week 6, so during that wait period we just researched more about the project and any tips or advice that we can find regarding how to use our peripherals and see if they are compatible with PYNQ.

Week 7:
Now that we got the parts that we need, we started constructing the car with the chassis kit we bought. After assembling the car the way we want it to look, we attached the battery box to the bottom of the car, which will be the power source. Once that is done, we tested the battery box on the PYNQ by connecting the battery box to the board, which successfully works and then we moved onto the bluetooth, which connected but we were still unsure on how to make it work with the phone so we put it to the side for now and look more into it.

Week 8:
We test the equipment and came up with a python code that will test the rotors in different cases (forward, backward, left, right), seeing if the wheels would turn accordingly but it did not work as we expected because of the wiring, so we did some research and asking around, finding out that we needed an H-Bridge to help us send voltage and signals from the H-Bridge to PYNQ which then controls the wheels. We also started trying to figure out how to make a phone application to control the car.

Week 9:
The extra parts that we ordered came in and the car is now fully assembled. We were able to use the H-Bridge to operate the car rotors, as well as have an application that we can use on our phone to control the car once the bluetooth PMOD was implemented. For some reason, the bluetooth PMOD was giving us a lot of trouble and we could not make it work with just the PYNQ on its own, so we resorted to using an Arduino board and then connect the bluetooth PMOD to the Arduino which will then be connected to the PYNQ board. We managed to get something working and the phone app connects to the bluetooth, but there are still bugs that we need to resolve because the car was not working / getting the inputs from the phone. We also placed an order for the sonar in case we finish with the car a little early.

Week 10:
We were hoping that by week 10, we would have everything done and ready to be presented, only tweaking things here and there, but that was not the case. Coming into week 10, we still had problems with getting the bluetooth picking up the buttons from the phone application. So we decided to do some rewiring, connecting the bluetooth to the Arduino which then communicates with the PYNQ board by being connected into PMODB and controlling it through programming the PMOD ports.
We had ordered a grove adaptor for the PYNQ board which was used to connect the sonar and we would receive analog from it, which we will pick a specific number and once the sonar picks up values less than that specified amount, the car auto-brakes. The wheel rotors were still connected to the H-Bridge, but the H-Bridge is now connected to PMODA so that we can send signals to the H-Bridge which will operate the wheels based on input from the phone application.

TL;DR (for peripheral and sensor connections):
Bluetooth→Arduino→PYNQ (PMODB)
Sonar →PYNQ (grove adapter)
Wheel Rotors →H-Bridge →PYNQ (PMODA)
Battery Box →PYNQ
Battery Box →Arduino

3 Peripherals and Sensors

1) **Bluetooth RN-42:** The bluetooth PMOD is basically used to communicate between the phone and the PYNQ board. Basically, when the bluetooth is powered on, the phone can connect to it just like any other bluetooth device and will then be able to send information from the phone input to the Arduino which communicates with the PYNQ which communicates with the rotors.
Connection: bluetooth mod connected to arduino which is then connected to PMODB on PYNQ board where will control and program the pins.

2) **Sonar:** The purpose of the sonar is to detect when the car approaches a wall or an obstacle that is in the way. For our project, once the sonar detects something within a distance that weve specified, the car will automatically stop. From there we cannot move the car forward anymore, but we can turn left, right, or reverse so that we can maneuver.
Connection: sonar is connected to a grove adapter which is connected to the PYNQ board. We just need to receive analog from it to know how much we should specify for the distance before the car autobrakes.

3) **H-Bridge:** Purpose of the H-Bridge was to help us send signals from the PYNQ board and signal which rotors turn on (i.e. forward (all rotors on), right (only left rotors on)). It is powered from the PYNQ board and we send signals in binary, where write(1) means rotor on and write(0) means rotor off.
Connection: the rotors are connected to the H-Bridge, power to one end and ground to another, separated by left and right rotors. So the left two rotors connect to same node and right two connect to same node. From there, the H-Bridge is connected to the PYNQ board (PMODA), where we will send voltage to the bridge, as well as the binary signals 0 and 1 depending on what the user input is.

4) **Arduino:** We originally did not want to use the arduino but because we were using the bluetooth so that we can communicate between our phone and the board, we had trouble with the serial when connecting the bluetooth PMOD to the PYNQ board. Since arduino had built in support for the bluetooth, that is what we decided to use it for. Its main purpose was to make the bluetooth work because we were in week 9 and could not figure out how to make bluetooth work with the PYNQ board.
Connection: the arduino is connected to PMODB on the PYNQ board so that we can power it as well as receive what the user is sending through bluetooth from their phone. From there we send signals to the H-Bridge specifying which rotors to turn on and spin the wheels.

4 Problems and How We Overcame Them

1) Confused on how to work the rotors with the PYNQ board because there was not enough connections and we did not know how to provide voltage to the rotors.
*We solved this problem by purchasing an H-Bridge which makes it easier for us to distribute voltage the rotors. The rotors would connect to the H-Bridge and the Bridge would connect to the PYNQ to receive signals and power. Based on the signal from PYNQ, rotors will turn on and off accordingly.

2) Bluetooth not working with PYNQ board. We had the drivers and everything, but for some reason there was no serial communication between the bluetooth PMOD and the PYNQ board. Because of this, the bluetooth was not working properly.
*We resorted to using an Arduino Uno board to handle the bluetooth PMOD device because there was already built in and supported serial communication between the bluetooth piece and arduino board. So we ended up connecting the bluetooth to the arduino and then connecting the arduino to PYNQ board to transfer data.

3) Bluetooth PMOD and sonar would not work at the same time because the sonar was producing analog and the bluetooth PMOD was producing digital and we were not able to use them both at the same time (separately was okay though). We had to include both arduino analog and arduinoio libraries, and that was when the error really occurred.
*This problem occurred back when we had the connection of the arduino to the IOPs of the PYNQ. This produced digital signals and did not want to work together with the sonar, which produced analog signals. So instead we connected the arduino to PMODB since PMODA was taken by the H-Bridge. By doing this we are able to remove the arduino io library since we can now control the arduino/bluetooth through the PMOD library.

4) Smaller problems and solutions:
*Needed jumper wires to make or extend connections.
*PYNQ board was touching wheels when plugged into battery box, so elevated using a random box/container.
*Powering our project with laptops was not sufficient plus we need the car to be able to move around without having to drag the laptops around so we had battery boxes attached to the chassis.
*Wanted to find way to make it so that the PYNQ board runs the specific code for this project the minute it turns on, but we were not sure if that was possible. Therefore, we would have to manually run the code through ethernet before start controlling the car.

5 Final Remarks

*This was a great learning experience for us. We learned a lot not only about the PYNQ board but about FPGAs in general and how they operate.

*There definitely were times where it got really frustrating because since PYNQ is a relatively new FPGA, it was hard to find a lot of documentations as well as examples and solutions to errors that our group would run into. The class could have been good if we used Arduino boards too, but PYNQ had its good sides too.

*Our group had to figure out a lot of stuff on our own (i.e. programming, how PYNQ works) with little resources online (mentioned earlier) but we understand at the same time that this is how real life works at times too, where we have to work with nothing to make something.

*Our group can see how this class can prepare us for our senior design project based off the other remarks we made.