Implementation of Position Control using Electronic Speed Controller (ESC) for Brushless Direct Current (BLDC) Motors

Abstract

This document explores the implementation of position control in BLDC motors using an ESC. It will examine all components of the project, including motor control theory, software design, electronics hardware design of the ESC and mechanical design of BLDC motors at a technical level. It shows that it is possible to implement position control on BLDC motors, however much work is needed to improve the performance of the control strategy.

Applications for this project will then be discussed and further improvements will be proposed.

Contents

Introduction

Business opportunity or challenge

Traditionally DC motors are used for applications that requires a mechanical energy output, such as a windscreen wiper system or an electric golf buggy. In the past decade, DC motors are being phased out and replaced by BLDC motors as they have many performance advantages but only minor drawbacks. In low-cost appliances, DC motors are still used to keep manufacturing costs low. In transportation, most hybrid and electric vehicles uses some form of BLDC motor. Most HVAC systems uses BLDC motors to increase efficiency since a lot of them are always powered. In car manufacturing, most robot arm uses BLDC motors.

In the motor sports industry, motors and actuators are widely use in different subsystems of the car, such as throttle motor, alternator, wastegate actuator. There is a desire to convert all these motors from traditional DC motors to BLDC motors for higher performance and lower cost. There is also a desire to use additional motors to help with manual tasks such as automatic windows and automatic closing doors.

Technical Challenge

The commutation and mechanical design of a DC motor is trivial compared to BLDC motors as it requires complex control algorithms. There are two types of control, sensored and sensorless of which both has many different control techniques depending on the application. This paper will only explore sensored position control of BLDC motors as it is not possible to implement position control without a sensor.

This paper will be broken down into 4 main sections, the physical construction and operation of BLDC motors, the electronic hardware to drive BLDC motors, the position control strategy implemented in software and the PC front end software. Combining these 4 components allows us to use BLDC motors in many applications, such as replacing a DC electronic throttle assembly (Figure 1 - Typical Electronic Throttle Body [10]) .

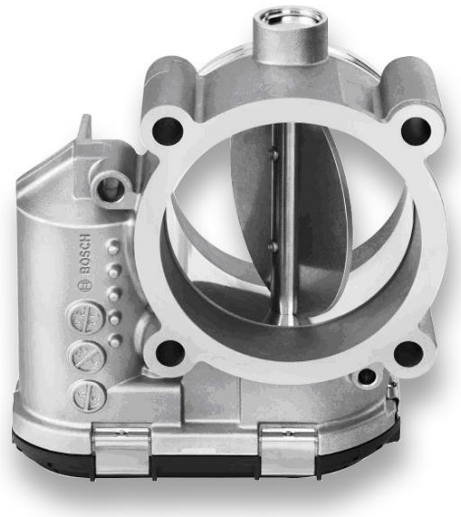

Figure 1 - Typical Electronic Throttle Body [10]

Technology Background

BLDC Advantages

BLDC motor uses electronics to commutate instead of brushes. [1] Brushes tend to burn out due to sparking which causes lots of unwanted electrical noises. Eliminating brushes increases the lifespan of BLDC motors exponentially as replacing brushes are no longer necessary. This would lead to lower cost in the long term. In terms of performance, BLDC motors draws very little current when under no load thanks to its almost frictionless mechanical design. Using complex software control techniques, BLDC motors can provide instant maximum torque at all points of one mechanical revolution, while torque in DC motors varies depending on the position of the rotor, as this is limited by the mechanical design. Combining all these advantages gives a much more efficient, reliable, compact, lower power consumption way of converting electrical energy to mechanical energy than DC motors. Since the development of electric vehicles there is a desire to increase efficiency of motors to increase driving range.

BLDC Disadvantages

[2] Although the construction of BLDC motors requires a smaller number of parts, it is significantly more expensive than a DC motor. This is largely because of the additional expense of an ESC. [3] Vibrations can also be observed when a BLDC motor is running at low speed, the frequency of this vibration can sometimes match the natural frequency of the motor, causing a resonance phenomenon which may contribute to additional noise. A sensor is also required in most cases for smooth commutation which increases production cost.

How does BLDC work

A brushed DC motor works by having permeant magnets on the stator and an armature as a rotor. The armature is an electromagnet with brushes which flips the magnetic field every 180 degrees so the motor can spin for 360 degrees.

There are generally two types of BLDC motor construction, inrunner and outrunner. [4] An outrunner has permeant magnets placed on the rotor of the motor, which is also the outer casing, the whole outer casing rotates when the motor is spinning, the internal stator windings stay in position. On the other hand, an inrunner has its windings placed on the casing of the motor while the rotor contains permanent magnets. The casing does not rotate, only the motor shaft. There are only minor performance differences between two configurations, and the operating principle is the same. All diagrams and theories discussed in this paper will be based on an outrunner, but it can be applied to an inrunner as well.

The basics of a BLDC motor is based on electromagnets. [5] When a coil of wire has an electrical current flowing through it, a magnetic field is created due to the Biot-Savart law (Figure 2 - Magnetic Field of Electromagnet).

Biot-Savart law: $B = \frac{\mu_0 NI}{2R}$

B = Magnetic Field Intensity

u_0 = Permeability of Free Space

N = Number of Turns

I = Current Intensity

R = Radius

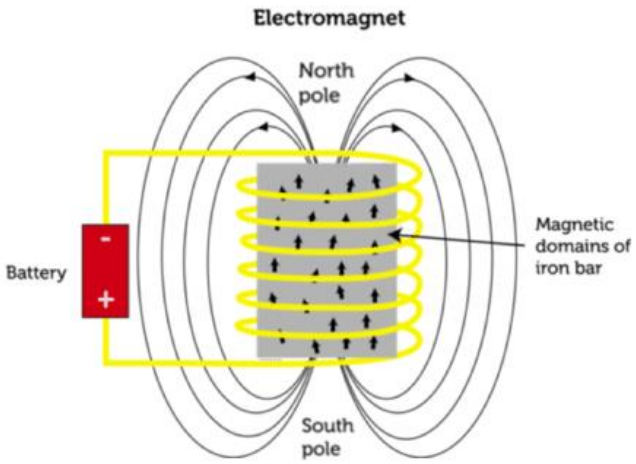

Figure 2 - Magnetic Field of Electromagnet

The magnetic field produced by the coil interacts with the permanent magnets which leads to an attractive or repulsive force depending on the polarity. (Figure 3 - Simplified construction of BLDC motor) When the coil is energized a magnetic field is created which interacts with the permanent magnet causing the rotor to spin. A BLDC motor has multiple coils and permanent magnets for smoother commutation.

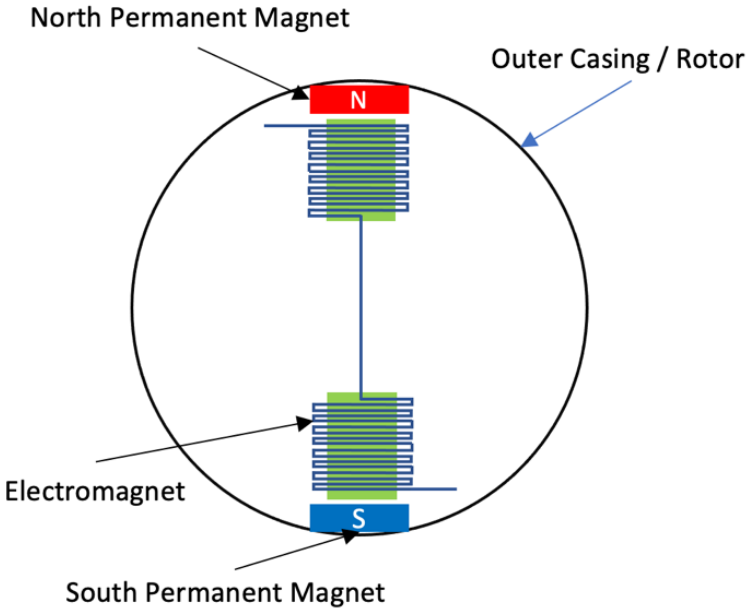

Figure 3 - Simplified construction of BLDC motor

To understand how to commutate a BLDC motor, we can look at the following simplified diagram (Figure 4 - BLDC Motor Commutation Sequence). In each of the commutation sequence, 1 set of coils are energized, and the other coils are left floating to create the correct magnetic field for the rotor to spin. There are 3 phases which are labelled A, B, C, these coils are connected to an ESC allowing the software to energize the coils when required. We will discuss how this is electronically achieved in the next section. The polarity of the energized coil is labelled with + and -. Non labelled coils are left floating. The purple arrows show the attractive forces due to the interaction between the permeant magnetic field and the electromagnetic field. These forces allow the rotor to rotate in the direction as indicated with the yellow arrows. As you can see in

the diagram, 6 commutation steps are required for one mechanical revolution as there are 6 pole pairs. However, this is not always the case. We will further explore why this is in the next section.

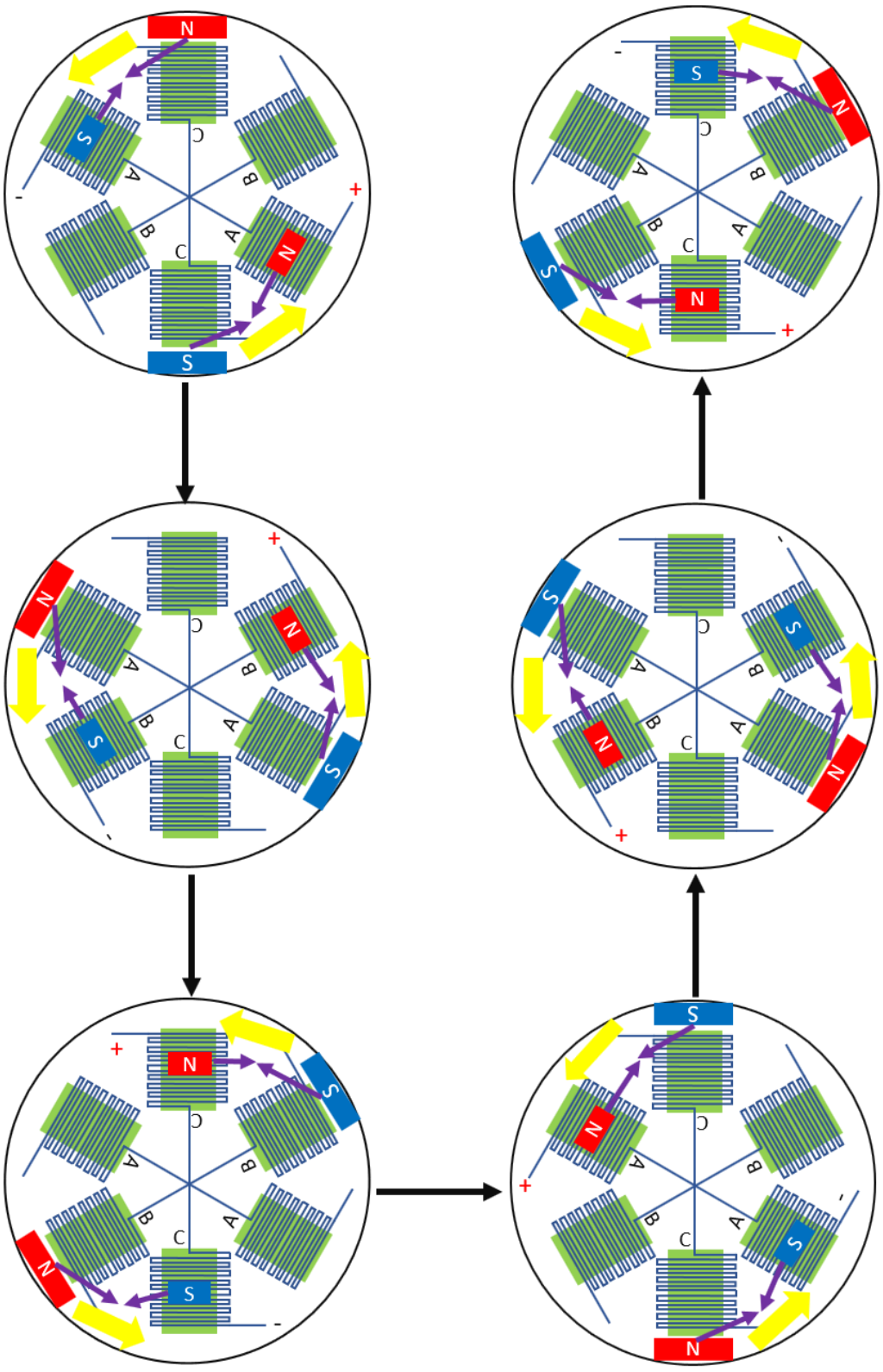

Figure 4 - BLDC Motor Commutation Sequence

To further improve the torque of the motor, rather than only energizing one coil, 2 coils can be energized at the same time with different polarity. (Figure 5 - Dual Coil Energize) Each permanent magnet is interacting

with two magnetic fields produced by two different coils. Black arrows represent repulsive force due to same polarity and purple arrows represents attractive force due to opposite polarity.

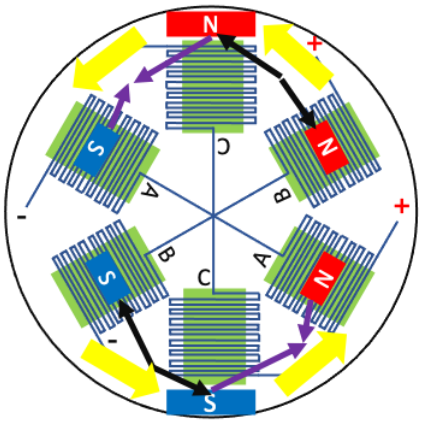

Figure 5 - Dual Coil Energize

To achieve a greater degree of accuracy for position control applications we can also adjust the voltage hence adjust the attractive force of each coil to allow the rotor to stay in between commutation steps.

BLDC Design Considerations

[6] There many different pole slot combinations, to choose the correct combination we can look at the slot to pole ratio denoted as q. The slot to pole ratio can then be used to calculate the winding factor which is the ratio of armature current to torque. The calculations required is beyond the scope of this paper, but this information can be found on [7] "Distribution, coil-span and winding factors for PM machines with concentrated windings" by S. E. Skaar, Ø. Krøvel, R. Nilssen. (Figure 6 - Winding factor for number of poles v number of slots) shows the winding factor values, where Nm is the number of poles while Ns is the number of slots.

Ns	Nm										
	2	4	6	8	10	12	14	16	18	20	22
3	0.866	0.866	0.000	-0.866	-0.866	0.000	0.866	0.866	0.000	-0.866	-0.866
6	0.500	0.866	1.000	0.866	0.500	0.000	-0.500	-0.866	-1.000	-0.866	-0.500
9	0.328	0.617	0.866	0.945	0.945	0.866	0.617	0.328	0.000	-0.328	-0.617
12	0.250	0.500	0.683	0.866	0.933	1.000	0.933	0.866	0.683	0.500	0.250
15	0.199	0.389	0.562	0.711	0.866	0.910	0.951	0.951	0.910	0.866	0.711
18	0.167	0.328	0.500	0.617	0.735	0.866	0.902	0.945	1.000	0.945	0.902
21	0.142	0.282	0.415	0.538	0.650	0.747	0.866	0.890	0.932	0.953	0.953
24	0.125	0.250	0.366	0.500	0.583	0.683	0.760	0.866	0.885	0.933	0.949
27	0.111	0.220	0.328	0.429	0.525	0.617	0.695	0.766	0.866	0.877	0.915
30	0.100	0.199	0.296	0.389	0.500	0.562	0.640	0.711	0.774	0.866	0.874

Figure 6 - Winding factor for number of poles v number of slots

The ideal combinations can be identified by process of elimination.

1. Slot to pole ratio (q) < 0.25

   If q is less than 0.25 then each coil is interacting with multiple opposing magnets poles, thereby reducing torque.

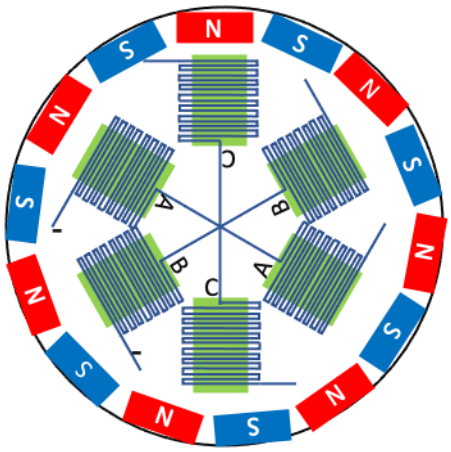

Figure 7 - BLDC motor with slot to pole ratio < 0.25

2. Slot to pole ratio (q) > 0.5

   If q is larger than 0.5 then multiple coils are interacting with the same magnet, this configuration is more suitable for a distributed winded motor instead of a concentrated winded motor. (Figure 8 - Difference between Concentrated and Distributed Windings)

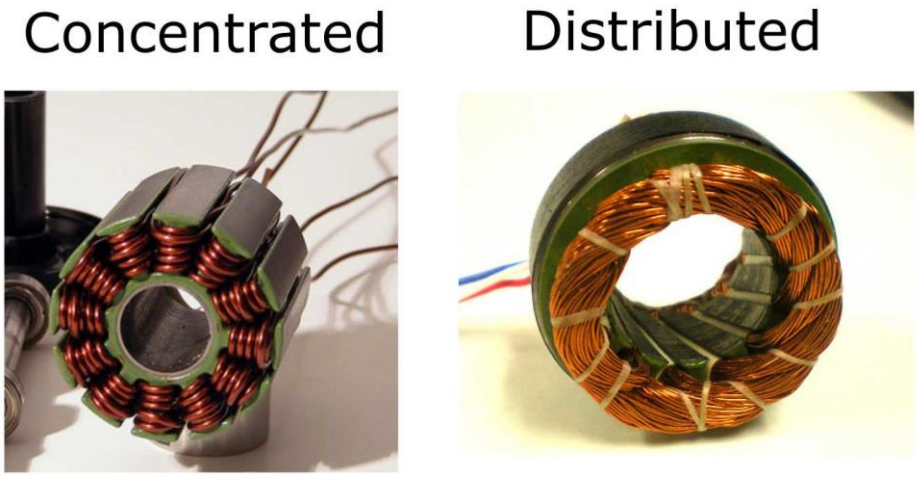

Figure 8 - Difference between Concentrated and Distributed Windings [6]

3. Slot to pole ratio = 1

   If the number of slots is equal to the number of poles there will be cogging torque.

4. Unbalanced windings

   The motor should have the same number of coils for each phase and divisible by 2 to allow torque to be produced on both sides of the motor keeping it balanced.

The control strategy discussed in the next section will be based on a motor with 12 stator slots (6 pole pairs) and 14 magnet poles, this is quite a common configuration in commercial BLDC motors.

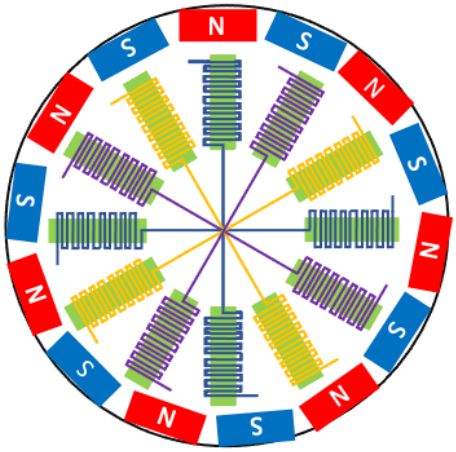

Figure 9 - BLDC motor for control strategy discussed in next section.

Problem Definition

Technical Problem

The current throttle motor system in production engines uses a DC motor. Given the many advantages as discussed previously on BLDC motors, there is a desire upgrade existing motors. This document will investigate the methodologies on both hardware and software level to achieve.

Existing Work / Work completed previously

The ESC is in the prototyping stage and can commutate a BLDC motor. Therefor no hardware modification is required to use the existing ESC for position control. Sensorless control is not implemented but since we will be using a hall effect position sensor we will be using sensored control.

Existing software is able to spin a BLDC motor at a specific RPM but position control is not implemented.

Existing user interface can display live parameters but uses outdated platform. Therefor it will be written from scratch using a modern programming language (C) and tailored for position control.

To program the existing ESC an additional piece of hardware is required. To further streamline the programming process this will be removed. The ESC will be reprogrammed from the PC using a USB to CAN adapter.

The boot code for the ESC is functional which allows programming via CAN so no modifications are required.

Technical Method

Overview of approach

Electronic Speed Controllers (ESC)

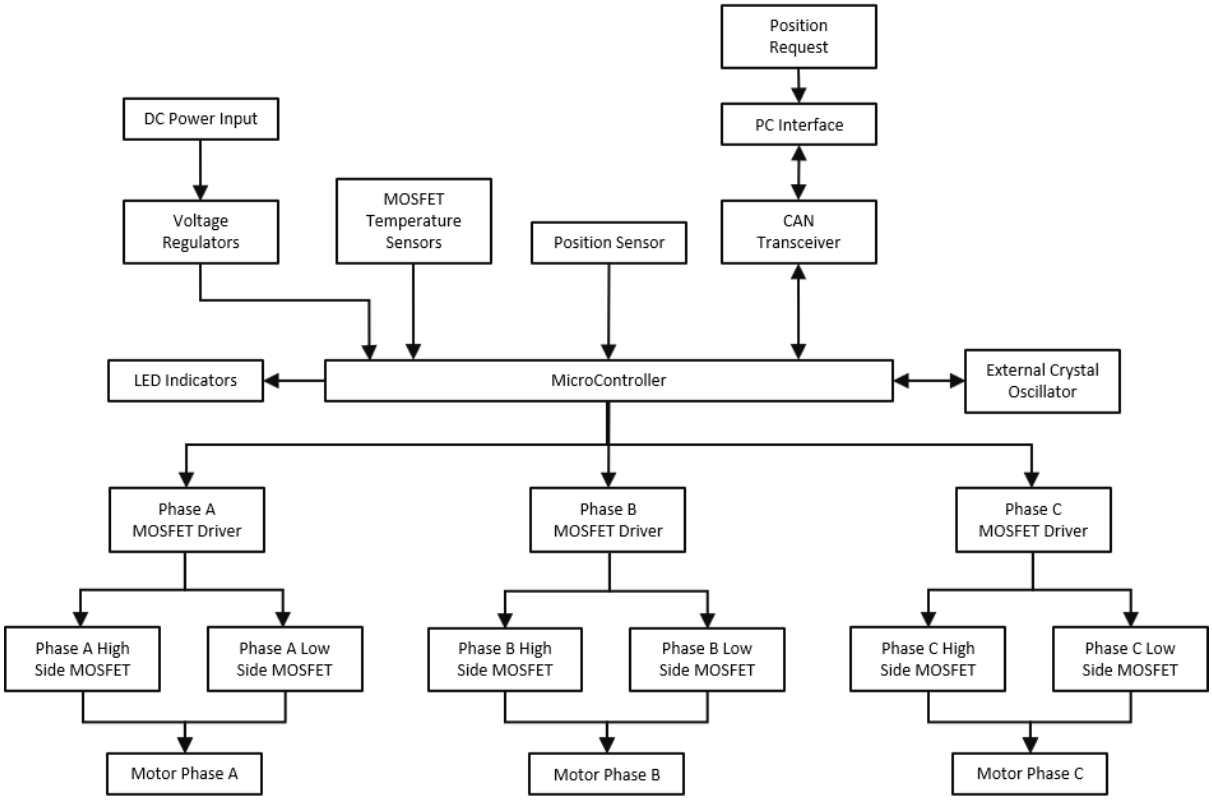

Figure 10 - ESC Block Diagram

The ESC is required to drive the BLDC motor by applying power to the 3 coils. The "brains" of the ESC is a microcontroller which handles all the signal processing. Factors such as clock speed, ability to calculate 32-bit floating-point numbers, number of input/output ports should be considered when choosing a suitable microcontroller. An external crystal oscillator is used to allow a higher and more stable clock frequency. To control the position of the motor a PC software is used to send a CAN message containing the desired position to the CAN Transceiver which then communicates to the microcontroller the position demand. The motor's live position is detected by an inductive position sensor (Figure 11 – Hall effect Position Sensor), this is superior to a potentiometer as it has a much longer live span due to no mechanical wear. The position sensor has an analogue output, and the microcontroller calculates the position of the shaft with that information. The MOSFET temperature sensors are a safety feature to allow the system to shut off if the MOSFET overheats. LED indicators are used to show if the device is powered and what mode it is in (boot mode or normal operation). Coils A, B, C are each connected to a pair of MOSFETS in a push pull configuration (Figure 12 - MOSFET with push pull configuration) to allow the coils to be pulled to supply or pulled to ground. A MOSFET driver is used to drive the gates as it requires a high current which cannot be supplied by the microcontroller.

Figure 11 – Hall effect Position Sensor [8]

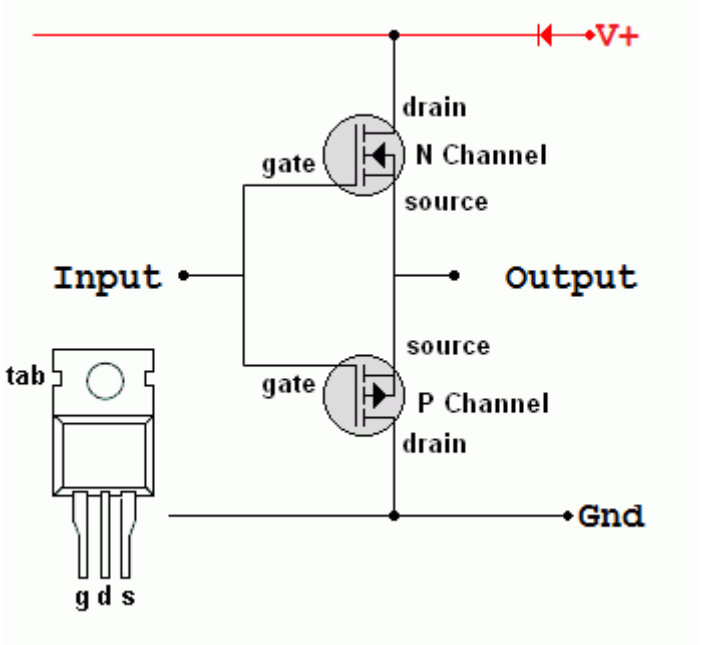

Figure 12 - MOSFET with push pull configuration [9]

Position Control Strategy

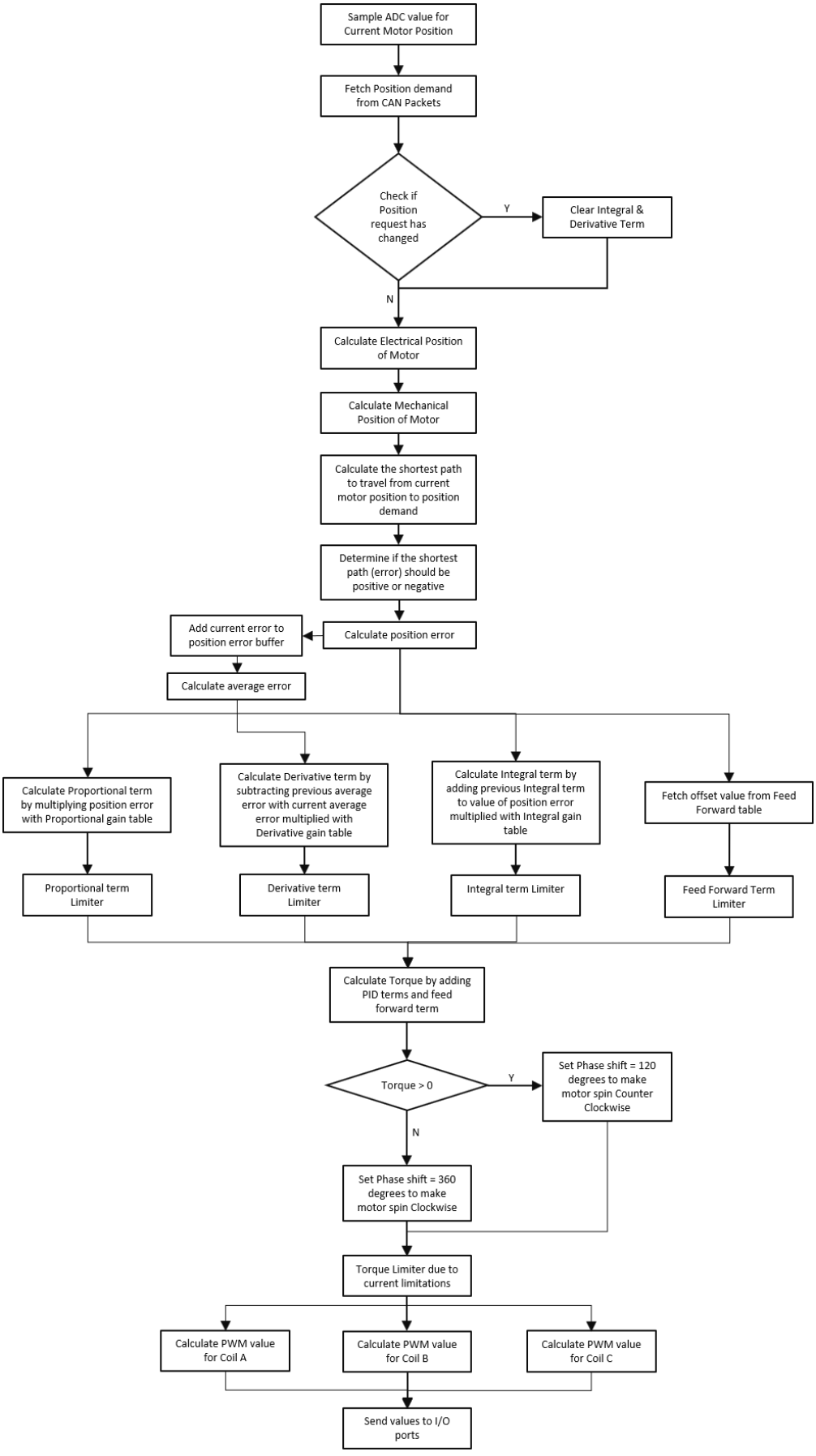

Figure 13 - Block Diagram for position control

The position control strategy is largely based on an array representing the values for a single cycle of the sine wave (Figure 14 - Array containing 480 samples representing one cycle of sine).

```
unsigned int SineWave[] = {
36619,36193,35766,35338,34911,34482,34054,33625,33196,32768,32339,31910,31481,31053,30624,30197,29769,29342,28916,28490,28066,27642,27218,26796,26375,25955,25536,25118,24702,24287,23873,23461,
23051,22642,22235,21829,21426,21025,20625,20228,19833,19440,19049,18661,18275,17891,17510,17132,16757,16384,16014,15647,15282,14921,14563,14208,13856,13507,13162,12820,12481,12146,11815,11487,
11162,10842,10525,10212,9903,9597,9296,8999,8706,8417,8132,7851,7574,7302,7035,6771,6512,6258,6008,5763,5522,5286,5055,4829,4607,4390,4178,3971,3769,3571,3379,3192,3010,2833,2661,2494,2333,
2176,2025,1879,1739,1604,1474,1349,1230,1117,1008,905,808,716,630,549,473,403,339,280,227,180,137,101,70,45,25,11,3,0,3,11,25,45,70,101,137,180,227,280,339,403,473,549,630,716,808,905,1008,
1117,1230,1349,1474,1604,1739,1879,2025,2176,2333,2494,2661,2833,3010,3192,3379,3571,3769,3971,4178,4390,4607,4829,5055,5286,5522,5763,6008,6258,6512,6771,7035,7302,7574,7851,8132,8417,8706,
8999,9296,9597,9903,10212,10525,10842,11162,11487,11815,12146,12481,12820,13162,13507,13856,14208,14563,14921,15282,15647,16014,16384,16757,17132,17510,17891,18275,18661,19049,19440,19833,20228,
20625,21025,21426,21829,22235,22642,23051,23461,23873,24287,24702,25118,25536,25955,26375,26796,27218,27642,28066,28490,28916,29342,29769,30197,30624,31053,31481,31910,32339,32768,33196,33625,
34054,34482,34911,35338,35766,36193,36619,37045,37469,37893,38317,38739,39160,39580,39999,40417,40833,41248,41662,42074,42484,42893,43300,43706,44109,44510,44910,45307,45702,46095,46486,46874,
47260,47644,48025,48403,48778,49151,49521,49888,50253,50614,50972,51327,51679,52028,52373,52715,53054,53389,53720,54048,54373,54693,55010,55323,55632,55938,56239,56536,56829,57118,57403,57684,
57961,58233,58500,58764,59023,59277,59527,59772,60013,60249,60480,60706,60928,61145,61357,61564,61766,61964,62156,62343,62525,62702,62874,63041,63202,63359,63510,63656,63796,63931,64061,64186,
64305,64418,64527,64630,64727,64819,64905,64986,65062,65132,65196,65255,65308,65355,65398,65434,65465,65490,65510,65524,65532,65535,65532,65524,65510,65490,65465,65434,65398,65355,65308,65255,
65196,65132,65062,64986,64905,64819,64727,64630,64527,64418,64305,64186,64061,63931,63796,63656,63510,63359,63202,63041,62874,62702,62525,62343,62156,61964,61766,61564,61357,61145,60928,60706,
60480,60249,60013,59772,59527,59277,59023,58764,58500,58233,57961,57684,57403,57118,56829,56536,56239,55938,55632,55323,55010,54693,54373,54048,53720,53389,53054,52715,52373,52028,51679,51327,
50972,50614,50253,49888,49521,49151,48778,48403,48025,47644,47260,46874,46486,46095,45702,45307,44910,44510,44109,43706,43300,42893,42484,42074,41662,41248,40833,40417,39999,39580,39160,38739,
38317,37893,37469,37045,
};
```

Figure 14 - Array containing 480 samples representing one cycle of sine wave.

The current mechanical position of the motor is calculated by scaling the analogue input of the inductive position sensor to 360 degrees. Due to the hardware limitations of the microcontroller, floating point calculations are not possible, therefor scaling is achieved by bit shifting (Figure 15 - Scaling Electrical and Mechanical Position using bit). To commutate the motor, we must first determine what the electrical position is based on the mechanical position, this is calculated by scaling the mechanical position to the number of magnet pole pairs multiplied by the number of discrete values in one sine wave.

```
ElecPos = (3*MotorAngle) + (unsigned int)(((unsigned long) MotorAngle * (unsigned long)23592)>>16);
MotorAngle = (unsigned int)(((unsigned long) MotorAngle * (unsigned long)23592)>>16);
```

Figure 15 - Scaling Electrical and Mechanical Position using bit shifting.

The shortest path from the current position to the position demand is determined and the error is calculated. The PID control loop is used to ensure the system response is optimized, with a quick response, minimal overshoot, and minimal oscillations. This is achieved using a standard PID loop with additional features. A look up table is used for the gains for each of the PID terms to allow gain scheduling. Gain scheduling can be used to further optimize the system response by adjusting the gain of each term depending on the size of the error, such as removing integral gain when the error is very small to prevent integral windup. Each of the PID terms are also limited. Additionally, a feed forward table is added which can be used to tune systems where the effects of disturbances can be predicted. The proportional gain is simply calculated by the corresponding gain from the proportional gain table depending on the error multiplied by the absolute value of the angle error. The integral gain is calculated by the previous integral gain stored in the memory from the last calculation added to the corresponding gain from the integral gain table depending on the error multiplied by the absolute value of the angle error. The derivative gain is slightly more complex as it is easily affected by noise. A circular buffer (Figure 16 - Derivative term buffer) is used to store the last ten angle errors, the average of these errors are then taken away from the previous derivative gain stored in the memory from the last calculation.

```
//Calculate D Term
Derivative = DerivativeFiltered - LastDerivativeFiltered;
LastDerivativeFiltered = DerivativeFiltered;
DerivativeLimited = DerivativeGain[ABSPIDError]*Derivative;
```

Figure 16 - Derivative term buffer

This value is then multiplied with the corresponding gain from the derivative gain table depending on the error. The stored integral and derivative values will be set to 0 if the position demand changes. The sum of the PID and feed forward terms are then added up to provide a torque. If the torque is a negative value, the motor has to spin Counter clockwise, likewise if the torque is a positive value, the motor has to spin clockwise.

The sine wave values for each coils are then calculated (Figure 17 - Calculate Sine Wave values for each individual). For Coil A, the sine wave value is calculated by the current electrical position of the motor added to the direction the motor to travel in (120 for Counter-clockwise, 360 for Clockwise) added to an offset which depends on how the inductive encoder is mounted. This offset is required due to the difference in position between the beginning of a mechanical revolution and the beginning of a electrical revolution. To make the motor travel in the Counter-clockwise direction the sine wave must be 90 degrees out of phase, as there are 480 samples in the sine wave, an offset of 120 is used. Similarly, to make the motor travel in the clockwise direction the sine wave must be 270 degrees out of phase, therefor an offset of 360 is used for 480 samples. The modulo (%) operator is then used to wrap around the samples. For coil B an offset of 160 is used as 480 samples divide by 3 phases equals 160, similarly for coil C.

```
CoilA = (ElecPos + MagDir + 50) % 480;
CoilB = (CoilA + 160) % 480;
CoilC = (CoilA + 320) % 480;
```

Figure 17 - Calculate Sine Wave values for each individual coil.

The sine wave values are then scaled by multiplying with the torque to provide the PWM values (Figure 18 - Scale Sine wave values to torque).

```
//Scale Torque to match Sine Wave Values
CoilA = (unsigned int)(((unsigned long) (SineWave[CoilA]) * (unsigned long)Torque)>>16);
CoilB = (unsigned int)(((unsigned long) (SineWave[CoilB]) * (unsigned long)Torque)>>16);
CoilC = (unsigned int)(((unsigned long) (SineWave[CoilC]) * (unsigned long)Torque)>>16);
```

Figure 18 - Scale Sine wave values to torque required.

Testing software

The testing software is written with a mixture of C and C++ using commercial DLL libraries. The user interface is designed specifically to control BLDC motors using position control. A commercial product (CANDo) is used to convert the serial data into CAN messages. The software can receive live CAN messages for debugging purposes such as MOSFET temperature, microcontroller run time and supply voltage (Figure 19 - PC software user interface). Users can send a position demand via the user interface to the ESC.

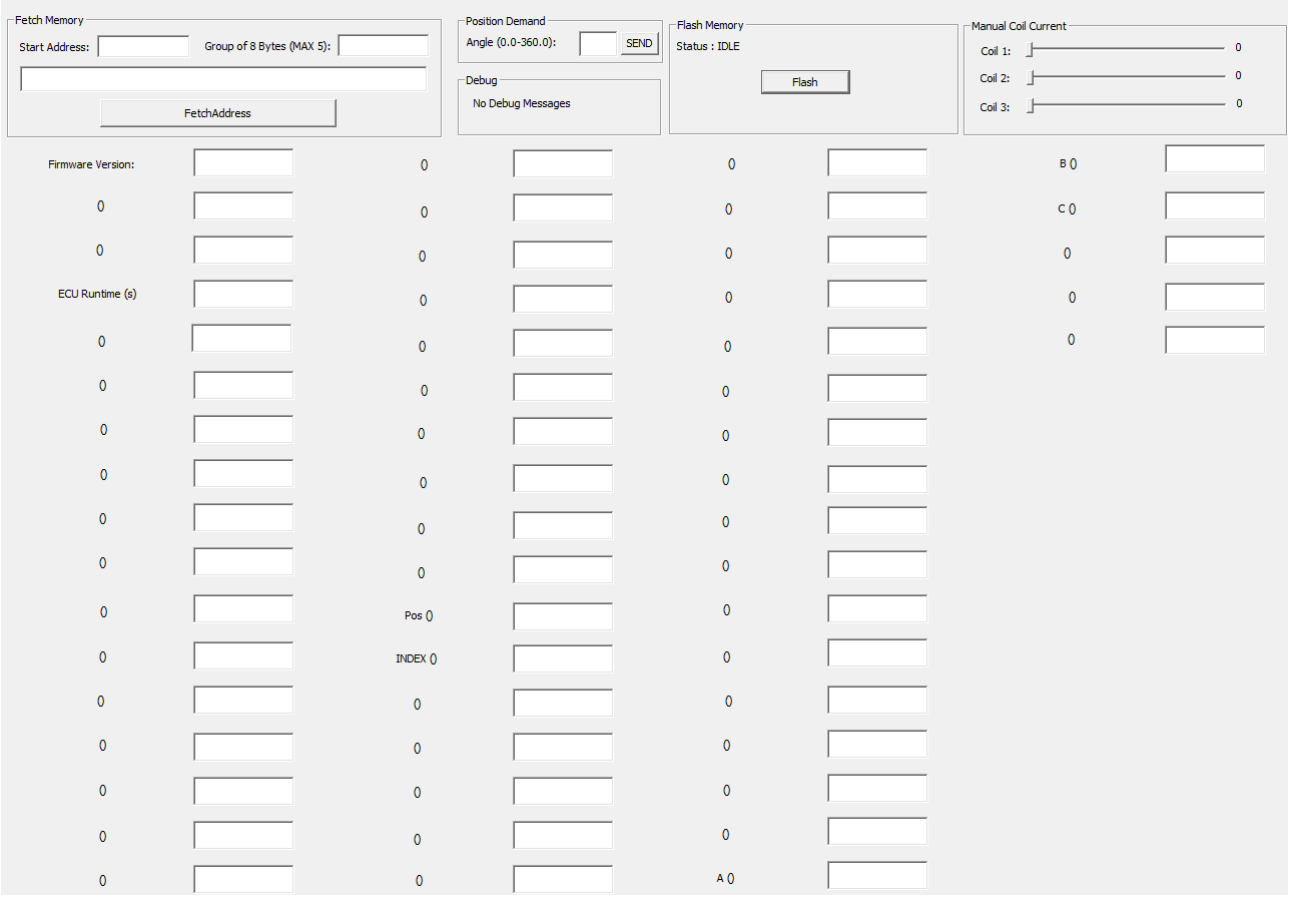

Figure 19 - PC software user interface

The graphical user interface (GUI) is designed using Resedit, which supports all Win32 controls such as static text, display, and buttons. This tool then generates C++ code for the dialogues which can then be imported into the main program.

```
16      LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
17      IDD_MAIN DIALOGEX 0, 0, 867, 557
18      STYLE DS_3DLOOK | DS_CENTER | DS_MODALFRAME | DS_SHELLFONT | WS_CAPTION | WS_GROUP | WS_SYSMENU
19      EXSTYLE WS_EX_APPWINDOW
20      FONT 8, "MS Shell Dlg", 0, 0, 0
21      {
22          PUSHBUTTON      "Flash", BUTTON_FLASH, 511, 44, 59, 15, 0, WS_EX_LEFT
23          GROUPBOX        "Flash Memory", False, 450, 12, 193, 72, 0, WS_EX_LEFT
24          EDITTEXT        EDIT_FIRMWARE, 133, 92, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
25          EDITTEXT        EDIT_MEMDIS, 18, 42, 271, 16, 0, WS_EX_LEFT
26          EDITTEXT        EDIT_ADDRESS, 69, 23, 62, 14, ES_AUTOHSCROLL, WS_EX_LEFT
27          LTEXT           "Group of 8 Bytes (MAX 5):", False, 141, 25, 96, 9, NOT WS_GROUP | SS_LEFT, WS_EX_LEFT
28          LTEXT           "Firmware Version:", False, 37, 98, 63, 9, NOT WS_GROUP | SS_LEFT, WS_EX_LEFT
29          GROUPBOX        "Position Demand", False, 309, 10, 137, 31, 0, WS_EX_LEFT
30          PUSHBUTTON      "SEND", BUTTON_RPM, 418, 21, 25, 14, NOT WS_TABSTOP, WS_EX_LEFT
31          EDITTEXT        EDIT_RPM, 390, 20, 26, 16, ES_AUTOHSCROLL, WS_EX_LEFT
32          LTEXT           "Angle (0.0-360.0):", False, 315, 23, 63, 9, NOT WS_GROUP | SS_LEFT, WS_EX_LEFT
33          LTEXT           "Start Address: ", False, 18, 26, 47, 9, NOT WS_GROUP | SS_LEFT, WS_EX_LEFT
34          GROUPBOX        "Fetch Memory", False, 9, 9, 290, 77, 0, WS_EX_LEFT
35          GROUPBOX        "Debug", False, 309, 46, 136, 39, 0, WS_EX_LEFT
36          LTEXT           "No Debug Messages", STATIC_DEBUG, 320, 60, 68, 9, SS_LEFT, WS_EX_LEFT
37          EDITTEXT        EDIT_BYTES, 229, 22, 61, 14, ES_AUTOHSCROLL, WS_EX_LEFT
38          PUSHBUTTON      "FetchAddress", BUTTON_ADDRESS, 71, 62, 157, 17, 0, WS_EX_LEFT
39          EDITTEXT        2001, 133, 119, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
40          EDITTEXT        2002, 133, 146, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
41          EDITTEXT        2003, 133, 173, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
42          EDITTEXT        2004, 132, 200, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
43          EDITTEXT        2005, 133, 229, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
44          EDITTEXT        2006, 133, 256, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
45          EDITTEXT        2007, 133, 283, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
46          EDITTEXT        2008, 133, 310, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
47          EDITTEXT        2009, 133, 337, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
48          EDITTEXT        2010, 133, 366, 67, 18, ES_AUTOHSCROLL, WS_EX_LEFT
```

Figure 20 - Code generated through Resedit

From Figure 20, Resedit generates code for the GUI by specifying location and size of each button/text.

This tool also includes a flash feature, which allows me to send my position control code through CAN messages to the microcontroller directly to be stored in the memory. There are a few steps involved in this:

1. Boot mode of target device is entered.
2. Existing memory is erased.
3. New code is imported and converted to transmittable format.
4. Code is divided into sections and send through CAN messages.
5. Code is temporary stored in RAM of the target device.
6. Code is written into the flash memory of the target device.
7. Device responds when code is received.
8. Exit boot mode.

List of Abbreviations

BLDC – Brushless Direct Current

ESC – Electronic Speed Controller

MOSFET – Metal Oxide Semiconductor Field Effect Transistor

PID Controller – Proportional Integral Derivative Controller

CAN – Controller Area Network

PC – Personal Computer

FPGA – Field Programmable Gate Array

RPM – Rotation per minute

DC – Direct Current

UAV – Unmanned Aerial Vehicle

APU – Auxiliary Power Unit

USP – Unique Selling Point

RAM – Random Access Memory

References

[1] electricityshock, “Advantages and disadvantages of brushless dc motor,” Electricity Shock, [Online]. Available: https://electricityshock.com/advantages-and-disadvantages-of-brushless-dc-motor/. [Accessed 16 03 2021].

[2] J. Flynt, “What are the benefits and limitations of brushless motors?,” 3D Insider, 23 January 2019. [Online]. Available: https://3dinsider.com/brushless-motors/. [Accessed 27 03 2020].

[3] Wheatstone, “Disadvantages of the BLDC Motor,” Wheatstone, 08 June 2020. [Online]. Available: https://www.bldc-wheatstone.com/news_a/Disadvantages-of-the-BLDC-motor.shtml. [Accessed 27 03 2020].

[4] Ryan , “Brushless Inrunner vs Outrunner motor,” General Electric, 02 08 2018. [Online]. Available: https://www.radiocontrolinfo.com/brushless-inrunner-vs-outrunner-motor/. [Accessed 28 03 2021].

[5] North Eastern, “Electromagnets,” North Eastern, [Online]. Available: https://ece.northeastern.edu/fac-ece/nian/mom/electromagnets.html. [Accessed 30 03 2021].

[6] R. Parsons, “Selecting the best pole and slot combination for a BLDC motor with concentrated windings,” Things in Motion, 27 01 2019. [Online]. Available: https://things-in-motion.blogspot.com/2019/01/selecting-best-pole-and-slot.html. [Accessed 03 04 2021].

[7] Ø. K. R. N. S. E. Skaar, “Distribution, coil-span and winding factors for PM machines with concentrated windings,” NTNU, [Online]. Available: http://www.elkraft.ntnu.no/eno/Papers2006/icem-skaar-krovel-nilssen06.pdf. [Accessed 03 04 2021].

[8] Environmental Engineering, “Non-Contact Angle Sensor For Motor Sport,” [Online]. Available: https://www.environmentalengineering.org.uk/news/non-contact-angle-sensor-for-motor-sport-1292/. [Accessed 01 04 2021].

[9] “MOSFET Push Pull Amplifier,” Reviseomatic, [Online]. Available: https://reviseomatic.org/help/s-push-pull/Push%20Pull%20MOSFET%20Amp.php. [Accessed 01 05 2021].

[10] High Speed Training, “Manual Handeling Awareness,” High Speed Training, [Online]. Available: https://www.highspeedtraining.co.uk/media/Products/manual-handling-training/manual-handling-awareness.pdf. [Accessed 28 04 2021].

[11] The IMI, “Electrical Behicle eLearning Course,” The IMI, [Online]. Available: https://www.theimi.org.uk/landing/ev/#0. [Accessed 28 04 21].

[12] “Electronic Throttle Body,” BOSCH, [Online]. Available: https://www.bosch-motorsport.com/content/downloads/Raceparts/en-GB/51017995147518219.html. [Accessed 2021 05 01].