Your New Companion

Wall-E

Frank Li, Joseph Teh, Annabella Macaluso

Table of Contents

Introduction

WALL-E is an Infrared Red remote controlled car. It is different from all the other remote controlled cars because it is using infrared sensors and remote control instead of a radio control. Though different, WALL-E can just do as much as a regular RC controlled car does, but simpler. Like the movie character WALL-E, this robotic car is simple, interesting and fun! WALL-E will be the ultimate entertainment choice for your child, and it will be the best friend for you child as well.

For hardware, WALL-E consists of a caterpillar chassis, a motor drive control unit, an arduino uno and other circuits components of your choosing. Most importantly, it must have an IR sensor installed to communicate with the IR remote. The center LED will indicate whether the car is on or off right now.

The software side of the WALL-E is all done in arduino C. We have used two external libraries: IRremote and NewTone. IRremote is there to decode the signals from IRremote and translate them into commands for the car. NewTone library is for our buzzer to communicate with IR remote properly.

Purpose and Motivation

During the pandemic, people have had enough social distancing. No place to have fun, nowhere to hangout with your friends … This is where WALL-E comes in. WALL-E is your personal companion to ease your boredom during the pandemic. It is easier to make, and yet super fun to be with. Another motivation for this robotic car is to see if we can use IR remote to control the motors instead of using traditional radio waves. And in the end, it is quite successful. The purpose of WALL-E is to entertain, have fun and make you laugh. When more modules are implemented, WALL-E will be a system of entertainment: it will just be like the arcade of the ancient time where people can play race games, shooting games and so on.

Timeline

Week 1
Exploring different project ideas related to entertainment and gaming activities.

Week 2
Finalizing project idea centered around robotics and began researching electronics components and how to wire and code them.

Week 3
Created a Bill of Materials and purchased the necessary materials. We began prototyping our robot design.

Week 4-5
Waited for shipments to arrive. Began general assembly of robot base and wiring motor and battery components. Planned design layout of electrical components on the base.

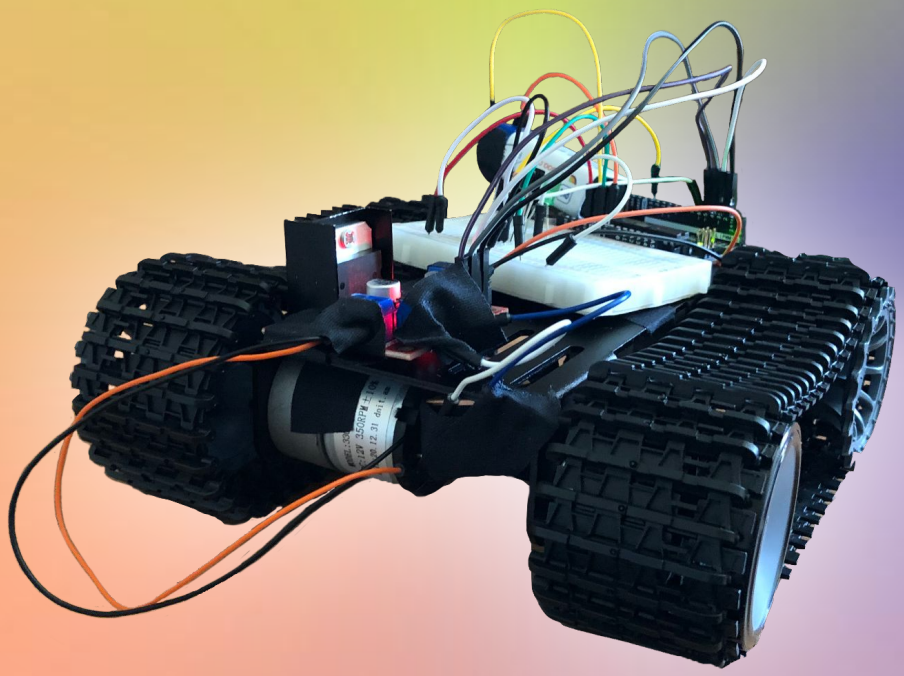
Week 6
Researched libraries to use and how to control motors in arduino. We wired and mapped an IR controller that controls forward, backwards and side to side motion of the robot.

Finished Documentation and Submitted Project!

How-To-Use

In order to use our remote-controlled car, you must have the remote control in hand. First, you have to power on the car with the power button and a green LED light should light up, indicating that the car is turned on. Once that LED is turned on, you can use the remote control to make the car move forwards, backwards, turn left, and turn right with the four directional buttons. The button in the middle makes the car come to a stop. We also have set the button labeled A to trigger a beeping/honking sound as an added feature.

Our Final Product



Teammate Contributions

With two of us living on campus, and one of us working from home we had to divide up the work accordingly. All of us took part in the planning of the project when figuring out what we should even begin to work on. After we decided on our project and we received the parts we would need we began to work.

Frank and Joseph were mainly responsible for assembling the kit. Frank also soldered the wires onto the motor and make sure all the hardwares were connected properly and making the code work on properly.

Annabella on the other hand was responsible for the software components. She coded the motor to drive forward, backward, and turns and all their respective IR commands. Joseph also helped on the software development side by implementing more features like buzzer and halt.

Real-World Applications

This type of idea can be utilized in many different forms. Our model is simple version that could be turned into pretty much anything. For example, if we were to add onto our project, we could add anything we wanted such as a basket to carry belongings, contraptions to create a game, and with a little bit more work, we could implement features such as a sensor that would allow our device to simply follow us. Or this could even be used for transportation much like booster-boards work using a remote control to control the speed forwards or backwards.

Difficulties and Challenges

On the hardware side, soldering the wires onto battery was pretty difficult. In the end, we determined that taping is much safer and better than soldering onto the battery. Another difficulty was to make sure every hardware component is in place while the car is moving. Because of the basic mechanics and sometimes the wires attached to the battery might drop, we have to constantly making sure everything is intact.

For software, a big challenge for us was getting used to the arduino language and learning how to implement various electronic components. We had to learn how motors were controlled and how different electrical signals would create different directions and speeds of motion. In addition, we took time to figure out how to implement libraries to make interfacing with electronics easier.

Conclusion

The current WALL-E is only the first step towards our idea of a new entertainment system. Yet, in its current form, we have already learned a lot about communication systems, embedded system design, software development, and hardware development. It is very exciting to see all of our efforts combined and make something innovative and interesting. There are surely still works to be done for this project, if it is possible, we intend to implement a ultrasonic module so that it can alert the user if it is on a collision course, and then we make a launcher module to make this robot launch something while moving. In the end, the modules are infinite and there surely more to do.

Resources

IRremote.h https://www.arduino.cc/reference/en/libraries/irremote/ Arduino library for wireless communication

NewTone.h https://bitbucket.org/teckel12/arduino-new-tone/downloads/ Arduino library for buzzer Tone.

Project Source Code: https://github.com/AnnabellaMacaluso/IEEESpringProject2021