HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

PROJECT 1

RC CAR MOTOR CONTROL CIRCUIT

PHAN THANH DANH

STUDENT ID: 18119214

NGUYEN VO THANH TRI

STUDENT ID: 18119047

Major: **COMPUTER ENGINEERING**

Advisor: **TRUONG NGOC SON, PhD.**

Ho Chi Minh City, January 2020

HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRAINING

PROJECT 1

RC CAR MOTOR CONTROL CIRCUIT

PHAN THANH DANH

STUDENT ID: 18119214

NGUYEN VO THANH TRI

STUDENT ID: 18119047

Major: **COMPUTER ENGINEERING**

Advisor: **TRUONG NGOC SON, PhD.**

Ho Chi Minh City, January 2020

APPENDIX 3:

THE SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom– Happiness

Ho Chi Minh City, January 20, 2020

PROJECT ASSIGNMENT

Student name: ____________________________ Student ID: _____________________

Student name: _____________________________ Student ID: _____________________

Student name: _____________________________ Student ID: _____________________

Major: ____________________________________ Class: ___________________________

Advisor: ________________________________ Phone number: ___________________

Date of assignment: _______________________ Date of submission: ______________

1. Project title: ____

2. Initial materials provided by the advisor: _______________________________________

3. Content of the project: ____

4. Final product: ____

CHAIR OF THE PROGRAM
(Sign with full name)

ADVISOR
(Sign with full name)

APPENDIX 4:

THE SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom– Happiness

Ho Chi Minh City, January 20, 2020

ADVISOR'S EVALUATION SHEET

Student name: .... Student ID: ....

Student name: .... Student ID: ....

Student name: .... Student ID: ....

Major: ....
Project title: ....
....
Advisor: ....

EVALUATION

1. Content of the project:

....
....
....

2. Strengths:

....
....
....

3. Weaknesses:

....
....
....

4. Approval for oral defense? *(Approved or denied)*

....

5. Overall evaluation: (Excellent, Good, Fair, Poor)

....

6. Mark:……………….(*in words*:..)

Ho Chi Minh City, month day, year
ADVISOR
(Sign with full name)

APPENDIX 5:

THE SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom– Happiness

Ho Chi Minh City, January 20, 2020

PRE-DEFENSE EVALUATION SHEET

Student name: .... Student ID: ....
Student name: .... Student ID: ....
Student name: .... Student ID: ....

Major: ....
Project title: ....
....
Name of Reviewer: ....

EVALUATION

1. Content and workload of the project

....
....
....
....

2. Strengths:

....
....
....

3. Weaknesses:

....
....
....

4. Approval for oral defense? *(Approved or denied)*

....

5. Overall evaluation: *(Excellent, Good, Fair, Poor)*

....

6. Mark:……………….(*in words*:...)

Ho Chi Minh City, month day, year
REVIEWER
(Sign with full name)

APPENDIX 6:

THE SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom– Happiness

EVALUATION SHEET OF
DEFENSE COMMITTEE MEMBER

Student name: .... Student ID: ....

Student name: .... Student ID: ....

Student name: .... Student ID: ....

Major: ....

Project title: ....

....

Name of Defense Committee Member:

....

EVALUATION

1. Content and workload of the project

....

....

....

....

2. Strengths:

....

....

....

3. Weaknesses:

....

....

....

4. Overall evaluation: *(Excellent, Good, Fair, Poor)*

....

5. Mark:……………….(*in words*:..)

Ho Chi Minh City, month day, year

COMMITTEE MEMBER

(Sign with full name)

Acknowledgment

To complete this research project, the first word that allows the group to do this is to thank all the teachers of Ho Chi Minh City University of Technology And Education in general and the teachers in the Faculty of Electrical - Electronics in particular, those who have dedicated to teaching and equipping the group with important background and specialized knowledge, helping the implementation team have a solid theoretical basis and have always created favorable conditions to help. best for the group during learning and research.

In particular, the group would like to thank Assoc. Prof Truong Ngoc Son enthusiastically helped, gave research directions as well as solutions to some problems so that the group could perform well the topic. During the time working with the teacher, the implementation team has continuously acquired more knowledge taught from the teacher, always showing a serious and effective research attitude and this is also essential in the process. study and work later.

Also, I would like to thank the Intelligent System Lab (ISLab) pertaining Department of Electrical and Electronic Engineering, supported the muscles equipment for group to do topic.

Despite trying our best, but due to the group's little experience, the topic cannot be avoided. Therefore, the group hopes to receive comments from teachers.

Team sincerely thank you!

Table of Contents

LIST OF IMAGES

LIST OF TABLES

LIST OF ACCRONYMS

CPU: Central Processing Unit

AVR: Automatic Voltage Regulator

UART: Universal Asynchronous Receiver / Transmitter

ACK: Acknowledgement

NACK: Negative Acknowledgement

Abstract

Recently, the topic of autonomous cars has been a lot of research and hunting applications to pick up. It is growing and complete as a strong trend of the world. These applications not only help reduce road accidents but also bring a lot of benefits to the driver's wing. This car does not use gas, instead it is equipped with a large capacity battery to ensure long journeys such as gasoline-powered cars. It is gradually replacing cars that run on gasoline and oil. But to complete an autonomous vehicle requires many stages of development such as the development from human control to a machine that can detect paths and obstacles to move by itself or replace human beings to identify signs, road markings, obstacles. Therefore, in this article, the implementation team created an RC car motor control circuit in combination with the Devo7 and RX601 controller handle.

Keywords: autonomous car, motor driver with H-bridge, DC motor controller

Chapter 1: INTRODUCTION

1.1 Overview:

In this project,we created a vehicle motor control circuit car based on Arduino NANO and L298N.This circuit is controlled by Devo 7 and RX 601. The car moves forward and reverse, left and right, and stops as the user desires. The user can also accelerate and decelerate using a joystick. Speed is displayed on LCD 1602 with the supports of I2C modules.

1.2 Objectives:

- Research, learn about Arduino NANO, L298N theory.
- Research, learn about the i2c communication standard combined with LCD 1602.
- Learn and implement the handcraft module as well as the control module for RC car.
- The object was to design the vehicle engine control circuit in conjunction with the control handle.

1.3 Scope of Works:

Our group have a lot of ideas about autonomous vehicles but we have limited capacity and time so we just make simple control circuits.

1.4 Included contents:

- **Chapter 1: INTRODUCTION**

In this chapter, we show the foremost look related to our project include.

- **Chapter 2: LITERATURE REVIEW AND THEORETICAL BASIS**

In this chapter, we mention about the basic knowledge related to our research and the needed things for accessing the topic easier like: knowledge about Arduino, H-bridge, I2C protocol.

- **Chapter 3: HARDWARE DESIGN**

In this chapter, we introduce the hardware we use in our project as well as the circuit we design from mentioned hardware.

- **Chapter 4: IMPLEMENTED RESULTS**

In this chapter, we show the results after complete product.

- **Chapter 5: CONCLUSION AND RECOMMENDATION**

In this chapter, we evaluate our results included advantages and disadvantages of this project. Besides, we propose the idea for future development.

Chapter 2: LITERATURE REVIEW AND THEORETICAL BASIS

2.1 Arduino-based introduction:

2.1.1 Introduction:

Arduino has really made headlines in the DIY user market (who are themselves invented its own products) worldwide in recent years, almost identical to those. What Apple has done in the mobile device market. The number of users is extremely large and multi. Form with degrees extending from high school up to university has made even these. Their creators must be surprised by how popular they are

What is Arduino that can induce even students and researchers at these. Famous universities such as MIT, Stanford, Carnegie Mellon must use; or even Google wants to help with the release of the Arduino Mega ADK kit for playing develop Android apps that interact with sensors and other devices?

The Arduino is actually a microprocessor board that is used for programming to interact with hardware device such as sensors, motors, lights or other devices. Highlights Arduino is an extremely easy-to-use application development environment, with a programming language that can be learned quickly even with less electronics and programming savvy. submit. And what makes the Arduino phenomenon is the very low price and the nature of the source open from hardware to software. For only about $ 30, users can own one. The Arduino board has 20 I / O terminals that can interact and control as many devices.

2.1.2 Atmega *328:*

Atmega is the main microcontroller integrated into the Arduino nano circuit which is small and compact as compared to Arduino Uno.

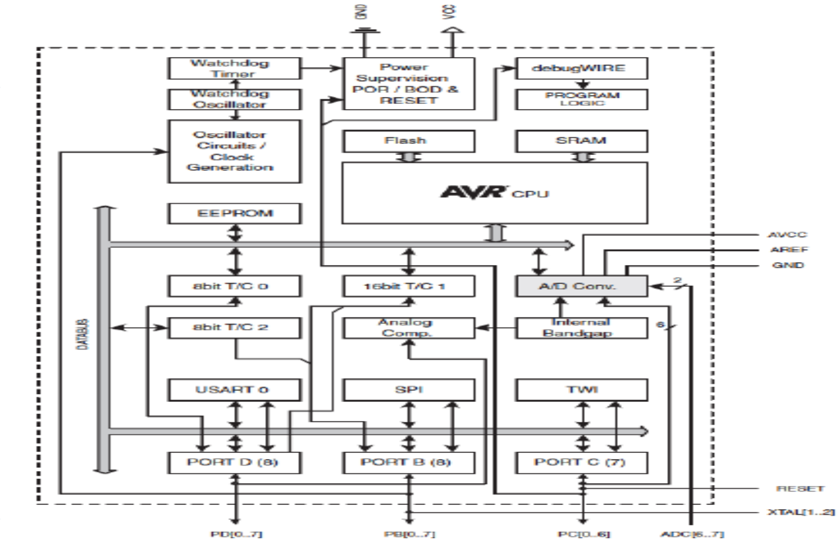

Figure 2.1: Structural block diagram inside the

This is general architecture in AVR core in general. [3] The CPU core's main function is to ensure correct program execution. The CPU must therefore be able to quickly access, perform calculations, control peripherals, and handle interrupts. To optimize performance, the AVR uses a separate Harvard architecture and bus route for the program and data. Data transfer direction in program memory executes at a certain rate.

2.1.2 Software structure

The program structure written for Arduino consists of the first two parts, the setup () and the loop () constructor.

Arduino is best known for its hardware, but must have software for programming the hardware, both hardware and software collectively called Arduino.

2.1.3 Arduino software

Arduino software called sketches, which was created on a development environment (IDE) computer. The IDE allows you to write, modify code, and convert it so that your hardware can understand it. IDE is used to compile and load into the Arduino (this processing is called UPLOAD).

2.1.4 Arduino hardware

Arduino hardware are Arduino boards where programming equations are executed. These boards can control or respond to electrical signals, so components are directly coupled to it to interact with the real world to sense and communicate.

2.2 Dual Motor Controller IC:

2.2.1 Dual H bridge operation:

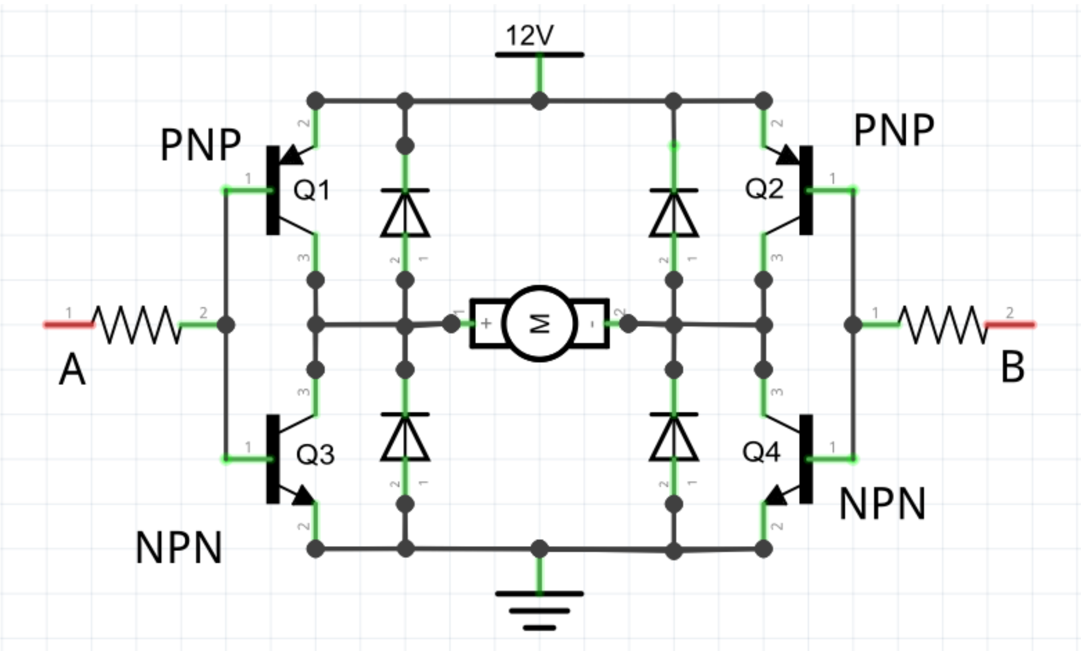

Figure 2.2: H-bridge circuit diagram using BJT transistor

[2] As shown in the diagram above, we have A and B are 2 control poles connected in series with 2 resistors of current, depending on the type of transistor you are using, this

resistance value is different. Make sure that the current through the base terminals of the transistors is not too great to damage them. On average, use a 1k Ohm resistor.

We control these two poles by signal levels HIGHT, LOW 12V or 0V respectively

With 2 control poles and 2 HIGH / LOW signals corresponding to 12V / 0V for each pole, there are 4 cases that happen as follows:

A is at LOW and **B is HIGH**

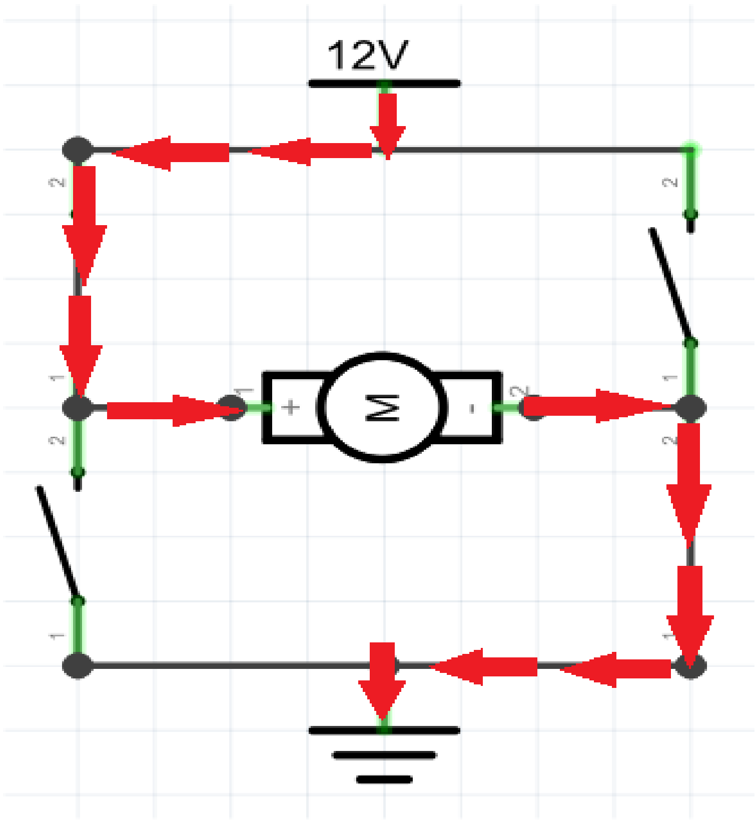

Figure 2.3: H-bridge operation in forward direction

On the A side, transistor Q1 is open, Q3 is closed. On the B side, transistor Q2 is closed, Q 4 is open. Therefore, the current in the circuit can flow from 12V to Q1, through the motor to Q4 to GND. At this time, the motor rotates in the forward direction.

A is HIGH and **B is LOW**

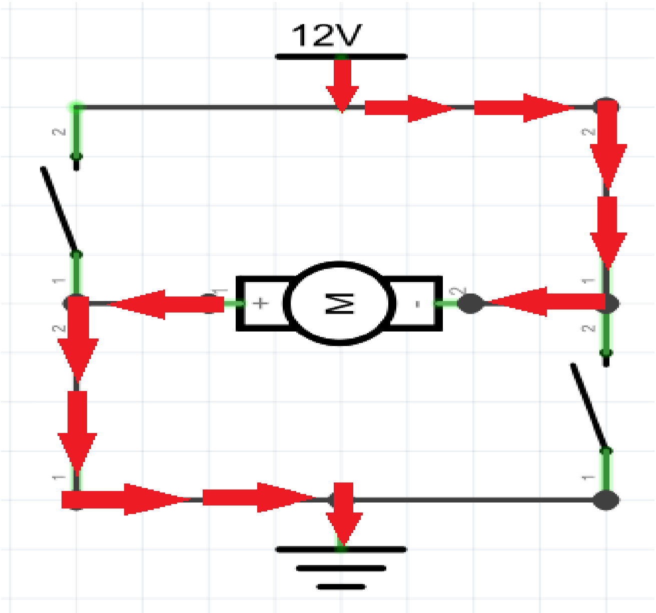

Figure 2.4: H-bridge operation in reverse direction

On the A side, transistor Q1 is closed, Q3 is open. On the B side, transistor Q2 is open, Q 4 is closed. Therefore, the current in the circuit can flow from 12V to Q2, through the motor to Q3 to return to GND. At this time, the motor turns in the opposite direction.

A and B are both at LOW

Then, transistors Q1 and Q2 are open but Q3 and Q4 are closed. Current has no path to return to GND so there is no current flowing through the motor - the motor does not work.

A and B are both at HIGH

Then, transistors Q1 and Q2 are closed but Q3 and Q4 are open. Current cannot flow from the 12V source so there is no current flowing through the motor - the motor does not work.

2.2.2 IC L298 operation and details:

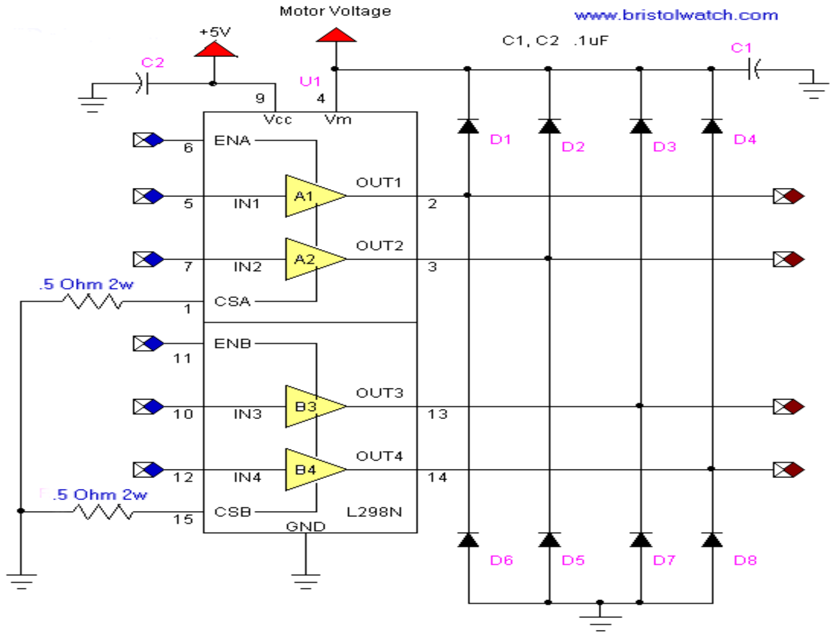

Figure 2.5: Schematics of L298N

Pictured above is the basic L298N circuit used to drive inductive/magnetic loads. One of the annoying features of the unit is the lack of internal parasitic (flywheel) diodes to deal with voltage spikes. We use diode 1N4001 for this purpose.

The four power amplifiers and grouped in pairs of two with individual enable pins (ENA, ENB) and individual current sense pins (CSA, CSB) for each pair. The current sense pins in general can be tied to ground, but one can insert low value resistor, whose voltage reading is proportional to current.

ENA, ENB, and In1-In4 are all standard 5-volt TTL logic making connection to most micro-controllers easy. ENA will turn on A1 and A2 when with a digital HIGH (5-volts) and off when LOW (0 volts); the corresponding outputs will be floating when off. Same is true of ENB, In3 and In4. ENA and ENB can be connected directly together to enable both channels at once or simply tied to +5 volts and both channels making all four outputs active at all times.

2.3 The I2C Communication Protocol.

2.3.1 Introduction I2C Communication Protocol.

I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD

Like UART communication, I2C only uses two wires to transmit data between devices:

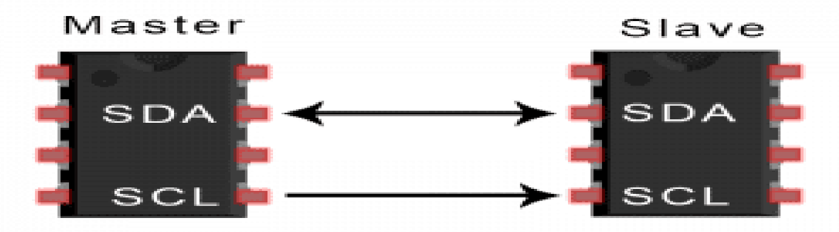

Figure 2.6: I2C communication protocol

SDA (Serial Data) – The line for the master and slave to send and receive data.

SCL (Serial Clock) – The line that carries the clock signal.

I2C is a serial communication protocol, so data is transferred bit by bit along a single wire (the SDA line).

Like SPI, I2C is synchronous, so the output of bits is synchronized to the sampling of bits by a clock signal shared between the master and the slave. The clock signal is always controlled by the master.

2.3.2 Activities of I2C Communication Protocol

With I2C, data is transferred in messages. Messages are broken up into frames of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame:

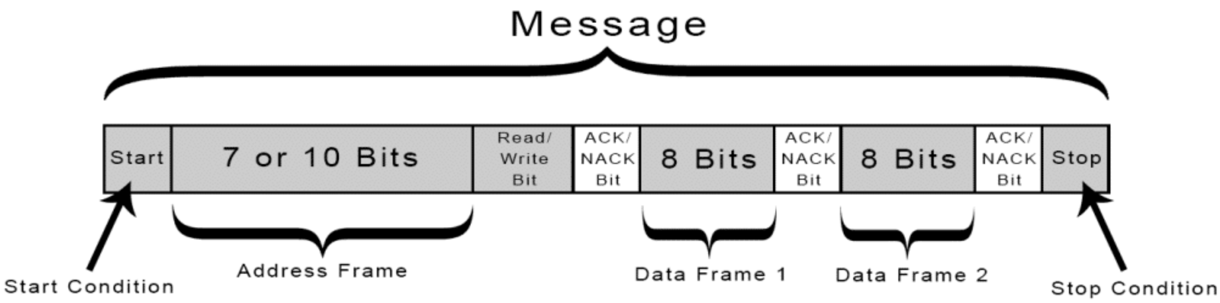

Figure 2.7: Message structure

Start Condition: The SDA line switches from a high voltage level to a low voltage level before the SCL line switches from high to low.

Stop Condition: The SDA line switches from a low voltage level to a high voltage level after the SCL line switches from low to high.

Address Frame: A 7 or 10 bits sequence unique to each slave that identifies the slave when the master wants to talk to it. The address frame is always the first frame after the start bit in a new message. The master sends the address of the slave it wants to communicate with to every slave connected to it. Each slave then compares the address sent from the master to its own address. If the address matches, it sends a low voltage ACK bit back to the master. If the address doesn't match, the slave does nothing and the SDA line remains high.

Read/Write Bit: The address frame includes a single bit at the end that informs the slave whether the master wants to write data to it or receive data from it. If the master wants to send data to the slave, the read/write bit is a low voltage level. If the master is requesting data from the slave, the bit is a high voltage level.

ACK/NACK Bit: Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

The Data Frame: After the master detects the ACK bit from the slave, the first data frame is ready to be sent. The data frame is always 8 bits long, and sent with the most significant bit first. Each data frame is immediately followed by an ACK/NACK bit to verify that the frame has been received successfully. The ACK bit must be received by either the master or the slave (depending on who is sending the data) before the next data frame can be sent. After all of the data frames have been sent, the master can send a stop condition to the slave to halt the transmission. The stop condition is a voltage transition from low to high on the SDA line after a low to high transition on the SCL line, with the SCL line remaining high.

2.3.3 Steps of I2C Data Transmission:

Step 1: [1] The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level before switching the SCL line from high to low.

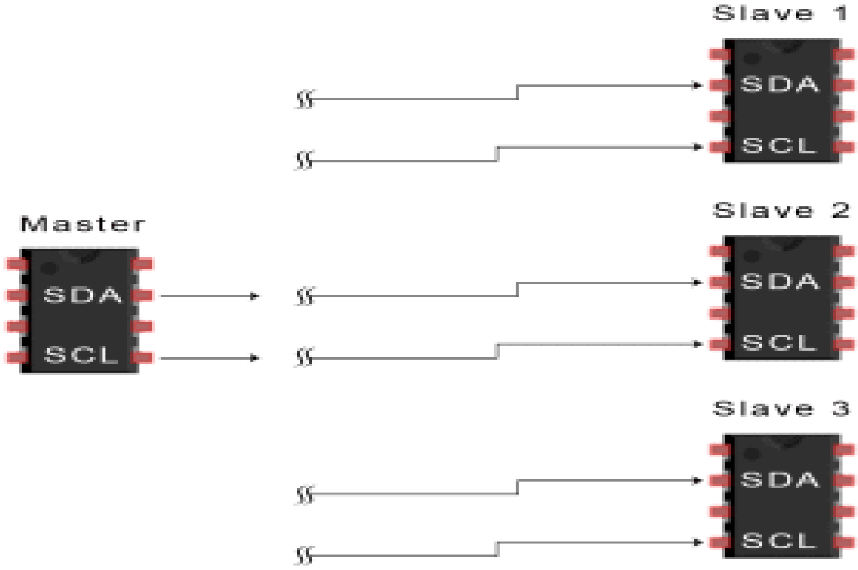

Figure 2.8: The master sends the start condition to slave

Step 2: The master sends each slave the 7 or 10 bits address of the slave it wants to communicate with, along with the read/write bit:

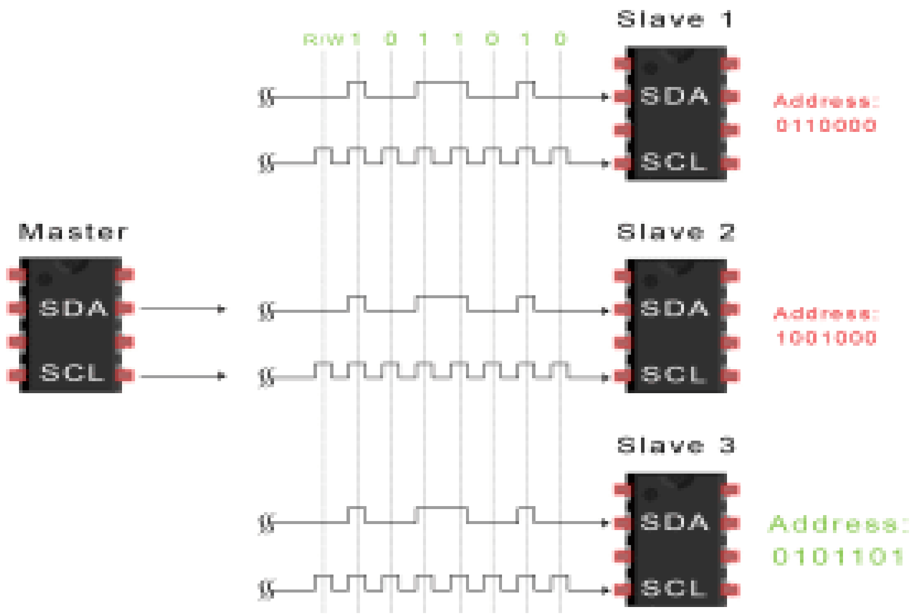

Figure 2.9: Master sends each slave the 7 or 10 bits address

Step 3: Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high

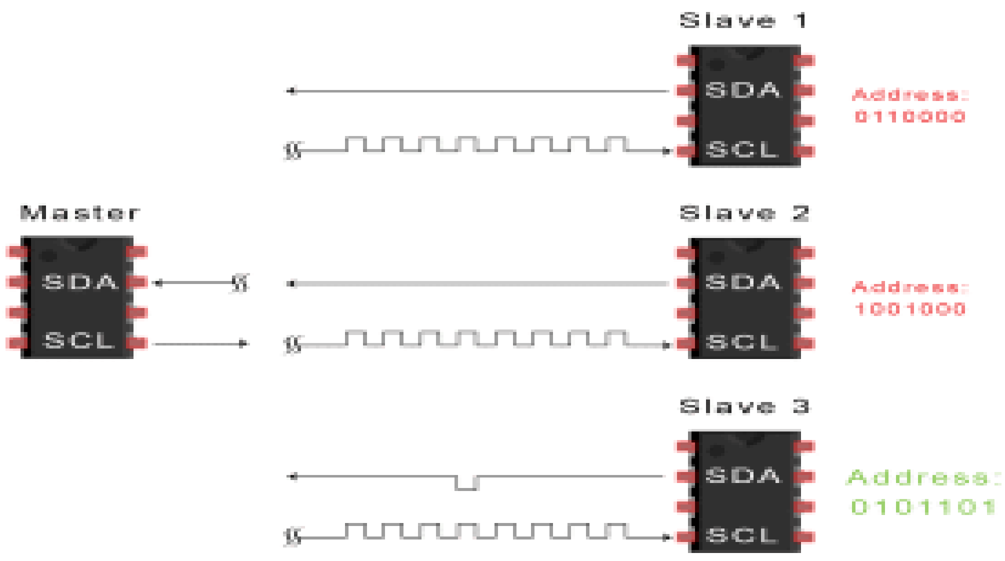

Figure 2.10: Each slave compares the address sent from the master

Step 4: The master sends or receives the data frame

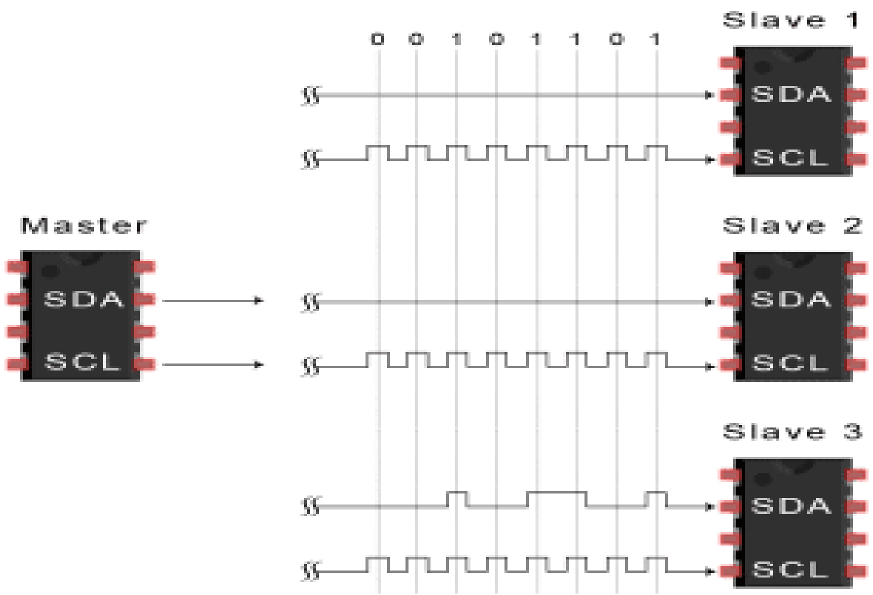

Figure 2.11: The master sends or receives the data frame

Step 5: After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame

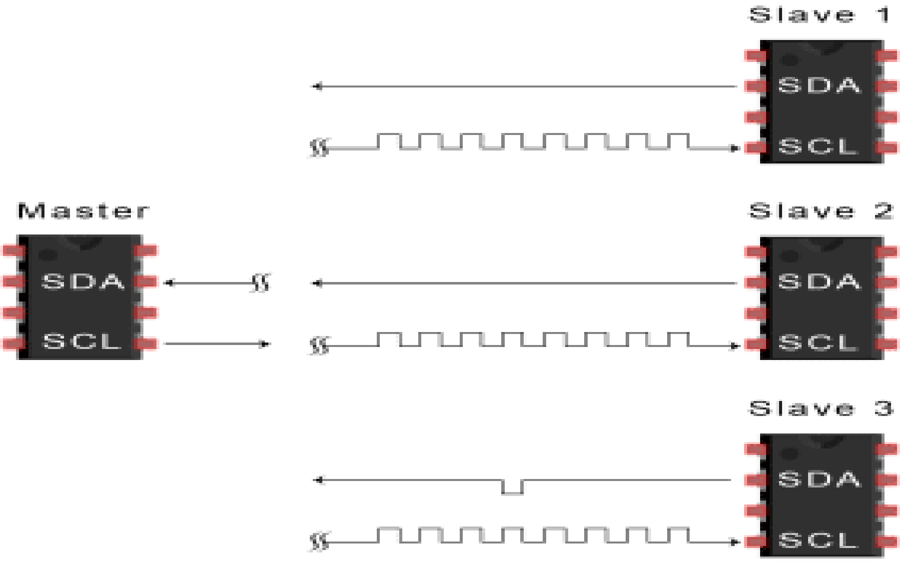

Figure 2.12: The receiving device returns another ACK bit

Chapter 3: HARDWARE DESIGN

3.1 Car RC

Figure 3.1: Car RC Land Rover 70 Anniversary Edition [4]

Length	385mm
Width	145mm
Height	155mm
Car Battery	7.4V 1200mAH
Speed	≥8.72km/h
Control distance	80m

Table 3.1: Specifications of car RC

- The simulation is true for real cars.
- Size 1/12 real car.
- The transmission structure is the same as the real car, including 2 sets of rear and middle differential.

3.2 Brushed Motor

Figure 3.2: Brushed Motor [5]

Length	550mm
Diameter	36mm
Shaft	3mm
RPM	Max 22000

Table 3.2: Specifications of Brushed Motor

Working principle: based on the contact mechanism between commutator, brush to help supply electricity to the coil. The brush parts on this type of motor are pushed by the leaf or coil springs to continuously contact and slide over the commutator surface or the slip ring electrically, maintaining the contact of the rotor.

Feature: is the main movement of the vehicle

3.3 Devo7

Figure 3.3: Control Handle Devo7 [6]

Encoder	ARM microcomputer system
Frequency	2.4 GHz (DSSS)
Output power	-5dBm ~ 20dBm
Battery	12V or 1.5VX8 AA battery

Table 3.3: Specifications of Devo 7

Feature

- Adopting 2.4GHz Direct Scan Frequency (DSSS) spectrum with fast response and strong anti-interference protection
- Two efficient recovery circuits ensure stability of singal reception
- The Micyoco single chip is the CPU that provides super powerful analytics
- It can be customized as ID set and automatic ID assignment

3.4 RX 601

Figure 3.4: RX 601 [7]

Dimensions	33*20*13mm
Channel	6CH
Sensitivity	-105dmb
Receiver Battery	4.8-6V1300mAh
Weight	5g

Table 3.4: Specifications of RX 601

Function: transmit and receive signals

3.5 IC L298N

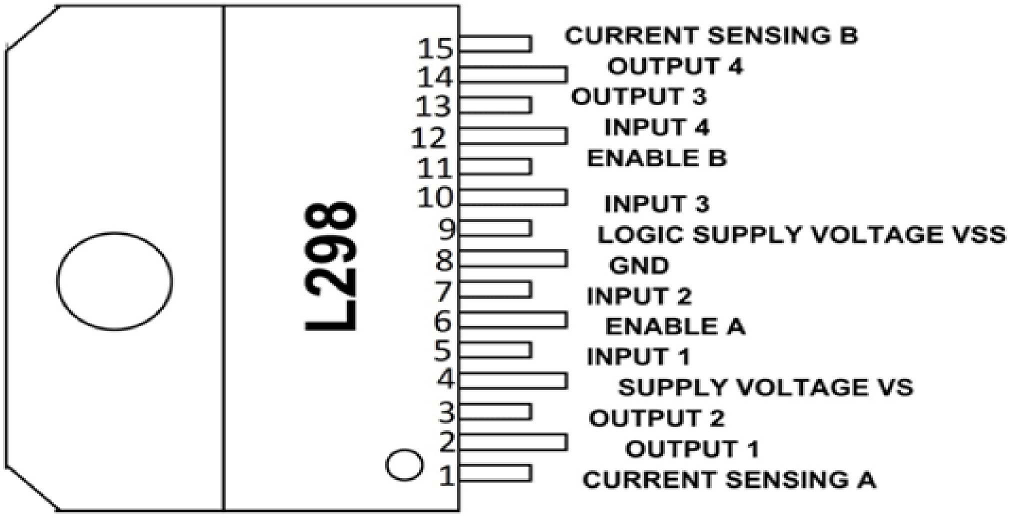

Figure 3.5 IC L298N [8]

Pins	**Name**	**Function**
1,15	Sense A, Sense B	Between this pin and the ground, a sense resistor is connected to control the current of the load
2,3	Out 1, Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1
4	VS	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5,7	Input 1, Input 2	TTL Compatible Inputs of the Bridge A
6,11	Enable A, Enable B	TTL Compatible Enable Input: the L state disables the bridge A(enable A) and/or the bridge B (enable B)
8	GND	Ground
9	VSS	Supply Voltage for the Logic Blocks. (A100nF capacitor

		must be connected between this pin and ground).
10,12	Input 3, Input 4	TTL Compatible Inputs of the Bridge B
13,14	Out 3, Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at the pin

Table 3.5: Pin Function of IC L298N

Features:
- Operating Supply Voltage up to 46V
- Total DC Current up to 4A
- Low Saturation Voltage
- Overtemperature Protection
- Logical "0" Input Voltage up to 1.5V (High Noise Immunity)

3.6 Servo

Figure 3.6: Servo [9]

Size	32 × 11.5 × 24mm (Include tabs) 23.5 × 11.5 × 24mm (Not include tabs)
Weight	8.5g (Not include a cable and a connector) 9.3g (Include a cable and a connector)
Speed	0.12sec/60degrees (4.8V) 0.10sec/60degrees (6.0V)
Torque	1.5kgf-cm (4.8V) 2.0kgf-cm (6.0V)
Voltage	4.8V-6.0V

Table 3.6: Specifications of Servo

Working principle**:** A servo motor is a motor formed by a closed loop feedback system. The motor output will be connected to a control circuit. When the motor operates, speed and position will be returned to this control circuit. If the motor rotation is prevented for any reason. The feedback mechanism detects that the output signal has not reached the desired position. Control circuit will continue to adjust the deviation for the motor to achieve the most accurate point.

3.7 Module I2C

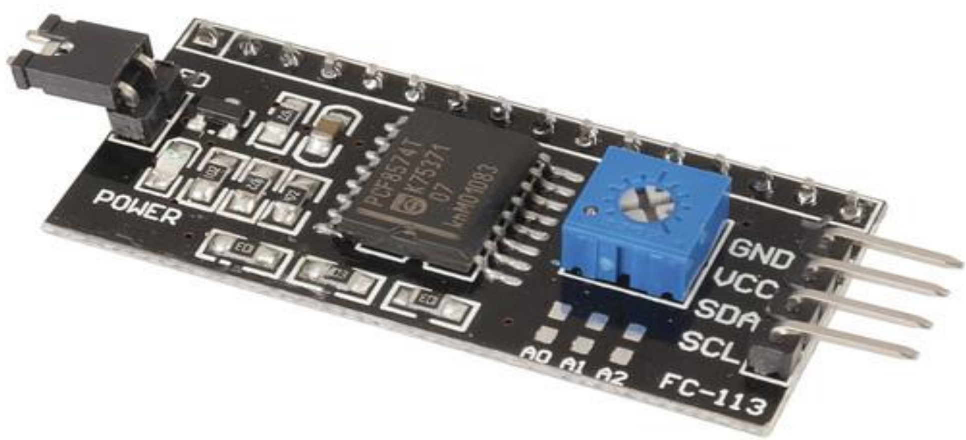

Figure 3.7: Module I2C [10]

Dimensions	41.5x19x15.3mm
Weight	5g
Operating voltage	2.5v-6v
Jump latch	Provide lights to LCD or disconnect

Table 3.7: Specifications of Module I2C

GND	Ground
VCC	Power supply 5V
SDA	Wire transmisstion data
SCL	Synchronize between devices
A0,A1,A2	Address

Table 3.8: Pins Function Table of Module I2C

Feature:

- Designed to solder directly to 16pin SIL headers
- Mounts behind LCD panel, taking up minimal space
- Arduino libraries offer direct LCD control

3.8 LCD 1602

Introduction LCD 1602

Figure 3.8: LCD 1602 [11]

LCD1602, or 1602 character-type liquid crystal display, is a kind of dot matrix module to show letters, numbers, and characters and so on. It's composed of 5x7 or 5x11 dot matrix positions; each position can display one character. There's a dot pitch between two characters and a space between lines, thus separating characters and lines. The model 1602 means it displays 2 lines of 16 characters

Pin	Function
VSS	Connected to ground
VDD	Connected to a +5V power supply

V0	To adjust the contrast
RS	Toggles between Command/Data Register
R/W	Toggles the LCD between Read/Write Operation
Enable	Must be held high to perform Read/Write Operation
Data Bits (0-7)	Pins used to send Command or data to the LCD.
A	Pins that control the LCD blacklight. Connect A to 3,3V
K	Pins that control the LCD blacklight. Connect K to GND

Table 3.9: Pin Function Table of LCD 1602

3.9 Arduino NANO

Arduino NANO is a board designed with the central processing unit being an AVR microcontroller Atmega328. The main structure of Arduino NANO includes the following parts:

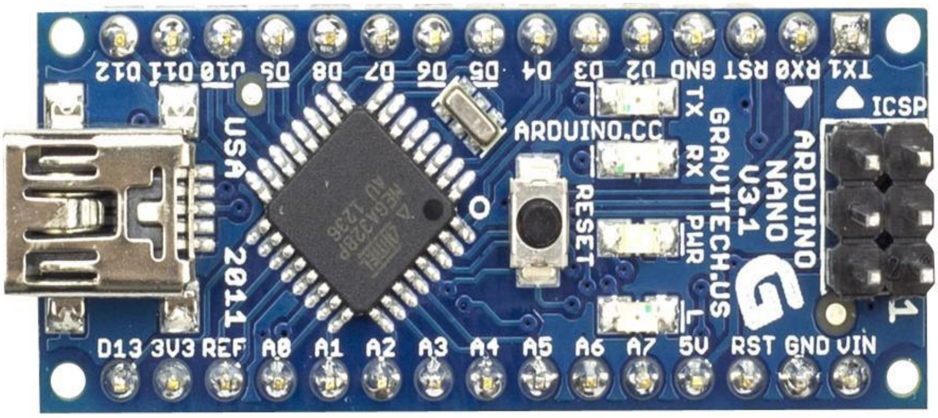

Figure 3.9 Arduino NANO [12]

USB port: This is the communication port for us to upload code from the PC to the microcontroller and power supply for Arduino. Copper it was also the serial communication for data transmission between the microcontroller and the computer.

Microcontroller: This is the heart of the development board, which works as a mini computer and can receive as well as send information or command to the peripheral devices connected to it. The microcontroller used differs from board to board; it also has its own various specifications.

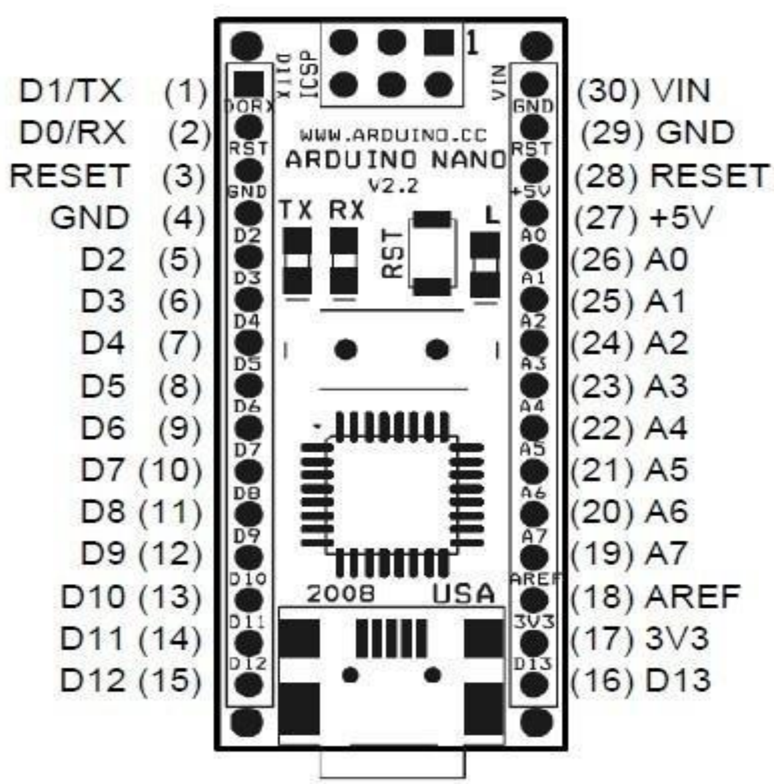

Figure 3.10 Topview of Arduino NANO

There are 14 input / output pins numbered from 0 to 13, in addition to a grounding pin (GND) andone reference voltage pin (AREF).

Analog Pins: There are some analog input pins ranging from A0 – A7 (typical). These pins are used for the analog input / output. The no. of analog pins also varies from board to board.

Digital I/O Pins: There are some digital input pins also ranging from 2 to 16 (typical). These pins are used for the digital input / output. The no. of these digital pins also varies from board to board

AVR microcontroller: this is the central processing unit of the entire board. With each Arduino model different chips are different. In this Arduino NANO, it use ATMega328

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"
Length	45 mm
Width	18 mm
Weigth	5 g

Figure 3.11 Specifications of Arduino NANO

Power:

The Arduino Nano can be powered via the Mini-B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source. The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory:

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM

Input and Output:

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions.

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the analogReference() function. Additionally, some pins have specialized functionality:

I2C: 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library (documentation on the Wiring website).

There are a couple of other pins on the board:

AREF. Reference voltage for the analog inputs. Used with analogReference().

Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication**:**

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A SoftwareSerial library allows for serial communication on any of the Nano's digital pins. The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus.

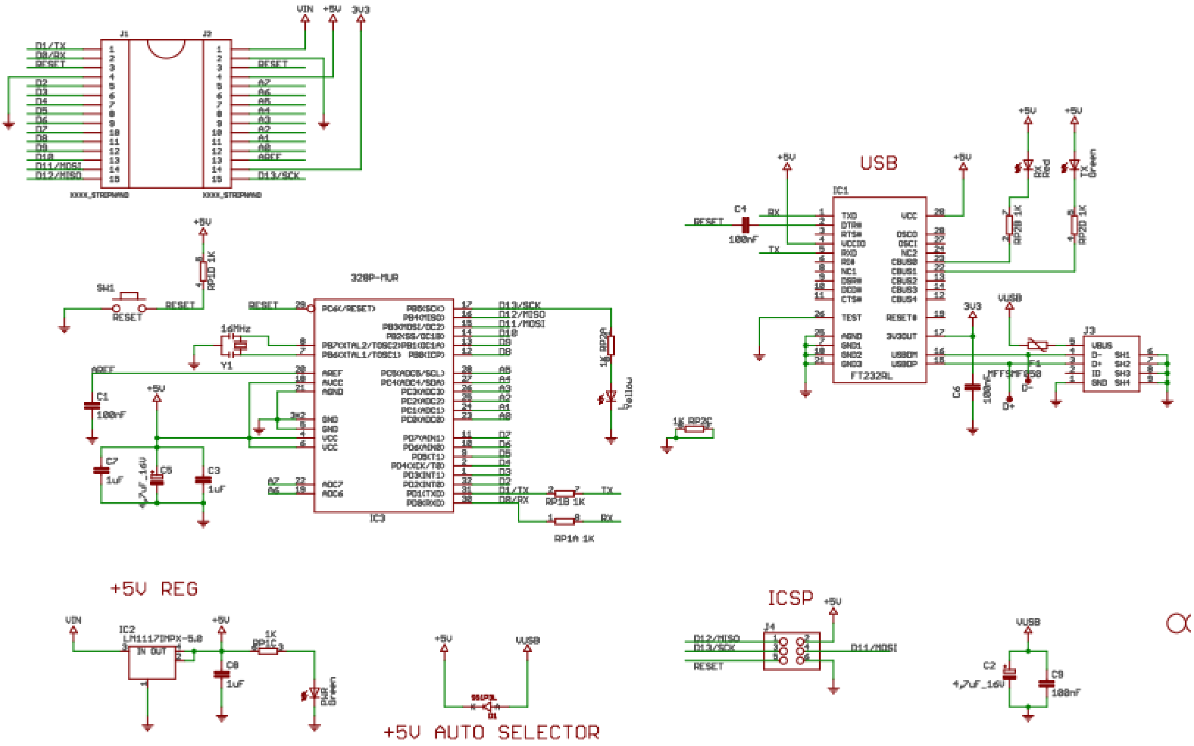

Figure 3.12 Schematic of Arduino NANO

3.10 Design Circuit

- Arduino NANO pins used to communicate with LCD and L298N
- Pin 5 connects to the IN1 pin of the L298N
- Pin 6 connects to the IN1 pin of the L298N

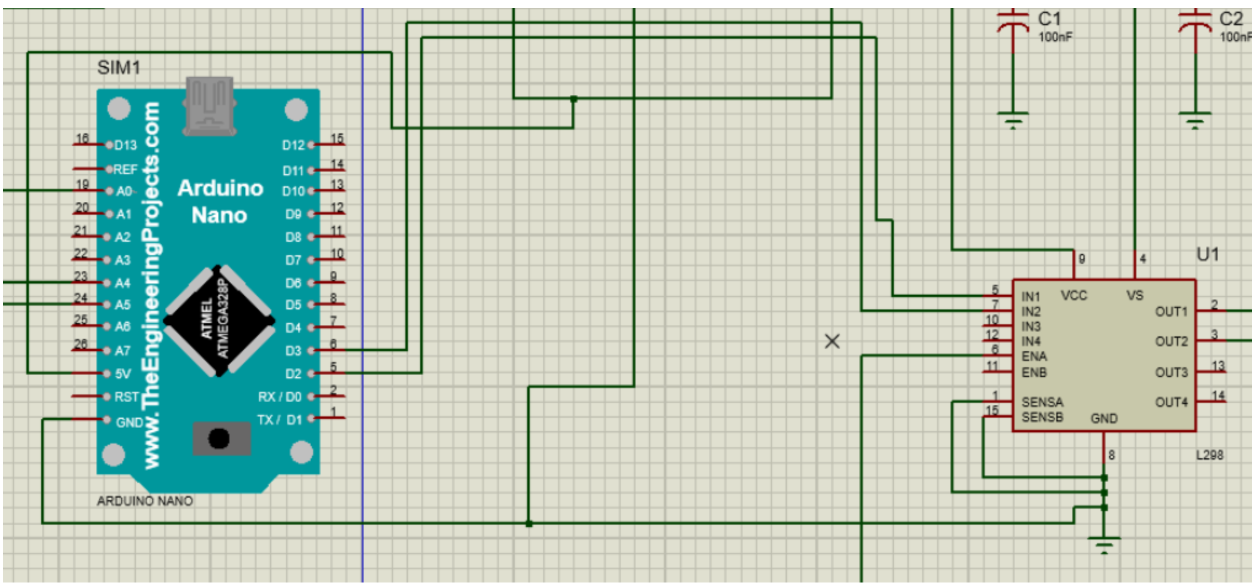

Figure 3.13: Connect Arduino NANO with L298N

- Pin 23 connects to the SDA pin of the LCD to transmit data to LCD
- Pin 24 connects to the SCL pin of the LCD to synchronize between devices when transmitting data
- Pin 27 connects to the VCC pin of LCD to power the LCD
- Pin 29 connects to the GND pin of LCD

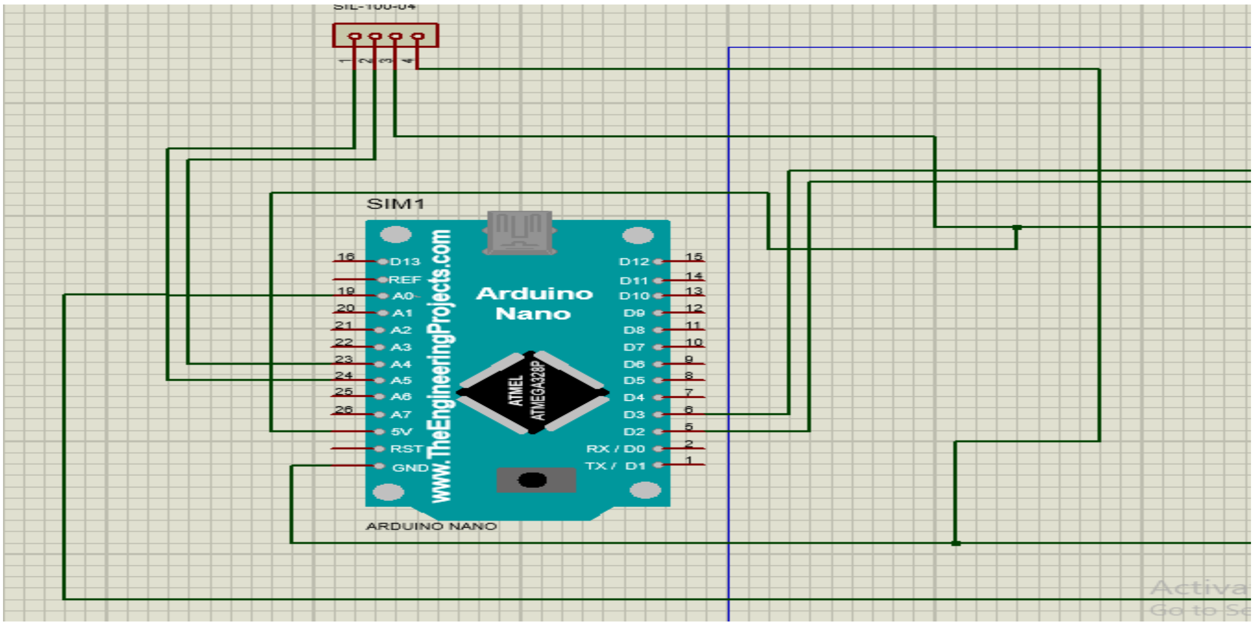

Figure 3.14: Connect Arduino NANO with LCD 1602 through I2C

From the schematic we designed the PCB base on that to implement the handcraft circuit:

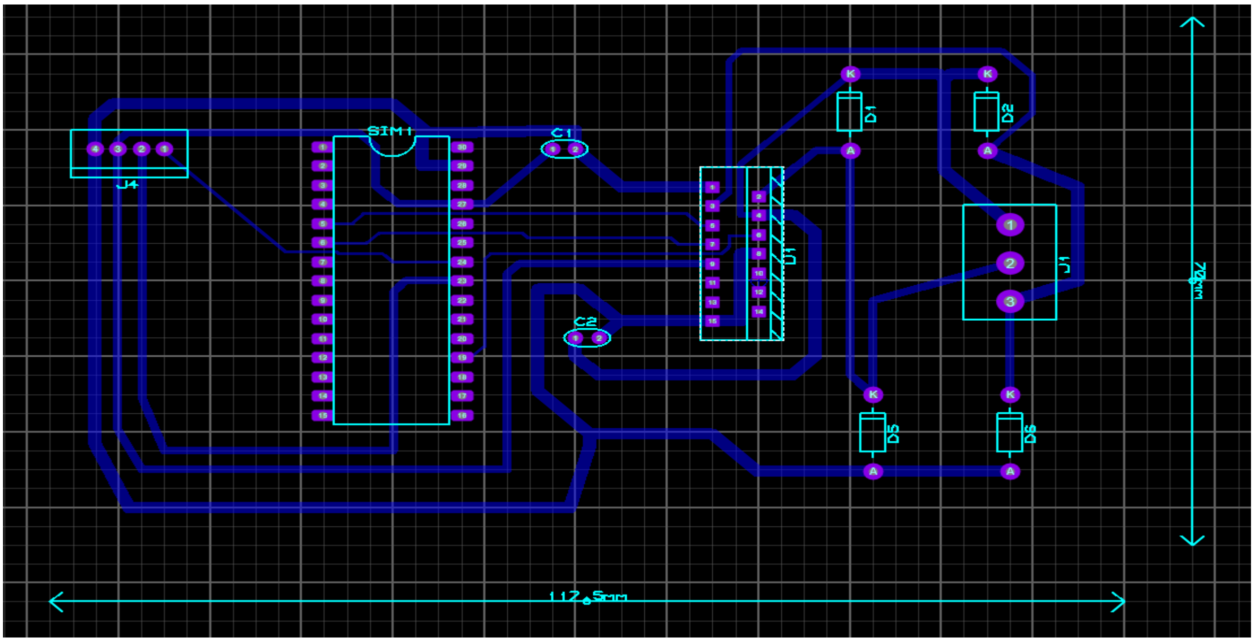

Figure 3.15: PCB of circuit

General diagram of hardware operation

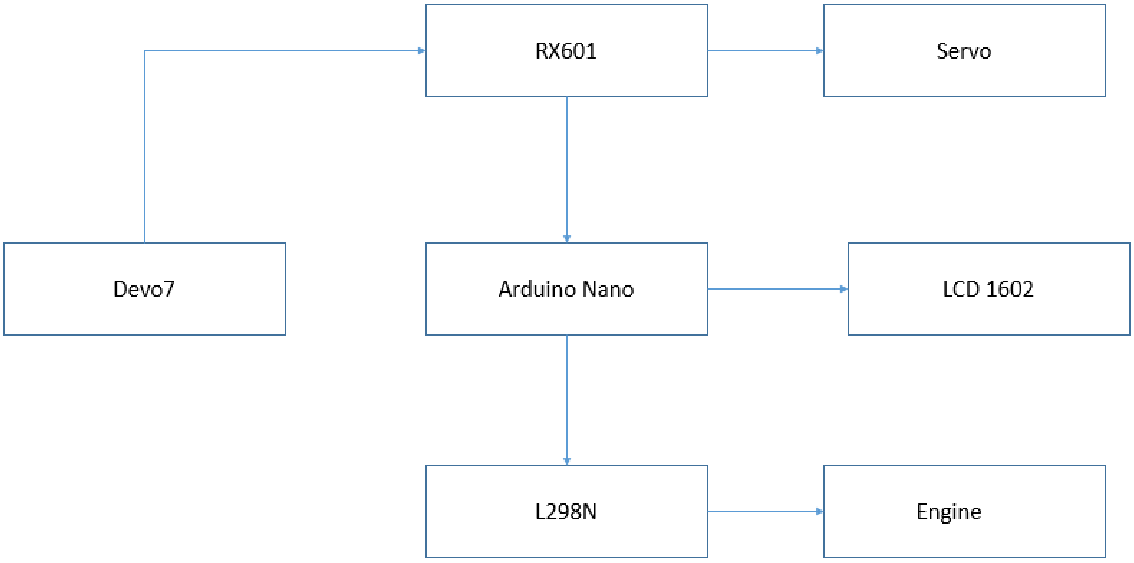

Figure 3.16: Operation diagram of hardware

Chapter 4: IMPLEMENTED RESULTS

4.1 The pulse measured from RX601:

The pulse is plot from Arduino serial monitor after receive a signal from TX (devo7). The range is of the pulse is approximately 1000 to 2000 and we scale that pulse to convert to speed:

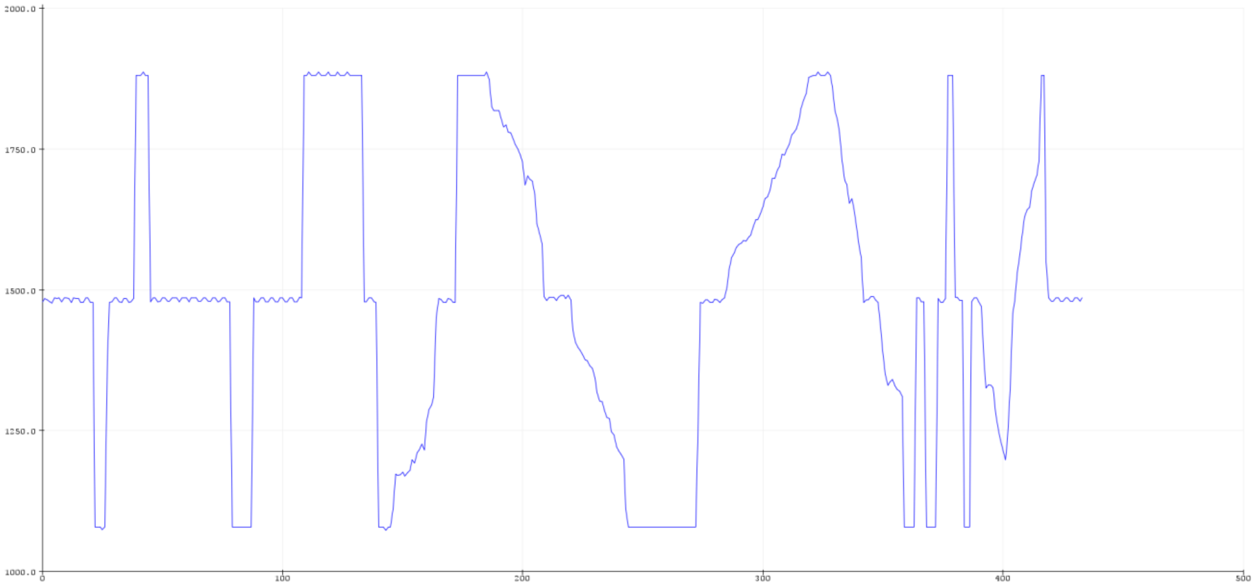

Figure 4.1: Pulse is received by RX601

4.2 The LCD display:

The LCD is used for displaying the speed which is transmitted to the motor and the corresponding pulse that Arduino receive from the RX601

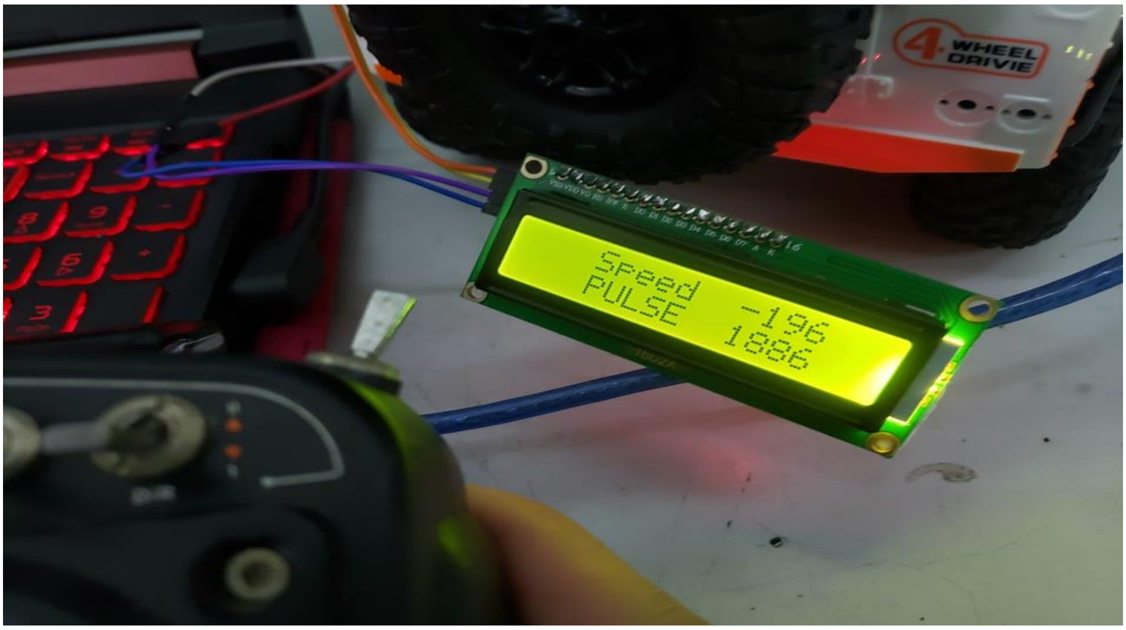

Figure 4.2 Display on LCD

4.3 Complete System

Figure 4.3: The whole system

Chapter 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

5.1.1 Complete works

In this project, our group built a circuit to control the DC (brushed) motor by based on the L298N theory. We had designed the handcraft circuit and integrated it onto the RC car. The all built product is tested with the stability and do not have any serious problems.

In the other hand, the combination of the TX, RX module with the control motor module help us develop our product in the future very much.

5.1.2 Incomplete works

We have not achieved our goals that we have set that is made a more integrated module and complex circuit, because the limit of one layer circuit so that we cannot route the wire completely. Finally, we decided to test the initial circuit on test board and make it simple when apply it to the car.

5.2 Future works

This project is to research the structure and the way to control a typical car. This is for develop to autonomous car in the senior project 2 and maybe the thesis. It can be used for the document for teach the university fresher who want to make the initial project like: Line following car, Bluetooth controller car, and so on.

References

[1] https://www.circuitbasics.com/basics-of-the-i2c-communication-protocol/
[2] http://arduino.vn/bai-viet/281-mach-cau-h-va-dieu-khien-dong-co-voi-sn754410
[3] https://predictabledesigns.com/atmega-versus-stm32-which-microcontroller-is-best-for-your-application/
[4] https://shopee.vn/MN-99-2.4G-1-12-4WD-RTR-Crawler-RC-Car-For-Land-Rover-70-Anniversary-Edition-Vehicle-Model-i.175761037.6110186687
[5] http://www.stormracer.com/sanpham/mo-to-dieu-toc/dong-co-dien-for-car/hbx-brushed-motor-rc-550ph-motor-choi-than_12010006.product
[6] http://myrcsaigon.com/san-pham/quadcopter/bo-dieu-khien-tx-rx/tay-dieu-khien-devo-7-2/
[7] http://www.vietrc.com/walkera-devo-rx601
[8] https://dientubachviet.com/l298-l298n-9h6-2b
[9] https://dientu360.com/dong-co-servo-sg90
[10] https://chotroihn.vn/module-i2c-lcd1602-lcd2004-tai-linh-kien-dien-tu-3m
[11] https://nshopvn.com/product/lcd-1602-kem-module-i2c-mau-xanh-la/
[12] https://www.elektor.com/arduino-nano-v3