Controlling a Robotic Car Using Mobile Phone

Smart Chilaka, Nataya Taylor
Dr. Xiaohua Xu, Professor

CS 4491: Deep Learning Cybersecurity
Department of Computer Science
College of Computing and Software Engineering
Kennesaw State University

Abstract

This project will focus mainly on building a robotic car with a NodeMCU. The robotic car connects directly to your mobile phone using Wi-Fi, similar to Bluetooth. You can use your mobile phone to control the robotic car by moving it forward, backward, to turn left, and to turn right. Hardware components for the robotic car include Arduino 2WD Smart Robotics Car Chassis, Arduino L298N Dual H Bridge Stepper Motor Driver Board, NodeMCU ESP8266 CP2102 WiFi Development Board, a 9V Battery, 9V Battery Connector, Male to Female Jumper Wires, Male to Male Jumper Wires.

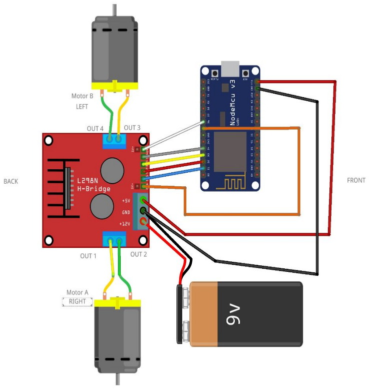
Introduction

The Internet of Things describes the network of physical objects that are embedded with sensors, software, and other technologies to connect and exchange data with other devices and systems over the internet. The Internet of Things has evolved due to the combination of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Our robotic car is interdisciplinary research at the interface of Computer Science and Engineering.

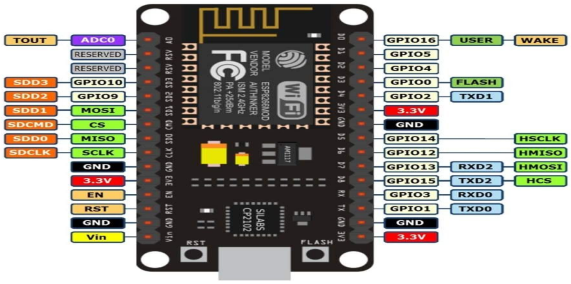
In Computer Science, an implementation is a realization of a technical specification or algorithm as a program, software component, or another computer system through computer programming. This project will hopefully lead to more thoughts and conversations about the Internet of Things assisting in the integration of communications, control, and information processing across various transportation systems. Application of the Internet of Things extends to all aspects of transportation systems. Dynamic interaction between these components of a transport system enables many different means, but the current most important is vehicle control.

System architecture

Below are a few pictures of what the system architecture looks like. This first image is the NodeMCU being connected to the Arduino L298N Dual H Bridge and 9V battery with the help of male to male wires. Here you can see more clearly that we used the 9V Battery Connector to test it out and try some trials.



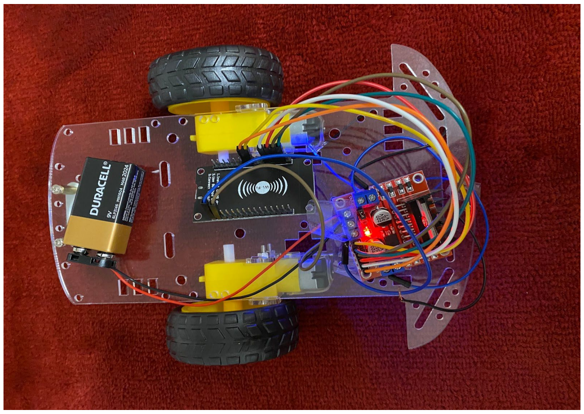
Robotic car architecture



NodeMCU ESP82

Algorithm design

This is the design of our robotic car.



Robotic car design

Algorithm analysis

Our code is written with arduino IDE. After downloading and installing the arduino IDE. We also configured the arduino IDE by adding the NodeMCU ESP8266 board manager.

```
#include <ESP8266WiFi.h>

#include <ESP8266WebServer.h>

#include "webpage.h"


const char* ssid = "directwifirobot";

const char* password = "11111111";

String  data ="";


#define rightMotorForward D1   //OUT1

#define rightMotorBackward D2  //OUT2

#define leftMotorForward D3  //OUT3

#define leftMotorBackward D4 //OUT4

#define  rightMotorSpeed  D5      //Motor-A Enabled

#define  leftMotorSpeed  D6      //Motor-B Enabled


const int SPEED = 380;     //380

const int TURNSPEED = 340; //340
```

```
//Static IP address configuration

IPAddress staticIP(192, 168, 8, 1); //ESP static ip

IPAddress subnet(255, 255, 255, 0); //Subnet mask


ESP8266WebServer server(80);


void setup() {

  // put your setup code here, to run once:

  pinMode(D1, OUTPUT);

  pinMode(D2, OUTPUT);

  pinMode(D3, OUTPUT);

  pinMode(D4, OUTPUT);

  pinMode(leftMotorSpeed, OUTPUT);

  pinMode(rightMotorSpeed, OUTPUT);


  Serial.begin(115200);

  delay(10);


  WiFi.mode(WIFI_AP);

  WiFi.softAPConfig(staticIP, NULL, subnet);

  WiFi.softAP(ssid, password);

  server.begin();


  IPAddress ServerIP = WiFi.softAPIP(); // Obtain the IP of the Server

  Serial.print("\nServer IP is: "); // Print the IP to the monitor window

  Serial.println(ServerIP);


  server.on("/", handleRoot); //Associate handler function to web requests

  server.on("/right", handleRight); //Associate handler function to web requests

  server.on("/left", handleLeft);

  server.on("/stop", handleStop);

  server.on("/forward", handleForward);

  server.on("/backward", handleBackward);

  server.begin(); //Start web server

  Serial.println("HTTP server started");

}


void loop() {

  server.handleClient();

}


void handleRoot() {

  server.send(200, "text/html", webPage); //webPage is defined in htmlPage.h file
```

```
}


void handleRight() {
  Serial.println("got right request");
    server.send(200,  "text/html",  webPage); //webPage is defined in htmlPage.h file
  analogWrite(leftMotorSpeed, TURNSPEED);
  analogWrite(rightMotorSpeed, TURNSPEED);
  digitalWrite(leftMotorBackward,LOW);
  digitalWrite(rightMotorBackward,HIGH);
  digitalWrite(leftMotorForward,HIGH);
  digitalWrite(rightMotorForward,LOW);
}


void handleLeft() {
  Serial.println("got left request");
    server.send(200,  "text/html",  webPage); //webPage is defined in htmlPage.h file
  analogWrite(leftMotorSpeed, TURNSPEED);
  analogWrite(rightMotorSpeed, TURNSPEED);
  digitalWrite(leftMotorBackward,HIGH);
  digitalWrite(rightMotorBackward,LOW);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(rightMotorForward,HIGH);
}

void handleStop() {
  Serial.println("got stop request");
    server.send(200,  "text/html",  webPage); //webPage is defined in htmlPage.h file
  analogWrite(leftMotorSpeed,0);
  analogWrite(rightMotorSpeed,0);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(leftMotorBackward,LOW);
  digitalWrite(rightMotorForward,LOW);
  digitalWrite(rightMotorBackward,LOW);
}


void handleForward() {
  Serial.println("got forward request");
    server.send(200,  "text/html",  webPage); //webPage is defined in htmlPage.h file
  analogWrite(leftMotorSpeed, SPEED);
  analogWrite(rightMotorSpeed, SPEED);
  digitalWrite(leftMotorBackward,LOW);
  digitalWrite(rightMotorBackward,LOW);
  digitalWrite(leftMotorForward,HIGH);
  digitalWrite(rightMotorForward,HIGH);
}


void handleBackward() {
  Serial.println("got backward request");
    server.send(200,  "text/html",  webPage); //webPage is defined in htmlPage.h file
  analogWrite(leftMotorSpeed, SPEED); //500 - 300
  analogWrite(rightMotorSpeed, SPEED);
  digitalWrite(leftMotorForward,LOW);
  digitalWrite(rightMotorForward,LOW);
  digitalWrite(leftMotorBackward,HIGH);
  digitalWrite(rightMotorBackward,HIGH);
}

#htmlPage.h
const char webPage[] PROGMEM = R"=====(
  <!DOCTYPE html><html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" href="data:,">

<style>
html {
  font-family: Helvetica;
  display: inline-block;
  margin: 0px auto;
  text-align: center;
}
.button {
  background-color: #008CBA;
  border: none; color: white;
  padding: 16px 40px;
  text-decoration: none;
  font-size: 30px;
  margin: 2px;
  cursor: pointer;
  }
  .button2 {
  background-color: #77878A;
  }
  </style>
</head>


<body><h1>Robot Wifi Direct</h1>
<div><a href="/forward"><button class="button">FORWARD</button></a></div>
<div style="margin-top: 20px"><a href="/stop"><button class="button">STOP</button></a></div>
<div style="float: left"><a href="/left"><button class="button">LEFT</button></a></div>
<div style="float: right"><a href="/right"><button class="button">RIGHT</button></a></div>
<div style="clear: both; margin-top: 100px"><a href="/backward"><button class="button">BACKWARD</button></a></div>
</body></html>
)=====";
```

Results

After extensive research and hard work we were able to get our robotic car to move forward and backward and also move right and left but not as smooth as it moves forward and backward. We will continue working on this project in the future because we hope to add a camera to the robot to enable it see it surrounding when it moves.

Reference

Anon.Retrieved September 1, 2020 from https://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx

TechTarget Contributors. 2020. What is IoT (Internet of Things) and How Does it Work? (February 2020). Retrieved September 1, 2020 from https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT

Alex Wood. 2015. The internet of things is revolutionising our lives, but standards are a must. (March 2015). Retrieved September 1, 2020 from https://www.theguardian.com/media-network/2015/mar/31/the-internet-of-things-is-revolutionising-our-lives-but-standards-are-a-must

The Arduino Team. Software. Retrieved September 1, 2020 from https://www.arduino.cc/en/software