**DC MOTOR PROJECT**

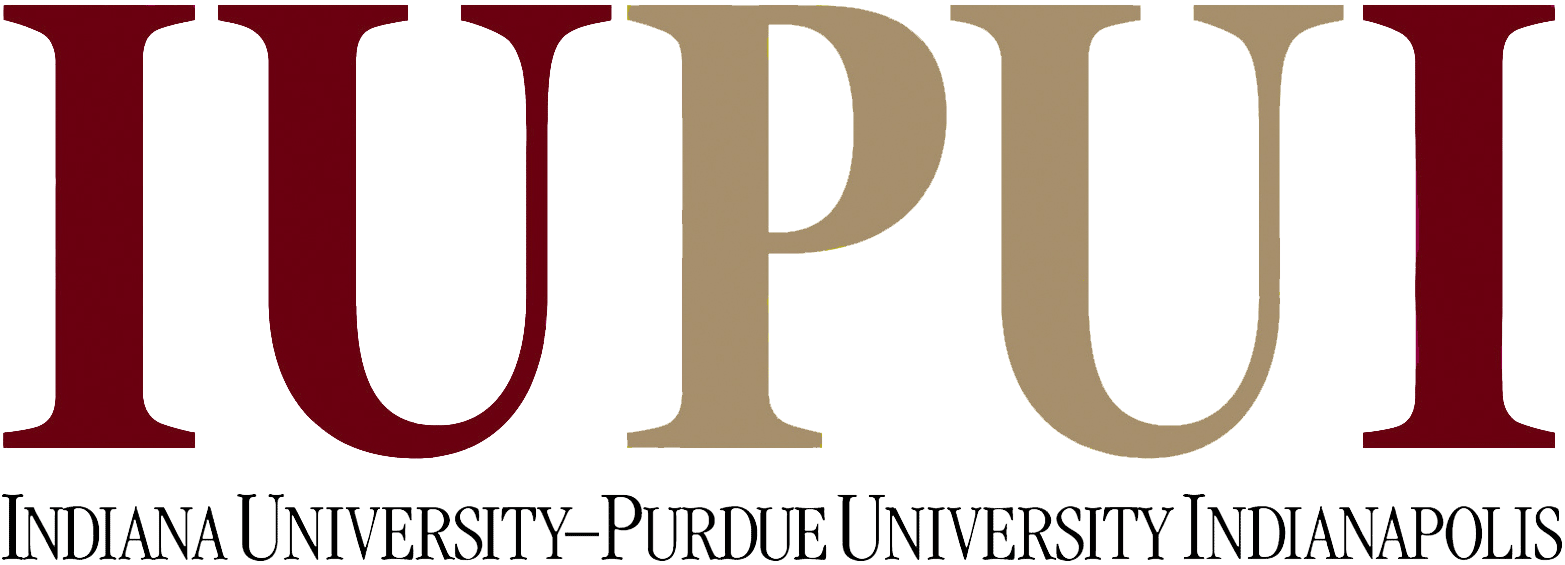
By

**Kyle J Cooper**

*Submitted to class ECE 47100 Spring 2018 at IUPUI*

*In Partial Fulfillment of the Requirements for the degree of*

**Computer Engineering**

****

Engineering, IUPUI

Indianapolis, Indiana

Graduate Spring 2019

**TABLE OF** **CONTENTS**

[LIST OF TABLES iii](#_Toc492993590)

[LIST OF FIGURES i](#_Toc492993591)v

[1. INTRODUCTION 1](#_Toc492993594)

[1.1 INTRO 1](#_Toc492993595)

[2. REQUIREMENT OF PROJECT 1](#_Toc492993608)

[2.1 REQUIREMENT 1](#_Toc492993609)

[3. THE DESIGN DETAILS](#_Toc492993608) 1

[3.1 HARDWARE DESIGN](#_Toc492993609) 1

[3.2 SOFTWARE DESIGN 4](#_Toc492993609)

[4. IMPLEMENTATION 5](#_Toc492993608)

[4.1 PARTS LIST](#_Toc492993609) 6

[4.2 WIRING DIAGRAM](#_Toc492993609) 7

[4.3 SOFTWARE TOOLS](#_Toc492993609) 7

[5. EXPERIMENT RESULTS 8](#_Toc492993608)

[6. DISCUSSION 8](#_Toc492993608)

[7. CONCLUSION 8](#_Toc492993608)

[8. REFERENCES 9](#_Toc492993608)

[9. APPENDICES 10](#_Toc492993608)

[9.1 CODE](#_Toc492993609) 10

[9.2 PROJECT VIDEO](#_Toc492993609) 11

LIST OF TABLES

[4.1 Parts List](#_Toc491256730) 6

[4.3 Software Tools](#_Toc491256730) 7

LIST OF FIGURES

[3.1 PIC18F4515 2](#_Toc491256732)

[3.1 L293D 2](#_Toc491256733)

[3.1 L293D – Pin Functions](#_Toc491256733) 3

[3.2 Flow-Chart 5](file:///S:\Thesis\Templates\Purdue%20University%20Thesis%20Template%20-%20IEEE.docx#_Toc491256735)

[4.2 Wiring Diagram 7](#_Toc491256733)

# INTRODUCTION

## Intro

The purpose of this project is to demonstrate how to control a DC Motor with a PIC18F4515 Microcontroller.

# 2. REQUIREMENT OF THE PROJECT

**2.1 Requirements**

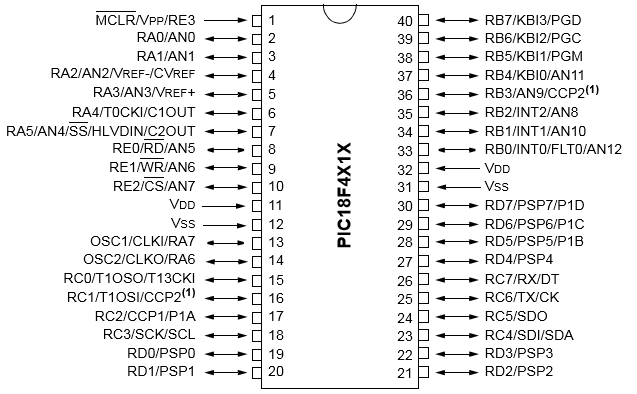
The objective of this project is to get at least two peripherals working on the PIC. The two peripherals I will attempt to get working will be Analog-to-Digital Circuit (ADC) and Pulse-Width-Modulation (PWM).

The PIC will take the signals from the potentiometer using ADC, which will then used to control the duty cycle for a PWM signal. The PWM will be sent to the enable pins of a L293D chip, to control motor speed. The L293D can also control motor direction by reading either 10 or 01 which will be also sent to the PIC and controlled by push buttons attached to the PIC. The DC Motor will also be turned off and on using a push button.

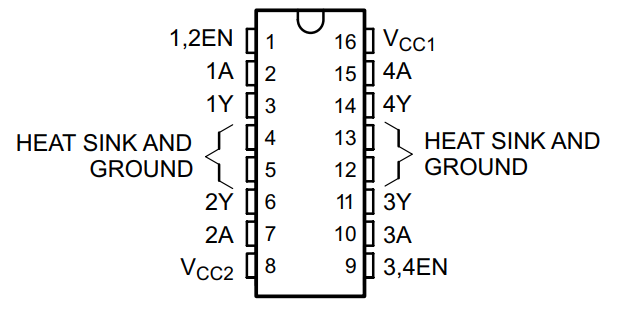
# DESIGN DETAILS

**3.1 Hardware Design**

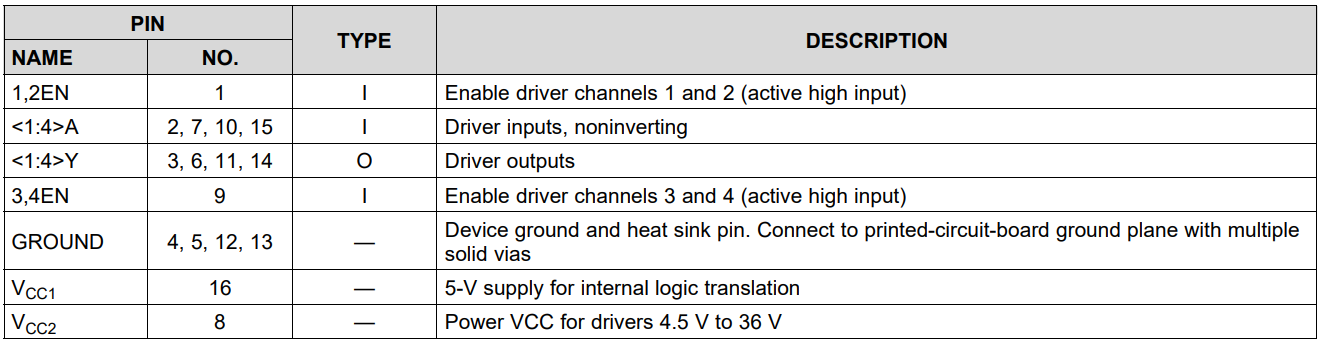
Below are a few figures which show the pin layout for both the PIC18F4515 and the L293D, as well as the pin functions for the L293D.



PIC18F4515



L293D



L293D – Pin Functions

Pins 1, 32, 31, and 38-40 on the PIC18F4515, are used to connect to the PicKit3, so that the PIC can be programmed, debugged, and have power provided to it.

Pin 17 on the PIC is used to send a PWM signal to pin 1 on the L293D, which enables/disables the DC Motor (important for speed control).

Pin 2 on the PIC is setup as an ADC to receive the signal from the potentiometer, which is then used to control the PWM duty cycle.

Pins 33 and 34 on the PIC are used as standard OUTPUT pins, which are connected to pins 2 and 7 on the L293D, which control the DC Motors direction.

Pins 19 and 20 on the PIC are used as standard INPUT pins, which are connected to push buttons

Pin 19 push button is used to toggle motor on and off. Pin 20 is used to toggle motor direction.

Pins 35-37 on the PIC are used as standard OUTPUT pins, connected to LEDS, pin 37 LED is always on if the PIC is on, pin 36-37 LEDS toggle being on or off, one will be off and the other will be on and vice versa depending on motor direction.

Pins 3 and 6 on the L293D are where the DC Motor is connected too.

**3.2 Software Design**

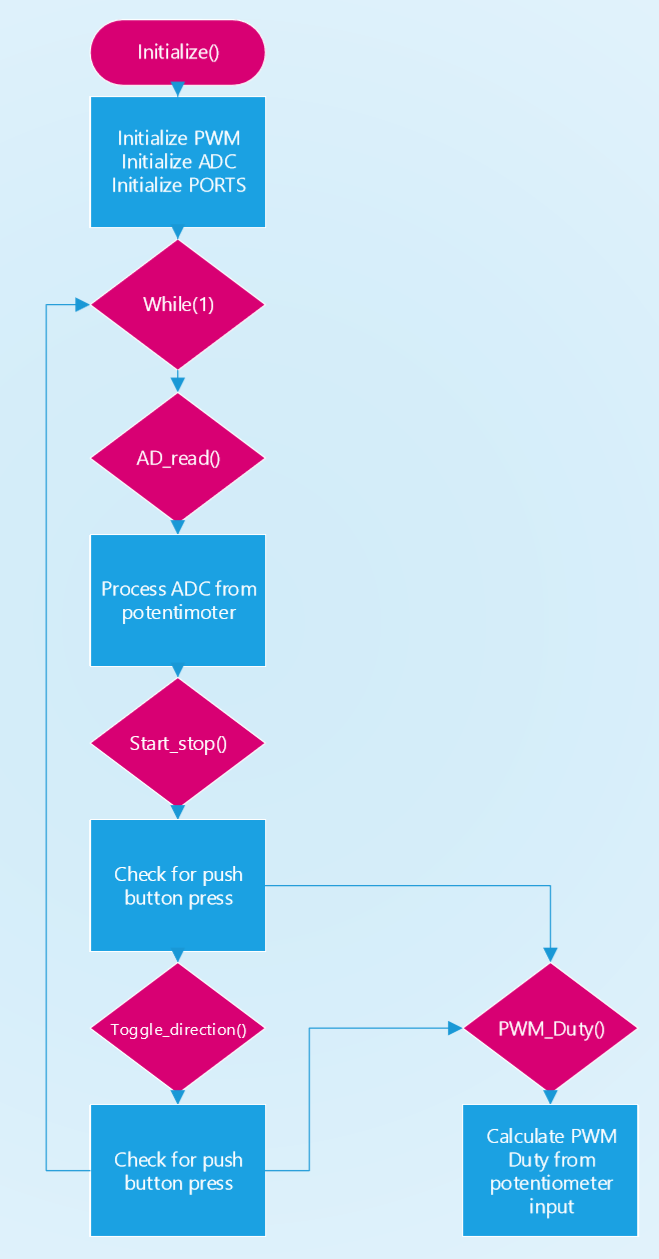
Initialize(), initializes the PWM and ADC, as well as PORTs used for standard IO.

AD\_read(), reads signal from potentiometer on pin 2 of the PIC, and saves result in analog\_reading variable.

Start\_stop(), reads pin 19 on PIC to check for push button press, and toggles on/off flag for motor depending on flags previous value.

Toggle\_direction(), reads pin 20 on PIC to check for push button press, and toggles direction flag for motor depending on flags previous value.

PWM\_Duty(), takes input from analog\_reading, and uses that to calculate PWM duty cycle and sets new PWM duty cycle.



Flow Chart

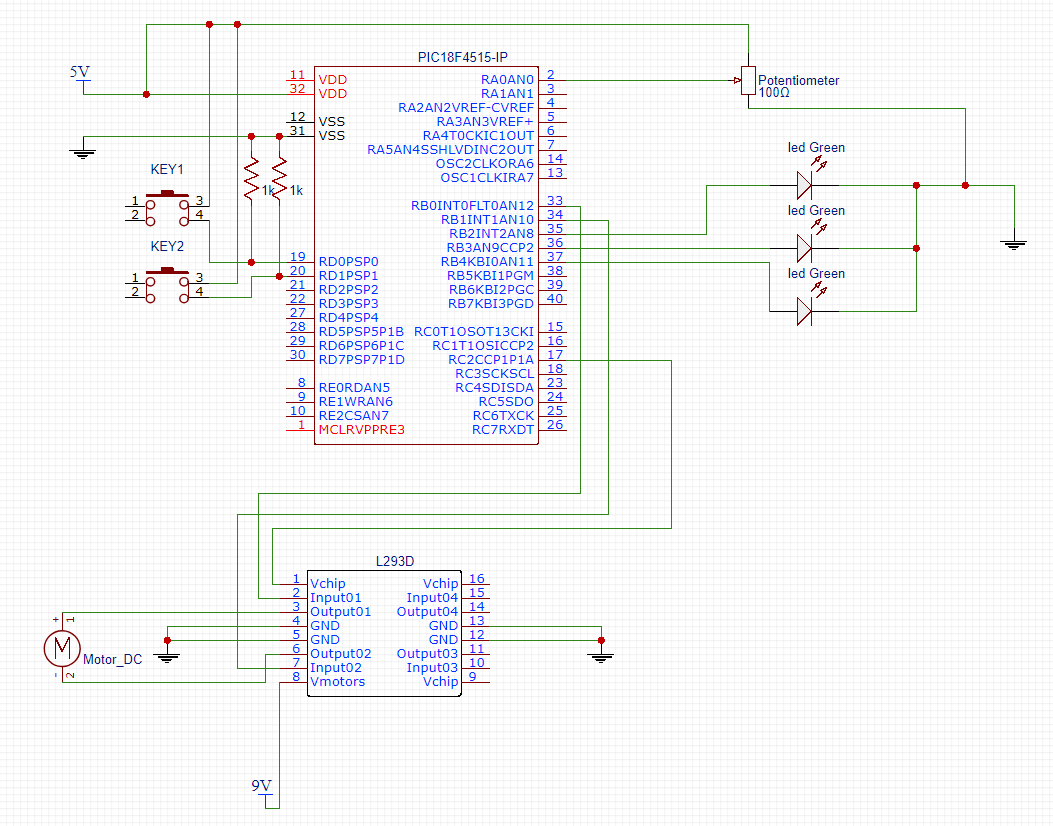
# IMPLEMENTATION

**4.1 Parts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Parts** | **Amount** | **From** | **Cost per unit** |
| PIC18F4515 | 1x | Microchipdirect.com | $4.42 |
| L293D | 1x | Amazon.com | $1 |
| LED lights | 3x | Amazon.com | $0.10 |
| DC Motor | 1x | Sparkfun.com | $1.95 |
| Servo Wires | 35x | Already had | N/A |
| 9V Battery | 1x | Walmart | $2.95 |
| 1.5V D+ Battery | 3x | Walmart | $2.95 |
| PicKit 3 | 1x | Microchipdirect.com | $47.95 |
| 10k Resistor | 2x | Amazon.com | $0.05 |
| Push Button | 2x | Sparkfun.com | $0.5 |
| Potentiometer | 1x | Sparkfun.com | $0.95 |

Parts List

**4.2 Wiring Diagram**

****

Wiring Diagram

**4.3 Software Tools**

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Configuration** |
| MPLAB X IDE | 4.15 | default |

Software tools

# EXPERIMENT RESULTS

After completing the project all project requirements have been met. An analog-to-digital conversion is done on the Potentiometer signal, which is then used to create the duty cycle for the PWM signal. Everything works as expected, DC Motor can be speed up by the Potentiometer. Push Buttons can be used to toggle the DC Motor on and off, as well as toggle the DC Motors direction.

# DISCUSSION

To run my PIC18F4515 without having to rely on the PicKit3 for power I use 3 D+ 1.5v batteries in series to get a total of 4.5V, which is enough to power the project on its own. Since the L293D also allows for the DC Motor to be receive power from a different voltage source than the PIC I use a 9V battery connected to the L293D DC Motor power pin, which increases the speed of the DC Motor significantly.

# CONCLUSION

Overall this project help me understand how microcontrollers work, and was a great learning experience.

# REFERENCES

Microchip PIC18F2X1X/4X1X Manual – Microchip Technology Inc. Retrieved April 18, 2018, from http://ww1.microchip.com/downloads/en/DeviceDoc/39636d.pdf

L293x Quadruple Half-H Drivers Manual – Texas Instruments. Retrieved April 18, 2018, from http://www.ti.com/lit/ds/symlink/l293d.pdf

# APPENDICES

**9.1 Code**

#include <stdio.h>

#include <stdlib.h>

#include "config.h"

#include <xc.h>

#define \_XTAL\_FREQ 20000000

#define TMR2PRESCALE 4

long PWM\_freq = 5000;

int init\_run = 1;

int start\_flag = 0;

int toggle\_flag = 0;

short unsigned int analog\_reading;

void PWM\_Duty(unsigned int duty){

if(duty<1023){

duty = ((float)duty/1023)\*(\_XTAL\_FREQ/(PWM\_freq\*TMR2PRESCALE)); //duty = (((float)duty/1023)\*(1/PWM\_freq)) / ((1/\_XTAL\_FREQ) \* TMR2PRESCALE);

CCP1X = duty & 1; //Store the 1st bit

CCP1Y = duty & 2; //Store the 0th bit

CCPR1L = duty>>2;// Store the remining 8 bit

}

}

void Initialize(){

ADCON0=0x01;

ADCON1=0x00;

TRISB = 0;

TRISD = 1;

LATB = 16;

PR2 = (\_XTAL\_FREQ/(PWM\_freq\*4\*TMR2PRESCALE)) - 1; //PR2 formulae using Datasheet

CCP1M3 = 1; CCP1M2 = 1; //configure the CCP1 module

T2CKPS0 = 1;T2CKPS1 = 0; TMR2ON = 1; //configure the timer

TRISC2 = 0; // make port pin on C as output

}

void AD\_read(){

ADCON0bits.GO\_DONE=1; //do A/D measurement

while(ADCON0bits.GO\_DONE==1);

analog\_reading= ADRESL + (ADRESH\*256);

analog\_reading=analog\_reading/256;

return analog\_reading;

}

void start\_stop(){

if(PORTDbits.RD0){ //check stop/start button

if(init\_run){

init\_run = 0;

LATB = 21;

}

if(start\_flag){ //toggle

start\_flag = 0;

}else{

start\_flag = 1;

}

while(PORTDbits.RD0); //de-bounce

}

if(start\_flag){ //run

PWM\_Duty(analog\_reading\*4);

}else{ //stop

PWM\_Duty(0);

}

}

void toggle\_direction(){ //toggle motor direction

if(PORTDbits.RD1){ //if push button pressed

if(toggle\_flag){ //toggle flag check

toggle\_flag = 0;

PWM\_Duty(0); //turn off motor before reversing direction

while(PORTDbits.RD1); //de-bounce

LATB = 21;

}else{

toggle\_flag = 1;

PWM\_Duty(0); //turn off motor before reversing direction

while(PORTDbits.RD1); //de-bounce

LATB = 26;

}

}

}

void main(){

Initialize(); //initialize everything

while(1){ //main loop

AD\_read(); //do analog to digital read

start\_stop(); //check push button to start/stop motor

toggle\_direction(); //check push button to toggle motor direction

}

};

**9.2 Project Video**

