

Educational Sensing Car

“Driving Growth in Young Minds”

Team 6

Revision	Description
1	Updated section 6 with N-connection tests, software checks and cable connector checks
2	Figures were added to section 4, including H-Bridge schematics and descriptions of their functioning.
3	Added movement block to section 3.
4	Added definitions to section 2: N-FET Q21, R/L_FORWARD

Team Members
Duy Dang
Christian Fischer
Rajwal Gautam
Sam Messick

Originator: Sam Messick

Checked: 2-25-2018 Released: 2-25-2016

Filename: Project_04.doc

Title: **Educational Sensing Car
Team 6**

This document contains information that is **PRIVILEGED** and **CONFIDENTIAL**; you are hereby notified that any dissemination of this information is strictly prohibited.

Date:
02/25/2018

Document Number:
02-05-2016-P4

Rev:
A

Sheet:
1 of 11

Table of Contents

1. Scope	3
2. Abbreviations.....	3
3. Overview.....	3
3.1. Power System Block.....	3
3.2. Control Board Block.....	4
3.3. User Interface Block	4
3.4. Movement Block.....	4
3.5. Software Block	4
4. Hardware	5
4.1. LCD/LED Push Button Interface	5
4.2. Power System.....	7
4.3. Microcontroller.....	8
4.4. Motor Control Board.....	10
6. Test Process	11
6.1. Voltmeter	11
6.2. LCD Functions	11

Figures and Tables

Figure 3.1 Sensing Car Design Overview	3
Figure 3.2 Power System	3
Figure 3.3 Control Board	4
Figure 3.4 Input Blocks.....	4
Figure 3.5 Movement	4
Figure 4.1 Push Buttons and LCD Screen....	5
Figure 4.2 LCD Screen and Backlite....	5
Figure 4.3 Push Buttons and Reactive LEDs ...	6
Figure 4.4 Buck-Boost Converter Circuit....	7
Figure 4.5 Buck-Boost Converter ...	7
Figure 4.6 MSP430 Microcontroller Board....	8
Figure 4.7 MSP430 Microcontroller ...	8
Figure 4.8 Microcontroller Pin Configuration ...	9
Figure 4.9 Motor Control PCB ...	10
Figure 4.10 H-Bridge for Right Motor ...	10

1. Scope

This document describes the electronics of the Educational Sensing Car. The car will help kids learn their shapes, as well as interact with roads that they draw or create. Also, children can learn from the shapes that they create, making it an educational tool in many environments. This document covers the general process of developing the system created, as well as the methodology of the design. A basic overview of the design, in addition to implementations of the hardware and software, are described as well.

2. Abbreviations

AA	Predefined measurement of specific battery type – “50.5mm x 14.5mm”
ADC	Analog-to-Digital Converter
DC	Direct Current
FRAM	Ferroelectric Random Access Memory
J10	Connection to the anode and cathode of the battery pack
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MSP	Mixed Signal Processing
MSP430	Model of specific Mixed Signal Processing board by Texas Instruments.
N-FET Q21	A n-type field emitting transistor that drives the car's motors forward
PCB	Printed Circuit Board
RAM	Random Access Memory
R/L_FORWARD	A digital input/output pin on the MSP430 powering the motors' forward drive
SEPIC	Single-Ended Primary-Inductor Converter
Wi-Fi	Trademarked term referring to a specific type of network connection

3. Overview

The major components of the car are shown below. These components allow the car to function efficiently and effectively. They are:



Figure 3.1 Sensing Car Design Overview

3.1. Power System Block

The car requires a distribution method of power to the rest of the system to ensure the car function correctly. The power system block includes the battery, the switch and the converter. The car uses 6V that comes from four 1.5 V AA batteries in series. There is a switch component that turn the system on and off. The SEPIC converter is used to regulate the voltage by its ability to change the output voltage to be different from the input voltage.

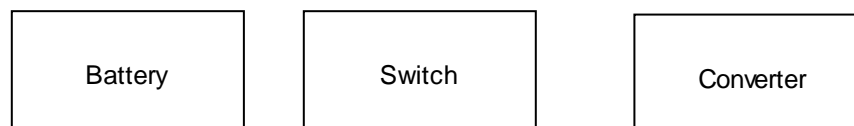


Figure 3.2 Power System

3.2. Control Board Block

The control board is used to control the operation of the car. The control board includes the power/ LCD board, H-Bridge board, Wi-Fi board, and the emitter/detector board.

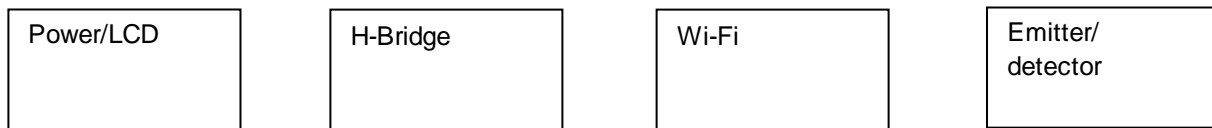


Figure 3.1 Control Board

3.3. User Interface Block

The user interface allows interaction between the car and the users. The user interface block includes LCD screen, LED(s) and the two push buttons. The LCD screen displays information of the car's operation status. The LEDs are used to show output signals from the car. The two push buttons are used to switch between the different information that is displayed on the LCD screen and are used as inputs.

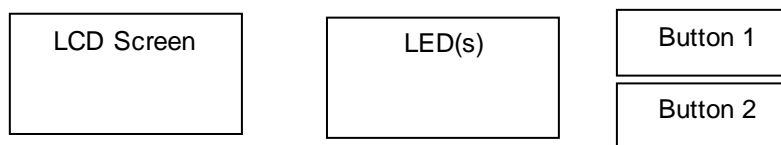


Figure 3.4 User Interface

3.4. Movement Block

The motor block is control by the FRAM board. It has two electric motors connected with wheels and a caster wheel in the back of the car. The FRAM board sends out instructions that are executed by the motors. The motors allow the car to go forward and make different shapes such as circle, figure 8 and triangle. The caster wheel is attached to the back of the car to keep the car level and easy to navigate.

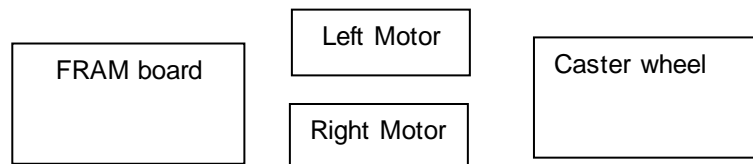


Figure 3.5 Movement

3.5. Software Block

The system is programmed with the C language. The code is stored in the control board which controls how the devices act.

4. Hardware

This section describes the peripheral components and microcontroller that make up the autonomous car.

4.1. LCD/LED Push Button Interface

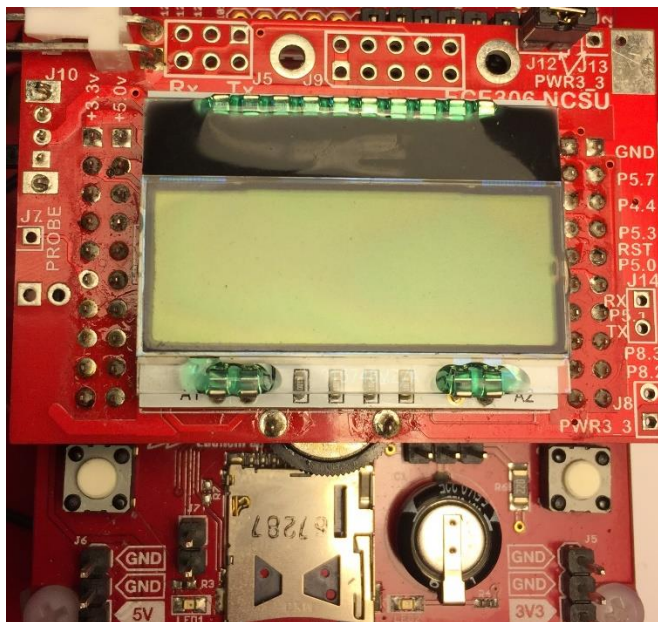


Figure 4.1 Pushbuttons and LCD Screen

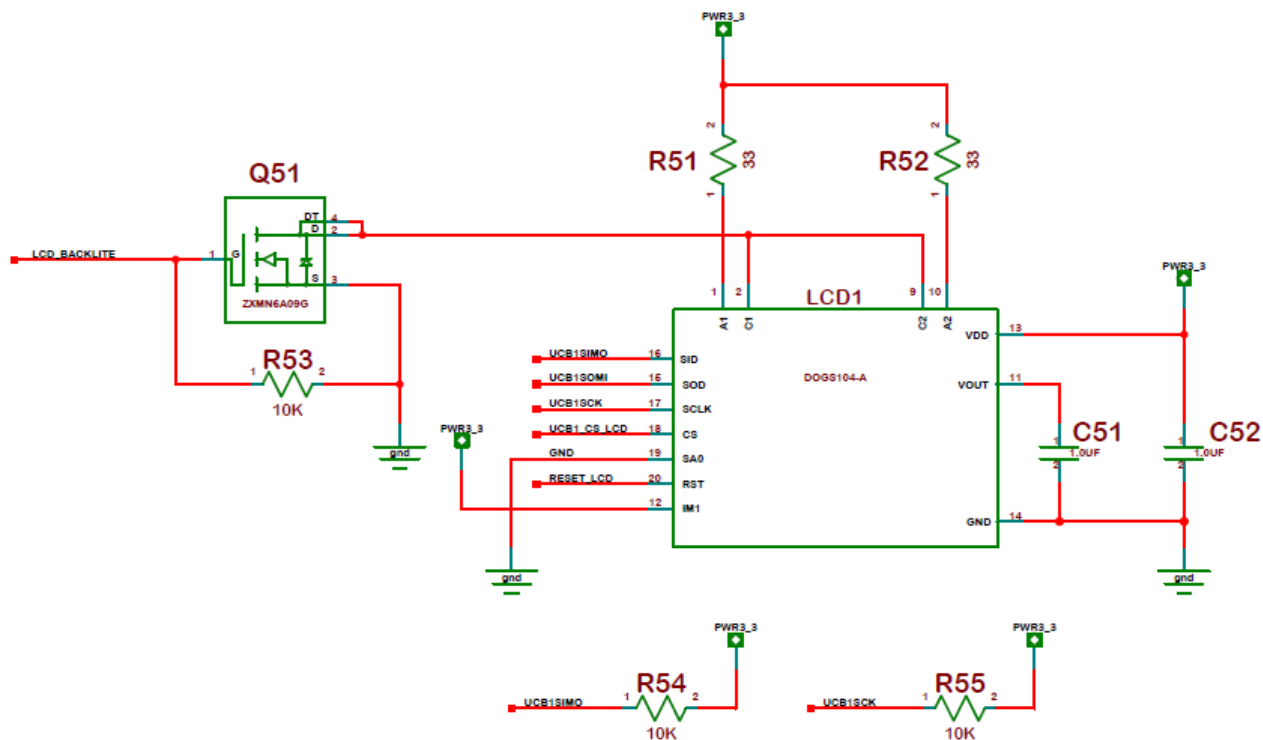


Figure 4.2 LCD Screen and Backlite

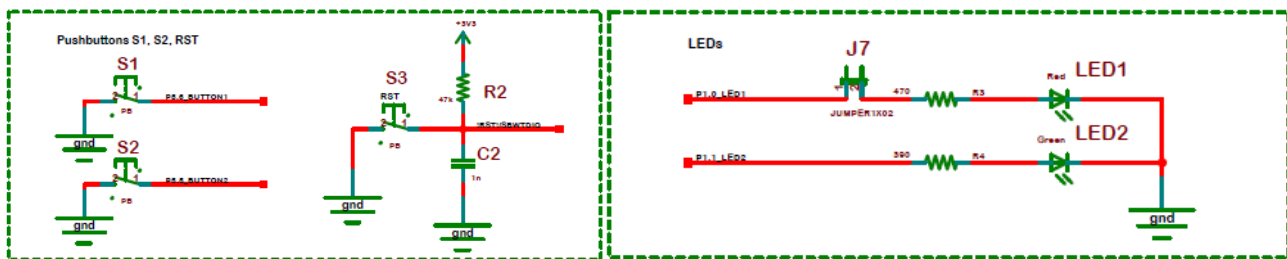


Figure 4.3 Push Buttons and Reactive LEDs

Users can modify the car's behavior through a responsive LCD screen and two LEDs. The screen switches between two messages to inform the user that a button has been pressed. To understand when input is received, the two primary switches are connected to pull-up resistors and “de-bounced” programmatically to respond accurately to each press.

In addition, the LEDs alternate on/off to inform the user when the car is “thinking,” or that the car’s operating system is running in its uninterrupted routine. The two LEDs are connected to resistors to reduce their brightness to a tolerable scale and extend playtime.

4.2. Power System

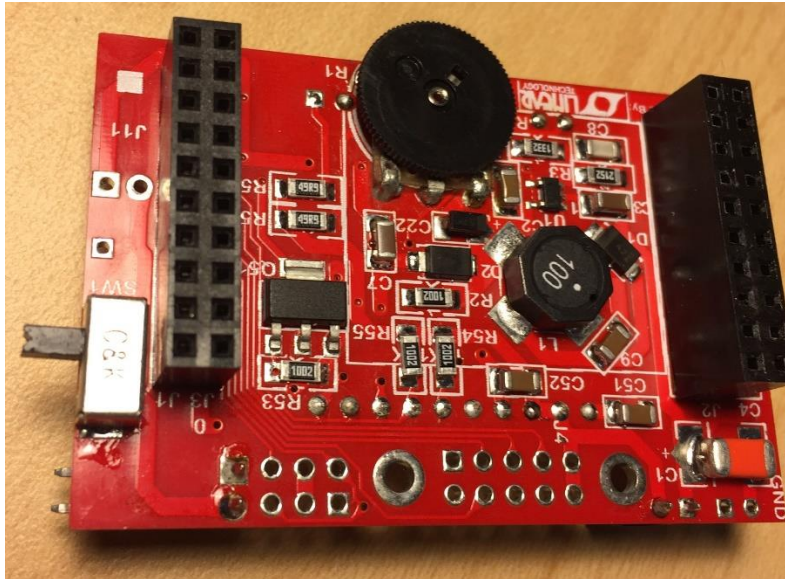


Figure 4.4 Buck-Boost Converter Circuit

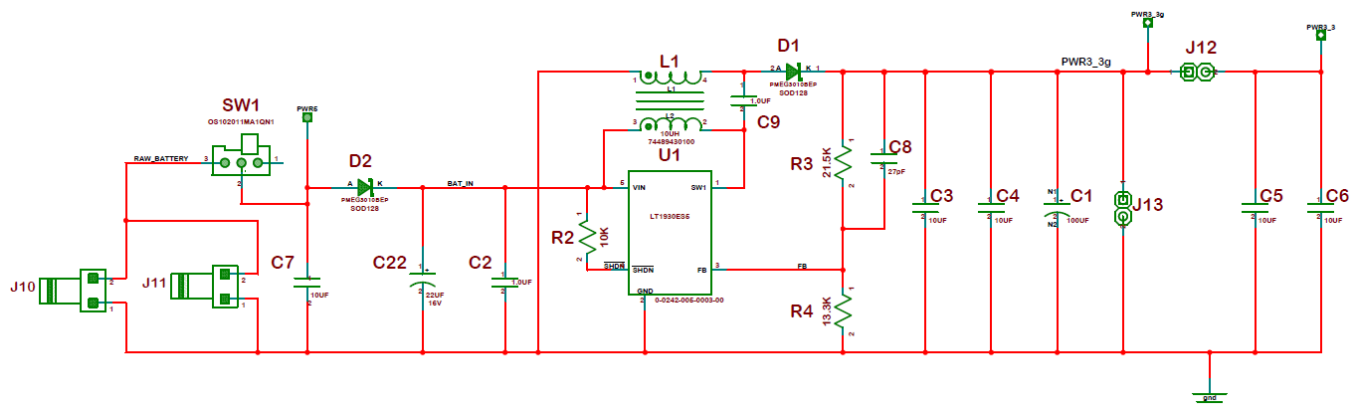


Figure 4.5 Buck-Boost Converter (DC Batteries to Car Microcontroller)

Power is received by 4 AA batteries connected to J10 . Since power can fluctuate between 2.8-6 Volts during regular use, a buck-boost converter adjusts the input voltage to 5 Volts. The MSP430 microcontroller then distributes the received power to the peripherals according to its processor, the user's input and the car's environment.

4.3. Microcontroller

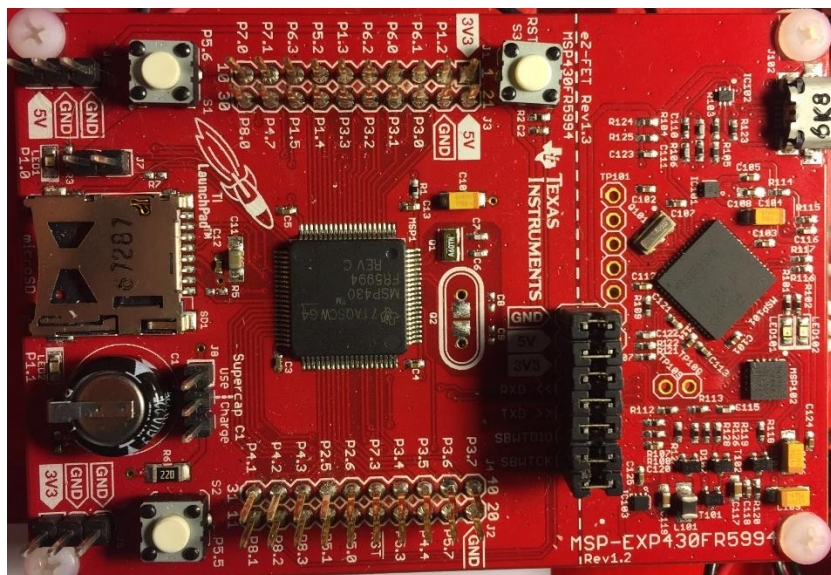


Figure 4.6 MSP430 Microcontroller Board

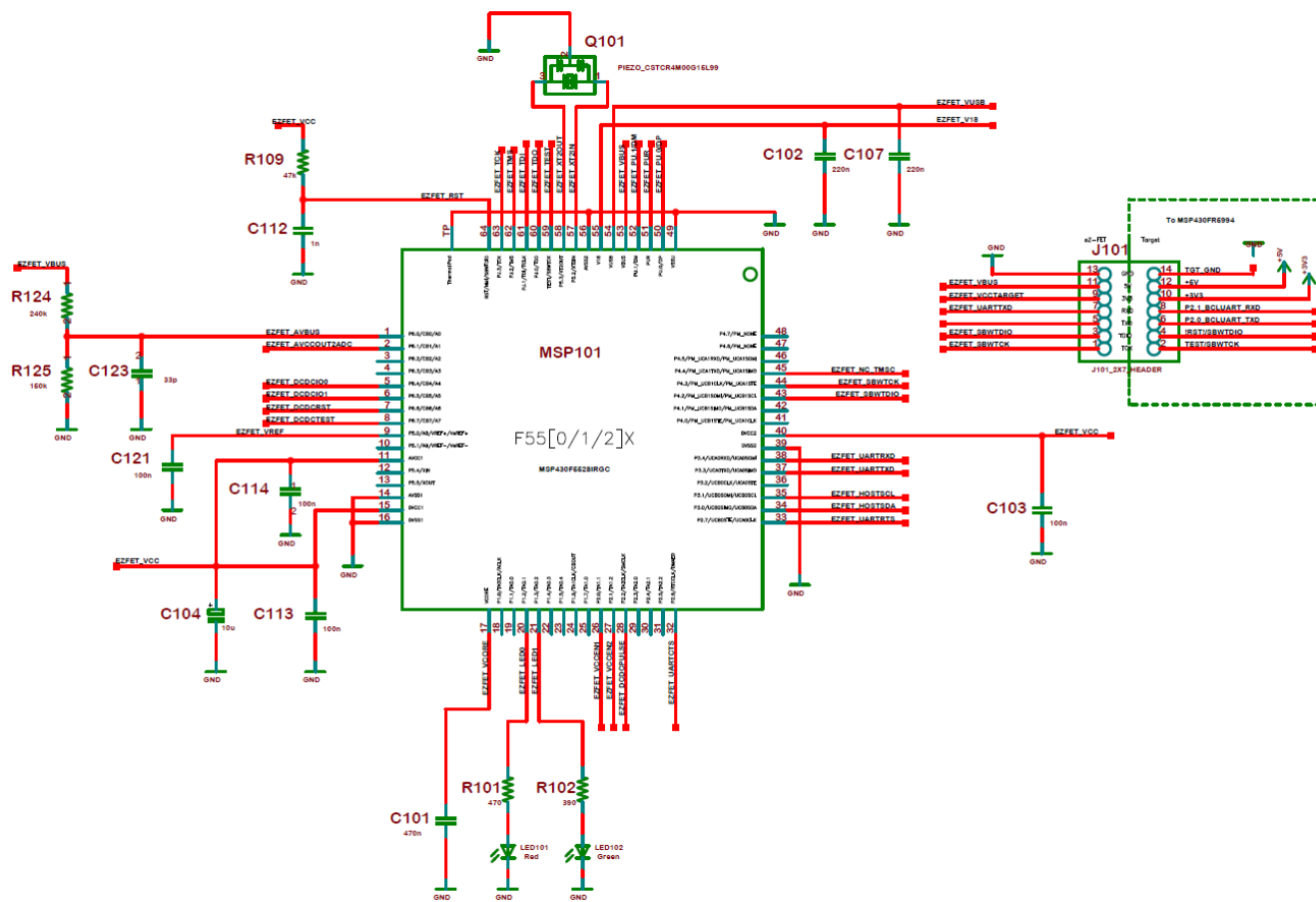


Figure 4.7 MSP430 Microcontroller

The Educational Sensing Car's peripherals respond to stimuli using a programmed MSP430 microcontroller. For example, the microcontroller receives a low signal from a button and displays a new message to the LCD screen in response. The programmable pins connecting the listed peripherals to the microcontroller's processor are listed below as proof of concept.

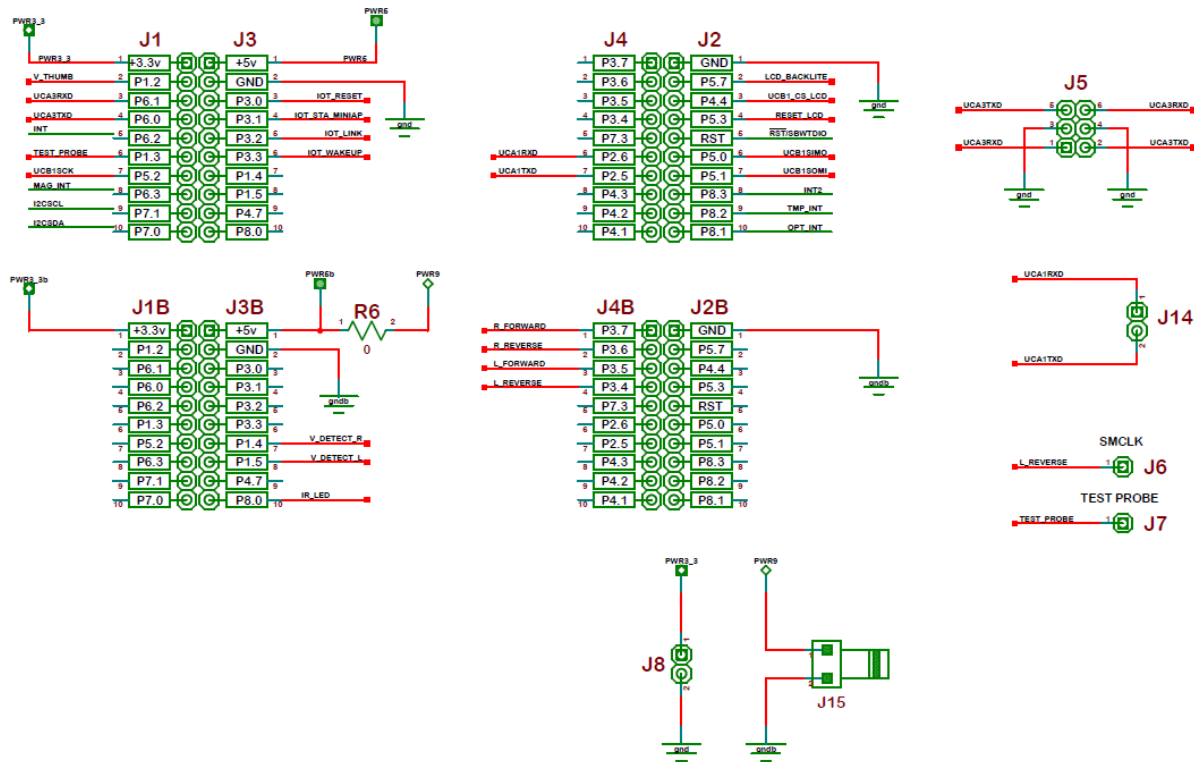


Figure 4.8 Microcontroller Pin Configuration

4.4. Motor Control Board

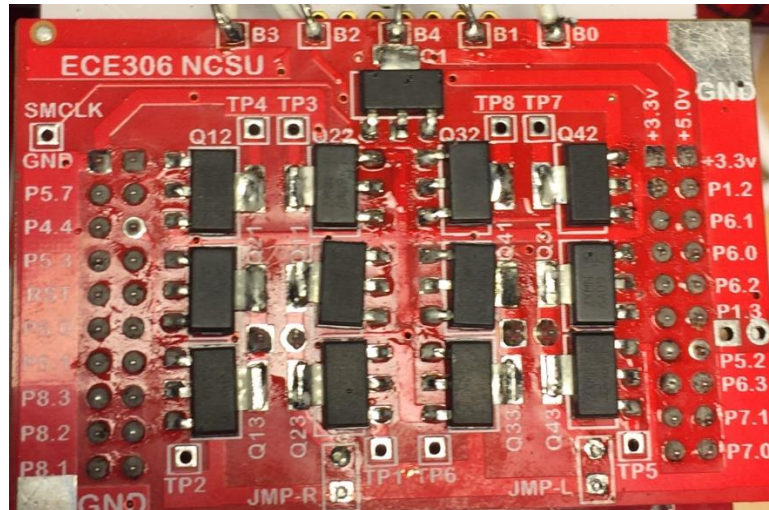


Figure 4.9 Motor Control PCB

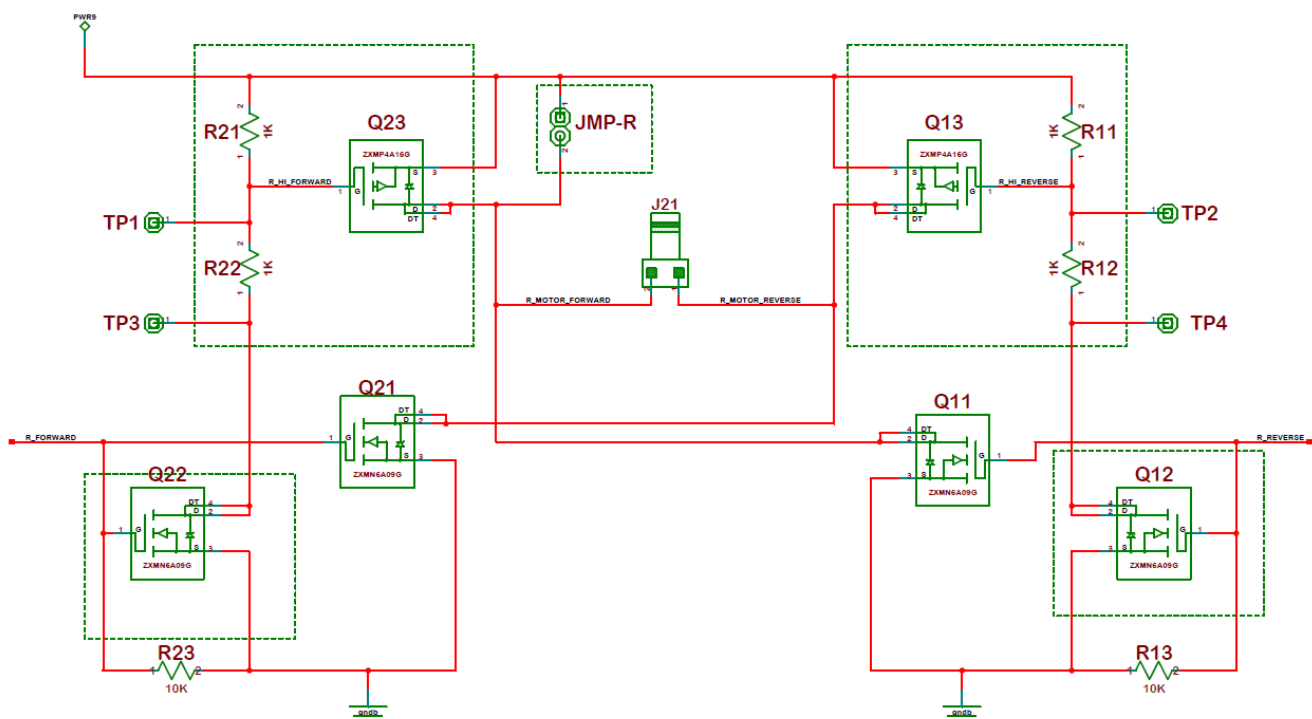


Figure 4.10 H-Bridge for Right Motor (mirror left H-Bridge schematic)

The Educational Sensing Car uses an h-bridge to drive each motor based on digital I/O signals from the MSP430. Although the above images display a fully-assembled h-bridge for each motor, the current implementation consists of only two half-h-bridges. The two half-h-bridges each consist of the N-FET Q21 and can only drive the motors in the forward direction based on the digital signal received from R_FORWARD and L_FORWARD.

6. Test Process

To test that the hardware was functioning properly, we performed various tests including probing the board with a volt meter to ensure the voltage across various terminals was correct and verifying that the LCD turned on and displayed the proper text.

6.1. Volt meter

To verify that all the resistors were secured in place and connected after the assembly and reflow, we connected probes to ground and the high voltage point on the power board. With a volt meter on a test probe site, we then verified that the proper amount of voltage was present at the test probe site. This voltage should be around 3.2-3.3 volts. In addition, we used a multimeter over the pins in J12 to ensure that when the system was connected to the battery pack, the pins were again receiving the proper voltage.

6.2. LCD Functions

The LCD is a very important part of the car as it allows the user to debug the system and therefore one must be certain that it works properly. To confirm that the LCD was connected properly, we carefully inspected all of the solder joints and checked that the backlight was snugly attached to the LCD.

6.3. Testing Cable Connectors

The jumper cables used to power the motors for the wheels were tested with a 5-volt input to one side of the connector and ground on the other side. Then using a voltmeter, we determined whether the wires were correctly crimped into the connector by checking if the voltage reached the ends of the wire.

6.4. Testing N-Connectors

To ensure that voltage from the batteries was reaching the newly installed N-Connectors, we plugged in the battery pack and turned it on. Then using a volt meter, we could check that the voltage was present between each terminal of the N-Connectors.

6.5. Software Checks

Verification that the code we wrote could control the cars forward movement was implemented by first writing functions that updated the motor port pins and then calling these functions to observe the output of the motors. Once the vehicle could drive, we utilized the above functions to control specific movement that allowed the car to turn and curve dependent of whether the ports were on or off. From there, observations of the drawn shape could be converted to code updates that improved the car's shapes.

6.6. Output LCD checks

We used functions that would display the shape that the car was attempting to draw in order to ensure that we were evaluating the correct shapes and updating the code to correct for any mistakes.