Implementation of Motor Control using ARM Cortex-M7 Microcontroller

A PROJECT REPORT

Submitted by

ANURAAG PAL

190130111026

In partial fulfillment for the award of the degree of

BACHELORE OF ENGINEERING

in

Electronics and Communication Engineering

Government Engineering College, Gandhinagar

Gujarat Technological University, Ahmedabad

May, 2023

Government Engineering College, Gandhinagar

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Internship / Implementation of Motor Control using ARM Cortex M7 Microcontroller** has been carried out by Mr. **Anuraag Pal** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **Electronics and Communication Engineering**, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2022-23.

Signature: _______________	**Signature:** ________________
Name: Prof. Suhas H. Patel	**Name:** Prof. Tejas B. Sheth
Internal Guide	Head of Department

Government Engineering College, Gandhinagar

DECLARATION

We hereby declare that the Internship / Project report submitted along with the Internship / Project entitled **Implementation of Motor Control using ARM Cortex M7 Microcontroller** submitted in partial fulfillment for the degree of Bachelor of Engineering in Electronics and Communication Engineering to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me / at **Space Application Center (SAC), ISRO** under the supervision of **Shri. Mohammad Waris** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

	Name of the Student	**Signature**
1	Anuraag Pal	

ACKNOWLEDGEMENT

I, Anuraag Pal, would like to express my sincere gratitude to all those who have supported me in completing this Final Year Internship Project Report. First and foremost, I would like to thank my external guide, Shri. Mohammad Waris and my co-guides Shri. Manish Kumar Dwivedi and Shri. Chunduri Sai Abhishek, for their valuable guidance, insightful feedback, and constant encouragement throughout the project. I am grateful for their expertise, time, and patience, which have been instrumental in shaping this report.

I would also like to acknowledge Shri. Sanjeev Mehta, the industrial division head with SEDA Department, for providing me with the opportunity to work on this project and for sharing their valuable insights and knowledge. Their support and guidance have been instrumental in enabling me to gain practical exposure and insights into the industry.

I extend my heartfelt thanks to my internal guide, Shri. Suhas Patel, for their constant support, constructive criticism, and valuable guidance. Their deep knowledge of the subject, patience, and encouragement have been pivotal in enabling me to complete this report to the best of my abilities.

I am grateful to Government Engineering College, Gandhinagar, and Gujarat Technological University, Ahmedabad, for providing me with the necessary resources, infrastructure, and academic support to complete this report. I am also grateful to the faculty members and staff who have taught me and helped me develop my skills and knowledge over the years.

Last but not least, I would like to express my gratitude to my family for their unwavering love, support, and encouragement throughout my academic journey. Their faith in me and their constant motivation have been the driving force behind my success.

ABSTRACT

The project titled "Implementation of Motor Control using ARM Cortex M-7 Microcontroller" falls under Advanced Research and Development. The objective of this project is to control a 3-phase BLDC motor via a 3-phase channel motor controller, LX7720DB and an ARM Cortex M-7 microcontroller. In order to program the microcontroller, concepts such as the PI controller, field-oriented control, and space vector pulse width modulation are used.

The simulation is carried out using MATLAB and Simulink software, while the implementation is carried out using Microchip's MPLAB X IDE, XC32, and Harmony v3 Framework/Configurator. Additionally, the project includes a mathematical analysis that proves the equivalence of SVPWM and Inverse Clarke Transformation. It is demonstrated that SVPWM is more efficient in case of PWM Generation than Clarke Transformation, when PWM is generated after ICT using SPWM.

The project's significance lies in the advancement of motor control techniques that provide better efficiency and control of motor speed and torque. The ARM Cortex M-7 microcontroller is renowned for its high processing speed and power, making it an excellent choice for motor control applications. Future motor control systems will be more efficient and reliable as a result of the implementation of the project.

As a whole, the project aims to provide an integrated approach to the design, simulation, and implementation of motor control systems based on the ARM Cortex M-7 microcontroller.

List of Figures

List of Tables

List of Symbols

$^\circ$	Degree
ϕ	Phase
θ	Theta
V_{ref}	Reference Voltage
α	Alpha Component of Stationary Reference Frame
β	Beta Component of Stationary Reference Frame
d	d-Component of Rotatory Reference Frame
q	q-Component of Rotatory Reference Frame
i_a	Stator Current of Phase-A
i_b	Stator Current of Phase-B
i_c	Stator Current of Phase-C

List of Abbreviations

ARM	Advanced RISC Machines
ADC	Analog-to-Digital Converter
BLDC	Brushless DC motor
CAN	Controller Area Network
DAC	Digital-to-Analog Converter
FOC	Field Oriented Control
GSAT	Geosynchronous Satellite System
ICT	Inverse Clarke Transform
IDE	Integrated Development Environment
ISRO	Indian Space Research Organization
JTAG	Joint Test Action Group
LED	Light Emitting Diode
LVM3	Launch Vehicle Mark-III
MCAPP	Motor Control Applications
MCC	Microchip Code Configurator
METSAT	Meteorological Satellite
MIL	Military from MIL-1553, a Military Standard connector
NMIC	Nano-Micro Integrated Circuit
PID	Proportional Integral Differential
PWM	Pulse Width Modulation
RISC	Reduced Instruction Set Computing
SAC	Space Application Center, ISRO
SEDA	Sensor Development Area, SAC, ISRO
SMB	Subminiature version B Connector
SPI	Serial Peripheral Interface
SPWM	Sinusoidal Pulse Width Modulation
SRAM	Static Random-Access Memory
SVC	Space Vector Control
SVM	Space Vector Modulation
SVPWM	Space Vector Pulse Width Modulation
TRACE	Tool for Real-time Analysis and Critical Control Execution
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
RPM	Revolutions Per Minute
PIC	Peripheral Interface Controller
SAM	Smart ARM-based Microcontroller
AVR	Alf-Egil Bogen and Vegard Wollan's RISC Proc
CPU	Central Processing Unit
MIPS	Microprocessor without Interlocked Pipeline Sta
CNC	Computer Numerical Control
HVAC	High Voltage Alternating Current System
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transi

THSPWM	Triangular Harmonic Source Pulse Width Modul
FLC	Fuzzy Logic Controller
EMF	Electromotive Force
FPGA	Field-Programmable Gate Array
VMPS	Volts per Meter per Second
VGS	Gate-to-Source Voltage
VCC	Collector Voltage
VDD	Drain Voltage
GND	Ground
BSP	Board Support Package
GPIO	General Purpose Input / Output
MAINCLK	Main Clock
MCK	Master Clock
FAULT	Fault Signal
TCLK	Transmit Clock
SNS	Sensor
OUT	Output
RXD	Receive Data
TIOB	Timer Input / Output B
BLO	Backlight Output
TXD	Transmit Data
TIOA	Timer Input / Output A
PCK	Programmable Clock
MOD	Modulator
CLK	Clock
SYS	System

Table of Contents

Chapter-1 : Company Profile

1.1. Indian Space Research Organization (ISRO):

The Indian Space Research Organization (ISRO) is the primary space agency of India. Established in 1969, ISRO has since made significant contributions to the development of space technology in the country. ISRO is responsible for various activities related to space research, such as the design, development, and launch of satellites, space exploration, and the development of indigenous space technology.

Fig 1.1 ISRO Logo

ISRO's achievements are noteworthy. One of its most significant achievements was the successful launch of the Chandrayaan-1 mission in 2008, which was India's first mission to the Moon. The mission was a success, and Chandrayaan-1 discovered water molecules on the Moon's surface. ISRO's Mangalyaan mission, launched in 2013, was India's first mission to Mars and was also a success. ISRO has also launched various other satellites for communication, remote sensing, and navigation.

ISRO has a vast workforce of employees and interns. The employees are scientists, engineers, and other professionals with various backgrounds and expertise. The interns are students pursuing undergraduate and postgraduate degrees in various disciplines such as engineering, science, and management. Both employees and interns work on various projects related to space research, such as the development of satellites and the design of space missions.

The work of employees in ISRO is varied and depends on their area of expertise. Scientists and engineers are involved in the design and development of space-based systems such as satellites, launch vehicles, and ground systems. They are also responsible for the testing and evaluation of these systems to ensure their reliability and effectiveness. ISRO's employees are also involved in the planning and execution of space missions, such as the design of the mission architecture, selection of the payload, and management of the mission.

Interns in ISRO are involved in various projects related to space research, such as the design and development of satellite components, software development, and data analysis. The interns are provided with opportunities to work on live projects under the guidance of experienced professionals, which provides them with valuable experience and exposure to the space industry. The interns also get the opportunity to attend workshops and training programs to enhance their knowledge and skills.

Fig 1.2 ISRO’s successfully launches 36 Satellites in its heaviest LVM3-M2 rocket

ISRO's employees and interns work towards a common goal of advancing space technology in India. Their contributions to the development of indigenous space technology have been significant, and their work has led to the advancement of various areas such as communication, navigation, and remote sensing. ISRO's continued efforts in the development of space technology will undoubtedly lead to further advancements in the future, and its contributions are a testament to India's commitment to the development of space technology for the socio-economic development of the country.

1.2. Space Application Center (SAC):

The Space Application Center (SAC) is a vital constituent of the Indian Space Research Organization (ISRO) that plays a pivotal role in India's space program. SAC is located in Ahmedabad, Gujarat, and was established in 1972 as a part of the Space Commission. Its primary objective is to develop and utilize space technology for various applications related to the socio-economic development of India.

SAC is responsible for designing, developing, and operating space-based systems for remote sensing, communication, meteorology, and navigation. The center is also involved in the research and development of various technologies related to space applications. SAC has made significant contributions in various areas of space technology.

Fig 1.3 SAC Logo

Remote sensing is one of the key areas where SAC has made significant contributions. Remote sensing involves the study of the earth's surface from space using satellites. SAC has designed and developed a number of remote sensing satellites, including the Indian Remote Sensing (IRS) series, which have been used for various applications such as natural resource management, disaster management, and weather forecasting. The center has also developed various ground-based systems for receiving and processing the data obtained from these satellites.

Another area where SAC has made notable contributions is in the development of communication satellites. Communication satellites are used for various purposes, including television broadcasting, telecommunication, and internet services. SAC has designed and developed a number of communication satellites, including the Indian National Satellite (INSAT) series, which have been used for providing telecommunication and television services across the country.

SAC has also been involved in the development of meteorological satellites, which are used for weather forecasting and monitoring. The center has designed and developed a number of meteorological satellites, including the METSAT series, which have been used for various applications such as cyclone monitoring, flood forecasting, and drought assessment.

In addition to the above, SAC has also been involved in the development of navigation systems. Navigation systems are used for various applications, including air and sea navigation. SAC has developed the Indian Regional Navigation Satellite System (IRNSS), which is a regional navigation system that provides positioning and timing services to users in India and its surrounding regions.

Fig 1.4 GSAT-30 being tested in a clean room at the Space Applications Center.

SAC has been actively involved in international collaborations in the field of space technology. The center has collaborated with various space agencies around the world, including National Aeronautics and Space Administration (NASA), European Space Agency (ESA), and Japan Aerospace Exploration Agency (JAXA), for various space missions and research activities. SAC has also been involved in the development of various international projects such as the Global Monitoring for Environment and Security (GMES) project, which is a joint initiative of the European Commission and ESA.

SAC has also been involved in the development of various applications related to disaster management. The center has developed various satellite-based systems for disaster management, including the Indian Satellites for Disaster Management (ISDM) program, which aims to provide timely information to various disaster management agencies in the country. SAC has also developed a number of applications related to forest fire monitoring, landslide monitoring, and flood monitoring.

SAC has been actively involved in the development of various technologies related to space applications. The center has developed various technologies related to satellite communication, remote sensing, and navigation. SAC has also developed various ground-based systems for receiving and processing data obtained from satellites.

1.3. Sensor Development Area (SEDA):

The Sensor Development Area (SEDA) is a part of the Indian Space Research Organization (ISRO) and is located in Ahmedabad, Gujarat. SEDA was established in 1983 with the objective of developing sensors for space-based applications. The center is responsible for designing, developing, and testing various types of sensors used in space-based systems, including imaging sensors, spectrometers, radiometers, and specialized sensors for atmospheric, oceanic, and land surface observations.

SEDA's contributions in the field of remote sensing are noteworthy. The center has developed various types of sensors for remote sensing applications, such as the Linear Imaging Self-Scanning Sensor (LISS), which is used in the Indian Remote Sensing (IRS) series of satellites. LISS is capable of capturing images of the Earth's surface with a resolution of up to 5.8 meters, making it suitable for various applications, such as natural resource management, disaster management, and urban planning. SEDA has also developed hyperspectral imaging sensors, which are capable of capturing images of the Earth's surface in hundreds of narrow spectral bands.

In addition to remote sensing, SEDA has also made significant contributions in the development of sensors for navigation systems. SEDA has developed various types of sensors for the Indian Regional Navigation Satellite System (IRNSS), which is used for positioning and timing services in India and its surrounding regions. SEDA has developed the Rubidium Atomic Frequency Standard (RAFS), which is used as a reference clock for the IRNSS satellites. SEDA has also developed various other sensors for the IRNSS, such as the Time-Stamped Signal Generator (TSSG) and the Navigation Receiver.

SEDA has been actively involved in research and development activities related to sensor technology. The center has developed various technologies related to sensor design, signal processing, and calibration. SEDA has also developed ground-based systems for receiving and processing data obtained from sensor.

Chapter-2 : Literature Review

A literature review in a report is a critical analysis of the existing literature on a particular topic, providing context for the research being presented and establishing the importance of the research question or problem being addressed. It identifies key themes, concepts, and findings from previous research and analyzes and evaluates the quality and relevance of the sources used. The purpose of a literature review is to demonstrate the author's familiarity with the existing literature on the topic and to position their research within the broader context of the field.

2.1. Hardware Components Used:

1) **Microcontroller:** ARM Cortex-M7 Microcontroller
2) **Motor Driver:** Microsemi LX7720DB Motor Controller
3) **Motor:** Hurst's Series DMA0204024B1010 BLDC Motor

2.1.1. Microchip's ARM Cortex-M7 Microcontroller-EK Board:

The Microchip's ARM7 Microcontroller-EK is an evaluation platform designed for evaluating the Radiation-Hardened Arm® Cortex®-M7 microcontroller.

This evaluation kit is supported by Microchip's MPLAB® X Integrated Development Environment (IDE) and provides easy access to the device features. It supports stand-alone debuggers and includes an on-board embedded debugger. When the on-board debugger is used, no external tool is required to debug the embedded code or to program the microcontroller.

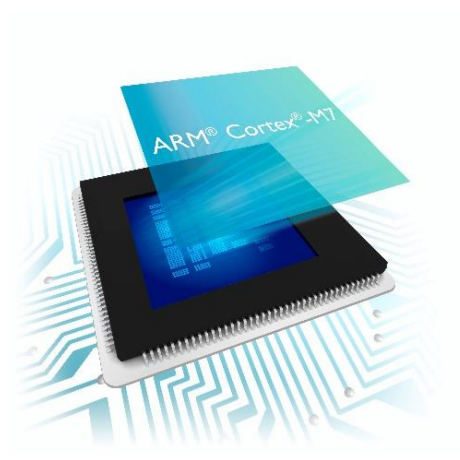


Fig 2.1 ARM Cortex-M7 Chip

The Xplained Pro headers enable support for additional peripherals to extend the features of the board and ease the development of custom designs.

2.1.1.1. Features of Microchip's ARM Cortex M7 Microcontroller-EK Board:

- **On-Board Memories**
  - 512 Kbytes (8-bit wide) Flash
  - 512 Kbits (8-bit wide) SRAM
  - 64 Mbit SPI Flash
- **On-Board Clock Management**
  - 32.768 kHz crystal
  - 10 MHz oscillator
- **Communication Interfaces**
  - UART emulation through USB interface
  - Two CAN ports supporting ATA6563 transceivers
  - Two SpaceWire connectors
  - Two MIL-1553 connectors
- **Analog Function Interfaces**
  - DAC outputs: Buzzer / SMB connector
  - ADC inputs: Potentiometer / SMB connector
- **Embedded Debug Access**
  - On-Board Embedded Debugger (PKOB4)
  - JTAG Debug connector
  - TRACE connector
- **Extension Capability**
  - Three headers compatible with Xplained mezzanine board
  - Four connectors with direct access to the microcontroller pins
- **On-Board End User Interface**
  - One mechanical Reset button
  - Four mechanical user push buttons
  - Six user LEDs
  - One pin for NMIC
- **5V Power Supply**
  - USB-C port
  - Power status LED

The main board offers a set of features that enables customers to prototype their own application with the Radiation Hardened features. The following figure shows the board layout of Microchip's ARM7-EK.

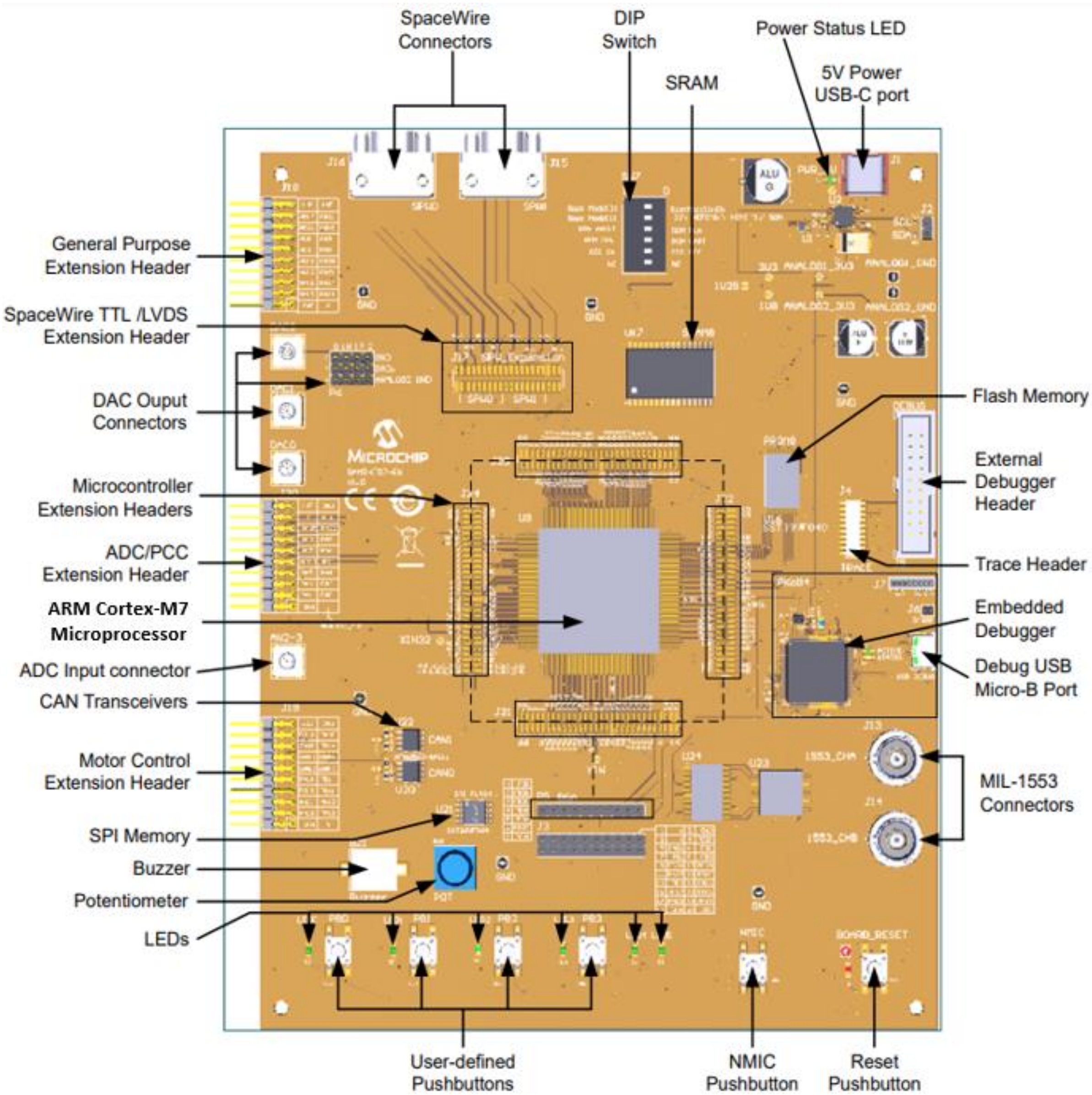


Fig 2.2 ARM Cortex-M7 Microcontroller - Evaluation Board

This microcontroller from the ARM7 family of devices developed by Microchip Technology Inc. It is designed for use in embedded applications that require high performance, low power consumption, and a small form factor.

The Microchip's ARM7 Microcontroller is based on an Arm Cortex-M7 processor running at a maximum clock speed of 480MHz. It has 1 MB of on-chip flash memory and 256 KB of SRAM, providing ample space for storing program code and data. The microcontroller also features a range of peripherals, including multiple UART, SPI, and I2C interfaces, as well as a USB 2.0 interface for communication with external devices.

One of the key features of the Microchip's ARM7 Microcontroller is its low power consumption. It includes multiple power-saving modes that enable it to operate with a very low current draw, making it ideal for use in battery-powered applications. The microcontroller also features an integrated voltage regulator that allows it to operate from a wide range of power sources, including single-cell Li-ion batteries.

Another important feature of this microcontroller is its security capabilities. It includes a range of hardware and software security features, such as secure boot, secure key storage, and tamper detection, that help to protect against unauthorized access and malicious attacks. These features make it well-suited for use in applications that require a high level of security, such as payment terminals and access control systems.

Overall, the Microchip's ARM7 Microcontroller is a powerful and versatile device that offers a range of features and capabilities for embedded applications. Its high performance, low power consumption, and security capabilities make it well-suited for a wide range of applications, from industrial control systems to consumer electronics devices.

2.1.2. Microsemi LX7720DB Motor Controller:

Microsemi's LX7720DB Motor Controller [15] is a high-performance, space-rated motor controller designed for use in satellite and spacecraft applications. The controller is radiation-hardened and is capable of operating in extreme temperatures and harsh environments.

The LX7720DB features a wide input voltage range, up to 50V, and can drive a variety of motor types, including 3-phase brushless DC (BLDC) motors, stepper motors, and DC brushed motors. The controller offers high-resolution sensorless motor control and supports a range of advanced control algorithms, including Field-Oriented Control (FOC), Space Vector Modulation (SVM), and sinusoidal commutation.

The controller is highly configurable, with a flexible software interface that allows for easy integration with a variety of microcontrollers and development platforms. It features a compact, lightweight design and low power consumption, making it an ideal choice for space-constrained applications.

2.1.2.1 Features of LX7720DB Motor Controller:

- High-precision voltage and current measurement
- Low power consumption
- Fast sampling rate
- Wide input voltage range
- High input impedance
- Built-in programmable gain amplifier
- Digital output interface
- Flexible power supply
- Small form factor
- Easy to use

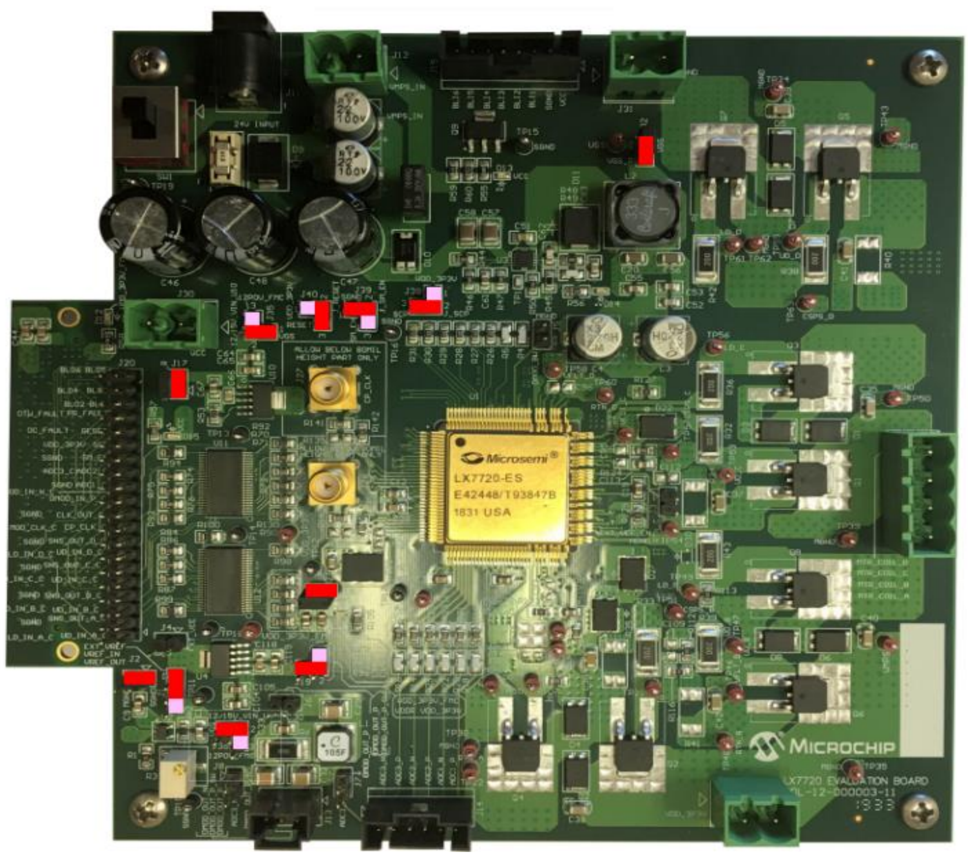


Fig 2.3 Microsemi LX7720DB Motor Controller

Overall, the Microsemi LX7720DB Motor Controller is a versatile and reliable motor driver IC that offers a range of features and capabilities for motion control applications. Its advanced control capabilities, protection features, and diagnostic and monitoring capabilities make it well-suited for use in a range of industrial and automation applications.

2.1.3. Hurst's Series BLDC Motor DMA0204024B1010:

The NT DYNAMO® is designed to provide:

- Maximum application flexibility
- Increased reliability
- Affordable Servo Control
- Smooth motion quality
- Matched motor and drive performance
- Plug-n-Play

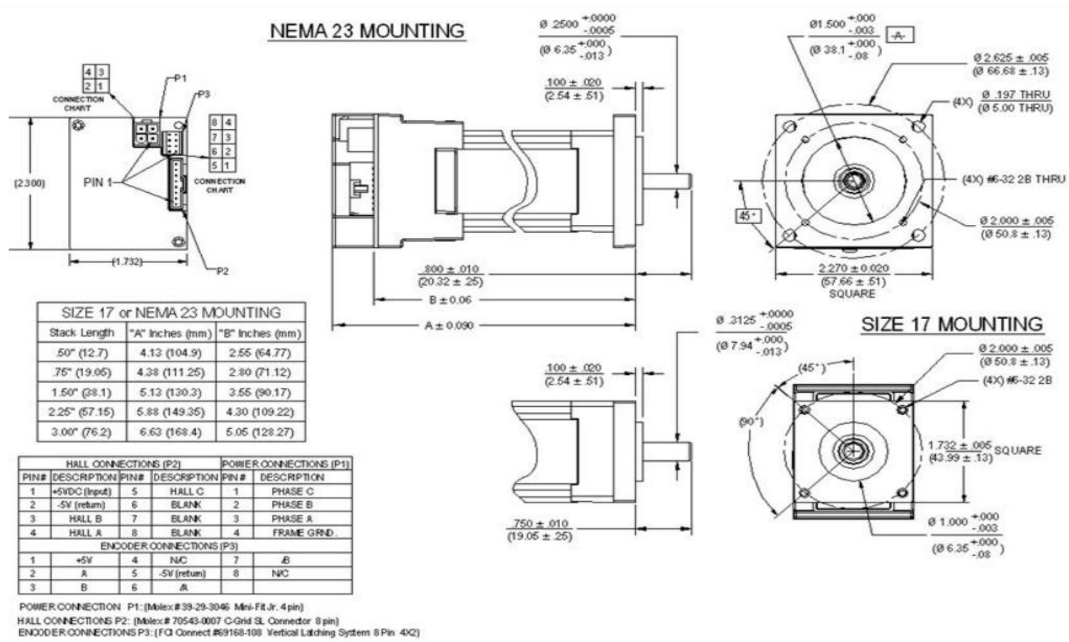


Fig 2.4 Hurst's BLDC Motor's External Control Type Drawings

The Hurst's BLDC motor [12][14] is a brushless DC motor designed for use in a range of applications, including automation, robotics, and industrial control systems. It is manufactured by Nidec Corporation, a leading global manufacturer of motors and related components.

The Hurst's motor features a compact and lightweight design, making it well-suited for use in applications where space is limited. It includes a range of advanced features, such as high torque output, low vibration and noise, and low power consumption, that make it ideal for use in applications that require high performance and efficiency.

Fig 2.5 Hurst's DMA0204024B1010 BLDC Motor

One of the key advantages of the Hurst’s motor is its high level of reliability and durability. It includes a range of protection features, such as overcurrent and overtemperature protection, that help to ensure safe and reliable operation over the lifetime of the motor. The motor also includes a range of diagnostic and monitoring features that enable it to detect and respond to potential issues before they can cause damage to the motor or surrounding equipment. Overall, the Hurst’s BLDC motor is a high-performance and reliable motor that offers a range of features and capabilities for a wide range of industrial and automation applications.

Table 2.1 Nomenclature of Hurst's Series DMA0204024B1010 BLDC Motor [9]

Nomenclature of Hurst’s DMA0204024B1010 BLDC Motor		
Product Family	DM	Dynamo
Operating Mode	A	External
Control Method (Input Signal)	0	None
Encoder (ppr)	2	250 ppr
Thermal Protection	0	None
Model Rated Torque (oz-inch)	4	30 oz-inch
Input Voltage (Vdc)	024	24V
Winding Resistance (ohms)	B	0.57 Ω
Mechanical Features	101	Size 17 with Cables
Preset Speed 10 (RPM)	0	None

2.2. Software Used:

- **For Practical Implementation:**
    a. MPLAB® X Integrated Development Environment **(**IDE)
    b. MPLAB® XC32 Compiler
    c. MPLAB® Harmony v3
- **For Simulation:**
    a. MATLAB & Simulink

2.2.1. MPLAB® X IDE and MPLAB®XC32 Compiler:

Fig 2.6 MPLAB® X IDE Logo

Microchip's MPLAB X IDE (Integrated Development Environment) and XC32 Compiler are software tools used for developing embedded applications using Microchip's 32-bit microcontroller family. MPLAB X IDE is a powerful, easy-to-use platform that provides an integrated development environment for creating, debugging, and programming applications for Microchip's microcontrollers. The MPLAB X IDE includes advanced features such as code profiling, code coverage, and integrated version control to help developers improve code quality and productivity.

The XC32 Compiler is a powerful optimizing C/C++ compiler that generates efficient and optimized code for Microchip's 32-bit microcontrollers. It includes advanced features such as inter-procedural optimization, loop optimization, and register allocation to help developers write efficient code for their applications. Additionally, the XC32 Compiler supports the use of inline assembly code, making it easy to include low-level code in C/C++ projects.

Fig 2.7 MPLAB® XC32 Logo

Together, the MPLAB X IDE and XC32 Compiler provide a comprehensive, reliable, and efficient toolset for developers working on Microchip's 32-bit microcontrollers. They are widely used in the embedded systems industry and are considered to be among the best software tools available for developing applications for Microchip's microcontrollers.

2.2.2. MPLAB® Harmony v3:

A software development platform from Microchip called MPLAB Harmony v3 offers an integrated and all-encompassing environment for creating embedded programs. It provides a collection of libraries, drivers, middleware, and tools that programmers may use to build programs that are effective, dependable, and high-quality for Microchip's microcontrollers.

Fig 2.8 MPLAB® Harmony Logo

A robust and user-friendly Configurator tool is included in MPLAB Harmony v3 to make tweaking and personalizing the framework components easier. Developers can choose and configure the desired libraries, drivers, and middleware for their applications using the Configurator's simple graphical interface. It also comes with a collection of pre-configured templates that are simple to edit to meet unique project requirements.

The MPLAB Harmony v3 Configurator enables programmers to construct applications using a variety of programming languages, including C and C++ and supports a wide range of Microchip microcontrollers, including the PIC32, AVR and SAM families like SAMRH71, SAMRH707, etc. It is also simple to design, debug, and deploy applications thanks to the Configurator's code generation which is fully integrated with the MPLAB X IDE.

Overall, the MPLAB Harmony v3 Framework and Configurator offer embedded application developers a strong and adaptable solution that enables them to accelerate time-to-market, enhance code quality and efficiency, and streamline the development process.

2.2.3. MATLAB & Simulink:

MATLAB & Simulink is a graphical programming environment developed by MathWorks for the modeling, simulation, and analysis of dynamic systems. It is a comprehensive software tool that allows engineers and scientists to design and simulate complex systems, providing a complete solution for the development of control systems, signal processing algorithms, and other applications.

Fig 2.9 MATLAB & Simulink Logo

Simulink uses block diagrams to represent and simulate systems, with each block representing a different component or subsystem of the system being modeled. The blocks can be connected together to form a complete model of the system, which can then be simulated and analyzed using a range of simulation tools.

One of the key advantages of Simulink is its ability to integrate with MATLAB, allowing users to take advantage of MATLAB's powerful mathematical and analytical capabilities. Simulink also includes a range of built-in tools and libraries for a wide range of applications, such as control systems, signal processing, and communications.

Simulink is used in a wide range of industries and applications, including automotive, aerospace, robotics, and telecommunications. It is particularly well-suited for the development of complex systems with multiple components, such as control systems for autonomous vehicles or signal processing algorithms for wireless communication systems.

Overall, MATLAB & Simulink is a powerful and versatile tool for the modeling, simulation, and analysis of dynamic systems. Its integration with MATLAB, range of built-in tools and libraries, and ease-of-use make it an ideal choice for engineers and scientists working on complex system design and analysis projects.

2.3. Concept Used:

1) **PID Controller**
2) **Field Oriented Control (FOC)**
3) **Space Vector Pulse Width Modulation (SVPWM)**

2.3.1. PID Controller:

A PID controller is a type of feedback control system widely used in engineering applications to control a wide range of processes, including temperature, speed, pressure, and position. PID stands for Proportional, Integral, and Derivative, which are the three main components of the controller.

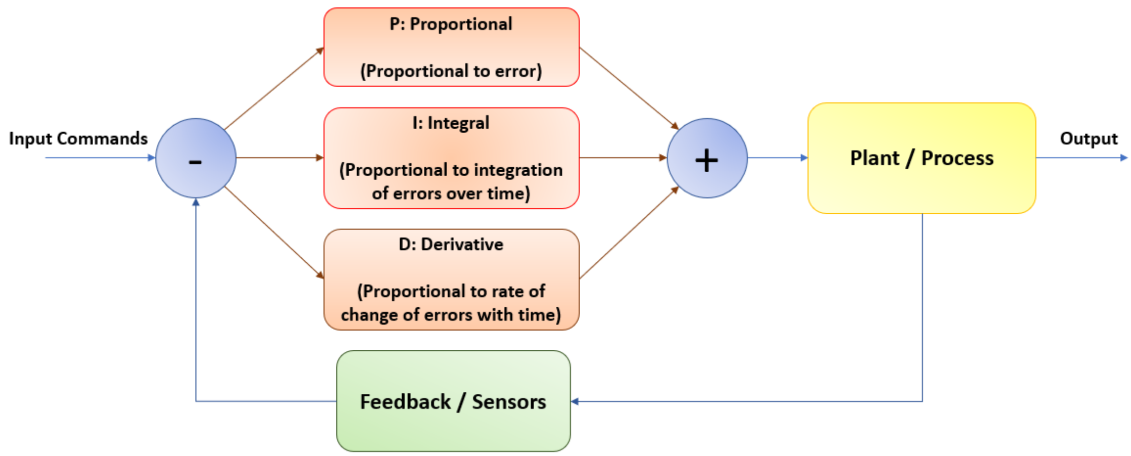


Fig 2.10 Block Diagram of PID Controller

The proportional component of the controller provides an output that is proportional to the difference between the desired setpoint and the actual value of the process being controlled. The integral component of the controller considers the history of the error and provides an output that helps to eliminate steady-state errors. The derivative component of the controller provides an output that helps to reduce the overshoot and settling time of the controlled variable.

The PID controller continuously monitors the process being controlled, compares the actual value with the desired setpoint, and calculates the appropriate control signal based on the three components of the controller. The control signal is then sent to the actuator, which adjusts the process variable to the desired setpoint. The PID controller is a powerful and versatile control system that can be used to control a wide range of processes. It is widely used in industrial automation, robotics, and process control applications due to its simplicity, effectiveness, and ease-of-use.

2.3.2. Field Oriented Control (FOC):

Brushless DC (BLDC) motors operate using the field-oriented control (FOC) technique. Electric motors known as BLDCs run without commutators and brushes, making them more dependable and effective than their brushed counterparts. The excellent performance and efficiency of FOC makes it a common technique of control for BLDC motors, which need an electronic controller to function. The rotating reference frame, also known as the D-Q reference frame, and the stationary reference frame, also referred to as the - reference frame, are used in FOC.

Stator currents and voltages are measured in the stationary reference frame and converted to the spinning reference frame. It is simpler to operate the motor since the currents and voltages in the spinning reference frame are in line with the rotor flux. Based on the required output, FOC employs a PI controller to modify the motor's torque and speed. The PI controller makes any necessary adjustments to the current and voltage after comparing the desired output to the actual output. This guarantees that the motor will run smoothly and efficiently at the desired speed and torque.

FOC allows precise control of the motor's speed and torque. This makes it the most suitable choice for applications requiring a high degree of accuracy and responsiveness. Additionally, FOC permits more effective energy use, which results in less energy being consumed and longer battery life. FOC is frequently utilized in many different applications, including industrial automation, robotics, and electric cars.

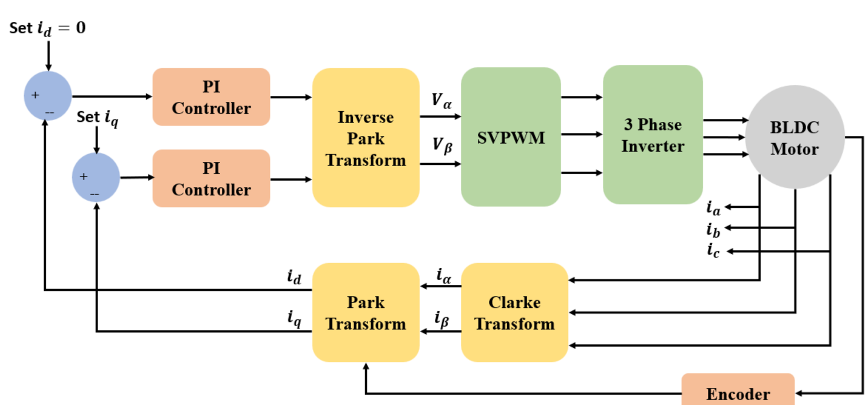


Fig 2.11 Block Diagram of Field Oriented Control (FOC)

2.3.3. Space Vector Pulse Width Modulation (SVPWM):

Space Vector Pulse Width Modulation (SVPWM) is a control method used to regulate the speed and torque of three-phase electric motors, particularly brushless DC (BLDC) motors. SVPWM utilizes the amplitude and angle of the voltage vector as the two control variables to create a three-phase voltage waveform.

Initially, the voltage vector is divided into a positive sequence component and a zero-sequence component in SVPWM, where the positive sequence component controls the voltage waveform's amplitude, while the zero-sequence component controls the voltage vector's angle. By adjusting the amplitude and angle of the voltage vector, it is possible to regulate the speed and torque of the motor with improved precision.

Compared to other modulation techniques, such as sinusoidal pulse width modulation (SPWM), SVPWM provides enhanced control over motor behavior. This is due to its ability to synthesize a more accurate three-phase voltage waveform. SVPWM is widely used in industrial automation, robotics, and electric vehicles, among other applications, where precise control over motor behavior is essential.

SVPWM offers improved efficiency, performance, and control, making it an ideal choice for applications that require accurate and precise control over motor behavior. Its widespread use in a range of industries is a testament to its effectiveness as a control method for regulating the speed and torque of electric motors, particularly in complex systems.

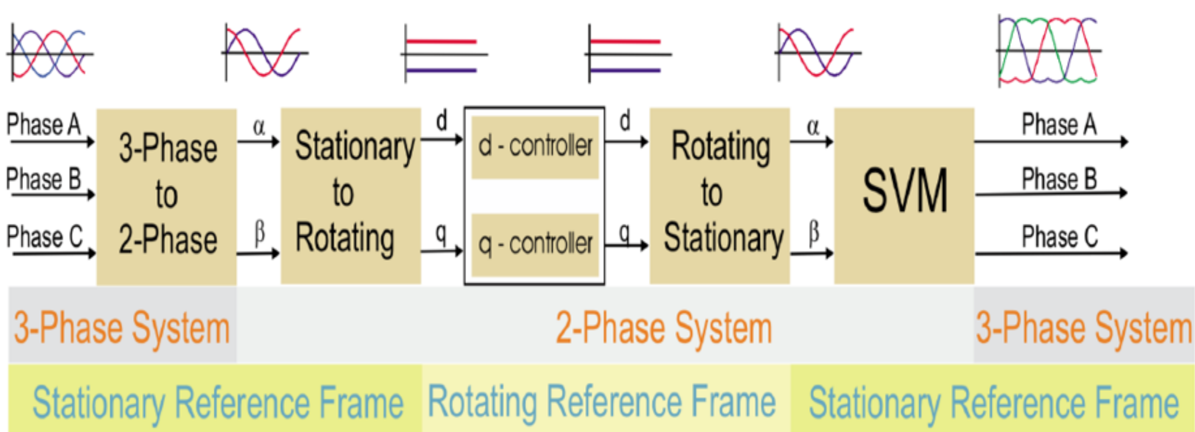


Fig 2.12 Field Oriented Control Transform with SVPWM and their waveforms [6]

2.4. Literature review and rationale for the current project:

The use of ARM Cortex microcontrollers for motor control has drawn a lot of attention in the field of embedded systems. The many methods and strategies for implementing motor control using microcontrollers have been studied in several studies. We will examine the pertinent papers about the implementation of motor control utilizing the ARM Cortex microcontroller in this literature study.

A method for providing brushless DC motor control using an ARM Cortex-M4F-based microcontroller was provided in [19]. The study showed that using the ARM Cortex-M4F for motor control was more effective and reliable than using conventional techniques.

A speed control system for a DC motor was designed and implemented in [10] utilizing an ARM Cortex-M3 microcontroller. The study proved that using the ARM Cortex-M3 microprocessor to provide motor control is both feasible and efficient.

An approach for controlling a permanent magnet synchronous motor with an ARM Cortex-M4F microcontroller was put forward in [4]. The study showed that the proposed algorithm enhanced the motor control system's effectiveness and efficiency.

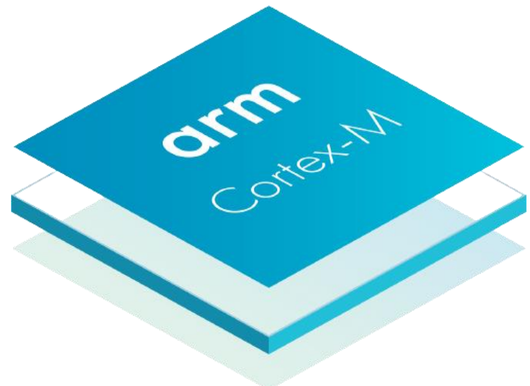
The design and implementation of a three-phase induction motor control system using an ARM Cortex-M4F microcontroller were also reported in [17]. The study showed that using the ARM Cortex-M4F to execute motor control was successful in establishing accurate control of the induction motor.

That is why, in this research, I demonstrate **the effectiveness and efficiency of using an ARM Cortex-M7 microcontroller to accomplish precise control of motor systems**. The experiments carried out in the present project, "Implementation of Motor Control using ARM Cortex-M7 Microcontroller," show that an ARM Cortex M7 microcontroller, was useful for improving the performance and efficiency of motor control applications.

Chapter-3 : Introduction to ARM Cortex-M7

3.1. ARM Cortex-M Processor Family:

The term "Cortex" is a brand name used by ARM to describe a specific family of microcontroller architectures in the ARM Cortex-M7 CPU. Cortex processors are the central processing unit for a wide range of embedded applications, hence the term "Cortex" was chosen to communicate the idea of a core or central processing unit.



The Cortex-M7 [2] is a member of the Cortex-M processor family, which is designed for microcontroller applications that require a balance of high performance, low power consumption, and ease of use. The "M" in Cortex-M stands for "microcontroller," indicating that this processor family is specifically developed for microcontroller applications.

The ARM Cortex-M processor family is a well-known architecture for embedded controllers. From low-power, economical options to high-performance, reconfigurable systems like the ARM Cortex-M7 processor, these CPUs offer the computing capability required for a variety of space and robotics applications. The Cortex-M processor family's scalability gives it a flexible option for many applications, and the instruction set upward compatibility guarantees simplicity of development across a variety of platforms.

The Cortex-M Microcontroller Software Interface Standard (CMSIS), which runs on top of the hardware, offers a vendor-neutral hardware abstraction layer for the Cortex-M processor series, while the recently unveiled mbed™ OS platform provides a potent software platform for creating Internet of Things and robotics applications. Robotics and space automation are only two of the embedded sectors where ARM Cortex-M processors are the go-to solution due to their software compatibility and robust development and support environment.

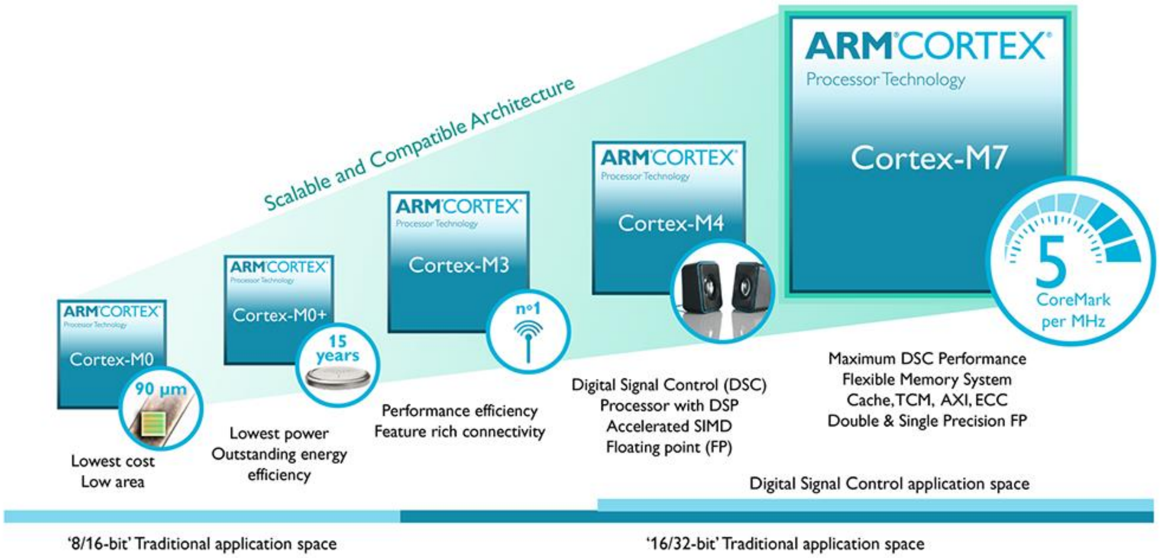


Fig 3.1 ARM Cortex-M Family [3]

3.1.1. Cortex-M7 an Excellent Choices:

Numerous embedded systems that require high processing performance, real-time response capability, and energy efficiency use the ARM Cortex-M7 processor in robotics, space automation, and other fields. These areas' needs can also be fulfilled by using its capabilities.

The Cortex-M7 is an excellent choice for applications that need real-time reaction and high-speed processing because of its high-performance, 6-stage pipeline with dual-instruction issue, which allows it to execute up to two instructions per clock cycle. Engineers and developers working on space automation, robotics, and other embedded systems have flexibility thanks to its flexible design, which enables chip designers to match the core with specific application needs.

Space automation, robotics, and other embedded systems that need dependable and deterministic access to peripherals and memory can be supported by the processor's optional instruction and data cache, tightly coupled memory, and low latency peripheral bus interface.

3.2. ARM's Architecture:

The RISC (Reduced Instruction Set Computing) architecture is used in ARM (Advanced RISC Machines) processors. A style of computer architecture known as RISC places an emphasis on the usage of a condensed, streamlined instruction set that can be executed more quickly and effectively. This method lessens the processor's complexity, which leads to a smaller chip size, less power use, and quicker execution times.

From smartphones and tablets to embedded systems and servers, ARM processors are employed in a variety of products due to their low power requirements. Since power consumption and performance are crucial considerations in mobile and embedded applications, the RISC architecture used in ARM processors has contributed to their popularity.

3.2.1. Key Points about RISC: [1]

- RISC stands for Reduced Instruction Set Computing.
- It is a type of computer architecture that emphasizes the use of a small and simple set of instructions that can be executed quickly and efficiently.
- The RISC approach aims to simplify the instruction set by removing unnecessary instructions and reducing the complexity of the processor.
- This approach results in a smaller chip size, lower power consumption, and faster execution times.
- RISC processors typically have a simpler pipeline and execute instructions in a single cycle, reducing the time required to execute each instruction.
- RISC processors also use a technique called register windowing, which allows for more efficient use of registers and faster context switching.
- RISC processors are designed to be highly modular, with separate units for instruction decoding, register file access, and arithmetic logic.
- Examples of RISC architectures include ARM, MIPS, and PowerPC.

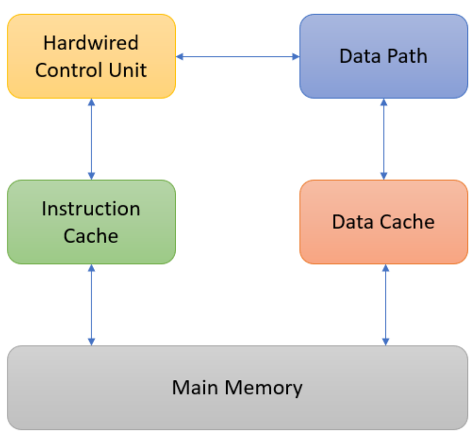


Fig 3.2 RISC Architecture

3.3. Features of ARM Cortex-M7 32-bit Microcontroller:

- **High-Performance Processing:** The Cortex-M7 features a high-performance 6-stage pipeline with dual-instruction issue, allowing it to execute up to two instructions per clock cycle. This makes it suitable for applications that require real-time response and high-speed processing.
- **Memory Options:** The Cortex-M7 supports a range of memory options, including instruction and data caches, tightly coupled memory (TCM), and external memory interfaces. This allows designers to optimize memory access for specific application requirements.
- **Flexible Instruction Set:** The Cortex-M7 instruction set is designed to be flexible and scalable, with support for up to 16 priority levels of interrupt handling. This allows developers to prioritize critical functions and optimize performance.
- **Low-Power Features:** The Cortex-M7 includes a range of low-power features, including power gating, dynamic voltage and frequency scaling, and sleep modes. This makes it suitable for battery-powered applications and other low-power environments.
- **Communication Interfaces:** The Cortex-M7 includes a range of communication interfaces, including UART, SPI, I2C, USB, and Ethernet. This makes it suitable for applications that require communication with other devices or networks.
- **Debugging and Trace:** The Cortex-M7 includes advanced debugging and trace capabilities, including real-time trace and data watchpoints. This makes it easier for developers to debug and optimize their applications.

3.4. Applications of ARM Cortex-M7 Microcontroller:

The ARM Cortex-M7 is a powerful and versatile 32-bit microcontroller that is suitable for a wide range of embedded applications. Here are some key applications of the Cortex-M7:

- **Internet of Things (IoT) Devices:** The Cortex-M7 is perfect for Internet of Things (IoT) devices that need fast processing, quick responses, and low power consumption. It is a popular alternative for IoT applications including smart home devices, wearables, and industrial automation thanks to its flexible memory options, communication ports, and low-power features.
- **Robotics:** Robotics applications that demand quick processing, prompt responses, and accurate control can also benefit from the Cortex-M7. It is a suitable alternative for robotics applications including drones, autonomous vehicles, and industrial robots due to its adaptable memory options and communication interfaces.
- **Automotive:** The Cortex-M7 is well-suited for automotive applications that require real-time response and high-speed processing. Its low-power features and communication interfaces make it a popular choice for automotive applications such as infotainment systems, advanced driver assistance systems (ADAS), and in-vehicle networking.

- **Medical Devices:** The Cortex-M7 is also suitable for medical devices that require high-speed processing, real-time response, and low power consumption. Its flexible memory options and communication interfaces make it a good choice for medical devices such as patient monitors, diagnostic equipment, and implantable devices.
- **Aerospace and Defense:** The Cortex-M7 is also used in aerospace and defense applications that require high-speed processing, real-time response, and precise control. Its low-power features and communication interfaces make it a popular choice for applications such as avionics, navigation systems, and unmanned aerial vehicles (UAVs).
- **Audio and Video Processing:** The Cortex-M7 is suitable for audio and video processing applications that require high-speed processing and real-time response. Its flexible memory options and communication interfaces make it a good choice for applications such as digital signal processing, audio and video codecs, and image processing.

Overall, the ARM Cortex-M7 is a versatile and powerful microcontroller that can be used in a wide range of embedded applications. Its high-performance processing, memory options, flexible instruction set, low-power features, communication interfaces, and debugging capabilities make it a popular choice for developers and engineers in a variety of industries.

Chapter-4 : Introduction to Motors

4.1. Motors:

An apparatus that transforms electrical or mechanical energy into rotational or linear motion is called a motor. From operating small household items to powering industrial machinery, motors are employed in a variety of applications.

There are many different types of motors, including brushless DC motors, stepper motors, AC motors, and others.

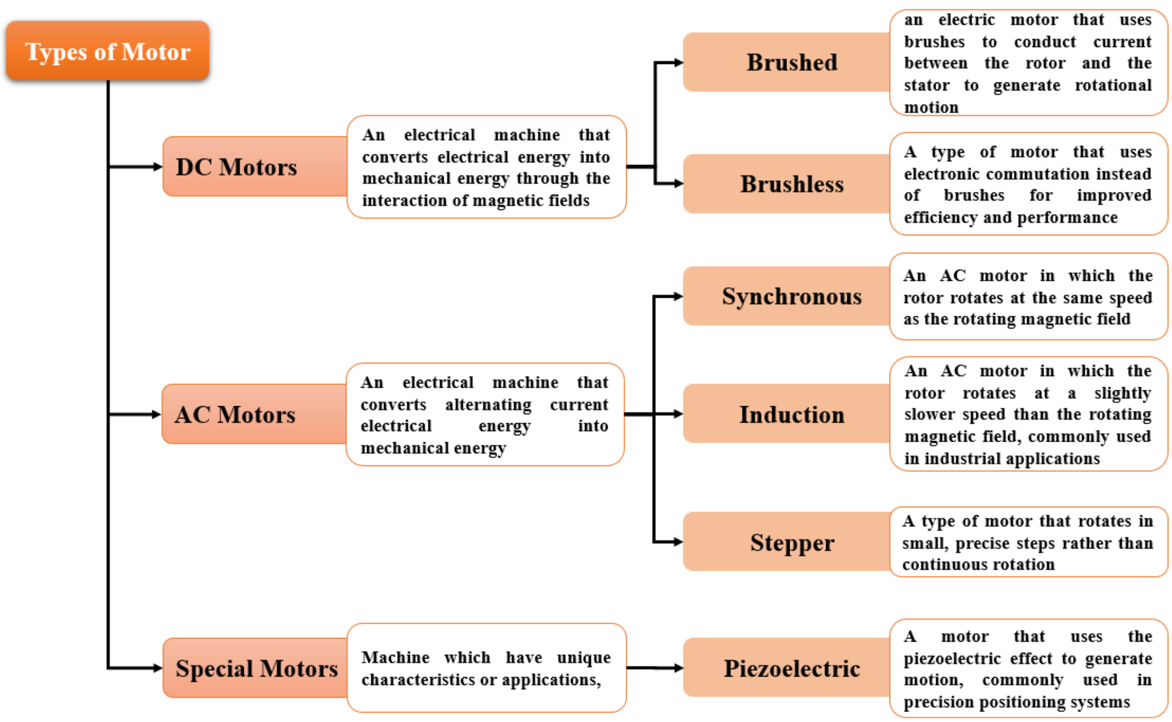


Fig 4.1 Types of Motors

Stepper motors are utilized in robotics and CNC machines, among other applications where precise motion control is necessary. Compared to conventional DC motors, brushless DC motors have improved efficiency and longer lifespan because they use electronic commutation rather than mechanical commutation.

A variety of motor control techniques, including pulse width modulation (PWM) and field-oriented control (FOC), are employed to obtain desired speed and torque outputs from a motor.

For motors to last a long time and perform at their best, proper maintenance is also essential. Regular checks, lubrication, and safeguards against overload, overvoltage, and overcurrent situations are part of this. Motors are becoming increasingly intelligent as a result of the development of smart motor technology and the Internet of Things (IoT), which allows for remote control, real-time monitoring, and proactive maintenance.

Fig 4.2 Different types motors

In space and robotics applications, motors play a crucial role in controlling the movement of robotic systems. They need to be compact, lightweight, and efficient while providing precise control over the movement. Here are some types of motors used in space and robotics applications:

- **DC motors:** These are commonly used in robotics applications due to their simple control mechanisms and compact size.
- **Stepper motors:** These motors are used in applications where precise control over the position of the motor shaft is required. They are commonly used in robotic arms and satellite positioning systems.
- **Brushless DC motors:** These motors are ideal for applications that require high torque, high speed, and high efficiency. They are commonly used in robotic systems and satellite attitude control systems.

- **Piezoelectric motors:** These motors use piezoelectric materials to generate motion, making them ideal for applications where space is limited. They are commonly used in micro-robotics and space applications.

In short, motors play a crucial role in space and robotics applications, where precise control and compact size are essential.

4.2. Types of Motor:

The two main categories of motors are **DC (Direct Current) motors and AC (Alternating Current) motors.**

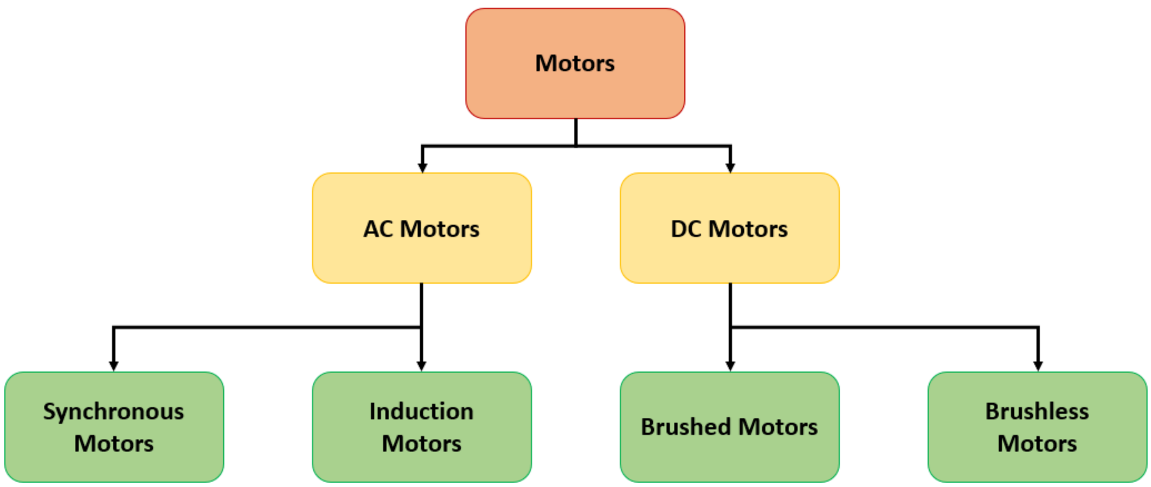


Fig 4.3 Main Categories of Motors

DC motors operate on a direct current power supply, and they are commonly used in a wide range of applications that require variable speed control. DC motors can be classified into several types:

1. **Brushed DC motors:** These motors have a mechanical commutator that transfers electrical energy to the rotor, producing rotational motion. They are widely used in low-power applications, such as toys, tools, and household appliances.
2. **Brushless DC motors (BLDC):** These motors use electronic commutation to control the speed and torque of the motor. They are more efficient and reliable than brushed DC motors and are commonly used in high-power applications, such as electric vehicles, drones, and HVAC systems.
3. **Stepper motors:** These motors are designed to rotate in small, precise steps, making them ideal for applications that require accurate positioning. They are commonly used in robotics, CNC machines, and automation systems.

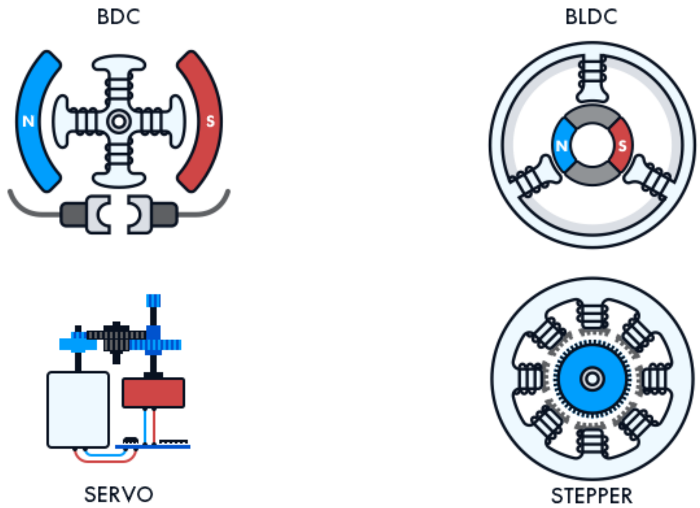


Fig 4.4 Types of DC Motors

AC motors operate on an alternating current power supply and are commonly used in a wide range of applications that require high torque and low speed. AC motors can be classified into several types:

1. **Induction motors:** These motors are the most commonly used type of AC motor. They operate on the principle of electromagnetic induction, which produces rotational motion. They are highly reliable and require little maintenance, making them ideal for industrial applications.
2. **Synchronous motors**: These motors operate at a fixed speed that is synchronized with the frequency of the AC power supply. They are commonly used in applications that require high precision, such as electric clocks and control systems.
3. **Brushless AC motors**: These motors use electronic commutation to control the speed and torque of the motor. They are highly efficient and reliable and are commonly used in high-power applications, such as industrial pumps and fans.

Fig 4.5 AC Motor and DC Motor

4.3. Brushless DC Motors: [18]

A BLDC (Brushless DC) motor is a type of electric motor that operates using electronic commutation to control the speed and torque of the motor. The basic working principle of a BLDC motor is to produce rotational motion using a magnetic field.

The motor consists of a stationary part called the stator, which contains the field winding, and a rotating part called the rotor, which contains the armature winding. The stator winding is energized with an alternating current, which creates a magnetic field that interacts with the magnetic field produced by the permanent magnets on the rotor. The interaction between the two magnetic fields produces rotational motion.

Unlike brushed DC motors, BLDC motors do not have a mechanical commutator. Instead, they use electronic commutation to control the current flow to the armature winding, producing rotational motion. This is done using a set of sensors that detect the position of the rotor and control the flow of current to the armature winding. The electronic commutation is typically achieved using a control circuit that converts the incoming DC voltage into a series of pulses of varying widths.

BLDC motors can be controlled using a variety of techniques. One common method is PWM (Pulse-Width Modulation), which adjusts the voltage and current supplied to the motor to control its speed and torque. Another method is sensorless control, which uses advanced algorithms and signal processing techniques to detect the position of the rotor without the need for external sensors.

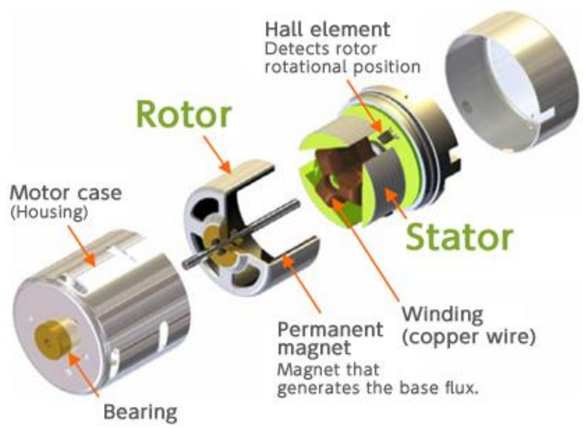


Fig 4.6 Structure of a BLDC Motor [7]

By far the most common configuration for sequentially applying current to a 3-ϕ BLDC motor is to use three pairs of power MOSFETs arranged in a bridge structure, as shown in below Figure. Each pair governs the switching of one phase of the motor. In a typical arrangement, the high-side MOSFETs are controlled using pulse-width modulation (PWM) which converts the input DC voltage into a modulated driving voltage. The use of PWM allows the start-up current to be limited and offers precise control over speed and torque. The PWM frequency is a trade-off between the switching losses that occur at high frequencies and the ripple currents that occur at low frequencies, and which in extreme cases, can damage the motor. Typically, designers use a PWM frequency of at least an order of magnitude higher than the maximum motor rotation speed.

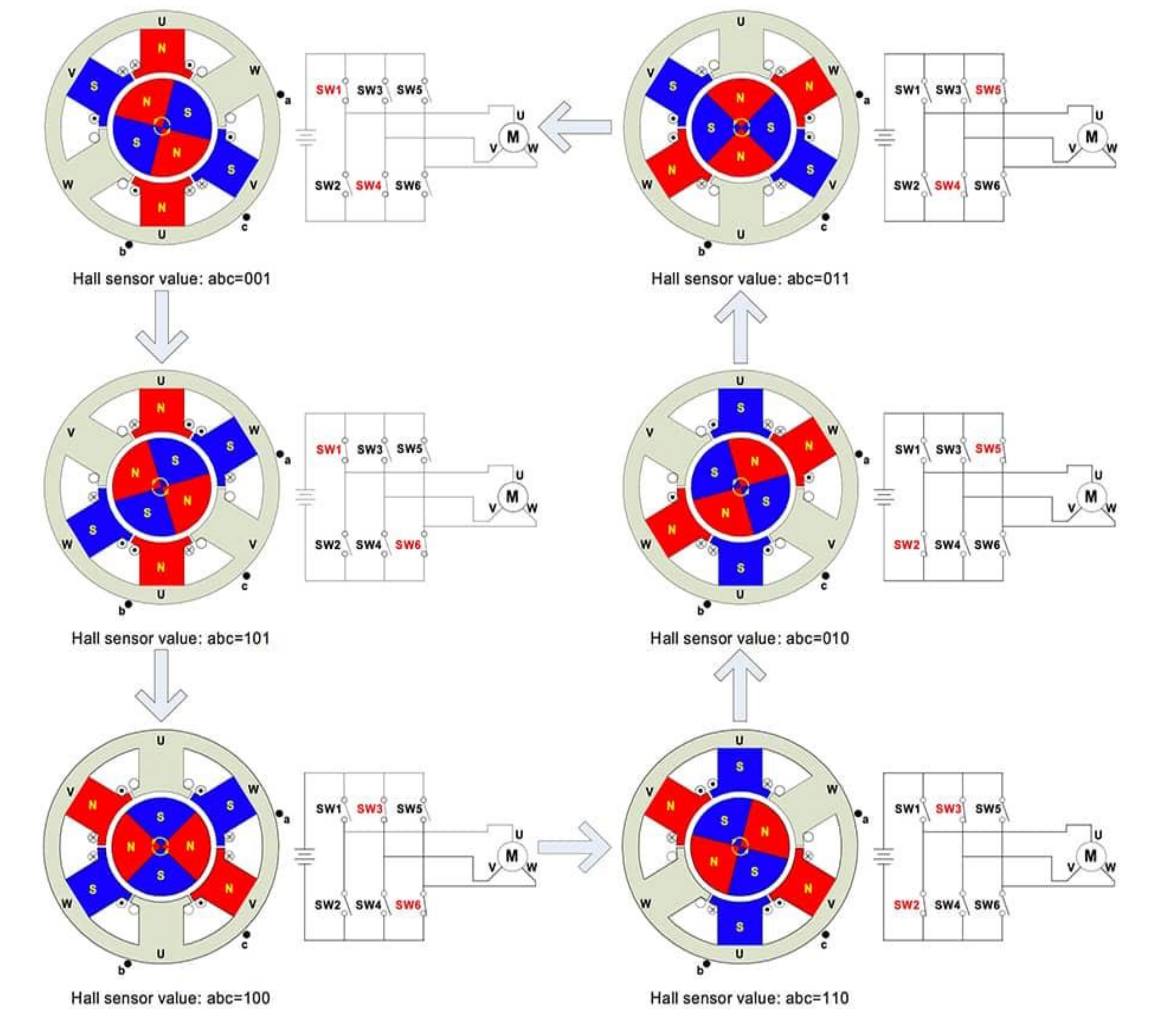


Fig 4.7 How to Power and Control a BLDC Motor [6]

4.3.1. Advantages of BLDC Motor over other types of motor: [9]

There are several advantages of BLDC (Brushless DC) motors over other types of motors, including:

1. High Efficiency
2. High Power Density
3. Precise Speed Control
4. Low Maintenance
5. Long Lifespan
6. Smooth and Quiet Operation

4.3.2. Application of BLDC Motors in Space and Robotics Fields:

BLDC (Brushless DC) motors are widely used in space, satellite, and robotics applications due to their high efficiency, power density, and precise control. Here are some examples:

- **Space Exploration:** BLDC motors are used in spacecraft for various applications, such as attitude control, positioning systems, and reaction wheels. The high-power density and precise control of BLDC motors make them ideal for these critical applications.
- **Satellite Antenna Control**: BLDC motors are used in satellite antenna control systems to accurately position the antenna for communication with the ground station. The precise speed control and high torque of BLDC motors make them suitable for this application.
- **Robotics:** BLDC motors are used extensively in robotics for various applications, such as actuating joints, driving wheels, and powering grippers. The high-power density and precise control of BLDC motors make them ideal for these applications, where weight, size, and efficiency are critical factors.
- **Electric Propulsion:** BLDC motors are also used in electric propulsion systems for satellites and spacecraft. The high efficiency and power density of BLDC motors make them ideal for this application, where minimizing the weight and size of the propulsion system is critical.

Chapter-5 : Introduction to Space Vector Control (SVC)

5.1. Introduction to Field Oriented Control (FOC):

FOC (Field Oriented Control), also known as **Space Vector Control**, is a technique used to control the speed and torque of AC electric motors, including BLDC (Brushless DC) motors. It is a complex control method that determines the ideal voltage and current needed to produce the desired speed and torque using a mathematical model of the motor.

When using FOC, the stator currents are managed in a rotating reference frame that is coordinated with the magnetic field of the rotor. In order to achieve exact control of the motor speed and torque, this enables independent control of the torque and flux components of the motor. FOC's primary benefit is that it gives the motor precise, effective control while minimizing distortion and ripple torque. This makes it perfect for high-performance and precise applications including robots, commercial machines, and electric cars.

FOC is commonly used in modern motor control systems, as it **provides a high level of accuracy and efficiency in controlling the motor**. Its effectiveness has been demonstrated in a wide range of applications, and it is considered to be one of the most advanced and reliable techniques for controlling AC electric motors, including BLDC motors.

5.2. Block Diagram of Field Oriented Control (FOC):

The block diagram of FOC (Field Oriented Control) with SVPWM (Space Vector Pulse Width Modulation) of a BLDC (Brushless DC) motor consists of three main blocks: **the rotor position estimator, the FOC control algorithm, and the SVPWM generator.**

- The rotor position estimator block uses sensors or algorithms to estimate the position and speed of the rotor. This information is essential for the FOC control algorithm to generate the optimal voltage and current vectors for the motor.
- The FOC control algorithm block uses the estimated rotor position and speed information to calculate the optimal voltage and current vectors required to achieve the desired speed and torque. The control algorithm uses PI (Proportional-Integral) controllers to regulate the motor's speed and torque based on the difference between the desired and actual values.
- The SVPWM generator block converts the voltage and current vectors generated by the FOC control algorithm into pulse width modulated signals that are applied to the

motor. The SVPWM technique provides efficient and precise control of the motor by adjusting the duty cycle and switching frequency of the pulse width modulated signals.

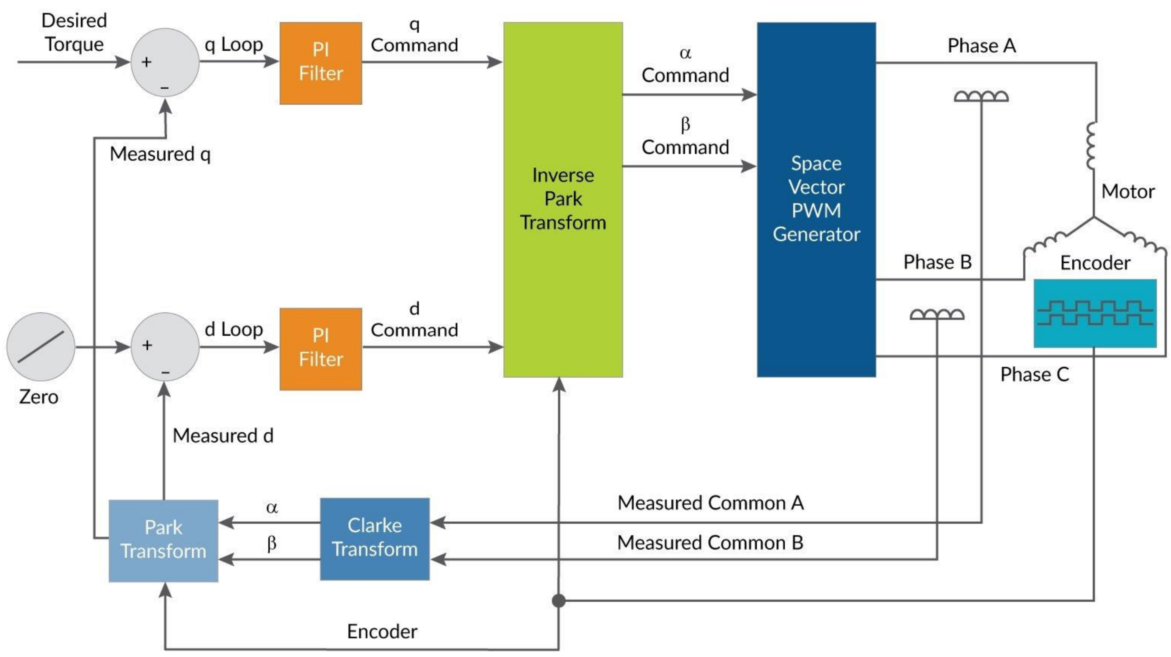


Fig 5.1 Block Diagram of Field Oriented Control System [5]

5.3. Mathematical Transformations in FOC Algorithm [13]:

The Field Oriented Control (FOC) algorithm uses various types of transformations to control the speed and torque of the motor. Here are the most common types of transformations used in FOC:

- **Clarke Transformation:** The Clarke transformation divides the three-phase stator currents into the Alpha (α) and Beta (β) components, two orthogonal components. The torque and flux components of the motor may now be independently controlled thanks to this transformation, which also makes the control method simpler.
- **Park Transformation:** This transformation creates a spinning reference frame that is synchronized with the magnetic field of the rotor from the two-phase direct and quadrature components. This modification makes the control algorithm simpler and enables independent control of the torque and flux components of the motor.
- **Inverse Park Transformation**: Using this transformation, the voltage vectors from the rotating reference frame are converted back into the two-phase direct and quadrature components. The ideal voltage vectors needed to generate the necessary speed and torque are produced using this transformation.

- **Inverse Clarke Transformation:** This technique is used to convert the three-phase stator voltage signals that are applied to the motor from the two-phase direct and quadrature voltage components. The voltage vectors must undergo this transformation in order to return to their original state.

These transformations work together in the FOC algorithm to generate the optimal voltage and current vectors required to achieve the desired speed and torque of the motor. By using these transformations, the FOC algorithm can provide efficient and precise control of the motor, making it ideal for applications that require high performance and precision, such as robotics, industrial machinery, and electric vehicles.

5.3.1. Clarke ($\boldsymbol{\alpha\beta}$) Transform:

The Clarke Transform is a mathematical technique used in the field of electrical engineering to convert the three-phase AC signals of an electric motor into two-phase signals, namely the α- and β-components, which are easier to analyze and control. It works by projecting the three-phase signals onto a two-dimensional space where the α- and β-axes are orthogonal to each other, while the γ-axis is perpendicular to the α-β plane. The resulting transformed signals are used to control the motor's speed, torque, and other parameters, making the Clarke Transform a fundamental component of modern motor control systems. The equations for the Clarke Transform involve simple linear operations on the three-phase signals, making it computationally efficient and widely used in various motor control applications.

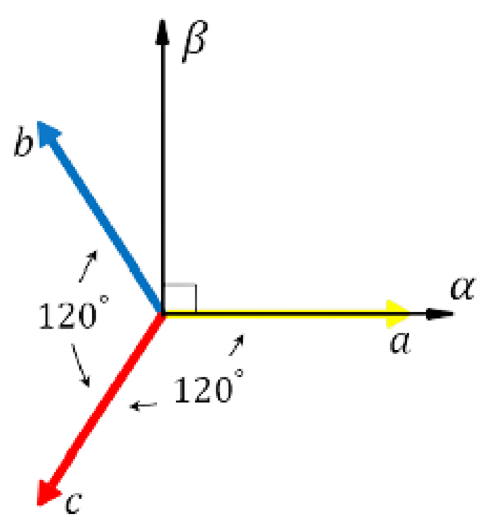


Fig 5.2 Clarke Transform Reference Frames

The following equation describes the Clarke transform:

$$\begin{bmatrix}\alpha\\ \beta\\ 0\end{bmatrix} = \left(\frac{2}{3}\right) \times \begin{bmatrix}1 & -\frac{1}{2} & -\frac{1}{2}\\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2}\\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}a\\ b\\ c\end{bmatrix} \qquad 5.1$$

For balanced systems like motors, the zero-sequence component calculation is always zero. For example, the currents of the motor can be represented as,

$$i_a + i_b + i_c = 0 \qquad 5.2$$

Therefore, you can use only two current sensors in three-phase motor drives, where you can calculate the third phase as,

$$i_c = -(i_a + i_b) \qquad 5.3$$

By using these equations, the block implements the Clarke transform as,

$$\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}1 & 0\\ \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{3}}\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix} \qquad 5.4$$

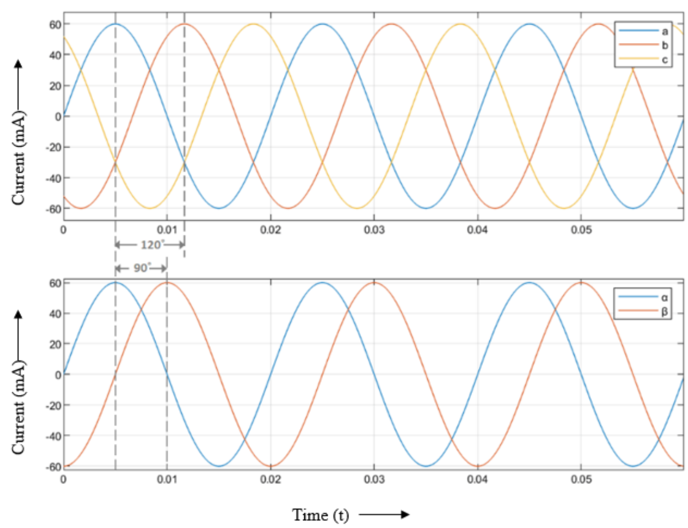


Fig 5.3 Clarke Transform abc to αβ Transformation

5.3.2. Park (DQ) Transform:

The Park transform, also known as the dq transform, is a mathematical technique used to transform a two-dimensional signal from a stationary reference frame to a rotating reference frame. This transform is commonly used in the field of electric machines and drives to simplify the control of the machines. The Park transform takes a two-phase signal (i_α, i_β) and the rotor angle (θ) as inputs, and outputs a DC signal (i_d) and a quadrature signal (i_q).

These transformed signals are easier to control as they correspond to the direct and quadrature axes of the rotating reference frame, which are perpendicular to each other. The Park transform is a key component of the Field-Oriented Control (FOC) technique, which is widely used in the control of electric machines.

There are 2 types of Park Transformation's Equations:

1) When the d-axis aligns with the α-axis.
2) When the q-axis aligns with the α-axis

5.3.2.1 Park Transformation, when the d-axis aligns with the α-axis:

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad 5.5$$

where,

- $[\alpha, \beta]$ is the column vector representing the three-phase voltage or current in the abc reference frame;
- [d, q] is the column vector representing the transformed voltage or current in the dq reference frame.

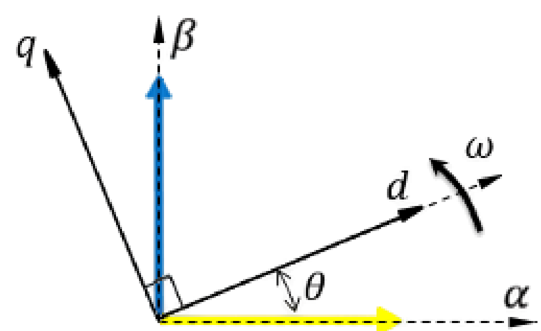


Fig 5.4 Park Transform Reference Frames, d-Axis Aligns with the α-axis

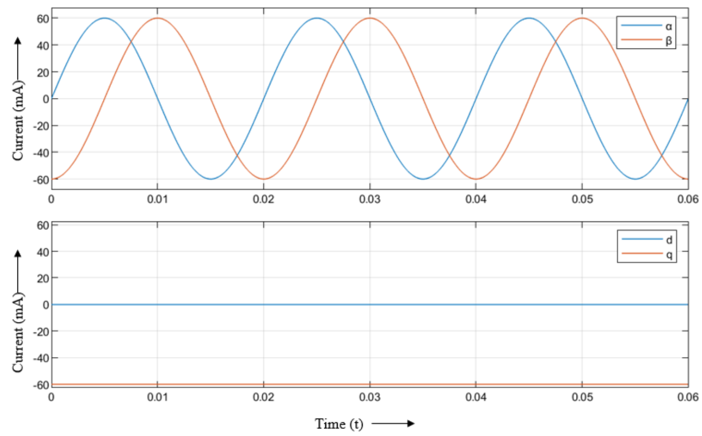


Fig 5.5 Park Transform αβ to dq Transformation, when the d-axis aligns with α-axis

5.3.2.2 Park Transformation, when the q-axis aligns with the α-axis:

$$\begin{bmatrix} d \\ q \end{bmatrix} = \begin{bmatrix} sin\theta & -cos\theta \\ cos\theta & sin\theta \end{bmatrix} * \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \quad 5.6$$

where,

- $[\alpha, \beta]$ is the column vector representing the three-phase voltage or current in the abc reference frame;
- [d, q] is the column vector representing the transformed voltage or current in the dq reference frame.

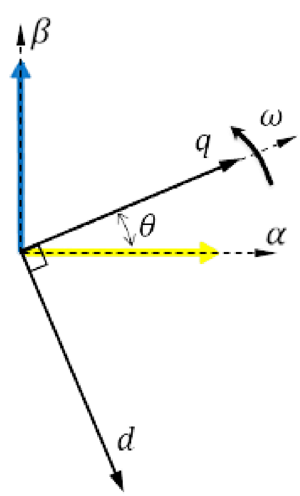


Fig 5.6 Park Transform Reference Frames, q-Axis Aligns with the α-axis

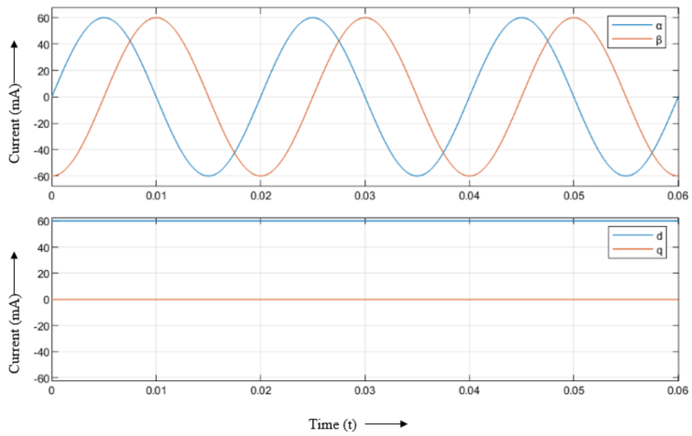


Fig 5.7 Park Transform αβ to dq Transformation, when the q-axis aligns with α-axis

5.3.3. Inverse Park Transform:

The Inverse Park transform is the reverse of the Park transform and is used to convert the DC current with a rotating reference frame back into two-phase currents. The Inverse Park transform takes the DC current (i_d) and the quadrature current (i_q) as input, along with the rotor angle (theta), and outputs the two-phase currents (i_α, i_β).

The Inverse Park transform is commonly used in motor control applications, where the DC current with a rotating reference frame needs to be converted back to two-phase currents in order to control the motor using the Field-Oriented Control (FOC) technique. The Inverse Park transform, along with the Park transform, is an essential part of FOC-based motor control systems.

There are 2 types of Park Transformation's Equations:

1) When the d-axis aligns with the α-axis.
2) When the q-axis aligns with the α-axis

5.3.3.1 Inverse Park Transformation, when the d-axis aligns with the α-axis:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} * \begin{bmatrix} d \\ q \end{bmatrix} \quad 5.7$$

where:

- [d, q] is the column vector representing the transformed voltage or current in the dq reference frame;

- $[\alpha, \beta]$ is the column vector representing the voltage or current in the abc reference frame.

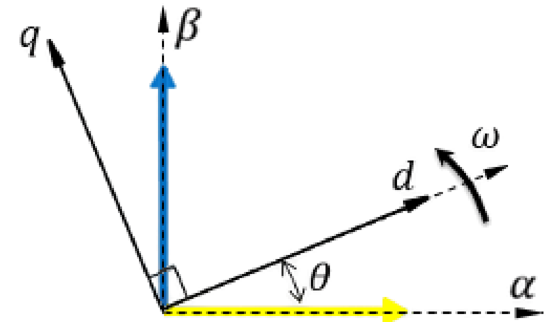


Fig 5.8 Inverse Park Transform Reference Frames, d-Axis Aligns with the α-axis

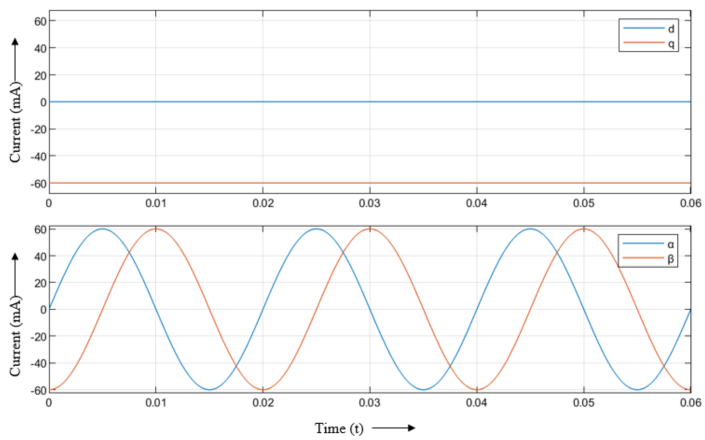


Fig 5.9 Inverse Park Transform dq to αβ Transformation, when the d-axis aligns with α-axis

5.3.3.2 Inverse Park Transformation, when the d-axis aligns with the α-axis:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} sin\theta & cos\theta \\ -cos\theta & sin\theta \end{bmatrix} * \begin{bmatrix} d \\ q \end{bmatrix} \quad 5.8$$

where:

- [d, q] is the column vector representing the transformed voltage or current in the dq reference frame;
- $[\alpha, \beta]$ is the column vector representing the voltage or current in the abc reference frame.

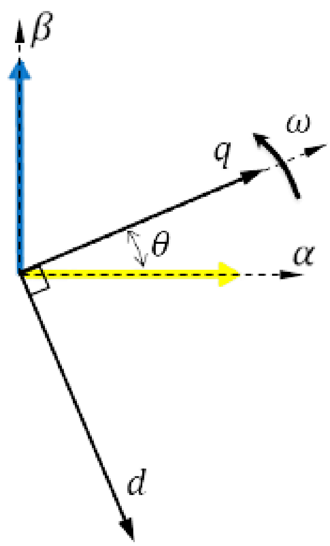


Fig 5.10 Inverse Park Transform Reference Frames, q-Axis Aligns with the α-axis

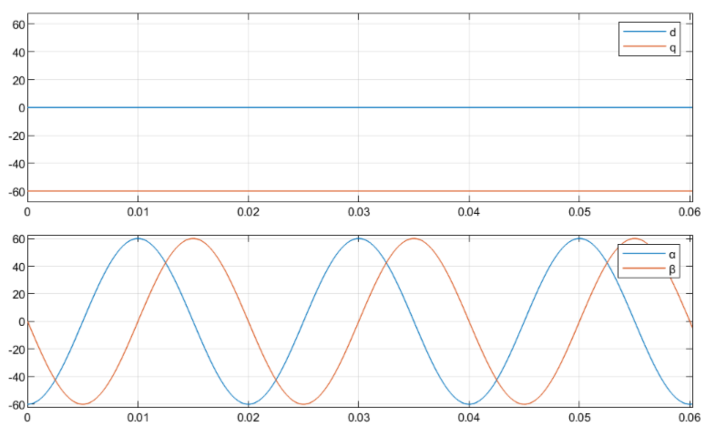


Fig 5.11 Inverse Park Transform dq to αβ Transformation, when the q-axis aligns with α-axis

5.3.4. Inverse Clarke Transform:

The Inverse Clarke Transform is used to convert the two-phase reference frame back to the three-phase reference frame. It is the inverse operation of the Clarke transform and is often used in motor control systems to convert the control signals from the rotating reference frame back to the stationary reference frame. The Inverse Clarke Transform takes the two transformed currents (i_α, i_β) as input and outputs the three-phase currents (i_a, i_b, i_c) in the stationary reference frame. The Inverse Clarke Transform equations are derived by using the Clarke Transform equations and solving for the original currents. The Inverse Clarke

Transform is an important tool in motor control systems that use FOC, as it allows the control signals to be transformed back to the original reference frame for implementation in the motor.

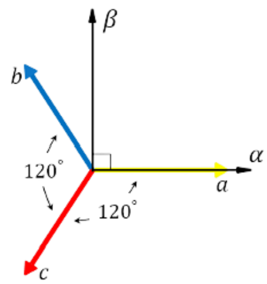


Fig 5.12 Inverse Clarke Transform Reference Frames

The following equation describes the Inverse Clarke transformation:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ -1/2 & \sqrt{3}/2 & 1 \\ -1/2 & -\sqrt{3}/2 & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ 0 \end{bmatrix} \qquad 5.9$$

where,

- $[\alpha, \beta]$ is the column vector representing the transformed voltage or current in the $\alpha\beta$ reference frame;
- [a, b, c] is the column vector representing the voltage or current in the abc reference frame.

For balanced systems like motors, the zero-sequence component calculation is always zero. For example, the currents of the motor can be represented as,

$$i_a + i_b + i_c = 0 \qquad 5.10$$

Therefore, you can use only two current sensors in three-phase motor drives, where you can calculate the third phase as,

$$i_c = -(i_a + i_b) \qquad 5.11$$

By using these equations, the block implements the Clarke transform as,

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \qquad 5.12$$

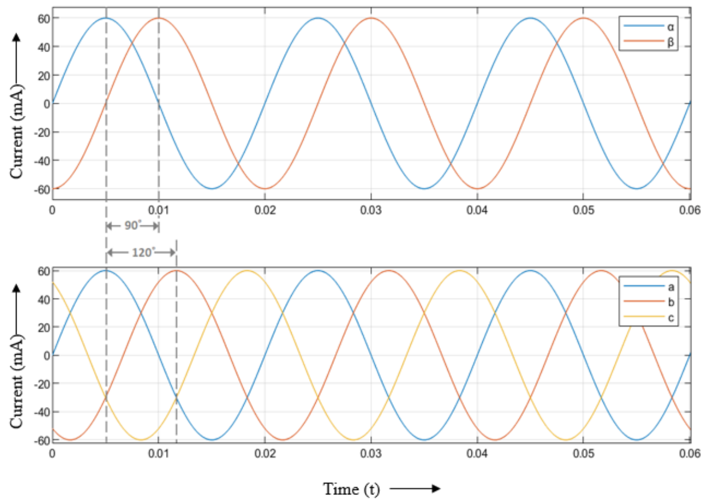


Fig 5.13 Inverse Clarke Transform αβ to abc Transformation

5.4. Sinusoidal Pulse Width Modulation (SPWM):

Sinusoidal pulse width modulation is a simple and most commonly used modulation technique for inverter and motor control applications which comes after the Inverse Clarke Transformation in Field Oriented Control system. In this method, the three sine signals are compared with high frequency carrier triangular signal to generate switching pulses for inverter. The three sinusoidal sine waves each having 120˚ phase shift is generated from dq to abc converter block are utilized for PWM generation. The Carrier Wave is chosen minimum 11 times the sinusoidal wave. The comparator provides switching pulses and these pulses are used to trigger respective inverter switches.

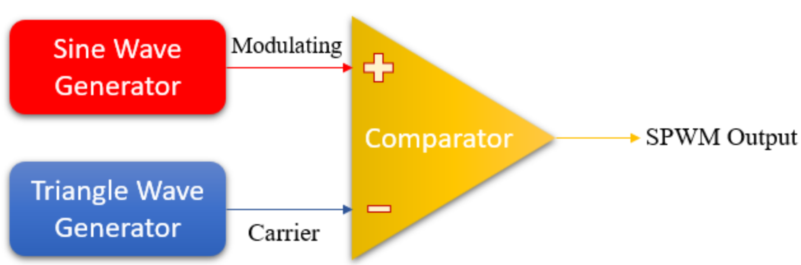


Fig 5.14 Simplified Schematic of Sinusoidal PWM Modulation

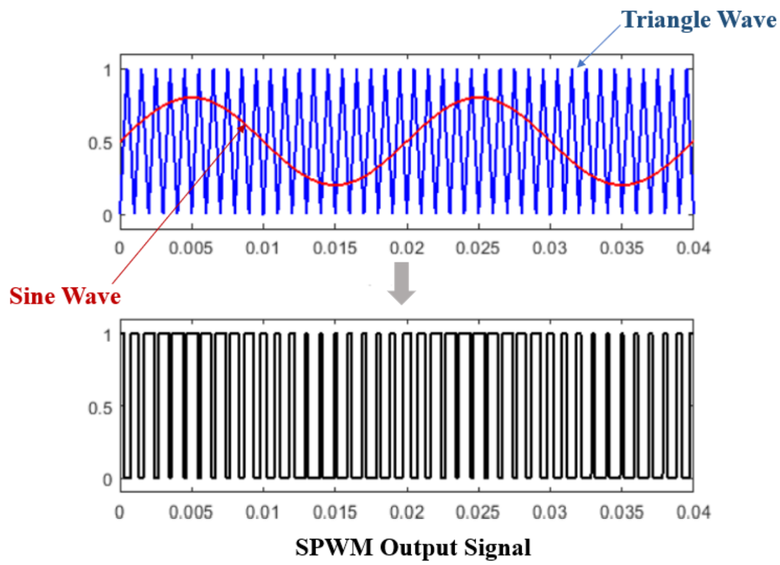


Fig 5.15 Sinusoidal PWM Modulation and Output Signal [20]

5.5. Space Vector Pulse Width Modulation (SVPWM):

The space vector pulse width modulation is an advanced PWM method. It provides better utilization DC bus voltage compared to Inverse Clarke SPWM. SVPWM has superior characteristics like reduced harmonic distortion, high output quality and low rating of filter components [11].

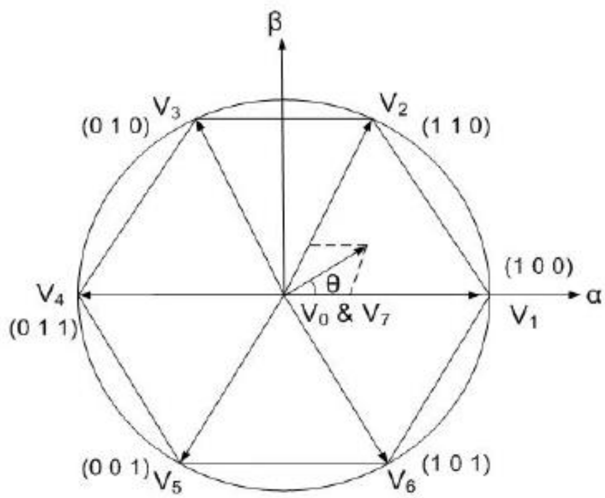


Fig 5.16 Vector Representation of SVPWM

Above Figure 5.16 shows the space vector representation of three phase system, in which the reference vector (V_{ref}) rotates with angular velocity with respect to stationary orthogonal α-β reference frame. Here α-axis coincides with phase voltage V_a axis. The magnitude and frequency of fundamental components are controlled by controlling magnitude and frequency of V_{ref}.

The desired three phase output voltage equations can be represented as below:

$$V_a = V_m \cos(\omega t) \qquad 5.13$$

$$V_b = V_m \cos(\omega t - 120°) \qquad 5.14$$

$$V_c = V_m \cos(\omega t + 120°) \qquad 5.15$$

where,

- V_m is the magnitude of the output voltage signals;
- ωt = θ is the angle of the reference vector in the abc reference frame.

Below Figure 5.17 shows the representation of sector-1 with the reference voltage. Here,

V_{ref} can be defined as,

$$V_{ref} = \sqrt{\frac{3}{2}} V_m e^{j\theta} \qquad 5.16$$

where,

- $\theta = \omega t = 2\pi f t$

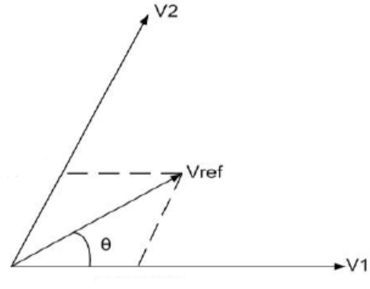


Fig 5.17 Vector Representation of V_{ref} and θ

In SVPWM, the reference voltage V_{ref} is compared with the voltage vectors V_1 and V_2 generated by the three-phase inverter to control the motor. The aim is to synthesize the desired voltage vector with the proper duration by applying appropriate switching signals to the inverter.

$$V_{ref}T_z = V_1T_1 + V_2T_2, \tag{5.17}$$

where,

T_z is the switching period, relates the reference voltage to the durations of the voltage vectors. It states that the area under the voltage waveform for one switching cycle is equal to the reference voltage multiplied by the switching period T_z.

The angle Φ represents the phase angle between the reference voltage vector and the voltage vector generated by the inverter. The angle θ is the electrical angle between the rotor position and the reference frame. The equation:

$$\Phi = \theta - (N\text{-}1)60 \tag{5.18}$$

relates the angle Φ to the electrical angle θ and the sector N (N=1, 2, 3...,6) in which the motor is operating. The switching periods T_1 and T_2 for the voltage vectors V_1 and V_2, respectively, are determined using the equations:

$$T_1 = T_Z \frac{2}{\sqrt{3}} \frac{V_{ref}}{V_1} \sin(60 - \Phi) \tag{5.19}$$

$$T_1 = T_Z \frac{2}{\sqrt{3}} \frac{V_{ref}}{V_2} \sin(\Phi) \tag{5.20}$$

The switching period T0 for the zero-voltage vector is simply:

$$T_0 = T_Z - (T_1 + T_2) \tag{5.21}$$

The equations essentially determine the duration of each voltage vector in a switching cycle based on the reference voltage and the angle between the reference voltage and the voltage vectors generated by the inverter. The switching signals are then applied to the inverter to synthesize the desired voltage vector and control the motor [11].

5.5.2. Compare SVPWM with SPWM:

Table 5.1 Comparison Between SVPWM and SPWM

Properties	SVPWM	SPWM
1. Utilization of DC bus voltage:	• Better utilization • Provide higher torque output.	• Don't utilize as good as SVPWM • Provide lesser torque output.
2. Dynamic response and torque ripple:	• Better dynamic response • Lower torque ripple.	• Inferior dynamic response • Higher torque ripple.
3. Harmonic performance and power factor:	• Better Harmonic Performance • Higher Power Factor.	• Inferior Harmonic Performance • Lower Power Factor.
4. Computational complexity:	• More Computationally intensive, • Requires more memory and processing power. • Also requires more complex hardware implementation.	• Less Computationally intensive • Requires Less memory • Requires Less processing power. • Requires Less complex hardware implementation.
5. Compatibility with different motor types:	• SVPWM, is primarily used for BLDC motors.	• Generally, more compatible with different types of AC motors, including induction motors, synchronous motors, and BLDC motors.

5.5.3. Mathematical Equivalence of SVPWM and ICT:

Both SVPWM and the inverse Clarke transform are methods for generating three-phase voltage waveforms from two-phase voltage signals. The outputs of both methods can be expressed in terms of the three-phase voltages V_a, V_b, and V_c.

The Equations of Inverse Clarke Transformations are:

$$V_a = V_\alpha \quad (5.22)$$

$$V_b = -\frac{1}{2}V_\alpha + \frac{\sqrt{3}}{2}V_\beta \quad (5.23)$$

$$V_c = -\frac{1}{2}V_\alpha - \frac{\sqrt{3}}{2}V_\beta \quad (5.24)$$

and, The Equations of Space Vector Pulse Width Modulation Transformation are:

$$V_a = V_m \cos(\omega t) \quad 5.25$$

$$V_b = V_m \cos(\omega t - 120°) \quad 5.26$$

$$V_c = V_m \cos(\omega t + 120°) \quad 5.27$$

where,

- V_a, V_b, and V_c are the voltages in the abc reference frame;
- V_α and V_β are the voltages in the alpha-beta reference frame;
- V_m is the magnitude of the output voltage signals;
- And, ωt = θ is the angle of the reference vector in the abc reference frame.

To prove, the equivalence of the two sets of equations, we substitute the expression for V_a from the SVPWM's equation set into the ICT's equation set, yielding:

$$V_a = V_\alpha \quad 5.28$$

$$V_m * \cos(\theta) = V_\alpha \quad 5.29$$

$$V_m = \frac{V_\alpha}{\cos(\theta)} \quad 5.30$$

We can then substitute this expression for V_m into the expressions for V_b and V_c from the SVPWM's equation set, yielding:

$$V_b = V_m * \cos(\theta - 120°) \quad 5.31$$

$$V_b = (V_\alpha / \cos(\theta)) * \cos(\theta - 120°) \quad 5.32$$

$$V_b = (-\frac{1}{2} V_\alpha \cos(120°)) + \frac{\sqrt{3}}{2} V_\beta \sin(120°) \quad 5.33$$

$$V_b = -\frac{1}{2} V_\alpha + \frac{\sqrt{3}}{2} V_\beta \quad 5.34$$

Similarly, we can show that V_c in the Inverse Clarke Transform's Equation set is equivalent to $V_m * \cos(\theta + 120°)$ in the SVPWM's equation set.

These equations show that the output of SVPWM is equivalent to the output of the inverse Clarke transform. Therefore, SVPWM and the inverse Clarke transform are mathematically equivalent.

5.5.4. Third Harmonic PWM:

The sinusoidal PWM with third harmonic injection (THSPWM) is a type of sinusoidal pulse width modulation where a harmonic component is added to the voltage signal so that the waveform of the modulating signal has its top flattened and one reduces the period of over modulation.

In this method**,** the reference signal is not a pure sinusoidal wave, but of fundamental wave and the third harmonic. In the Below Figure 5.18, the Blue colored Distorted Wave is the SVPWM's Resultant waveform.

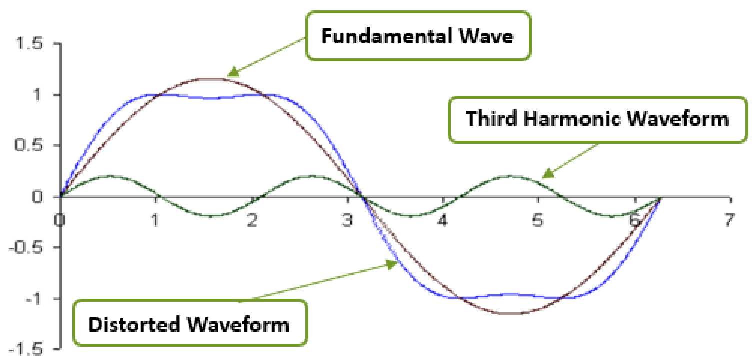


Fig 5.18 Third Harmonic Waveform

5.5.5. PWM Generation by SVPWM is more efficient than SPWM [20]:

The Space Vector Modulation technique uses around 15% more of the available bus voltage than third harmonic-injected sinusoidal PWM, boosting the efficiency of motor operation.

$$V_{SVPWM} = 1.15 \times V_{SPWM} \qquad 5.35$$

The neutral point of the phase voltages is limited to half of the bus voltage, unlike a sinusoidal PWM that does not inject a third harmonic, as seen in Figure 5.19.

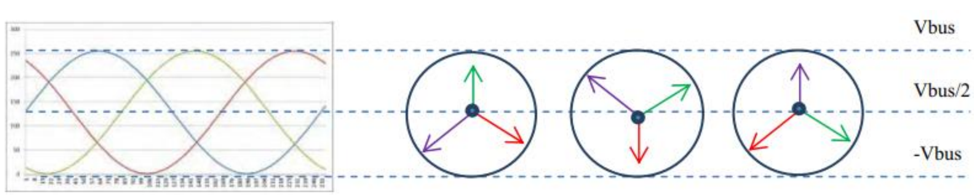


Fig 5.19 The Rotating Vectors are constrained by $\pm V_{bus}$ and the Center of the V_{bus}

Space vector modulation can float in space, as seen in Figure 5.20, and is not limited by the boundaries of the V_{bus} and the center voltages.

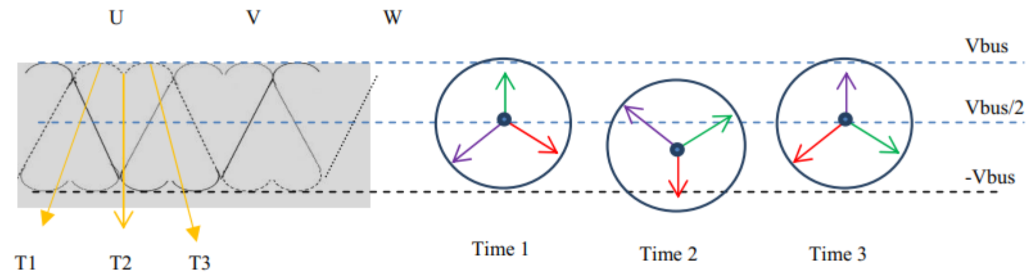


Fig 5.20 Center Voltage of the Space Vector Floats in Space

Space vector modulation controls the entire inverter as a single unit to produce the sinusoidal currents, in contrast to sinusoidal PWM, which generates sinusoidal currents independently in each push/pull stage of the inverter. Thus, the inverter operates in eight different states within the hexagon, two of which are known as zero vectors because they produce no voltages, and six of which produce non-zero voltages.

The rotating reference voltage V_s (also represent as V_{ref}) within this hexagon, is represented by a space vector using the equation, as seen in Figure 5.21:

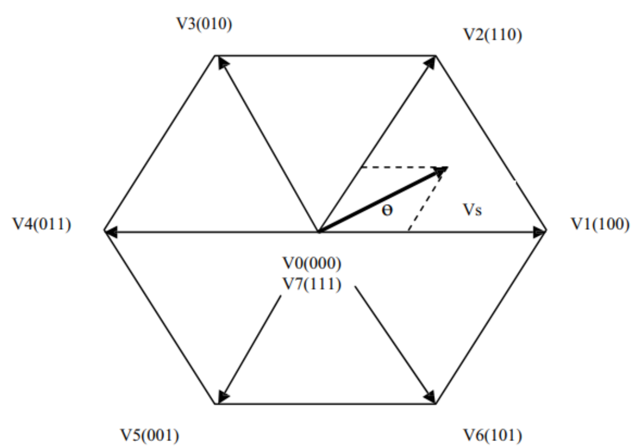


Fig 5.21 The Rotating Reference Vector V_s (V_{ref}) within the Hexagon

5.6. Simulations of Field Oriented Control with SVPWM of BLDC Motor:

The simulation of the implementation of Field Oriented Control (FOC) with Space Vector Pulse Width Modulation (SVPWM) on Brushless Direct Current (BLDC) motors is an important step in the development and optimization of motor control systems. The simulation allows engineers to test and analyze the performance of the system before the actual implementation, which saves time and resources.

The simulation model, as shown in below Figure 5.22, includes various components such as FLC with derivative filter, PI controller, Clarke, Park, and Inverse Transformation, user-defined function named "SVPWM" with input alpha and beta, and output as a gating signal, and BLDC motor with Trapezoidal Back EMF takes in input as T_m, and 3 Phase A, B, and C. The model is designed to work with MATLAB & Simulink software, which are widely used in the field of control systems.

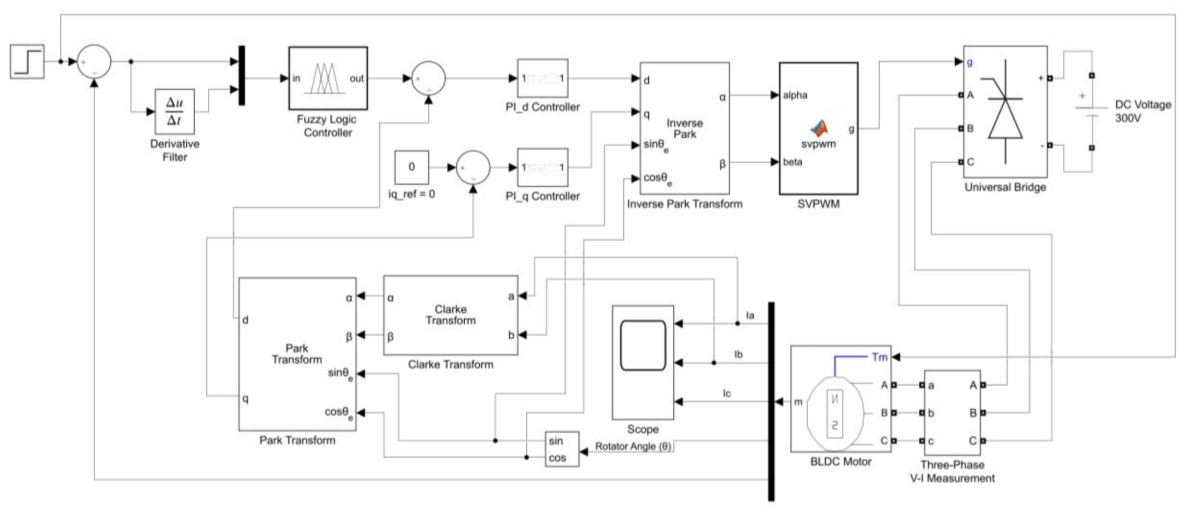


Fig 5.22 Simulation Model of Field Oriented Control (with SVPWM) on BLDC

The FOC technique is used to control the speed and torque of the motor by manipulating the stator current in a way that aligns the magnetic field produced by the stator with the rotor's magnetic field. This technique requires the measurement of the rotor position, which is done using sensors or sensorless methods. The SVPWM is a technique used to generate the gating signals that control the switching of the inverter that drives the motor. This technique provides better utilization of the DC bus voltage and reduces the harmonic distortion of the output waveform.

The simulation results are shown in Figure 5.23, which displays the scope reading of stator current. The y-axis shows the stator current, and the amplitude of the current is 0.5A (500mA). The yellow-colored wave is I_a, the blue-colored wave is I_b, and the red-colored wave is I_c. All these waves have a phase shift of 120° with each other, which indicates that the three-phase current is balanced and produces a rotating magnetic field that interacts with the rotor's magnetic field.

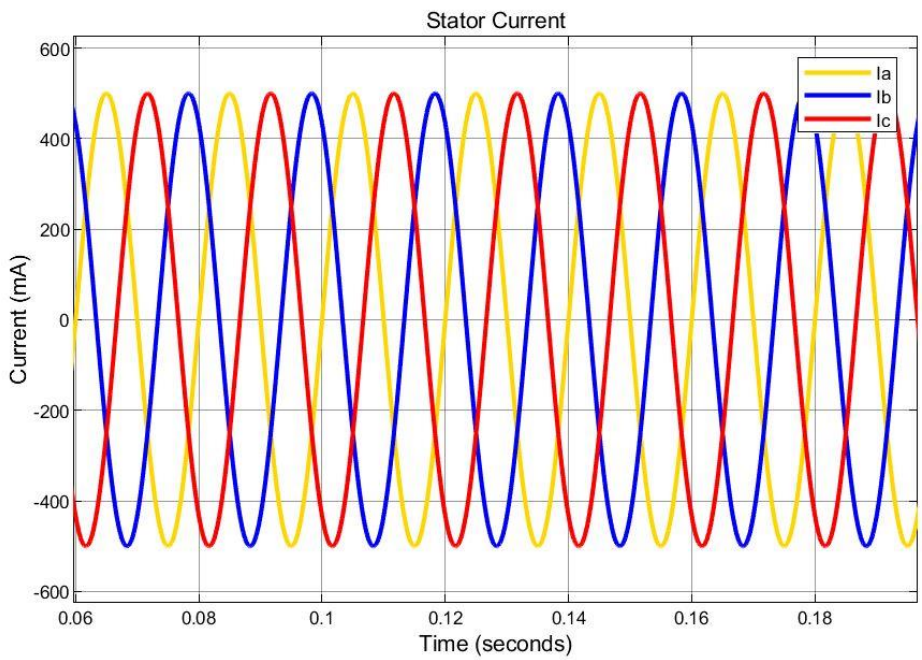


Fig 5.23 Simulation Results shows Sinusoidal 120° Phase Shift

The simulation allows the engineers to analyze the performance of the system under different operating conditions such as varying load and speed. The simulation results can be used to optimize the system's parameters such as PI gains and FLC coefficients, which can improve the system's performance and efficiency. The simulation also allows the engineers to test different fault conditions such as overvoltage, overcurrent, and short circuit, which can help in designing robust control systems that can handle such situations.

In conclusion, the simulation of the implementation of FOC with SVPWM on BLDC motors is an important step in the development and optimization of motor control systems.

Chapter-6 : Hardware Results of Motor Control Implementation

6.1. Introduction:

The implementation of motor control is a fundamental aspect of many engineering applications, and as such, it is an area of significant interest and development. This chapter focuses on the hardware implementation of motor control, with specific attention given to the use of an ARM Cortex-M7 microcontroller to control a 3-phase BLDC motor using the LX7720DB motor controller. The chapter provides detailed information on the hardware setup schematics and connection pinout used in the implementation process. Additionally, the chapter outlines a comprehensive methodology for the implementation process, including step-by-step instructions and a code flowchart. Finally, the chapter presents the results of the motor control implementation, highlighting the performance and efficiency of the system.

6.2. Project Overview:

The project aims to implement a motor control system using ARM Cortex-M7 microcontroller and Microsemi LX7720DB Motor Controller to control the Hurst's BLDC motor. The software used for this project includes Microchip's MPLAB X IDE, XC32 compiler, and Harmony v3 Framework/Configurator.

6.2.1. Goal of the Project:

The primary goal is to demonstrate the practical implementation of a motor control system using the ARM Cortex-M7 microcontroller and 3 Phase Channel Motor Controller. The project aims to show the versatility and effectiveness of the ARM Cortex-M7 Microcontroller for motor control applications, especially in controlling the BLDC motor. Additionally, the project aims to showcase the use of advanced motor driver technology to provide precise and efficient control of the motor.

6.2.2. Objective of the Project:

The project objectives include selecting and integrating the hardware components, designing and implementing the control algorithm, and testing the motor control system under various operating conditions. The project also aims to evaluate the performance of the motor control system and to identify any limitations or challenges encountered during the implementation process. Finally, the project aims to provide valuable insights into the practical considerations and methodologies involved in designing and implementing motor control systems using advanced microcontrollers and motor drivers.

6.3. Getting Started with MPLAB X IDE and Harmony v3 Framework:

Here is a Step-By-Step Guide on how to set up the tools required for the project, and start learning by implementing a simple example project like LED_Blinking that will show you how to do it:

- **Step-1:** Visit Microchip's website and download the required software tools, including MPLAB X IDE, XC32 compiler, and Harmony v3 Framework/Configurator.
- **Step-2:** Connect the ARM Cortex M7 microcontroller to your computer using a USB cable. You can power the microcontroller through USB or an external power supply.
- **Step-3:** Open MPLAB X IDE and create a new project. Select the microcontroller model, specify the clock frequency, and set up the I/O pins for the LED_Blinking project.
- **Step-4:** Write the code using the XC32 compiler and Harmony v3 Framework/Configurator to implement the LED_Blinking project. Use C language and include the required libraries and functions to control the LEDs.
- **Step-5:** Compile and build the project in MPLAB X IDE to generate the necessary files for programming the microcontroller.
- **Step-6:** Program the microcontroller using MPLAB X IDE and test the LED_Blinking project. Connect the LED to the microcontroller's I/O pin and observe it blinking.

As a result of following these steps, users will be able to set up the necessary tools and begin learning by implementing a simple example project, such as LED_Blinking, starting with the installation of the required tools. Microchip's software tools provide a practical way of understanding the basics of programming the ARM Cortex M7 microcontrollers using Microchip's software tools.

6.4. An overview of the hardware connection schematics and pinouts:

6.4.1. Hardware Components:

This report focuses on the implementation of motor control using ARM Cortex microcontroller, and this chapter specifically details the hardware components used in the project. The following is a detailed description of the hardware components used in this project:

1. **Microcontroller:** The microcontroller used in this project is Microchip's ARM Cortex M7 Microcontroller. This microcontroller is ideal for use in space technology due to its radiation-hardened and SpaceWire features. It has a clock speed of 10 MHz and offers high-performance computing capabilities.

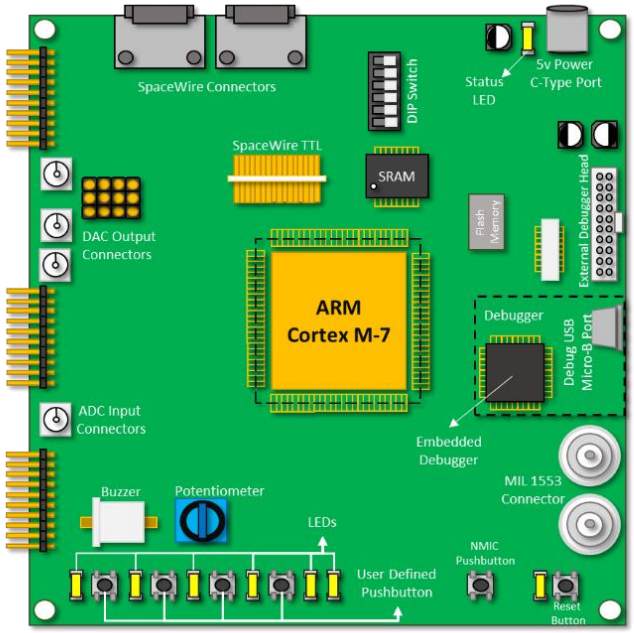


Fig 6.1 ARM Cortex-M7 Microcontroller-Evaluation Kit Diagram

2. **Motor Driver:** The motor driver used in this project is Microsemi LX7720 Motor Controller. It is a Rad Tolerant Spacecraft Motor Controller, that provides a wide range of motor driving functions from open-loop cardinal step driving to space vector modulation using field-oriented control. The motor driver comes with FPGA IP modules to support motor driving functions, making it an ideal choice for the project.

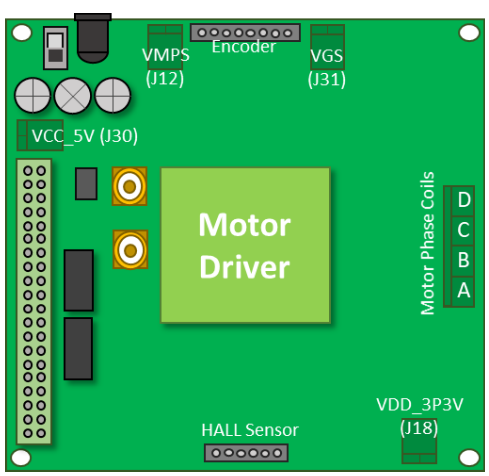


Fig 6.2 Microsemi LX7720 Motor Controller Diagram

3. **BLDC Motor:** The Hurst's 3-phase BLDC motor was used in this project. This motor offers high efficiency, high torque density, and is easy to control. It has a power rating of 250 W and operates at 24 V DC.

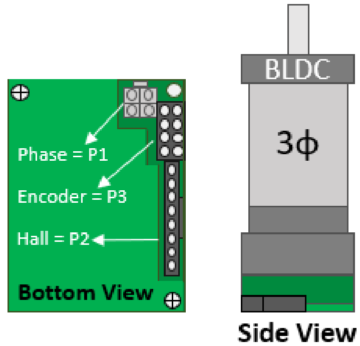


Fig 6.3 Hurst's DMA0204024B1010 BLDC Motor Diagram (Bottom and Side View)

In conclusion, the hardware components used in this project were carefully selected to ensure optimal performance and reliability.

6.4.2. Harness Setup and Connections:

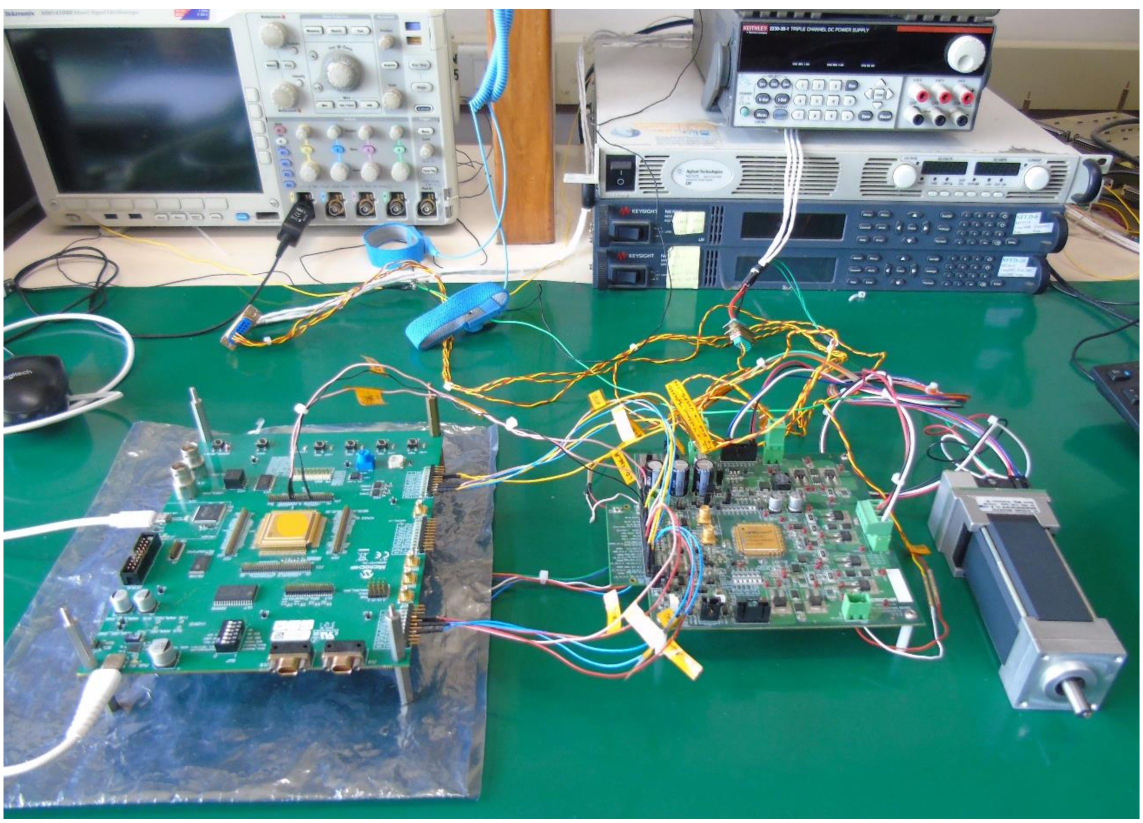


Fig 6.4 Harness Setup at the workspace

The harness setup is a crucial component of the practical implementation of the motor control project. This subtopic will provide a detailed description of the hardware used in the project as shown in the below figure, including the ARM Cortex M7 Microcontroller, 3 Phase Channel Motor Controller, Hurst's BLDC Motor, and associated power supplies and connectors.

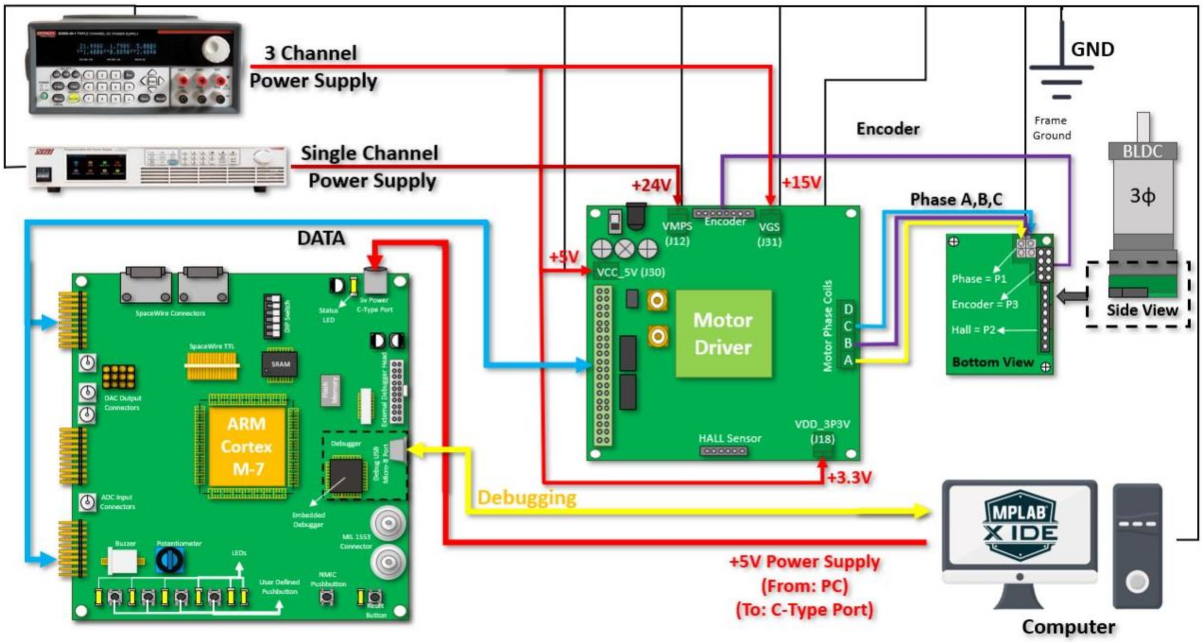


Fig 6.5 The Harness Setup Schematic Diagram

Starting with the Microcontroller, the power supply is sourced from a computer using a C-type power cable. Debugging is done using a Micro USB cable with the computer and debugging in MPLAB X IDE software. The data jumper cables, made by simple wires and HD sockets, are connected to the Motor Controller's J20 Headers from the J19 and J18 headers of the microcontroller evaluation kit board.

The 3 Phase Channel Motor Controller is connected to the microcontroller, with power supply coming from a single supply (+24V from VMPS), +15V from Channel 1 of Triple Channel Power Supply for VGS, +5V from Channel 2 of Triple Channel Power Supply for VCC, and +3.3V from Channel 3 of Triple Channel Power Supply for VDD. The encoder connectors (VDD, GND, A, and ~A) are connected with BLDC Motor's P3 Connectors. The motor coils (Phase A, B, and C) are connected with the BLDC Motor's P1 Connectors.

Finally, the Hurst's BLDC Motor is connected to the Motor Driver with its P1 Connectors Frame GND connected to the common ground of the complete harness. There is a common ground between computers, oscilloscopes, power supplies, microcontrollers, motor drivers, and BLDC motors.

The harness setup is vital to the success of the project, and a step-by-step guide on how to set up the tools and start learning by an example project like LED_Blinking can help others understand how to replicate this setup for similar projects. It is essential to ensure that all connections are made correctly and that the correct power supply is provided to each component.

In conclusion, the practical implementation of motor control using an ARM Cortex microcontroller requires careful planning and attention to detail in the harness setup.

6.4.2.1 Connection Pinouts:

For a detailed illustration of the pinouts and connections required for proper hardware communication, it can refer from [Appendix 1].

6.5. Methodology:

The methodology section of a project report is crucial as it outlines the steps and processes followed to achieve the project objectives. It provides a clear understanding of the approach used to collect, analyze and interpret data, and the methods used to reach conclusions. The methodology section also includes the research design, sampling methods, data collection techniques, data analysis procedures, and ethical considerations. It is important to ensure that the methodology is accurately documented to enable others to replicate the study and assess its validity.

This is where the Step-by-Step Methodology begins, Open MPLAB® and execute the following steps:

Step-1: From the top-level menu, select File > New Project.

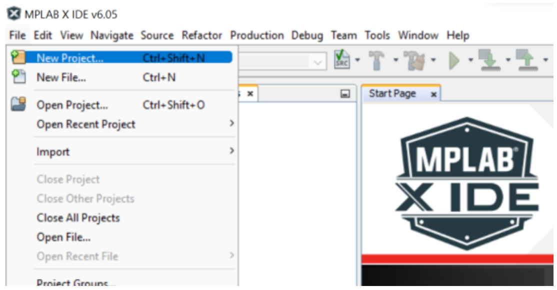


Fig 6.6 Open a New Project in MPLAB X IDE

Step-2: In the New Project/Choose Project window, select Microchip Embedded from Categories menu then select 32-bit MCC Harmony Project from the Projects menu. Click the Next button.

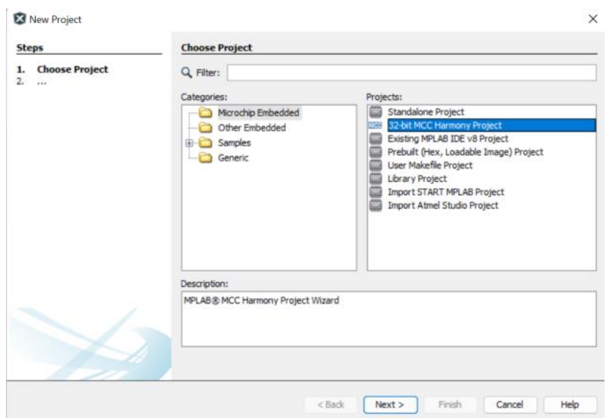


Fig 6.7 Select 32-bit MCC Harmony Project in Microchip Embedded

Step-3: In the New Project / Manage Framework window, update the Framework Path field if necessary. Note: Even if the displayed Framework Path is correct, for the first time through this wizard, user must click on browse (yellow folder icon on the right) and navigate to the path to save it correctly. Click the Next Button.

Fig 6.8 Enter the Harmony Framework Repository and its path

Step-4: In the New Project/Name and Location window, enter the Location and Folder fields. For Instance, my_project for the folder name and bldc_foc_encoder_arm7 for Name of project. Click the Next Button.

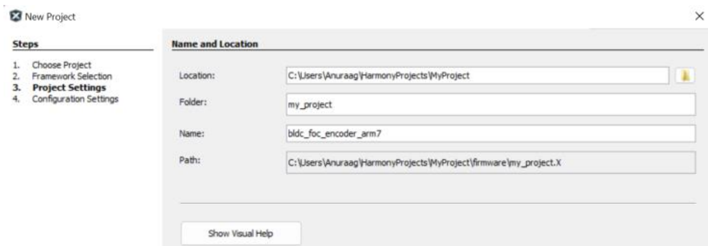


Fig 6.9 Enter the Name and Location of the Project

Step-5: In the New Project/Configuration Settings window:

1. Leave the project Name as default.
2. Enter the Name of ARM7 Microcontroller in the Target Device field.

Step-6: Click Finish Button. A loading window opens warning that MCC is starting in the background.

Fig 6.10 Loading Window of MCC

Step-7: Click Select MPLAB Harmony, then click Finish on the MCC Content Manager Wizard.

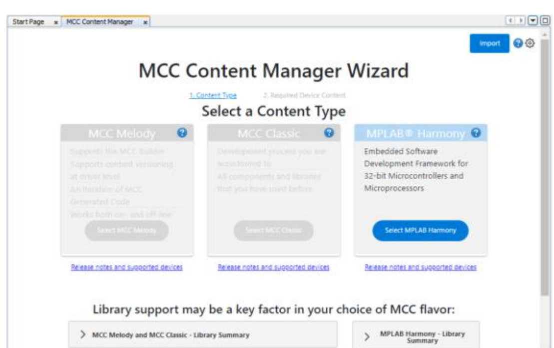


Fig 6.11 MCC Content Manager Wizard

Step-8: Wait until the project graph tab is displayed.

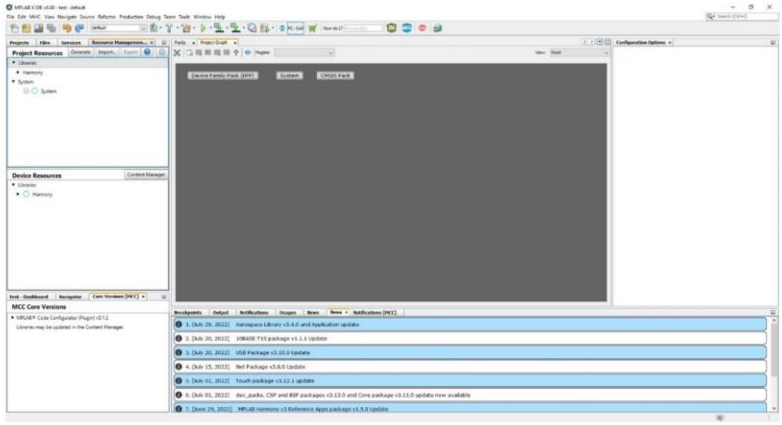


Fig 6.12 Project Graph of the Project

Step-9: Port Configuration:

a) Add the Evaluation Kit's BSP to the project resources to configure the ports connected to LEDs and buttons.
b) Open the Pin Configuration window to check that GPIOs connected to Motor Controller, LEDs and Buttons are configured, correctly.

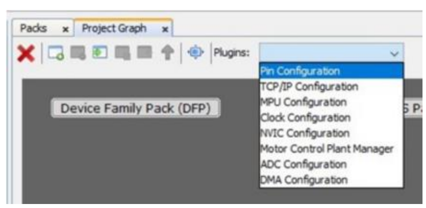


Fig 6.13 Pin Configuration in Project Graph

Step-10: From the Project Graph window, click on the System block.

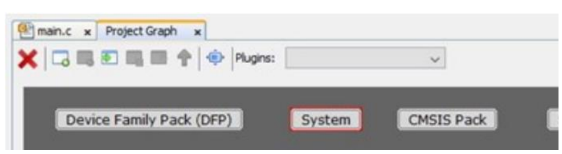


Fig 6.14 Open System Block from project graph

Step-11: The System parameters are displayed on the Configuration Options on the right-hand window. Scroll down to the Main Clock (MAINCLK) and the Master Clock (MCK) parameters list, and set that according to below figure:

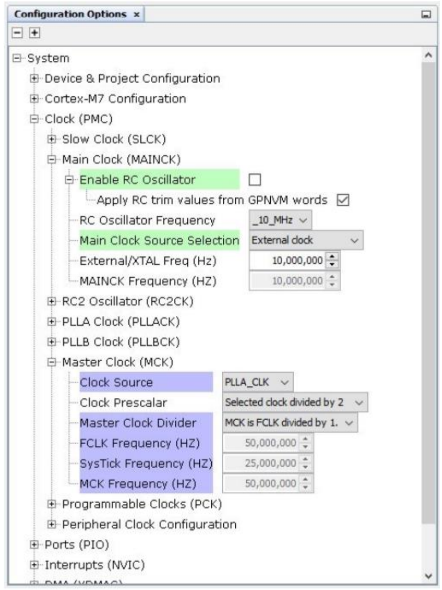


Fig 6.15 Set Clock Parameters in Configuration Option in Project Graph

Step-12: From the Configuration Options, scroll down to the Cortex-M7 Configuration parameters, and set it according to below figure.

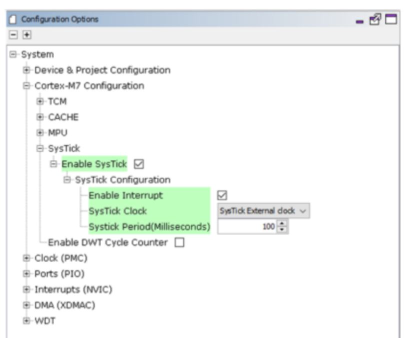


Fig 6.16 Set the Cortex M7 Configuration Parameter in Configuration Options

Step-13: From the MCC Project Resources, select Generate.

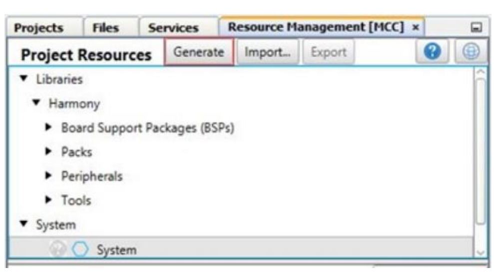


Fig 6.17 Generate the configuration code by pressing Generate in Project resources

The code must be implemented in the main.c file only. The rest of the project's files must not be changed because they are built and updated by MPLAB Harmony Configurator. If generated files are modified, Harmony will display a merge dialog interface when the project is generated again.

Step-14: From the MPLAB interface, select the Project tab on the left-hand side and scroll down to the Source Files item to locate the main.c file. Double click the file name to open it. And write the code of Motor Control, the main code can be referred from [Appendix 2].

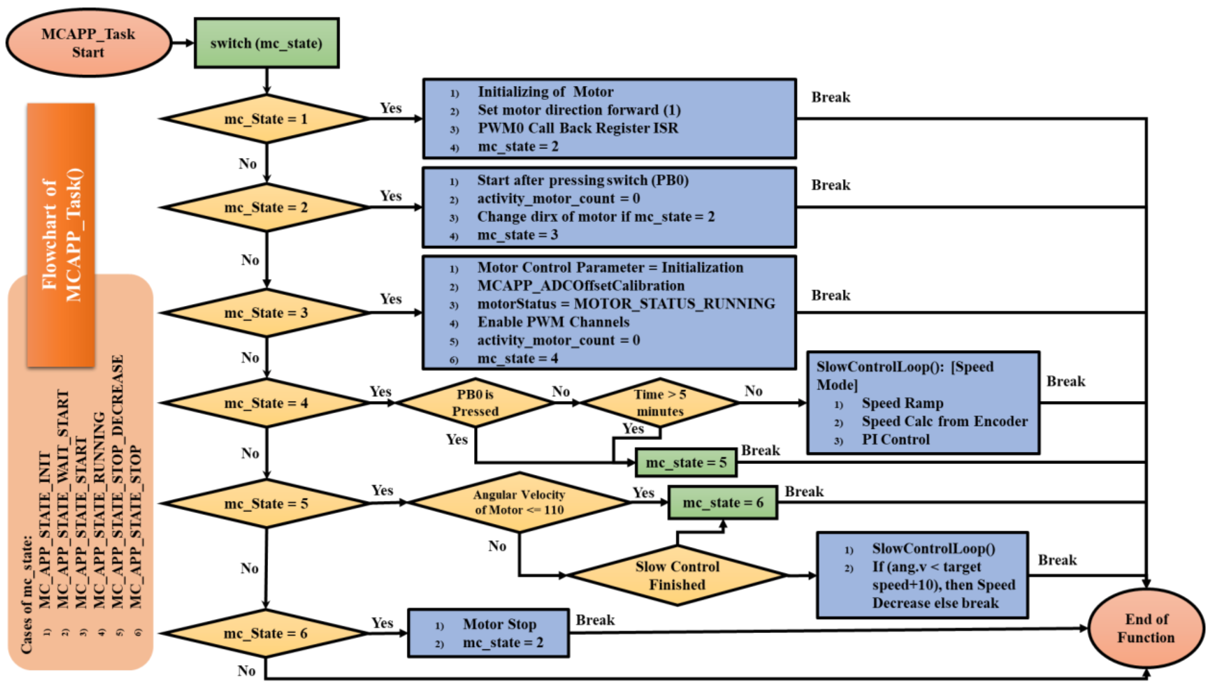


Fig 6.18 Flowchart of MCAPP_Task () Function in while (true) of main.c.

Step-15: This step explains MPLAB settings and how to compile, program and debug it:

Open MPLAB and execute the following steps.

1. Select the project name at the top of the project tree in the leftmost window.
2. Right click the project name and select properties from the drop-down menu.
3. From the Project Properties window that comes up:
    a. Check that the Device field contains ARM7 Microcontroller name.
    b. Select Connected Hardware Tool from the drop-down menu. If the embedded or an external debugger/programmer device is attached to your PC, it must be listed in the drop-down menu.

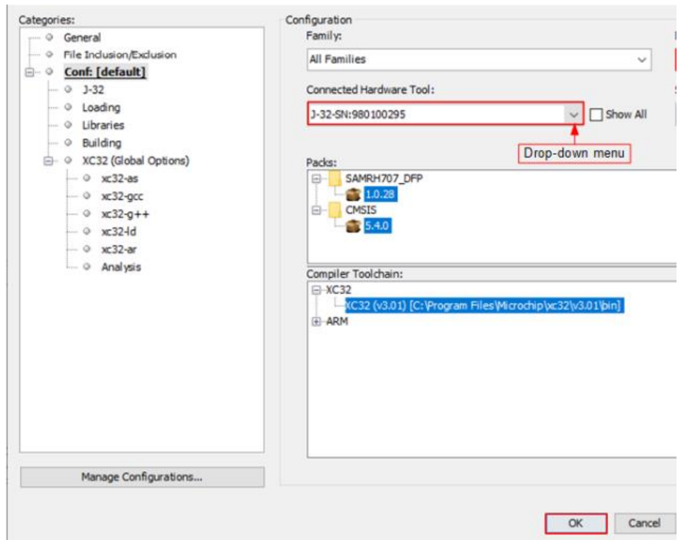


Fig 6.19 Selection of Connected Hardware Tool and Compiler Toolchain

Step-16: For Compiling, Programming, and Debugging.

a. Clean and compile the project by clicking on the Clean and Build button.

Fig 6.20 Clean and Compile Button

b. Program the application to the device by clicking on the Make and Program button.

Fig 6.21 Make and Program Button

c. Run the code by clicking on the Debug Project button.

Fig 6.22 Debug Project Button

d. The code can then be stopped by clicking on the Finish Debugger Session button.

Fig 6.23 Finish Debugger Session Button

e. Or paused by clicking on the Pause button.

Fig 6.24 Debug Pause Button

In summary, the methodology of this project involves a step-by-step process of designing, implementing, and testing the motor control system using the ARM Cortex-M7 Microcontroller and 3 Phase Channel Motor Controller, as well as ensuring the accuracy and reliability of the system through various performance metrics and evaluations.

6.6. Hardware Results:

The result of this project demonstrates a successful implementation of the Field Oriented Control (FOC) algorithm using Proportional Integral (PI) and Space Vector Pulse Width Modulation (SVPWM) techniques for the control of a 3-phase Brushless DC (BLDC) motor. The system is controlled by an ARM Cortex-M7 Microcontroller and a 3-Phase Channel Motor Controller. An encoder is used as feedback to provide the position and speed of the motor.

The main objective of the project was to achieve a 120-degree phase shift in each of the three phases of the motor. The implementation of the FOC algorithm with PI and SVPWM techniques enables the control system to achieve this objective with high accuracy and efficiency. By using a mixed signal oscilloscope and current probes, the current waveforms of each of the three phases were observed and recorded.

The recorded waveforms show that the motor is operating with a 120-degree phase shift in each of the three phases. The three phases are in perfect synchronization with each other, indicating that the FOC algorithm is working properly. The PI controller ensures that the motor follows the reference signal accurately, while the SVPWM technique provides smooth and efficient power delivery to the motor.

The Figure 6.25 shows the sinusoidal profile of motor currents waveform of all three phases of the motor, which shows the 120-degree phase shift. And, the Figure 6.26 shows the current waveform of each of the three phases of the motor separately, with a scale of current 5A per division.

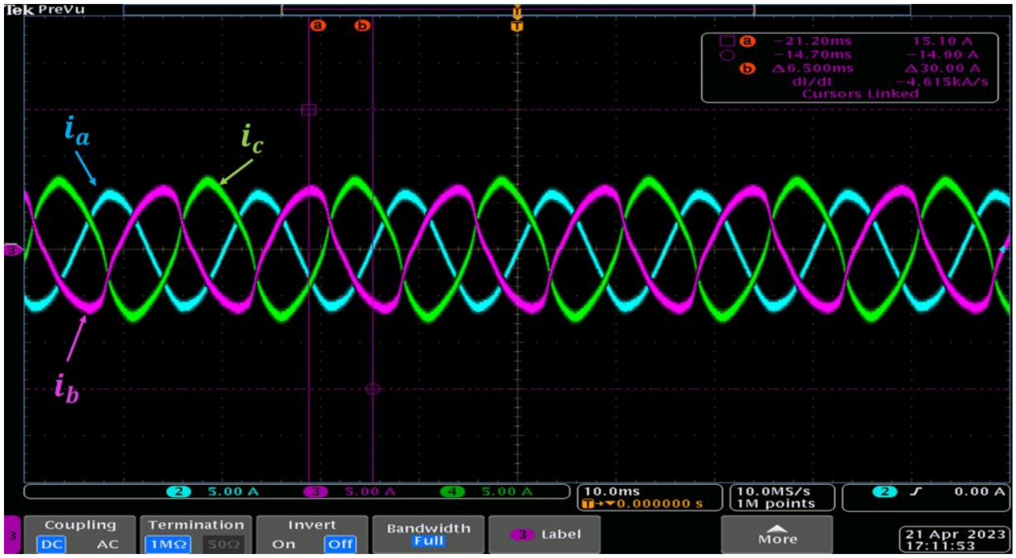


Fig 6.25 This shows a 120° phase shift in motor current's sinusoidal profile [Scale: 5A/div]

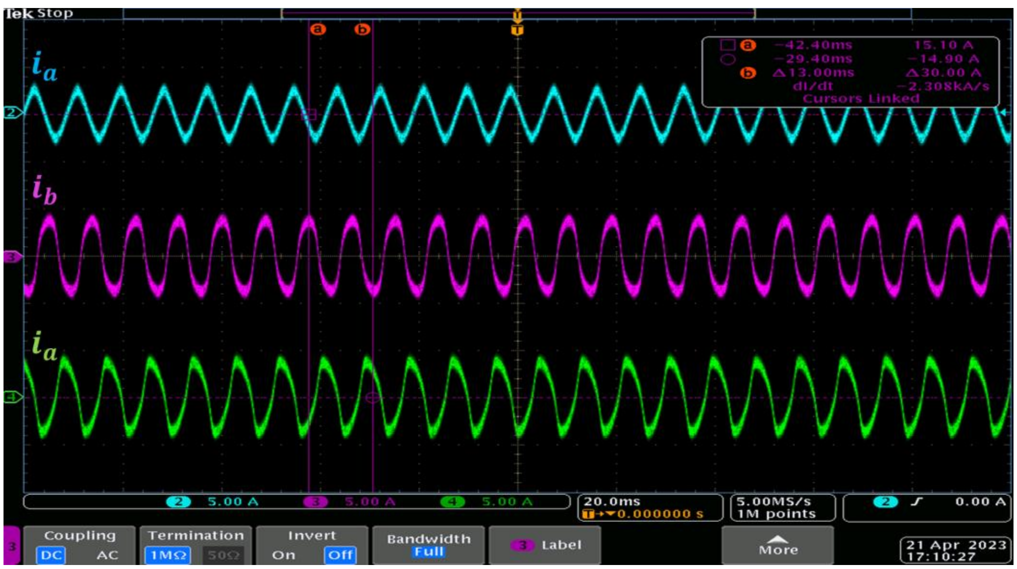


Fig 6.26 i_a, i_b and i_c refer to the currents in Phase A, B, and C, respectively [Scale: 5A/div]

Finally, the successful implementation of the FOC algorithm using PI and SVPWM techniques resulted in an approx 120° phase shift in each of the three phases of the BLDC motor, which can be further tuned to maximize the result output. The present waveforms' mixed signal oscilloscope readings show that the motor is in sync and the FOC algorithm is functioning properly. The combination of the ARM Cortex-M7 with the LX7720DB resulted in a dependable and efficient motor control system that can be used in a wide range of applications.

6.7. Limitations showing through Results:

The results of this project have demonstrated that there is a limitation in the synchronization of the amplitude of the three-phase currents in the BLDC motor. This limitation is due to imperfect adjustment of the proportional, integral and derivative gains (Kp, Ki, Kc) of the FOC control algorithm. The values of Kp, Ki, and Kc determine the performance of the FOC algorithm in controlling the motor's position, speed, and torque. Kp, Ki, and Kc are the three tuning parameters of the PID Controller:

- **Kp** is the proportional gain that affects how much the controller reacts to current error (the difference between desired and actual motor current).
- **Ki** is the integral gain, which governs how much the controller responds to cumulative mistake over time. It is used to eliminate steady-state error and assist the controller in reaching the setpoint faster.
- **Kc** is used in the project instead of Kd, because Kd can amplify noise, whereas Kc can prevent the integral term from exceeding a certain limit in the case of motor saturation, known as anti-windup.

These parameters, when combined, dictate the behavior of the PID controller and are critical in providing accurate and steady control of the BLDC motor. To obtain the required performance and stability of the control system, the values of Kp, Ki, and Kc are normally calibrated through trial and error

Further, there is a limitation in the accuracy of the analog-to-digital converter (ADC) used to measure the motor feedback signal. The ADC is not perfectly calibrated, which affects the accuracy of the measurements of the motor's position, speed, and torque. This limitation results in a deviation of the actual motor performance from the expected performance.

To address these limitations, some corrections are needed. Firstly, the values of Kp, Ki, and Kc need to be adjusted to their optimal values through a rigorous testing process. The optimal values for these gains are dependent on the specific motor parameters and the desired performance requirements. Secondly, the ADC needs to be calibrated accurately to ensure that the measured feedback signals are reliable and accurate. The calibration process involves adjusting the scaling factor of the ADC and compensating for any non-linearities in the conversion process.

Chapter-7 : Conclusion and Applications

7.1. Conclusion of the Project:

In conclusion, this project aimed to design and implement a motor control system for a Hurst's Series DMA0204024B1010 3-phase BLDC motor using ARM Cortex-M7 microcontroller and Microsemi LX7720DB motor controller. The system was designed to control the position, speed, and torque of the motor with high accuracy and efficiency, and was implemented with Field Oriented Control (with PI Controlling and SVPWM) control algorithm.

The project successfully achieved its goals, demonstrating the ability to control the motor's position, speed, and torque with high accuracy and efficiency. The system was able to generate appropriate control signals for the motor driver based on the desired position, speed, and torque of the motor, while also including feedback mechanisms to respond to changes in the motor's behavior and adjust the control signals accordingly. Further, in this project, the mathematical equivalence is shown between SVPWM and Inverse Clarke Transformation (ICT), and also that SVPWM is more efficient than SPWM in case of PWM Generation.

The results obtained through the project showed that a 120-degree phase shift in each of the three phases of the motor was achieved. This phase shift proved that the motor was in sync and the Field Oriented Control (FOC) in the BLDC motor was working properly. This was supported by readings obtained from the mixed signal oscilloscope showing the 3 current waves of each of the 3 phases using current probes.

However, there were some limitations and areas for improvement that were identified during the project. One limitation was the amplitude not being perfectly synchronized due to not perfectly adjusting the Kp, Ki, and Kc. This could be improved through further calibration and adjustment of these parameters. Additionally, the ADC could be calibrated more precisely to improve the accuracy of the system.

Despite these limitations, the project has demonstrated the potential of the designed system for a range of applications, including in robotics, automation, and space applications. The system could be integrated into robotic systems to provide precise control of motors for movement and manipulation tasks. It could also be used in automation systems to control the speed and position of motors in manufacturing and industrial processes. Additionally, the system could be used in space applications where precise control of motors is required for spacecraft propulsion and other tasks.

Overall, this project has successfully designed and implemented a motor control system for a Hurst's 3-phase BLDC motor using ARM Cortex-M7 microcontroller and Microsemi LX7720DB Motor Controller. The results demonstrate that the system using SVPWM is reliable and efficient, making it a promising solution for motor control in various fields. Further research and development could be undertaken to improve the accuracy and functionality of the system, opening up opportunities for wider applications in the future.

7.2. Applications of Project in Different Fields:

7.2.1. Space Applications:

The project of implementing a motor control system for a Brushless DC (BLDC) motor using ARM CortexM7 microcontroller has wide-ranging applications, including space applications. This system can be integrated into spacecraft propulsion systems, robotic systems, automation systems, and many other applications where precise motor control is required. In this section, we will discuss the potential applications of this project in space.

1. **Vibration Wheels:** Vibration wheels are used in spacecraft for attitude control. The motor control system developed in this project can be used to control the vibration wheels, providing accurate control and precise positioning, which is essential for spacecraft stabilization.
2. **Satellite Antennas:** Satellite antennas require precise positioning to ensure accurate communication with the ground station. The motor control system developed in this project can be used to control the positioning of the satellite antennas, ensuring accurate communication with the ground station.
3. **Solar Array Drives:** Solar arrays are essential for powering the spacecraft. The motor control system developed in this project can be used to control the solar array drives, providing accurate control and positioning, which is essential for efficient power generation.
4. **Reaction Wheels:** Reaction wheels are used in spacecraft for attitude control. The motor control system developed in this project can be used to control the reaction wheels, providing accurate control and precise positioning, which is essential for spacecraft stabilization.
5. **Propellant Valves:** Propellant valves are used in spacecraft for controlling the flow of propellant. The motor control system developed in this project can be used to control

the propellant valves, ensuring accurate control and flow of propellant, which is essential for efficient spacecraft propulsion.

6. **Gyroscopes:** Gyroscopes are used in spacecraft for attitude control. The motor control system developed in this project can be used to control the gyroscopes, providing accurate control and precise positioning, which is essential for spacecraft stabilization.
7. **Sample Collection Arm:** Sample collection arms are used in planetary missions to collect samples from the surface of the planet. The motor control system developed in this project can be used to control the sample collection arm, providing accurate control and positioning, which is essential for successful sample collection.

In conclusion, the motor control system developed in this project using ARM CortexM7 microcontroller has wide-ranging applications, including space applications. The potential applications of this project in space include vibration wheels, satellite antennas, solar array drives, reaction wheels, propellant valves, gyroscopes, and sample collection arms. By implementing this motor control system, space missions can be made more efficient and reliable, ensuring the success of the mission.

7.2.2. Robotic Applications:

The implementation of the motor control system for the brushless DC motor using ARM Cortex-M7 microcontroller has potential applications in various fields, including robotics. The system's high accuracy and efficiency in controlling the position, speed, and torque of the motor can be utilized in different robotic applications.

1. **Robotic Arms:** The motor control system can be integrated into robotic arms to provide precise control of the arm's movement and manipulation tasks. The system's ability to accurately control the position, speed, and torque of the motor can ensure the robotic arms' smooth and precise movement.
2. **Autonomous Wheels:** The motor control system can be used to control the position, speed, and torque of the motors in autonomous wheels, such as those used in autonomous vehicles. The system's ability to respond quickly to changes in the motor's behavior and adjust the control signals accordingly can ensure safe and efficient operation of the autonomous wheels.
3. **Drones:** The motor control system can be integrated into drones to provide precise control of the drone's motor for efficient and stable flight. The system's ability to control the position, speed, and torque of the motor can ensure that the drone's movements are precise and stable.
4. **Humanoid Robots:** The motor control system can be integrated into humanoid robots to provide precise control of the motor for smooth and natural movements. The system's ability to control the motor's position, speed, and torque can ensure that the movements of the humanoid robot are precise and natural.

5. **Exoskeletons:** The motor control system can be used to control the motors in exoskeletons used to assist people with limited mobility. The system's ability to control the position, speed, and torque of the motor can ensure that the exoskeleton's movements are smooth and provide the necessary support to the user.
6. **Industrial Robots**: The motor control system can be integrated into industrial robots to provide precise control of the motors for manufacturing and industrial processes. The system's ability to control the position, speed, and torque of the motor can ensure that the industrial robot's movements are precise and efficient like in work cells.
7. **Prosthetics:** The motor control system can be used to control the motors in prosthetics to provide precise control of the movement of the prosthetic limb. The system's ability to control the position, speed, and torque of the motor can ensure that the movements of the prosthetic limb are smooth and natural.

In conclusion, the implementation of the motor control system for the brushless DC motor using ARM Cortex-M7 microcontroller has potential applications in various robotic fields. The system's high accuracy and efficiency in controlling the position, speed, and torque of the motor can be utilized in different robotic applications, from robotic arms and autonomous wheels to drones and humanoid robots. The system's ability to respond quickly to changes in the motor's behavior and adjust the control signals accordingly can ensure safe and efficient operation of the robots.

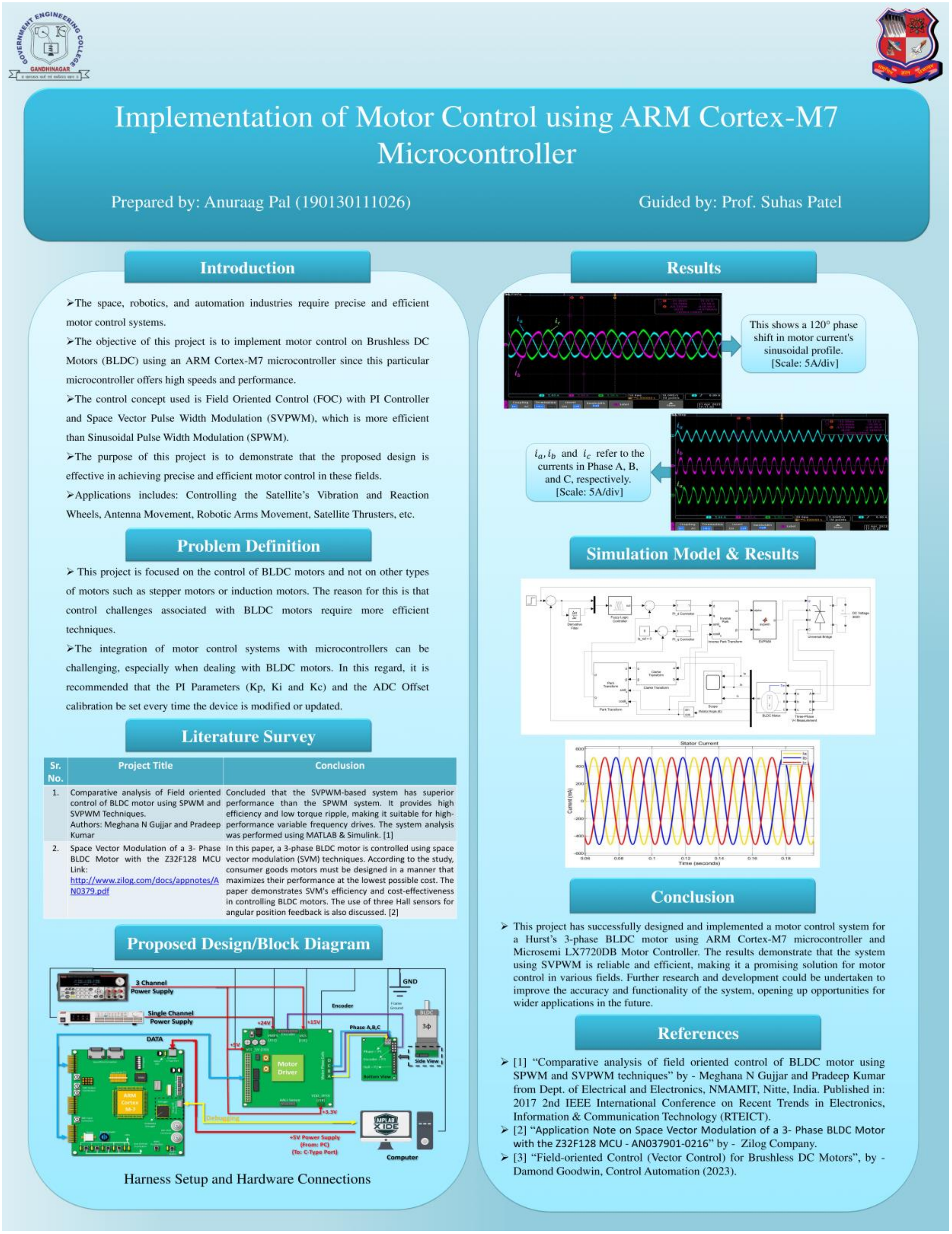
REFERENCES

1. Agrawal, R. (n.d.) ‘RISC and CISC’, Coding Ninja, Link: [https://www.codingninjas.com/codestudio/library/risc-and-cisc]
2. ARMDeveloper (n.d.) ‘Cortex-M7’, Link: [https://developer.arm.com/Processors/Cortex-M7]
3. Barrett, S. (2014) “Cortex-M7 Launches: Embedded, IoT and Wearables”, Anand Tech, Link: [https://www.anandtech.com/show/8542/cortexm7-launches-embedded-iot-and-wearables/2]
4. Chen, F.F., Xu, Q. and Zhang, J. (2019) 'Permanent Magnet Synchronous Motor Control System Based on ARM Cortex-M4F', International Journal of Online Engineering, vol.15, no. 4, pp. 43-51.
5. Damond Goodwin (2023) “Field-oriented Control (Vector Control) for Brushless DC Motors”, Control Automation, Link: [https://control.com/technical-articles/field-oriented-control-vector-control-for-brushless-dc-motors/]
6. Danielle Collins (2019) "What is Space Vector Pulse Width Modulation (SVPWM)", Motion Control Tips, Link: [https://www.motioncontroltips.com/what-is-space-vector-pulse-width-modulation-svpwm/]
7. Danielle Collins (2018) "External rotor motor basics: Design and applications", Motion Control Tips, Link: [https://www.motioncontroltips.com/external-rotor-motor-basics-design-applications/]
8. Digi-Key (2016) “How to Power and Control Brushless DC Motors”, Link: [https://www.digikey.com/en/articles/how-to-power-and-control-brushless-dc-motors]
9. Edward H. Herbert, Jr. and Ronald J. Riedel (2006) "Brushless DC Motors – Part I: Construction and Operating Principles", Power Conversion and Intelligent Motion.
10. Gawande, V.M., Moharil, S.R. and Raut, N.A. (2017) 'Speed Control of DC Motor using ARM Cortex-M3', International Journal of Advanced Research in Computer Engineering & Technology, vol.6, no. 9, pp. 58-62.
11. Gujjar, M.N. and Kumar, P. (2017) “Comparative analysis of field-oriented control of BLDC motor using SPWM and SVPWM techniques”, IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Link: [https://ieeexplore.ieee.org/document/8256733]
12. Hurst Motors, Link: [https://www.hurst-motors.com/ntdynamo.html]
13. MatWorks (n.d.) “Develop field-oriented control algorithms using simulation”, Link: [https://www.mathworks.com/solutions/electrification/field-oriented-control.html]
14. Microchip Technology (n.d.) “Hurst Dynamo Family Datasheet”, Link: [https://www.microchip.com/content/dam/mchp/documents/MCU16/ProductDocuments/DataSheets/Hurst_Dynamo_Family_Datasheet.pdf]

15. Microchip Technology (n.d.) ‘Production Datasheet of LX7720DB: Motor Driver with Four Half-Bridge Drivers, Rotation and Position Sensing for Space’, Link: [https://www.microsemi.com/existing-parts/parts/152475]
16. Microchip Technology (n.d.) 'Sensorless Control of Brushless DC Motor Using Back-EMF Technique', Link: [https://ww1.microchip.com/downloads/aemDocuments/documents/AERO/ApplicationNotes/ApplicationNotes/00004478.pdf]
17. Kumar, A., Gautam, R. and Singh, K.K. (2020) 'Design and Implementation of Three Phase Induction Motor Control System Using ARM Cortex-M4F', International Journal of Recent Technology and Engineering, vol.9, no. 3, pp. 1698-1702.
18. Kim, H., Kim, S., and Choi, J. (2018) 'Analysis and control of a brushless DC motor for variable-speed pump applications', Journal of Electrical Engineering & Technology, vol.13, no. 4, pp. 1423-1430.
19. Sathyanarayanan, S., Rajendran, R. and Srinivasan, K. (2016) 'Brushless DC Motor Control Using an ARM Cortex-M4F-Based Microcontroller', International Journal of Applied Engineering Research, vol.11, no. 14, pp. 7452-7456.
20. Zilog Company (n.d.) “Application Note on Space Vector Modulation of a 3 Phase BLDC Motor with the Z16FMC MCU”, Link: [http://www.zilog.com/docs/appnotes/AN0354.pdf]

Poster

The Appendix illustrates the Poster created in order to explain this project:



APPENDIX

Appendix 1

The following tables in Appendix A illustrate the connections and pinouts necessary for proper communication between the microcontroller, motor controller, and power supply. These pinouts and connections are essential for the successful operation of the hardware and should be carefully reviewed and followed during assembly and testing. The tables include details on the connection between the microcontroller and motor controller, the connection between the motor and motor controller, and the pinouts for power supply and debugging.

Connection Between ARM Cortex-M7 Evaluation Kit and LX7720DB Motor Controller:

Pinouts Between Microcontroller and Motor Controller		
ARM Cortex-M7 Microcontroller	**LX7720DB Motor Controller**	**Description of Connection**
PB11 (J19.7)	LD_IN_A_C (J20.2)	To Command Lower N-channel MOSFET of Phase A
PB12 (J19.8)	UD_IN_A_C (J20.1)	To Command Upper N-channel MOSFET of Phase A
PB13 (J19.18)	LD_IN_B_C (J20.6)	To Command Lower N-channel MOSFET of Phase B
PB14 (J19.15)	UD_IN_B_C (J20.5)	To Command Upper N-channel MOSFET of Phase B
PB15 (J18.7)	LD_IN_C_C (J20.10)	To Command Lower N-channel MOSFET of Phase C
PB16 (J18.8)	UD_IN_C_C (J20.9)	To Command Upper N-channel MOSFET of Phase C
PA29 (J19.16)	OC_FAULT (J20.32)	To detect the Over Current in Motor Driver
PB18 (J18.15) (TCLK0)	SNS_OUT_A (J20.3)	To sense the output voltage signal of Phase A
PB19 (J18.10) (TCLK2)	SNS_OUT_B (J20.7)	To sense the output voltage signal of Phase B
PB8 (CAN RXD) (TIOB3)	BLO2 (J20.36)	To read the 2nd Channel of encoder for feedback of direction and position of Motor
PB7 (CAN TXD) (TIOA3)	BLO1 (J20.35)	To read the 1st Channel of encoder for feedback of direction and position of Motor

PD8 (J18.11) (PCK2)	MOD_CLK (J20.18)	To generate the modulating waveform for the PWM signal that controls the motor's speed and direction
PA30 (J19.17) (PCK1)	CP_CLK (J20.17)	To control of the motor driver internally, by generating output of the clock signal

Connection between Hurst Series DMA0204024B1010 BLDC Motor and LX7720DB Motor Controller:

Connections for Hurst's Series BLDC Motor/Encoder to Motor Controller			
Hurst's Series BLDC Motor Connection	**Wire Color**	**LX7720DB Motor Controller Connector**	**LX7720DB Motor Controller Signal**
Motor winding phase A	White	J9 Pin1	Phase A
Motor winding phase B	Black	J9 Pin2	Phase B
Motor winding phase C	Red	J9 Pin3	Phase C
Encoder VDD	Red	J15 Pin1	VCC(+3.3V)
Encoder GND	Black	J15 Pin2	GND
Encoder Output A	White	J15 Pin3	BL11
Encoder Output A	Blue	J15 Pin4	BL12

Pinouts for Power Supply inputs for ARM Cortex-M7 Microcontroller and LX7720DB Motor Controller:

Power Supply and Debugging			
Hardware	**Pin Number or Connector Type**	**Voltage**	**Description**
Microcontroller	C-Type Port (J1)	+5V	Power directly from Computer
	Micro USB Port (J12)	-	Debugging
Motor Controller	J12	VMPS = +24V	For Motor Power Supply
	J31	VGS = +15V	For MOSFET Supply
	J30	VCC = +5V	For VCC
	J18	VDD = +3.3V	For VDD

Appendix 2

Appendix B contains the code for the "main.c" file in MPLAB X IDE. This code is an integral part of the system and contains confidential and proprietary functions that must be protected. It is important to maintain the confidentiality of this code and to ensure that it is only shared with authorized personnel who require access for the purpose of system development and maintenance.

Code:

```
/***************************************************************************

Main Source File

File Name: main.c

Summary: This file contains the “main” function for a project.

***************************************************************************/

//*************************************************************************

Section: Included Files

//*************************************************************************

#include <stddef.h>                     //Defines NULL

#include <stdbool.h>                    //Defines true

#include <stdlib.h>                     //Defines EXIT_FAILURE

#include “definitions.h”                //SYS function prototype

#include “mc_app.h”

#include “X2CScope.h”

#include “X2CScopeCommunication.h”

//*************************************************************************

//*************************************************************************
```

Section: Main Entry Point

```
//*************************************************************************
//*************************************************************************
int main () {
	/* Initialize all modules */
	SYS_Initialize (NULL);
	X2CScope_Init ();

	/* Filter any unexpected initial state on start switch */
	while (! SWITCH_START_Get ());
	while (true) {
		X2CScope_Update ();		//Update X2CScope Variables
		MCAPP_Task ();			// Main Function
		X2CScopeCommunicate ();	//Communication between Embedded System -
						//- and X2CScope Tools.
}
	/* Execution should not come here during normal operation */
	return (EXIT_FAILURE);
}
/*************************************************************************
End of File
*/
```

Appendix 3

Appendix 3 contains the softcopy of the following files on the attached Compact Disc (CD):

1. Industrial Internship Completion Certificate (.pdf)
2. Report (.pdf)
3. Presentation (.ppt)
4. Poster (.pdf)
5. Weekly Records (of all 12 weeks) – Annexure 1(.pdf)
6. Feedback Form by Industrial Expert – Annexure 2 (.pdf)
7. AICTE Formats (Format – 1, 3, 6, 7, & 10) (.pdf)
8. No Objection Certificate (NOC) (.pdf)

Appendix 4

Appendix 4 of this report is in separate spiral copy named “Appendix” consists of 4 parts:

1. The First is Annexure – 1, which shows weekly records of the student's internship for a period of 12 weeks.
2. The Second is Annexure – 2, which shows the feedback provided by my industry expert.
3. The Third consists of five forms that must be filled out in accordance with the internship policies of the AICTE. The five filled formats are:
    a. Student Internship Program Applications (Format-1),
    b. Internship Synopsis (Format-3),
    c. Supervisor's Evaluation of Interns (Format-6),
    d. Student Feedback on Internships (Format-7), and
    e. 12-Week Attendance (Format-10).
4. The fourth shows the No Objection Certificate (NOC).

These parts provide a thorough and detailed description of the internship experience of the student. This information includes progress, achievements, feedback, and evaluations from relevant stakeholders.