CMPT 433 – Project Write Up

Parmveer Dayal (301282695), Jashanraj Singh Gosain (301435386), Oliver Yalcin Wells (301350814)

We created a remote-controlled wireless rover where you can drive it with an Xbox controller or the keyboard. The rover can be controlled to be driven in four directions and a Logitech camera placed on a servo motor can move in four directions as well. The inspiration behind this project was the Mars rover, where we would be able to drive it using a camera and controller.

The hardware set up of our rover included receiving a four-wheel car from the professor and constructing it ourselves, but had help from the professor with soldering the wires. Once the car was constructed, we securely placed our BeagleBone Green (BBG) on top of the car with the correct wiring. We also placed a servo motor in the front of the car with our camera placed on top of it. The Xbox controller was connected using a USB-A wire. Finally, a power bank was used to give our rover power so it can run wirelessly.

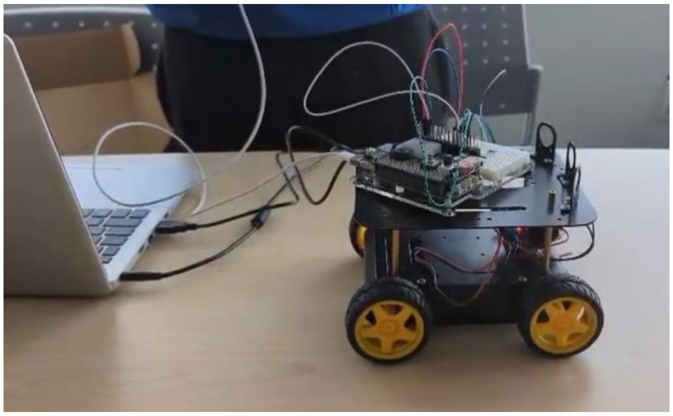

Figure 1: Rover in Progress.

Once we figured out the hardware portion of our project, we moved on to the software part. We used some guides made by past students and our prior CMPT 433 knowledge to learn how to control each hardware item. The most challenging part of the code was controlling the servo motor as the guides were not useful, however, eventually we were able to figure it out by using open PWM channels. After we were able to control each individual hardware piece, we brought the code together and synchronized it using multiple threads, similar to how we approached the assignments throughout the semester.

Overall, we were able to implement everything we wanted but did run into a few issues. The first issue was we were not able to get the rover to run over the SFU Wi-Fi due to certain security measures. We resolved this issue by using mobile hotspot from any one of the group members, and it worked perfectly. However, the one limitation of this was our rover was limited by the connection of the hotspot, so it would not be able to move as far as it would have if it were connected to the SFU Wi-Fi. Although, it was still able to move fairly far away from the phone it was connected to.

Another issue we ran into was our Xbox One controller was not able to connect to the BBG wirelessly. We tried to connect the controller over Bluetooth, however, the connection was weak and would disconnect after a certain amount of time. We would have preferred to use an Xbox 360 controller as those connect better, but none of the group members had permission to borrow it from the school. To solve this problem, we connected the controller with a USB wire so we can get inputs from the controller without it disconnecting. We also implemented driving the rover with the keyboard so it would still have wireless functionality.

Finally, the last issue we encountered in this project was the camera would not work wirelessly. A lot of time was spent trying to fix this issue; however, we were not able to get the camera to work wirelessly. We tried to make the camera work over Wi-Fi and hotspot but were not able to get it to send images to our webpage. The camera would indicate it is on and the code seemed correct as it would work when the camera was connected with a wire to a computer, but just would not work over Wi-Fi. We also tried to send still images over the server so we can manually push the images to our webpage, but the images were too big for this to work. Therefore, we settled for the camera to only work when it is plugged into the laptop. The camera would work well at 10-20 FPS, even with the servo motor moving it in different directions. The camera working over Wi-Fi was the only challenge that remained unsolved, despite a large portion of time being spent on trying to make it work.

Description	Host / Target	Comp	Code	Author(s)	Notes
4-Wheel Car	T	5	C	Jashanraj, Oliver	Able to build and control the car in four directions; fast and responsive
Servo Motor	T	5	C	Jashanraj, Parmveer	Move the motor in four directions; fast and responsive
Xbox Controller	T	4	C	Oliver	Control car with the D-pad and motor with A/B/X/Y buttons; responsive but not wireless
Camera	T	4	C	Parmveer, Jashanraj	Works very well and responsive (10-20 FPS), but only wired (no wireless connection)

Web page	T	5	JS / HTML / CSS	Parmveer	Works well and responsive
Keyboard	T	5	C	Jashanraj	Controls both car and motor in all four directions; responsive and works wirelessly
Wi-Fi Adapter	T	5	C	Oliver	Fast and responsive, no issues
Arduino	T	3	C	Jashanraj	Used it as a placeholder, when trying to get BBG to work, to make sure car was working correctly; abandoned it after
Bluetooth	T	3	C	Oliver	Implemented the code to connect the controller to the system wirelessly, but was inconsistent so we abandoned it
BBG Joystick	T	5	C	Parmveer	Originally added this to control car using a different BBG but abandoned it
BBG LED	T	5	C	Parmveer	Change LED patterns based on inputs but abandoned it as well since

					we were not using the BBG joystick anymore
UDP Server	T	5	C	Jashanraj, Oliver	Takes the inputs correctly and executes the tasks quickly, including stopping the program correctly with no memory leaks

We used an H-bridge in our car to be able to control the car and make it stop and go when we want it to. Without this bridge, our car would just keep running without any breaks the second we gave it power. As already mentioned in this report, the additional hardware beyond the BBG and Zen Cape was the 4-wheel car, Xbox controller, Logitech camera, servo motor, and a keyboard. We did not have to download any additional drivers for these hardware systems. All we had to do was learn how to control them and implement them into our code, while ensuring they work in synchrony.

For the Wi-Fi set up we had an adapter and tried following the guides on the course website but none of them worked for us. So, we used an RTL8188EU driver for the Wi-Fi adapter instead. We had to configure the interface with a username and password before loading the module into the BBG. Also, we had to blacklist any pre-existing modules if they were present, otherwise they would not let our new driver work. Once we completed these steps, we were able to connect our system via Wi-Fi.

Overall, we would consider our project a success. Outside of not being able to get the camera and Xbox controller to work wirelessly, we were able to implement everything we wanted and it worked very well. Despite the issues with the camera and controller, we were still able to use them flawlessly when they were connected with a wire. Had we not wasted so much time trying to get our camera and controller to work wirelessly, we could have implemented a few more features. However, we were still able to accomplish a lot and incorporate a ton of different features as mentioned previously in this report. Our rover was fast and very responsive when it received an input from both the controller and keyboard. It would start up and quit without any issues or bugs. It was also able to travel as far as its connection reached when tethered to the mobile hotspot. We spent a lot of time testing our code to ensure it was robust with no memory leaks. We are proud of our work and the final product, which was received very well during the project demos, earning us many compliments from our classmates.