Final Project

LabVIEW Programming

1741344

Benoit Moreau

16 May 2019

Contents

[Table of Figures 3](#_Toc8911710)

[Table of Tables 3](#_Toc8911711)

[Introduction 4](#_Toc8911712)

[Theory 4](#_Toc8911713)

[H-Bridge 5](#_Toc8911714)

[Implementation 5](#_Toc8911715)

[Nucleo code explanation 6](#_Toc8911716)

[LabVIEW Code Explanation 7](#_Toc8911717)

[Part 1: Direction Control 7](#_Toc8911718)

[Part 2: Motor control 8](#_Toc8911719)

[Part 3: Measurement 8](#_Toc8911720)

[Part 4: Front Panel 9](#_Toc8911721)

[Testing and Results 9](#_Toc8911722)

[Discussion 12](#_Toc8911723)

[Conclusion 12](#_Toc8911724)

[Annex 1: LabVIEW code 13](#_Toc8911725)

[Annex 2: Nucleo Code 14](#_Toc8911726)

[Annex 3: Testing Procedure 15](#_Toc8911727)

[Report evaluation 16](#_Toc8911728)

# Table of Figures

[figure 1 Data Flow 4](#_Toc8900475)

[figure 2 Project set-up 5](#_Toc8900476)

[figure 3 Nucleo Pinout 6](#_Toc8900477)

[figure 4 H-bridge 6](#_Toc8900478)

[figure 5 Nucleo Code structure 6](file:///C:\Users\benap\Downloads\D07_Report%20(1).docx#_Toc8900479)

[figure 6 Direction 7](#_Toc8900480)

[figure 7 Motor control code 8](#_Toc8900481)

[figure 8 Measurement code 8](#_Toc8900482)

[figure 9 LabVIEW front Panel 9](#_Toc8900483)

[figure 10 PWM graphs 10](#_Toc8900484)

[figure 11 LabVIEW Code 13](#_Toc8900485)

[figure 12 Nucleo Code 14](#_Toc8900486)

[figure 13 15](#_Toc8900487)

# Table of Tables

[Table 1 H-Bridge Truth table 5](#_Toc8899994)

[Table 2 H-Bridge truth Table 5](#_Toc8899995)

[Table 3 Materials Used 5](#_Toc8899996)

[Table 4 Nucleo Commands 7](#_Toc8899997)

[Table 5 Test Results 9](#_Toc8899998)

[Table 6 10](#_Toc8899999)

[Table 7 Motor Speed Test Procedure 15](#_Toc8900000)

[Table 8 Testing Motor Direction 16](#_Toc8900001)

# Introduction

The primary goal of this project was to control and monitor a Direct current (DC) motor using the LabVIEW programming language. This goal of controlling the DC motor will be accomplished by using a Nucleo microcontroller along with an h-bridge and a DC power supply. The secondary requirement of this is laboratory is to monitor certain elements of the DC motor circuit such as voltage, current and duty cycle. This goal is going to be accomplished through the use of a MyDAQ, an oscilloscope and a Multimeter.

# Theory

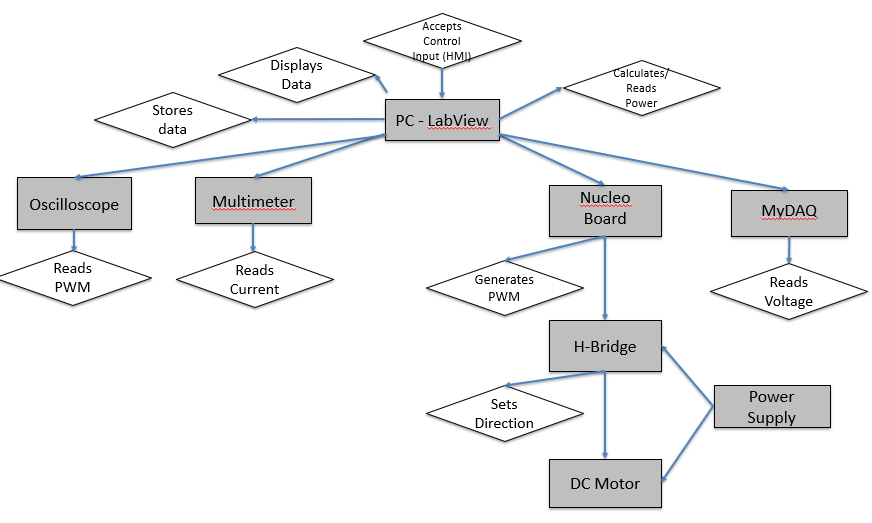


figure 1 Data Flow

The project starts with the personal computer (PC) with LabVIEW on it. This device serves two functions. The first function of this device is to receive data from the measurement devices, display the data on the LabVIEW front panel and store the data into an Excel spreadsheet. The second function of this device is to take human inputs for the speed and direction of the motor, transform them into code that can be read by the Nucleo and send that converted code to the microcontroller.

The Nucleo then receives this code and sends out the appropriate digital and analog signals to the H-bridge. The signals that the Nucleo sends to the H-Bridge are: a PWM signal for the speed of the motor and 2 digital signals for the direction of the motor

The h-bridge receives these signals and then outputs a corresponding voltage to the motor.

At this point, the oscilloscope multimeter and myDAQ will have recorded their measurements and will have sent them off to the PC with LabVIEW, thereby restarting this process. Nucleo Board

## H-Bridge

Table 1 H-Bridge Truth table

An H-bridge is a type of motor driver that uses 4 transistors in an H pattern, with the motor forming the center bridge of the H. These 4 transistors can be switched on and off depending on the inputs given to it. The h-bridge used in this project uses pins labeled IN1A and IN0A in order to control direction and the table below will provide a list of all possible direction states:

Table 2 H-Bridge truth Table

|  |  |  |
| --- | --- | --- |
| IN0A | IN1A | OUTPUT |
| 1 | 0 | Turns clockwise |
| 0 | 1 | Turns Counterclockwise |
| 1 | 1 | Brake |
| 0 | 0 | Brake |

# Implementation

Materials used:

* 1 H-bridge
* 1 myDAQ
* 1 Multimeter
* 1 Oscilloscope
* 1 Nucleo F411RE microcontroller
* 1 PC with LabVIEW installed
* 1 DC power supply
* 1 DC Motor

Table 3 Materials Used

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | **Could be useful to do what (mark with an x)** | | |
|  | **Control** | **Monitor** | **Store** |
| H-Bridge | X |  |  |
| MyDAQ |  | X |  |
| NucleoBoard | X |  |  |
| Multimeter |  | X |  |
| Power Supply | X |  |  |
| Oscilloscope |  | X |  |
| PC with LabView | X | X | X |

figure 2 Project set-up

For a description of how the circuit was set-up, please refer to Annex 3: Test Procedure.



GND

To 3.3/5V pin

To Oscilloscope

To PWMA

To IN1A

To IN0A

figure 3 Nucleo Pinout

Figure 3 shows the pinout of the Nucleo board. The pins with blue arrows go into the H-bridge connectors in figure 4 that share the label, i.e.: IN1A goes to connector IN1A.

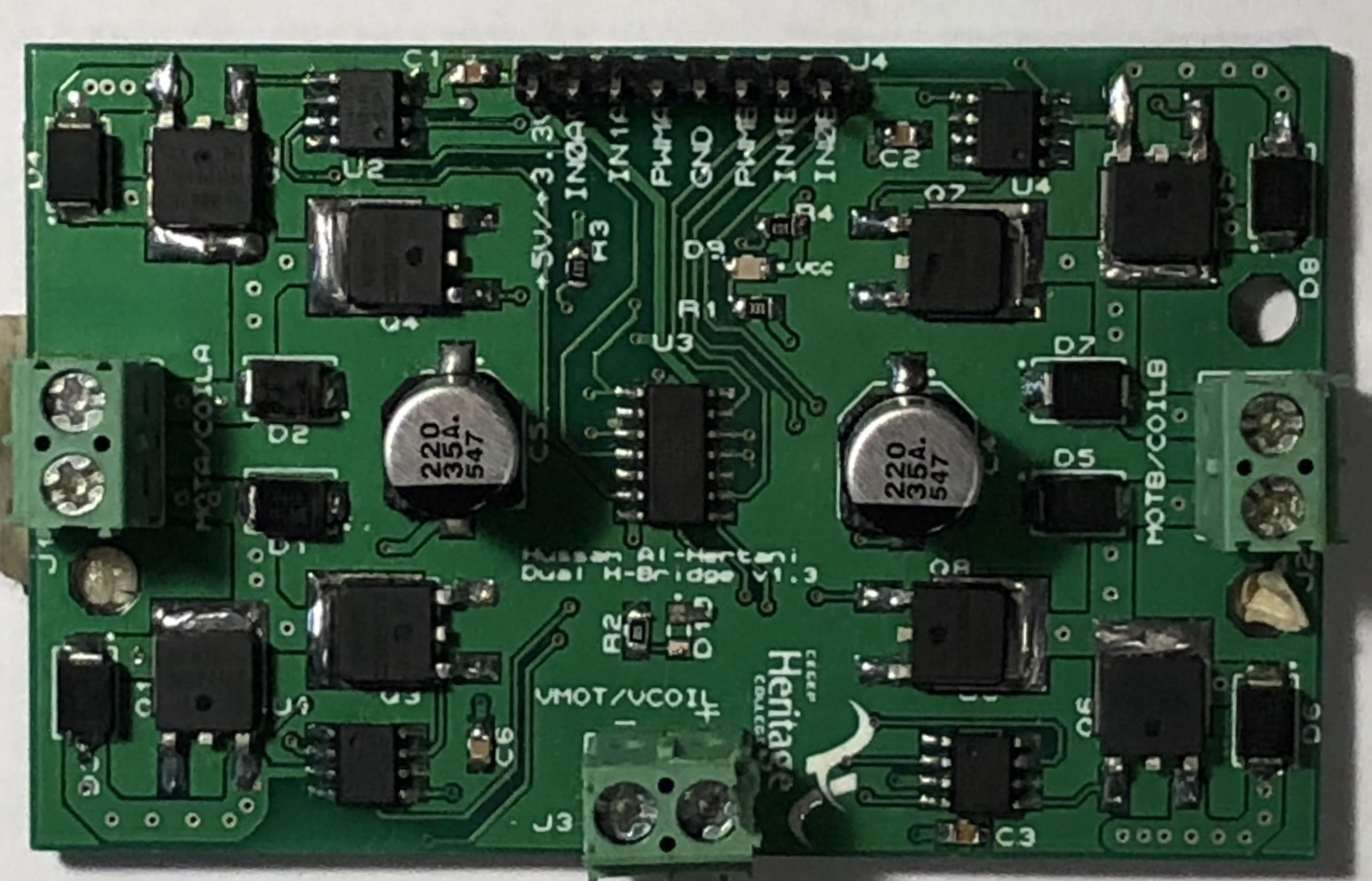


figure H-bridge

Figure 4 is a picture of the h-bridge used in this project.

## Nucleo code explanation

The code that LabVIEW transmits to the Nucleo is made up of 4 parts.

Tail

Command

Head

80XXYY81

Command Argument

figure 5 Nucleo Code structure

The purpose of the “head” block is to tell the nucleo to start reading the code

The purpose of the “command” is to tell the nucleo what to do. (For a complete list of commands available in this project, please refer to table 4.)

The purpose of the “command argument is to assign a value to the command.

The purpose of the “tail” is to tell the nucleo to stop reading the code/ close the session.

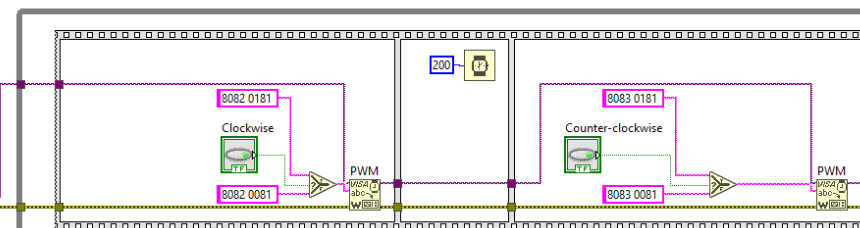
Table 4 Nucleo Commands

|  |  |  |
| --- | --- | --- |
| Command | Purpose | Command Arguments |
| 82 | Change the state of pin IN0A | 00 (off) OR 01 (on) |
| 83 | Change the state of pin IN1A | 00 (off) OR 01 (on) |
| 86 | Change the duty cycle of PWMA | 00 (0%) to FF (100%) |

## LabVIEW Code Explanation

The explanation of the LabVIEW code will be broken up into four parts: direction control, speed Control measurement and front panel. To view the complete code, please refer to annex 1: LabVIEW code

### Part 1: Direction Control

* 

C

B

A

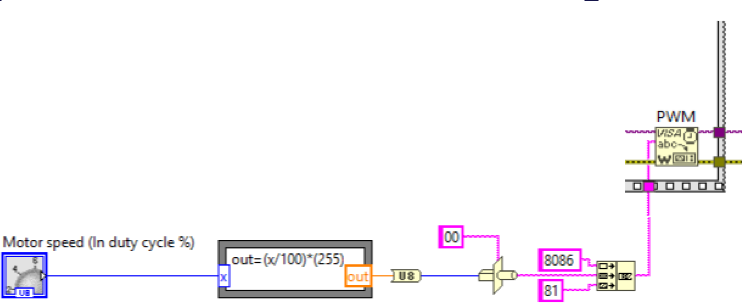
figure 6 Direction

Label “A” shows the first half of the code required to determine the direction of the motor. If the block labeled “Clockwise” returns true, it will send 80820181 to be written to the Nucleo. If that block returns false, it will send 80820081 to the Visa write block (block labeled PWM in figure 3).

Label B shows a 200-millisecond delay. This delay gives the Nucleo enough time receive the code and store it in its buffer

Label C shows the second part of the code required to determine the direction of the motor. if the button labeled “clockwise” in figure 3. If the block labeled “Clockwise” returns true, it will send 80830181 to be written to the Nucleo. If that block returns false, it will send 80830081 to the Visa write block (block labeled PWM in figure 3).

## Part 2: Motor control



E

D

C

B

A

figure 7 Motor control code

Part A in figure 4 is the user input for the duty cycle. This input is a knob that goes from 0 to 100

Part B in figure 4 changes the user input from a number between 0 & 100 to a number from 0 to 255 in order to facilitate the conversion to hexadecimal.

Part C in figure 4 converts the converted number into a hexadecimal string.

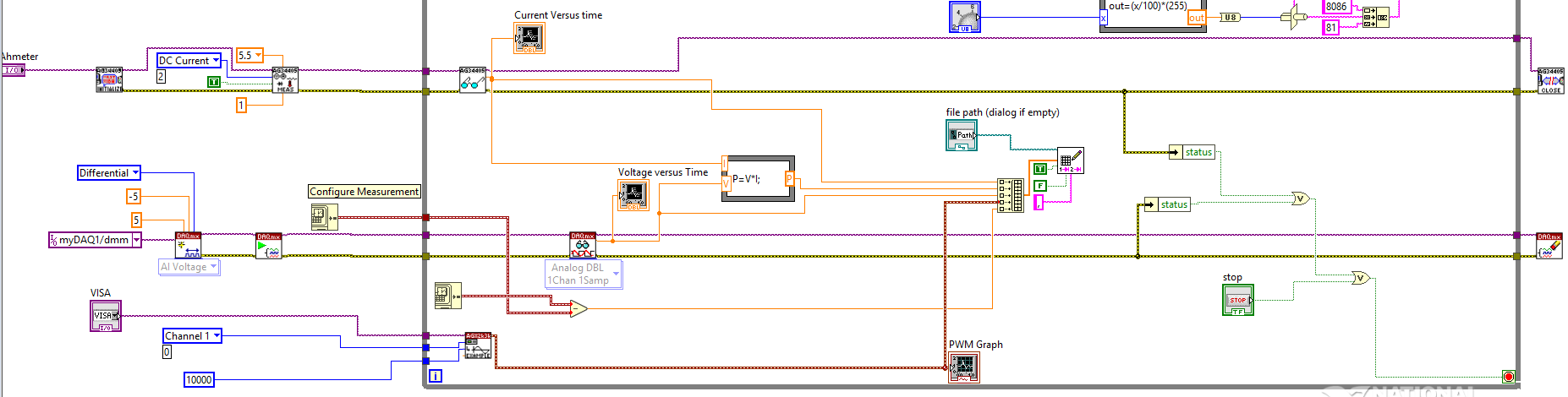
Part D in figure 4 concatenates the string with the other 2 strings required by the microcontroller.

Part E in figure 4 sends the concatenated string to the microcontroller.

## Part 3: Measurement

I

A



H

G

F

E

D

C

B

figure Measurement code

Part A Specifies all the device specific parameters, such as device location, timeout… This part also sends signals to the different devices to tell them to open a session.

Part B reads from the ammeter and sends the data to a waveform chart.

Part C reads from the myDAQ and sends the data to a wave form chart.

Part D calculates the sample time

Part E reads the data from the oscilloscope and sends it to waveform graph.

Part F calculates power.

Part G gathers all the different data (time, current, voltage, power, PWM), places them into an array. Once they are in the array, they are sent to an excel spreadsheeted.

Part H is the error check. If any of the devices or the user return an error, the program will stop running.

Part I tells the different devices to close the session.

## Part 4: Front Panel



H

G

F

E

C

**B**

A

figure 9 LabVIEW front Panel

The LabVIEW front panel is the part of the program that the end user interacts with. This panel contains all the user controls such as direction and speed control of the motor. This panel also displays all the data being gathered by all the different measurement equipment.

Part A is the Waveform chart of the PWM signal being output by the Nucleo.

Part B is a chart that plots the h-bridge input voltage as it relates to time.

Part C is a chart that plots the motors current draw as it relates to time.

Part E shows the two buttons that control the motor direction. When only one of them is pressed, the motor will rotate in the corresponding direction. When either neither of them or both of them are pressed, the motor will stop.

Part F shows the box in which the user specifies a path for where to save the excel spreadsheet.

Part G shows the device inputs the user must specify for the different connected devices such as the oscilloscope or multimeter.

Part H shows the motor control knob. This know controls the duty-cycle of the PWM and allows the user to change duty-cycle anywhere from 0 % all the way to 100%.

# Testing and Results

This section will only discuss the results of the testing process. To see what the testing process was, please refer to annex 3: the testing process.

Table 5 Test Results

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Protocol** |  |  |  |  |
|  |  |  |  |  |
| **Control** | **Result** | |  |  |
| Motor turns | 1 | / | 1 |  |
| Speed can be controlled | 1 | / | 1 |  |
| Direction can be controlled | 1 | / | 1 |  |
| **Monitoring** | **Result** | |  |  |
| Supply Voltage | 1 | / | 1 |  |
| Supply Current | 1 | / | 1 |  |
| Supply Power | 1 | / | 1 |  |
| Generated PWM signal | 1 | / | 1 |  |
| PWM signal sent to the motor | 1 | / | 1 |  |
| **Data stored to excel** | **Result** | |  |  |
| Supply Voltage | 1 | / | 1 |  |
| Supply Current | 1 | / | 1 |  |
| Supply Power | 1 | / | 1 |  |
| Generated PWM signal | 0 | / | 1 |  |

Table 5 demonstrates that nearly every component was working as intended. The only part of the circuit that did not function correctly during testing was exporting the PWM signal to excel.

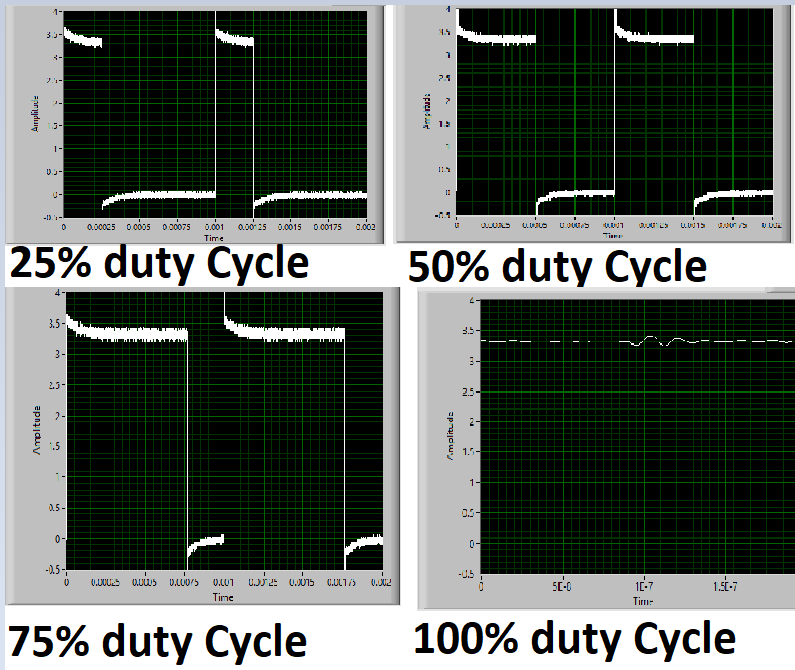


figure 10 PWM graphs

Figure 9 demonstrates the capability of the program to display the correct PWM signal. The signals shown in figure 7 are shown to be correct as the wave’s peak voltages matches the output of the nucleo board (3.3V)

The current and voltage graphs where also proven to work, however no screenshots of the graphs where taken.

Table 6 Excel data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Current | Power | Voltage | PWM | Time |
| 0.064 | 0.359 | 5.602 | 0.032 | 1 |
| 0.062 | 0.279 | 4.523 | -0.008 | 3.677 |
| 0.028 | 0.166 | 6.015 | -0.008 | 6.394 |

Table 6 is a small snippet of data from the excel file that this program uses to store the data captured. This table demonstrates that this program is indeed capable of storing data relating to the voltage, current and power of the circuit as it stores results that match what would be expected. This table also proves that it is unable to store PWM values as the values it has are complete nonsense.

Over the course of testing, a couple trends where noticed:

* The motor speed increased in a linear fashion as the duty cycle was increased linearly
* The current increased in a linear fashion as the duty cycle was increased linearly
* The voltage remained constant regardless of the duty cycle

During the course of testing, the results gained matched those described in the test plan.

# Discussion

The biggest thing that went wrong during this project was PWM data capture. I was unable to correctly gather the data. This might be due to the limited number of samples that the program takes (1 sample ever 2.7 seconds). In future iterations of this lab, I would probably try to increase the number of samples taken per iteration.

Next time that I have to do a similar problem, I’d try to make the program a lot more user friendly. This is to say that I would make the front panel a lot simpler to understand to the average person and reduce the number of spots that could cause an unexpected user error. This project taught me a lot about using LabVIEW to communicate with external devices, especially those whose structures or timing conflict with one another. This project also presented a new challenge as I had to contend not only with LabVIEW but also all the hardware that came along. Trying to figure out if the problem was caused by LabVIEW or some piece of the hardware was difficult to get used to initially. Even though the project was full of surprises and new challenges, the results where completely unsurprising, as the project did exactly what it was supposed to do.

# Conclusion

In conclusion, the challenge of this project was a build a LabVIEW program that would be able to control a DC motor and monitor certain aspects of it such as the input voltage, current draw and PWM duty cycle. This project should be considered a near complete success as it managed to successfully reach nearly all the goals set out initially. This is to say that the program and related circuits where able to control the speed and direction of a DC motor as well monitor and store data relating to the motor input voltage, motor current draw and motor power. The only thing that could considered a “failure” in this project would be the fact that the program is unable to save the PWM waveform graphs in excel. In future iterations of this lab, it would be advisable to make the LabVIEW program as user friendly as possible as well as make use of the signal generator to generate the PWM rather than using the Nucleo board.

## Annex 1: LabVIEW code

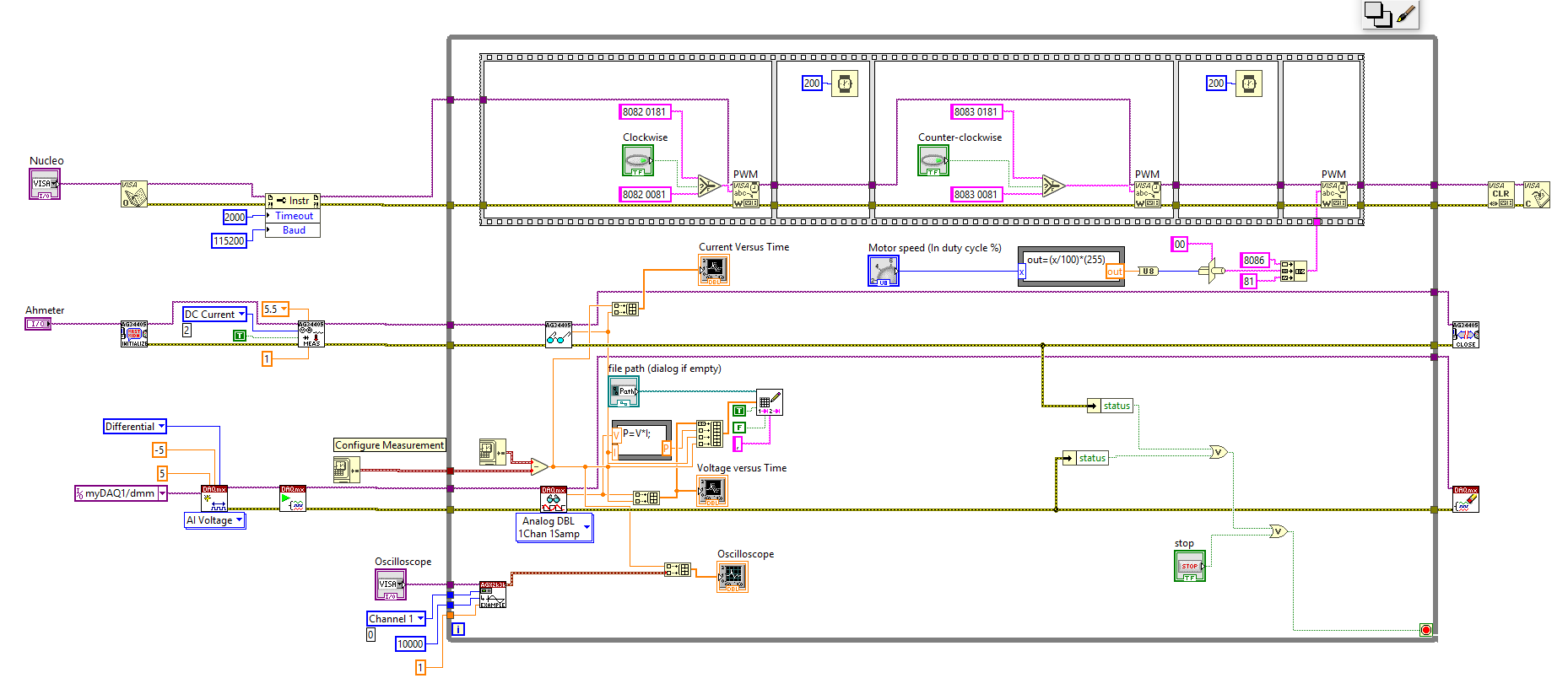


figure 11 LabVIEW Code

## Annex 2: Nucleo Code



figure 12 Nucleo Code

## Annex 3: Testing Procedure

USB

cables

Cables

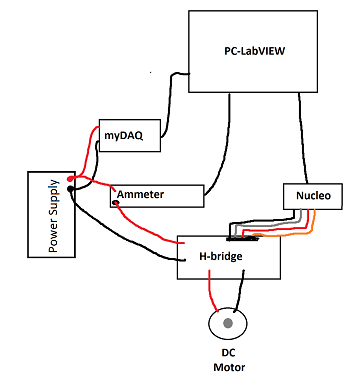


figure 13 Circuit diagram

Table 7 Motor Speed Test Procedure

|  |  |  |
| --- | --- | --- |
| Testing the Motor Speed. There is a slight delay of approximately 1.5seconds before any change is effectuated | | |
| **Step** | **Procedure** | **Result** |
| **1** | Select Clockwise & set duty cycle to 0% | No Change |
|  | Wait 3 seconds |  |
| **2** | Change duty cycle to 25% | The motor should start turning and the current should increase |
|  | Wait 3 seconds |  |
| **3** | Change duty cycle to 50% | The motor should spin slightly faster and the current should increase |
|  | Wait 3 seconds |  |
| **4** | Change duty cycle to 75% | The motor should spin slightly faster and the current should increase |
|  | Wait 3 seconds |  |
| **5** | Change duty cycle to 100% | The motor should spin slightly faster and the current should increase |
|  | Wait 3 seconds |  |
| **6** | Reduce duty cycle to 75% | \*The current and motor speed should slightly decrease |
|  | Wait 3 seconds |  |
| **7** | Reduce duty cycle to 50% | \*The current and motor speed should slightly decrease |
|  | Wait 3 seconds |  |
| **8** | Reduce duty cycle to 25 % | \*The current and motor speed should slightly decrease |
|  | Wait 3 seconds |  |
| **9** | Reduce duty cycle to 0% | \*The current decreased and the motor stopped |
|  | Wait 3 seconds |  |
| **10** | Deselect clockwise | No Change |
|  | Repeat for counter Clockwise rotation |  |
|  |  | **NOTE: \* means after a slight delay** |

Table 8 Testing Motor Direction

|  |  |  |
| --- | --- | --- |
| Testing Motor direction | | |
| **Step** | **Procedure** | **Result** |
| 1 | Set duty cycle to 50% | \*Current increases but no movement speed occurs |
|  | Wait 3 seconds |  |
| 2 | Toggle clockwise on | \*Motor should start spinning clockwise |
|  | Wait 3 seconds |  |
| 3 | Toggle counter-clockwise on | \*Motor should stop and be hard to spin manually |
|  | Wait 3 seconds |  |
| 4 | Toggle clockwise off | \*Motor should spin counter-clockwise |
|  | Wait 3 seconds |  |
| 5 | Toggle counter-clockwise off | \*Motor should stop and be easy to spin manual |

## Report evaluation

The main goal of the report-evaluation is to have you check yourself if you miss anything important. If you realize that you lose points in a section (and you have time), try to fix the issue before proceeding.

Highlight with YELLOW color, the statement that describes your team’s performance for each element.

|  |  |  |  |
| --- | --- | --- | --- |
| **Element** | **Achievement Level** | | |
| **Perfect** | **Good** | **Poor** |
| Identification | 0.5: Date, Class, Title, Student names and ID's present |  | 0.0: One of the above items missing or wrong. |
| Abstract | 1.0: Summarizes what you did and what you found. Short, fluent and without major grammar mistakes. | 0.5: Summarizes what you did and what you found. Too long, difficult to read, not fluent or many grammar mistakes. | 0.0: Missing, incomplete, hard to read, confusing or not aligned with the rest of the project. |
| Introduction | 2.0 : Covers all of the requested elements (Goal of the project, materials used, data-flow). Is clear and without major grammar mistakes. | 1.0: Covers all of the requested elements. Not always clear or many grammar mistakes. | 0.0: One or more requested elements missing. Hard to read, confusing or not aligned with the rest of the project. |
| Body | 2.0: Covers all of the requested elements (Hardware setup, Control Interface, Test results). Is clear (good picture quality) and without major grammar mistakes. | 1.0: Covers all of the requested elements. Not always clear (bad picture quality) or many grammar mistakes. | 0.0: Missing, incomplete, confusing or not aligned with the rest of the project. |
| Conclusion | 2.0: Covers all requested elements. Is clear, aligned with the rest of the project and without major grammar mistakes. | 1.0: Covers all requested elements. Is not always clear, but aligned with the rest of the project. | 0.0: Missing, incomplete, hard to read, confusing or not aligned with the rest of the project. |
| References | 1.0: All references given. (Or “no external references” written if there are none.) References are formatted in the requested format. | 0.5: All references given. (Or “no external references” written if there are none.) References are formatted in ANOTHER format, but the format is kept consistent for all references. | 0.0 Not all references given or the reference format is changing throughout the paper. |
| Annexes | 0.5: Covers all requested elements. Is clear and readable. |  | 0.0: Not clear (screenshot quality) or not readable (source code), Missing, incomplete, confusing or not aligned with the rest of the project. |
| Report-evaluation | 1.0: Present and appropriate (matching the teachers evaluation for more than 6 elements) | 0.5: Present but unrealistic (matching the teachers evaluation in less than 6 elements) | 0.0 Missing or present without self-evaluation (highlighting). |

Total amount of points: / 10