EE 40 Final Project Report

Charles Lee, Xinhe Ren

1. Feature Descriptions
2. *H-Bridge*

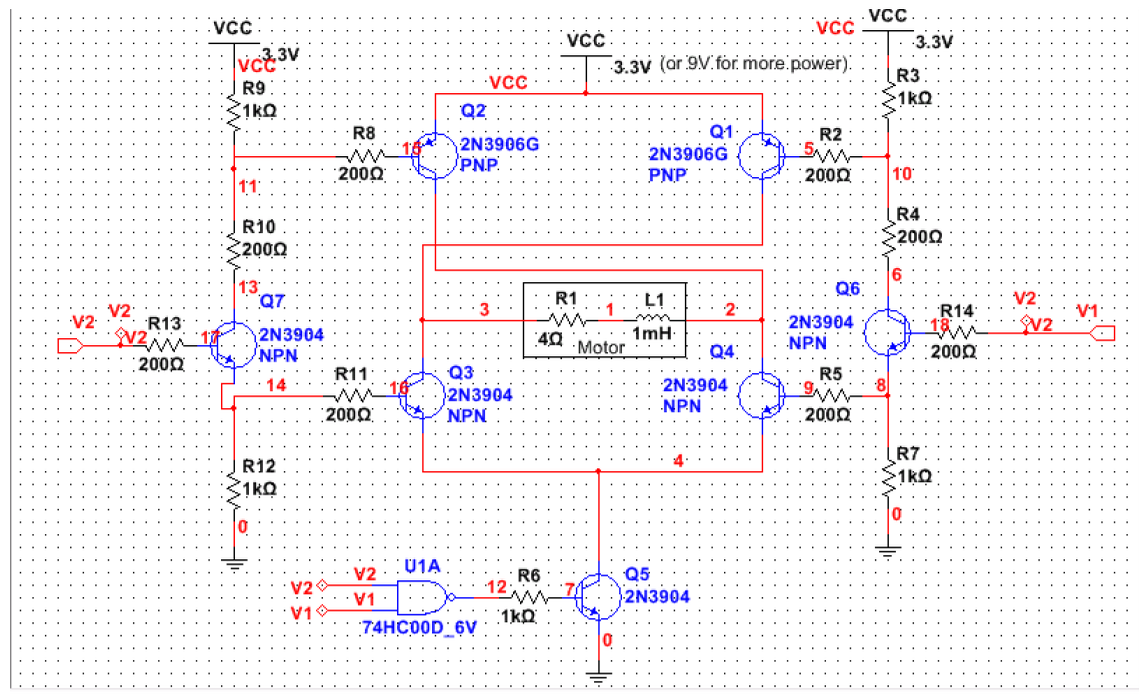
Two H-bridges (one for each wheel) control the steering of the robot car. The H-bridge consists of 2 PNPs and 4 NPSs. The input Vcc is 9V, and is used to power the motors. The two control signal inputs are from MSP430, and are at 3.3V at HIGH, 0V at LOW.

1. *Myo Armband + Bluetooth*

A Myo armband uses user’s EMG data to control the motion of the car. Five different hand gestures are mapped to different driving commands (forward, left turn, emergency stop, etc). A HC-05 Bluetooth module is used with MSP430 to receive wireless commands from laptop, which reads inputs from Myo.

1. Circuit Schematics

H-Bridge (courtesy of Michael):



1. MSP430 Code

// Constants for H-bridge inputs.

const int LEFT\_L = P1\_4;

const int LEFT\_R = P1\_5;

const int RIGHT\_L = P2\_5;

const int RIGHT\_R = P2\_4;

void setup()

// Initial pin setup.

{

pinMode(LEFT\_L, OUTPUT);

pinMode(LEFT\_R, OUTPUT);

pinMode(RIGHT\_L, OUTPUT);

pinMode(RIGHT\_R, OUTPUT);

pinMode(RED\_LED, OUTPUT);

pinMode(GREEN\_LED, OUTPUT);

Serial.begin(9600);

}

void loop()

{

// Digitally check for shorting Vcc to Gnd.

if (shortCircuit())

stopAll;

// Reading inputs from Bluetooth serial.

int buf\_size = Serial.available();

if (buf\_size > 0) {

char input = Serial.read();

if (input == '0') {

stopAll();

digitalWrite(RED\_LED, HIGH);

digitalWrite(GREEN\_LED, LOW);

// Drives car forward.

} else if (input == 'U') {

leftForward();

rightForward();

digitalWrite(RED\_LED, LOW);

digitalWrite(GREEN\_LED, HIGH);

// Drives car backward.

} else if (input == 'D') {

leftBackward();

rightBackward();

digitalWrite(RED\_LED, LOW);

digitalWrite(GREEN\_LED, HIGH);

// Turns car counterclockwise.

} else if (input == 'L') {

leftBackward();

rightForward();

digitalWrite(RED\_LED, LOW);

digitalWrite(GREEN\_LED, HIGH);

// Turns car clockwise.

} else if (input == 'R') {

leftForward();

rightBackward();

digitalWrite(RED\_LED, LOW);

digitalWrite(GREEN\_LED, HIGH);

}

}

}

// Turns left wheel forward.

void leftForward() {

digitalWrite(LEFT\_L, HIGH);

digitalWrite(LEFT\_R, LOW);

}

// Turns left wheel backward.

void leftBackward() {

digitalWrite(LEFT\_L, LOW);

digitalWrite(LEFT\_R, HIGH);

}

// Turns right wheel forward.

void rightForward() {

digitalWrite(RIGHT\_L, HIGH);

digitalWrite(RIGHT\_R, LOW);

}

// Turns right wheel backward.

void rightBackward() {

digitalWrite(RIGHT\_L, LOW);

digitalWrite(RIGHT\_R, HIGH);

}

// Returns TRUE if both inputs for a H-bridge are HIGH (shorting Vcc to Gnd).

boolean shortCircuit() {

return LEFT\_L == HIGH && LEFT\_R == HIGH ||

RIGHT\_L == HIGH && RIGHT\_R == HIGH;

}

// Kill all motor motions.

void stopAll() {

digitalWrite(LEFT\_L, LOW);

digitalWrite(LEFT\_R, LOW);

digitalWrite(RIGHT\_L, LOW);

digitalWrite(RIGHT\_R, LOW);

}

1. Picture

