Department of Computer Science
Technical University of Cluj-Napoca

DMP

Laboratory activity

Name: Bălaș Andrei-Ioan/Baiaș Ioan
Grupa: 30434
Email: balas.andrei32@gmail.com/ ionut10.baias@yahoo.com

Contents

1 Introduction

1.1 Porpuse

The purpose of this project is to implement a **hand-controlled robot** using Arduino technology. This idea of human-controlled robots is a well-known idea in this field of study and has lots of very advanced projects that respond to very complex tasks. We will only focus on the simple task of controlling the movement of the robot by hand gesture.

1.2 Motivation

The motivation behind it is to explore human-machine interaction through intuitive hand gestures, offering a unique and engaging way to control a robotic vehicle. Our educational motivation is to gain more knowledge about the Arduino microprocessor.

2 Bibliographic research

As was said above this area of study has lots of projects some of them being very complex but some of them being more for the educational part and more easy to implement. So we tried to find an already-implemented solution for this project and to get our ideas for there.

Our main inspiration can be found here: Click to go to the youtube video

The attached implementation had to be adapted because we didn't have to our availability all the hardware components, we had to analyze this solution and extract the main idea so we could adapt it to our tools.

Presented components	Our components
Arduino Nano(2x)	Arduino Uno (2x)
RF Nano (optional) (2x)	Not used
NRF24L01+ RF Module (2x)	NRF24L01 (2x)
L298N Motor Driver	The same
ADXL335 Module	MPU-9250/6500
TT Gear Motor (4x)	DC motor (2x)
Rubber Wheels (4x)	Rubber Wheels (2x) and 360 assisting wheel(1x)
Male and Female Header Pin	Male Pin and Female wires
18650 li-ion battery (2x)	9v battery (2x)
18650 battery holder	Not used
Screw Terminal	Not used
Code implementation	Different implementation
5mm Acrylic Sheet	A piece of wood
Not specified	Sock

3 Proposed solution and implementation

We had to divide the solution into 2 parts: the receiver part, and the transmitter part.

3.1 Transmitter part

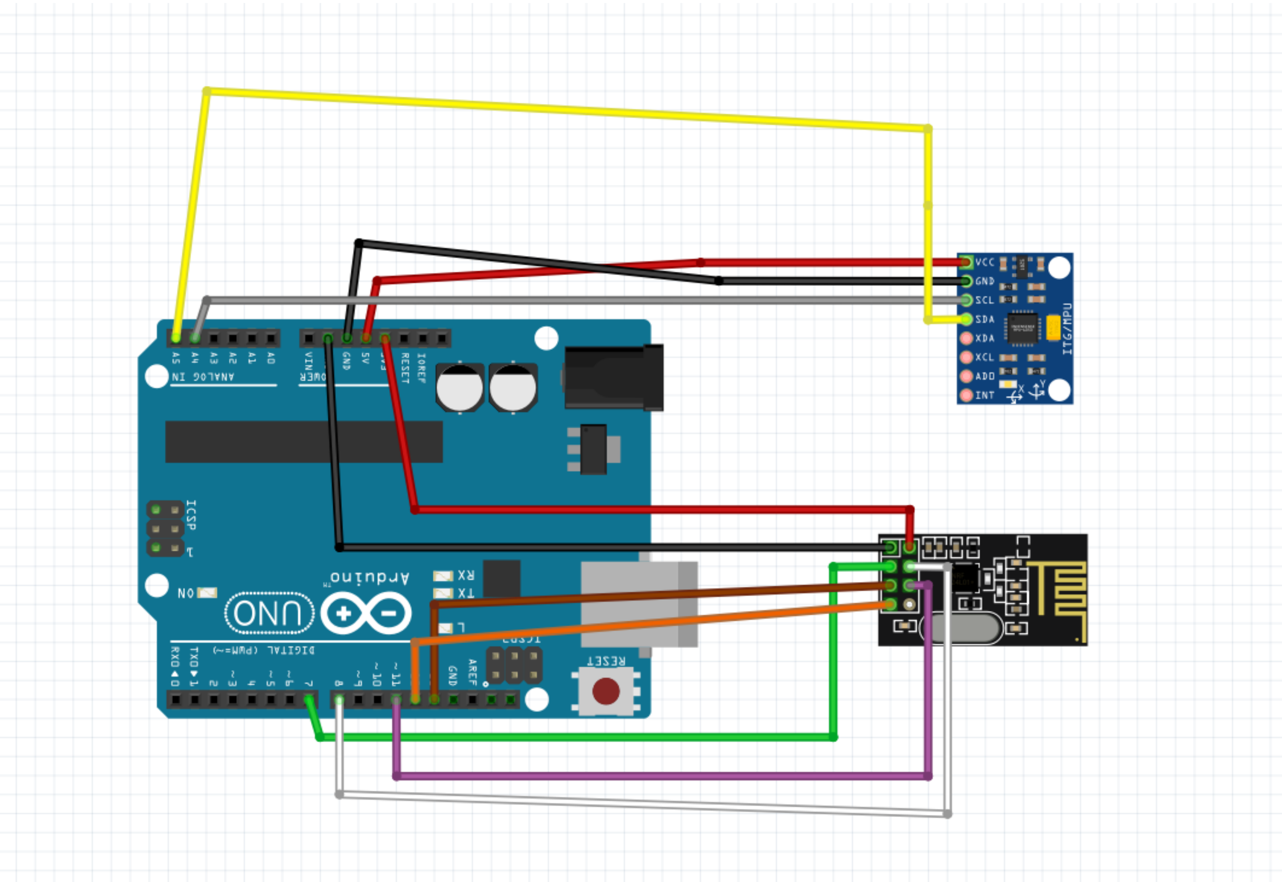
The transmitter in real life represents the glove that will control the robot. The transmitter is composed of the following components: Arduino Uno (1x), NRF24L01 (1x), MPU-9250/6500, male pin, and sock.

The program reads motion data from the MPU6050 sensor, specifically the acceleration along the X and Y axes, as well as the gyroscope data.

The ESP8266 module is configured to establish a wireless communication channel. The program defines a data structure, 'Package,' which holds information about the desired direction of the robot. The wireless communication is achieved through the RF24 library, where the Arduino acts as the transmitter.

Within the main loop, the code continuously reads motion data from the MPU6050 sensor. The acceleration values along the X and Y axes are used to determine the orientation of the sensor. Based on these readings, the 'Package' structure is updated with a direction code (1 for forward, 2 for backward, 3 for right, 4 for left, and 0 for stay). A threshold was chosen for each coordinate to restrain the sensitive part. If the X coordinate is smaller than -9000 it will represent the backward movement, if it is bigger than 9000 it will be forward. If the Y coordinate is bigger than 9000 it is right, smaller than -9000 left.

The program then transmits this 'Package' data wirelessly using the NRF24L01 module. The serial monitor is utilized for debugging and displays the selected direction. The code effectively transforms hand gestures into directional commands for the robot, offering a simple yet effective means of human-machine interaction.



The debugging part was done using trial and error to choose the best threshold.

3.2 Receiver part

The receiver part in real life represents the moving robot. It is composed of the following components: Arduino Uno (1x), NRF24L01 (1x), DC motor (2x), Rubber Wheels (2x) and 360 assisting wheel(1x), Female wires, 9v battery (2x), and a piece of wood.

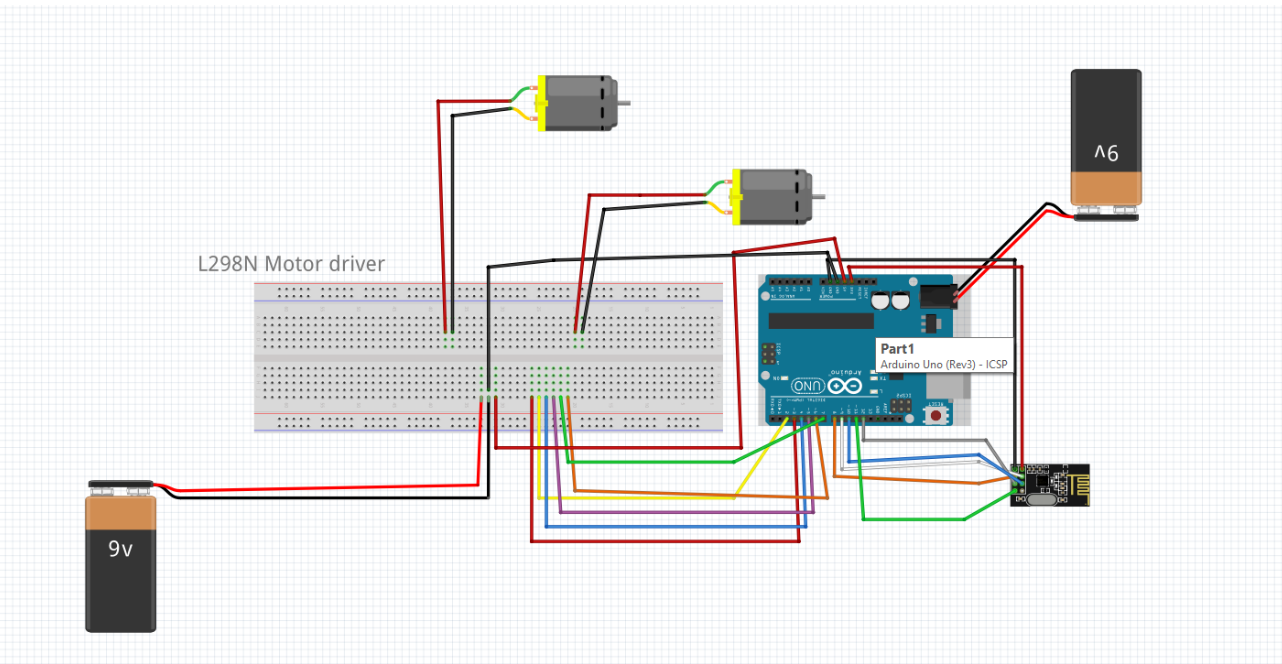
The receiver code complements the previous one, forming a two-part system for a hand-controlled robot. While the first code reads motion data from an MPU6050 sensor and transmits directional commands wirelessly, this second code receives those commands using an NRF24L01 module and controls the movement of the robot accordingly.

The program initializes an NRF24L01 module for wireless communication, defines a 'Package' structure to hold the received directional data, and sets up the motor driver connections for controlling the robot's movement. The motor driver is configured with two H-bridges, each controlling a pair of motors for the left and right sides of the robot.

In the main loop, the program checks for available data from the NRF24L01 module. If data is available, it reads the 'Package' structure, extracts the direction code, and executes the corresponding motor control logic.

Depending on the received direction code (1 for forward, 2 for backward, 3 for right, 4 for left, and 0 for stay), the program adjusts the motor control signals to drive the robot in the desired direction. The Serial monitor is used for debugging and displays the received direction.

Together with the first code, this system translates hand gestures into directional commands, enabling the robot to respond to human input wirelessly. This two-part implementation demonstrates a basic but functional hand-controlled robot using Arduino and NRF24L01 wireless communication.



The breadboard represents the motor driver because we didn't find the actual driver in the program.

4 Testing and validation

4.1 Transmitter part problems

The main problem that was taking us very long to solve was to find the right library to work with the MPU-9250/6500. We tried many libraries for example MPU9250 library, but all these libraries were useless in reading data from the sensor. Until we tried the with the MPU6500 library. Another problem that we encountered was to find the best threshold so the robot doesn't move at each slight move of the hand. This was solved by trial and error very fast by seeing how the robot responds to different thresholds.

4.2 Receiver part problems

For the receiver part the problems were created more by the hardware part. We first found that the NRF24L01 module has to be connected to 3V instead of 5V to make it work. The next problem was to find a solid frame to put the components on, so we found a not-that-solid frame: a piece of cardboard from a shoe box. After giving the robot a few tries, and playing around with it the cardboard started to fail. We changed the frame to a more solid and better one a cut from a parquet. Firstly the robot was composed of 2 rubber wheels not looking that good and after our teacher gave us the assisting 360 wheel the project got another look.

5 Conclusion

In conclusion, this documentation outlines the development process of a hand-controlled robot using Arduino technology. The project's primary goal was to create a simple system for human-machine interaction through simple hand gestures. Our motivation behind this endeavor was to explore the possibilities of human-machine interaction and gain a deeper understanding of the Arduino microprocessor.

Throughout the development process, we encountered challenges such as finding the right sensor library and determining the optimal threshold for gesture recognition. Hardware issues, such as the correct voltage supply for the NRF24L01 module, also required attention.

There are several promising avenues for future improvements to enhance the capabilities and functionality of this hand-controlled robot project. Firstly, the addition of more advanced sensors, such as ultrasonic sensors or cameras, could enable the robot to detect obstacles and navigate its environment autonomously. This would elevate its practicality and make it suitable for real-world applications. Moreover, incorporating machine learning algorithms for gesture recognition could enhance the precision and expand the range of recognizable gestures, making the interaction even more intuitive. Additionally, integrating a smartphone app or a dedicated remote control interface would provide users with more options for controlling and monitoring the robot remotely. Lastly, enhancing the robot's mobility by adding more wheels or even tracks could improve its ability to traverse different terrains. These future improvements would not only make the project more sophisticated but also open up possibilities for a broader range of applications, from home automation to surveillance and beyond.

In summary, this project served as an educational and hands-on exploration of Arduino technology, wireless communication, and gesture-based human-machine interaction. While the final implementation may have some limitations and areas for improvement, it successfully demonstrated the feasibility of controlling a robot through hand gestures. This project can serve as a starting point for further developments in the field of robotics and human-machine interfaces, encouraging further exploration and experimentation in this exciting domain.