Department of Electronics and Telecommunication Engineering
University of Moratuwa

EN2160 Electronic Design Realization

Design Document

Group No: 32

210152E	Dulnath W.H.R.
210599E	Shamal G.B.A.
210600D	Shamika K.A.M.
210642G	Thennakoon T.M.K.R.

Content

Introduction

Brief Overview of the Project

The stepper motor driver project involves the design and development of a sophisticated motor control system with a built-in feedback mechanism. This closed-loop stepper motor driver aims to provide precise control over the motor's rotation, ensuring it accurately reaches and maintains the desired position. Unlike traditional open-loop systems, our driver continuously monitors the motor's position and makes real-time adjustments to correct any deviations caused by external loads or other factors.Basically we are approaching to implement the stepper motor driver which will gain control inputs and encoder inputs and precise

Objectives and Goals

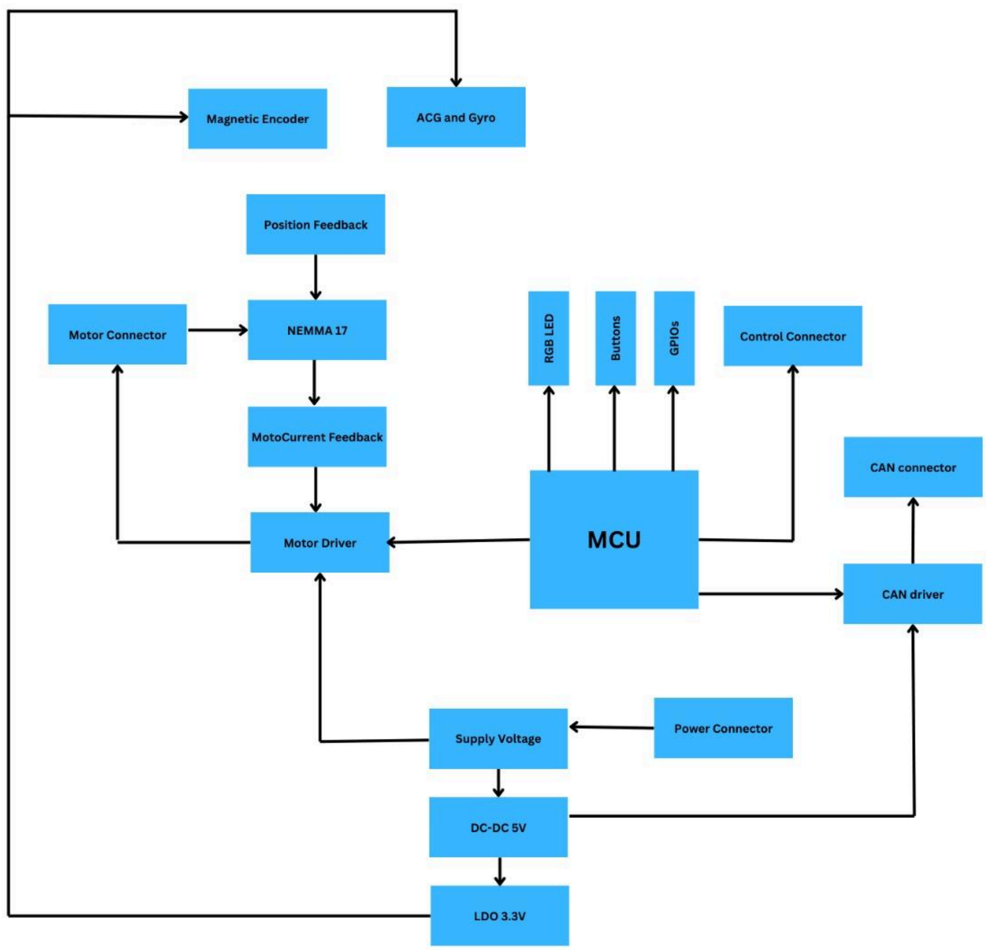
1. **Precise Motor Control:**
    - Develop a driver that ensures the stepper motor reaches the exact target position with high precision.
    - Implement a feedback mechanism to continuously monitor the motor's position and make necessary adjustments.
2. **Error Detection and Correction:**
    - Integrate sensors to detect positional errors caused by external loads or other disturbances.
    - Design a control algorithm that uses feedback data to correct these errors and return the motor to the reference position.
3. **Robust and Reliable Operation:**
    - Ensure the stepper motor driver operates reliably under various load conditions.
    - Minimize the impact of external disturbances on the motor's performance through efficient error correction.
4. **User-Friendly Interface:**
    - Provide a simple and intuitive interface for users to set target positions and monitor the motor's status.
    - Include diagnostic features to help users identify and troubleshoot issues.
5. **Scalability and Flexibility:**
    - Design the driver to be adaptable to different types and sizes of stepper motors.
    - Ensure the system can be easily scaled or modified for various applications.

By achieving these objectives, our closed-loop stepper motor driver aims to deliver precise, reliable, and adaptable motor control for a wide range of industrial and commercial applications.

System Architecture

Block Diagram

The provided block diagram illustrates the overall architecture of your closed-loop stepper motor driver system. Here's a detailed explanation of each component and its role in the system:



Block diagram explanation

1. **Microcontroller Unit:**
    - This central processing unit of the system that controls and coordinates all other components.
    - Here we are using STM32F103FBT6 microcontroller of STMicroelectronics as it suits with our meeting requirements to implement the feedback system and motor controller part.
    - It processes feedback data and generates control signals for the motor driver.
2. **Motor Connector:**
    - Connects the stepper motor (Nema 17) to the system.
    - Allows the motor to receive driving signals from the motor driver.
3. **Motor Driver:**
    - Here we are using two TB67H450FNG PWM Chopper Type DC Brushed Motor Drivers.
    - Receives control signals from the MCU and drives the stepper motor.
    - Provides the necessary current and voltage to operate the motor.
4. **Magnetic Encoder:**
    - Here we are using TZT AS5600 magnetic encoder which should get connected with the stepper motor.
    - Provides precise position feedback to the MCU.
    - Enhances the accuracy of the motor's position control.
5. **RGB LED:**
    - Indicates the status of the system through different colors.
    - Helps in visual diagnostics and monitoring.
6. **Buttons:**
    - User interface for manual control and input.
    - Allows the user to set target positions or initiate specific actions.
7. **GPIO (General Purpose Input/Output):**
    - Provides additional input and output pins for custom functionalities.
    - Allows for expansion and customization of the system.
8. **Control Connector:**
    - Interface for external control signals.
    - Enables integration with other control systems or devices.
9. **CAN Connector:**
    - Interface for CAN (Controller Area Network) communication.
    - Allows the system to communicate with other devices on a CAN bus.
10. **CAN Driver:**
    - SN65HVD23x 3.3-V CAN Bus Transceiver is used.
    - Transceiver that handles communication over the CAN bus.
    - Converts data between the MCU and the CAN network.

11. **Supply Voltage:**
    - Main power input for the system.
    - 12V input voltage is expected.
    - Provides the necessary voltage for all components.
12. **Power Connector:**
    - Interface for connecting the power supply to the system.
    - Ensures all components receive stable power.
13. **DCDC 5V:**
    - DC-DC converter that steps down the input voltage to 5V.
    - Switching Voltage Regulators 12-76V 1Ch Buck Conv w/ Integrated FET is used here.
14. **LDO 3.3V:**
    - Low Dropout Regulator that steps down the voltage to 3.3V.
    - LDO Voltage Regulators Micropower 250mA Lo- Noise Ultra LDO Reg is used.

System Operation

1. **Initialization:**
    - Upon power-up, the MCU initializes all components and sensors.
    - The system checks the status of the motor and sets the initial position.
2. **Position Control:**
    - The user sets a target position using external control signals.
    - The MCU processes the input and calculates the necessary motor movements.
3. **Feedback Loop:**
    - The motor driver receives control signals from the MCU and drives the motor.
    - Position feedback from the magnetic encoder and current feedback are continuously sent to the MCU.
    - The MCU adjusts the control signals in real-time to correct any deviations from the target position.
4. **Error Correction:**
    - If an error is detected (e.g., due to an external load), the MCU recalculates the control signals.
    - The feedback loop ensures the motor is driven back to the reference position accurately.
5. **Communication:**
    - The CAN driver allows the system to communicate with other devices on a CAN network.
    - Enables integration with larger systems or networks for coordinated control.
    - So user can gain the ability to control multiple stepper motors simultaneously.

By incorporating these components and their interactions, your closed-loop stepper motor driver system achieves precise and reliable motor control, capable of correcting errors and maintaining the desired position even under varying load conditions.

Component Selection

1. STEM Microcontroller:

- **Justification:** The STM32F103CBT6 microcontroller from STMicroelectronics was selected for its robust processing capabilities, low power consumption, and extensive I/O options, making it ideal for a closed-loop stepper motor driver. Based on the 32-bit Arm Cortex-M3 CPU operating at 72 MHz, this MCU offers up to 128 KB of Flash memory and 20 KB of SRAM. These features provide the necessary computational power to handle real-time feedback processing and control algorithms essential for precise motor control. Its integrated peripherals, including USB, CAN, seven timers, and two ADCs, enhance its utility in motor control applications. Additionally, the MCU supports multiple communication interfaces (I2C, USART, SPI), facilitating extensive connectivity and efficient data exchange. The low-power modes and DMA controller further contribute to its efficiency and performance, ensuring reliable and precise stepper motor operation.
- Datasheet: https://www.mouser.com/datasheet/2/389/stm32f103c8-1851025.pdf

4. PWM Chopper Type DC Brushed Motor Driver:

- **Justification:**The TB67H450FNG by Toshiba, a PWM chopper type DC brushed motor driver, was chosen for its efficiency in controlling motor speed and torque. Featuring BiCD process integration, it supports forward, reverse, brake, and stop modes, delivering up to 50V and 3.5A through its low-resistance MOSFETs. This IC provides precise control of stepper motors by varying the PWM signal's duty cycle, making it ideal for high-precision applications. It includes essential protections like thermal shutdown, overcurrent detection, and undervoltage lockout, ensuring robust and reliable motor management. Packaged in HSOP8, it supports both constant current and direct PWM drive capabilities.
- Datasheet: https://www.mouser.com/datasheet/2/408/TB67H450FNG_datasheet_en_20201126-1604947.pdf

6. Switching Voltage Regulators (12-76V 1Ch Buck Converter with Integrated FET, RoHS Compliant):

- **Justification:** The switching voltage regulators are used to step down the input voltage to 5V. The integrated FET design offers high efficiency and low power loss, which is essential for maintaining the system's overall efficiency and reliability.
- Datasheet : https://www.mouser.com/datasheet/2/348/bd9g341aefj_lb_e-1874278.pdf

7. LDO Voltage Regulators (Micropower 250mA Low-Noise Ultra LDO Regulator, RoHS Compliant):

- **Justification:** The LDO (Low Dropout) voltage regulators provide a stable 3.3V supply with low noise characteristics, which is critical for powering up the MCU and for sensitive analog and digital circuits. Their micropower operation helps in minimizing power consumption, making them suitable for low-power applications.
- Datasheet: https://www.ti.com/lit/ds/symlink/lp2992.pdf?ts=1718903067976&ref_url=http%253A%252F%252Fpavouk.org%252F

3. 3.3V CAN Interfaces:

- **Justification:** The 3.3V CAN interfaces facilitate communication with other devices and systems on a Controller Area Network (CAN) bus. This allows for reliable data exchange and integration into larger control systems, enhancing the system's versatility and scalability.
- Datasheet: https://www.ti.com/lit/ds/symlink/sn65hvd232.pdf?ts=1718954842524&ref_url=https%253A%252F%252Fwww.mouser.com%252F

2. 12MHz Crystal Oscillator:

- **Justification:** The 12MHz crystal oscillators ensure a stable and accurate clock signal for the microcontroller, which is crucial for maintaining timing accuracy in control tasks. Precise timing is essential for generating accurate PWM signals and processing feedback data.

5. Resistors and Capacitors:

- **Justification:** Resistors and capacitors are fundamental components used for various purposes, including filtering, signal conditioning, and voltage regulation. They ensure stable operation of the circuit by smoothing out voltage fluctuations and noise.

By carefully selecting these components, the design ensures a reliable, efficient, and precise stepper motor driver system capable of performing under various conditions while maintaining high accuracy and stability.

Circuit Design

Circuit Parts

The circuit design for the stepper motor driver consists of three main parts: the power circuit, the motor controller part, and the communication part. Each of these parts plays a crucial role in ensuring the overall functionality and performance of the system.

1. Power Circuit:

- **Components Involved:**
  - Switching Voltage Regulators (12-76V 1Ch Buck Converter with Integrated FET)
  - LDO Voltage Regulators (Micropower 250mA Low-Noise Ultra LDO Regulator)
- **Functionality:**
  - The power circuit is responsible for converting the input voltage to stable 3.3V and 5V outputs required by various components of the system.
  - The switching voltage regulator efficiently steps down the input voltage to 5V. It uses a buck converter design with an integrated FET to provide high efficiency and low power loss.
  - The LDO voltage regulator further steps down the 5V to a stable 3.3V, ensuring low noise and stable operation for sensitive components such as the microcontroller and communication interfaces.
  - By providing stable and clean power supplies, the power circuit ensures the reliable operation of the entire system, preventing voltage fluctuations and noise from affecting performance.

2. Motor Controller Part:

- **Components Involved:**
  - STM Microcontroller
  - Two PWM Chopper Type DC Brushed Motor Drivers
  - Resistors and Capacitors (for filtering and signal conditioning)
  - Magnetic Encoder (for position feedback)
  - Nema 17 Stepper Motor

- **Functionality:**
    - The motor controller part is the core of the system, handling the control and driving of the stepper motor.
    - The STM microcontroller processes input signals, feedback data, and control algorithms to generate precise PWM signals for the motor driver.
    - The PWM chopper type DC brushed motor driver receives PWM signals from the microcontroller and controls the current flowing through the stepper motor coils, enabling precise control of the motor's speed and position.
    - The magnetic encoder provides real-time position feedback to the microcontroller, allowing it to adjust the motor's position accurately.
    - Resistors and capacitors are used for filtering and signal conditioning, ensuring clean and stable signals for the motor driver and feedback sensors.

3. Communication Part:

- **Components Involved:**
    - 3.3V CAN Interfaces
    - CAN Driver
- **Functionality:**
    - The communication part enables the stepper motor driver to connect and communicate with other devices and systems on a CAN (Controller Area Network) bus.
    - The 3.3V CAN interfaces facilitate reliable communication by converting the microcontroller's digital signals into CAN-compatible signals.
    - The CAN driver acts as a transceiver, handling the transmission and reception of data on the CAN bus.
    - This communication capability allows multiple motor drivers to be connected and controlled simultaneously, enabling coordinated operation and synchronization of multiple motors in larger systems.
    - By enabling communication with other devices, the CAN interfaces enhance the scalability and flexibility of the system, making it suitable for complex and distributed applications.

Overall Functionality

By integrating these three parts, the stepper motor driver system achieves the following:

- **Precise and Reliable Motor Control:** The motor controller part ensures accurate positioning and smooth operation of the stepper motor, with real-time feedback and error correction.
- **Stable and Efficient Power Supply:** The power circuit provides clean and stable voltages necessary for the reliable operation of all components.

- **Scalable and Coordinated Operation:** The communication part allows multiple motor drivers to be connected and controlled over a CAN bus, enabling complex and coordinated motor control applications.

This comprehensive design ensures that the stepper motor driver system is robust, efficient, and capable of delivering high precision and reliability in various applications.

PCB Layout

Design Considerations

1. Trace Width:

- **Range:** The trace widths in the design range from 0.25mm to 1mm, with larger widths used for power tracing.
- **Justification:**
    - **Current Capacity:** Wider traces (up to 1mm) are used for power lines to handle the higher current flow (nearly 4 Amperes) without excessive heating or voltage drop.
    - **Signal Integrity:** Narrower traces (0.25mm) are used for signal lines where the current is much lower, ensuring minimal interference and maintaining signal integrity.

2. Grounding:

- **Ground Net:** A dedicated ground net is used to provide a common reference point for all components, ensuring stable operation.
- **Purpose:**
    - **Noise Reduction:** A solid ground plane helps reduce electromagnetic interference (EMI) and provides a low impedance path for return currents, improving overall signal quality.
    - **Current Return Path:** Ensures that all return currents have a low-resistance path, reducing ground loops and potential noise issues.

3. Coupling Capacitors:

- **Techniques Used:** Coupling capacitors are strategically placed near the ICs.
- **Purpose:**
    - **Protection Against Current Spikes:** These capacitors help filter out transient spikes and noise from the power supply, protecting sensitive components.
    - **Power Supply Stability:** They stabilize the power supply voltage by providing local energy storage and smoothing out voltage fluctuations.
    -

Layers Used and Their Purposes

1. Top Layer:

- **Purpose:**
  - **Component Placement:** The top layer is used for placing most of the components, including the microcontroller, motor driver IC, sensors, and connectors.
  - **Signal Routing:** Primary signal traces are routed on the top layer to minimize the complexity of the layout and maintain short signal paths for critical signals.

2. Ground Layer:

- **Purpose:**
  - **Ground Plane:** The entire layer is dedicated to the ground plane, providing a continuous and low-impedance ground reference for the entire circuit.
  - **Noise Reduction:** The ground plane helps in shielding and reducing electromagnetic interference (EMI) by providing a uniform return path for signals and power supply currents.

The design considerations ensure that the stepper motor driver is both robust and efficient. By carefully selecting trace widths, grounding techniques, and using coupling capacitors, the design addresses key electrical and thermal challenges. The use of dedicated layers for component placement and grounding further enhances the reliability and performance of the circuit. This meticulous approach ensures that the stepper motor driver can handle high currents, maintain signal integrity, and operate reliably in various conditions.

Feedback Mechanism

Description of the Feedback Mechanism Used

The feedback mechanism in our stepper motor driver system is designed to ensure precise control and accurate positioning of the motor. This is achieved by using a magnetic encoder to monitor the motor's position and provide real-time feedback to the control system.

Components Involved:

- **Magnetic Encoder:** The primary sensor used for capturing the position of the stepper motor.

- **Microcontroller :** Processes the feedback signals from the magnetic encoder.
- **Motor Driver:** Adjusts the motor's operation based on the feedback processed by the MCU.

Functionality:

1. **Position Sensing:**
    - **Magnetic Encoder:** The magnetic encoder is mounted on the stepper motor and continuously monitors its rotational position. It generates signals corresponding to the motor's position, which are then fed into the microcontroller.
2. **Signal Processing:**
    - **Microcontroller:** The MCU receives the position signals from the magnetic encoder. It processes these signals to determine the current position of the motor and compares it with the desired position. The MCU executes control algorithms to calculate the necessary adjustments needed to correct any deviation from the desired position.
3. **Error Detection and Correction:**
    - **Error Identification:** If there is a discrepancy between the actual position (as reported by the encoder) and the target position, the MCU identifies this as an error.
    - **Feedback Control:** The MCU uses a feedback control loop to correct this error. It sends control signals to the motor driver, instructing it to adjust the motor's operation (e.g., speed, direction) to bring the motor back to the desired position.
4. **Motor Control:**
    - **Motor Driver:** The motor driver receives control signals from the MCU and adjusts the motor's current and voltage accordingly. This ensures that the motor follows the correct path and reaches the target position accurately.

Benefits of the Feedback Mechanism

- **Precision:** The feedback mechanism allows for precise control of the stepper motor, ensuring it reaches the exact desired position.
- **Error Correction:** Real-time feedback enables immediate detection and correction of errors, improving the overall accuracy and reliability of the system.
- **Stability:** Continuous monitoring and adjustment of the motor's operation lead to stable performance, even under varying load conditions.
- **Efficiency:** By optimizing the motor's performance based on real-time feedback, the system operates more efficiently, reducing power consumption and wear on components.

In summary, the feedback mechanism involving a magnetic encoder and a microcontroller is crucial for achieving high precision and reliability in our stepper motor driver system. It ensures that the motor operates accurately, correcting any positional errors in real-time, thus maintaining the desired performance and stability.

Software Design

Introduction

This outlines the design and implementation of a closed-loop stepper motor driver using the STM32F103CBT6 microcontroller unit (MCU) and programmed via STM32CubeIDE. The purpose of this driver is to achieve precise control over the position and speed of a stepper motor by employing feedback mechanisms.

Development Environment

- **IDE:** STM32CubeIDE
- **HAL Library:** STM32CubeMX

Firmware Architecture

Initialization

The initialization process involves setting up the system clock, peripherals, GPIO pins, timers, encoder interface, and communication interfaces.

1. **System Clock Configuration:** Configures the MCU clock for optimal performance.
2. **Peripheral Initialization:** Initializes all necessary peripherals using HAL functions.
3. **GPIO Configuration:** Sets up the GPIO pins for motor control and encoder inputs.
4. **Timer Configuration:** Configures timers for PWM signal generation and encoder reading.
5. **Encoder Interface Configuration:** Initializes the encoder interface to read motor position.
6. **UART/I2C/SPI Configuration:** Configures communication interfaces for debugging or external communication if needed.

Main Loop

The main loop runs continuously to control the stepper motor based on the feedback from the encoder.

1. **Read Encoder Feedback:** Continuously reads the encoder to get the current position of the motor.
2. **Compute Error:** Calculates the error as the difference between the desired position and the current position.
3. **Apply PID Control Algorithm:** Uses the PID algorithm to compute the control signal based on the error.

4. **Generate PWM Signals for Motor Control:** Converts the control signal to PWM signals to adjust the motor speed and direction.
5. **Update Motor Driver:** Sends the PWM signals to the motor driver to control the stepper motor.

Interrupt Service Routines (ISRs)

ISRs handle real-time tasks and ensure responsive and precise control.

1. **Timer Interrupts for PWM Generation:** Ensures accurate and consistent PWM signal generation for motor control.
2. **Encoder Interrupts for Position Update:** Handles encoder signals to update the motor position in real-time.
3. **Communication Interrupts (if applicable):** Manages communication tasks such as data transmission and reception over UART/I2C/SPI.

Conclusion

Summary of the Project

In this project, we hope to successfully designed and implemented a closed-loop stepper motor driver with a feedback mechanism. The primary objective was to create a motor driver that can accurately control the position of the stepper motor and correct any positional errors through real-time feedback. The system comprises a robust power circuit, an efficient motor control unit, and a reliable communication interface. Key components used include the STM microcontroller, magnetic encoder, PWM chopper type DC brushed motor driver, and CAN interfaces.

The project involved meticulous design considerations, such as selecting appropriate trace widths for current capacity, ensuring stable power supply through efficient voltage regulators, and implementing a solid grounding strategy to reduce noise. The feedback mechanism, central to our design, utilized a magnetic encoder to provide real-time positional feedback, enabling precise and stable motor control.

Key Takeaways and Learning Outcomes

1. **Integrated Design Approach:**
    - Learned the importance of integrating various subsystems, such as power, control, and communication, to achieve a cohesive and functional design.
2. **Component Selection and Justification:**
    - Gained insights into selecting appropriate components based on performance requirements and design constraints. Each component was chosen for its specific role and contribution to the overall system efficiency and reliability.
    - All the components except connectors were ordered from mouser electronics and
3. **Feedback Mechanism Implementation:**
    - Developed an understanding of how to implement and utilize feedback mechanisms for error correction and precise control. The use of a magnetic encoder and real-time signal processing by the microcontroller was crucial in achieving accurate motor positioning.
4. **Power Management:**
    - Learned about designing efficient power circuits to provide stable voltage levels, which are essential for the reliable operation of electronic components. The use of switching and LDO regulators was key to maintaining a clean power supply.
5. **Signal Integrity and Grounding:**
    - Recognized the importance of proper grounding techniques and trace width considerations to ensure signal integrity and minimize electromagnetic interference.
6. **CAN Communication:**
    - Acquired knowledge on implementing CAN communication interfaces for connecting and controlling multiple motor drivers, enabling synchronized and coordinated operation in larger systems.
7. **Practical Challenges and Solutions:**
    - Encountered and addressed various practical challenges, such as managing current spikes and ensuring accurate feedback. This involved implementing coupling capacitors and optimizing feedback loops.
8. **Collaboration and Project Management:**
    - Improved skills in collaboration and project management, working as a team to design, test, and refine the stepper motor driver system. Effective communication and task delegation were critical to the project's success.

In conclusion, this project provided valuable hands-on experience in designing a complex electronic system with precise control capabilities. The knowledge and skills gained from this project will be beneficial for future endeavors in electronics and embedded systems design.

Appendix

Daily Log Entries

Feb 5-11:

Activities:

- Through the second round of proposal submission, under guidance of the professor we began the project "Closed loop stepper motor driver"
- As we start we were given the requirements that our final product must reach such as the presion in micro stepping to 6000 , size of the product and we were taught to arrange a meeting with the professor for theoretical guidance .

Outcomes:

- Gained insights into the project and started to research and background learning process on the project .

Feb 12-18:

Activities:

- Had a meeting with the professor where we gained a brief understanding about the theoretical background of the project. Here we learned about space vector PWM which we can use with microstepping, then PID control feedback,Clarke and Park transformations… etc.
- Conducted initial research on the project, focusing on understanding existing solutions in the market related to stepper motor drivers.
- As in the market there was both open and closed loop stepper motor drivers, Though open loop drivers were cost effective there were certain disadvantages when consider with the functionality as follows,
    1. Lack of Real-Time Feedback:

    Open-loop control lacks real-time feedback from the motor. Without feedback, it cannot precisely monitor the motor's position or detect errors.

    In contrast, closed-loop systems use feedback (usually from encoders) to constantly adjust and correct the motor's position, ensuring accuracy and reliability.

2. Stepping Errors:

   In open-loop systems, stepping errors can occur if the input signal is incorrect or if the load changes unexpectedly. These errors reduce acceleration and accuracy2.

   Closed-loop systems continuously adjust the motor's position based on feedback, minimizing stepping errors.

3. Continuous Current Flow

   In open-loop systems, current continues to flow through the motor even when it holds a position (e.g., when the rotor is locked). This results in high power consumption and heat generation.

   Closed-loop systems optimize current flow based on real-time feedback, reducing unnecessary power consumption.

4. Loss of Synchronization:

   Open-loop systems can lose synchronization when the load changes unexpectedly, especially at high speeds or during sudden speed changes.

   Closed-loop systems actively correct for any deviations, maintaining synchronization and accuracy.

Outcomes:

- Gained insights into the current state of stepper motor driver technologies and identified key features and functionalities.
- After identifying the importance of closed loop stepper motor drivers for more precise stepping with comparison to the open loop drivers we came to the conclusion that closed loop stepper motor drivers are much preferable but cost and Complexity is higher so that it is needed to implement the driver cost effective and much more simpler in order for practical use.

Feb 19-25:

Activities:

- Experimented with different approaches for calculations using FPGA and microcontrollers to determine the most suitable technology within budget constraints.
- As the driver should get feedback signals in order to control the driver movements we had to consider whether we use Microcontroller or a FPGA, We understood that whatever we choose should has high frequency abilities, so after considering many FPGAs and micro controllers we came to the final decision to use STM chip as it meet our requirements.

Outcomes:

- Identified that an STM32F103CBT6 microcontroller is cost-effective and meets the project's computational requirements.

Feb 26- March 3:

Activities:

- Developed multiple conceptual designs for the stepper motor driver, evaluating each for feasibility and effectiveness.Here we identified making a separate motor driver block is the most viable design among the ones we proposed for the conceptual designs.

Outcomes:

- Selected a conceptual design that optimally separates the motor from other components for better functionality and control.

March 4 - 10:

Activities:

- Researched and selected circuit designs that align with the chosen conceptual design, focusing on reliability and performance.
- After considering many approaches towards required design the final circuits designs were concluded as to require following requirements

Using CAN Bus Transceiver ,so that multiple drivers can be controlled by user simultaneously.

To use two PWM Chopper Type DC Brushed Motor Drivers to control two poles

Use an external magnetic encoder (if we use design no 2 in conceptual design we had to attach it to the pcb) that should fix behind the stepper motor , thus itis need to make an small pcb to fix behind the stepper motor.

Outcomes:

- Identified and adopted circuit designs that integrate well with the selected conceptual approach.

March 11-17:

Activities:

- Explored and finalized component selections based on performance, compatibility, and availability.
- Most of the components were selected from mouser electronics and order was placed.
- Other components such as screw connectors, jst conncters , headers are purchased from electronic shops.

Outcomes:

- Compiled a list of components necessary for the successful implementation of the stepper motor driver.

March 18-24:

Activities:

- Initiated the design phase by creating circuit diagrams using Altium and visualizing the designs in solidworks for further analysis.
- First we did the component placement without much reasoning , so for again a time we had to make it properly because we identified we had to place the component more conveniently. Placed the coupling capacitors,crystal oscillator as near as the ports of MCU and tried to keep power action separated , to design with sufficient trace widths.

Outcomes:

- Progressed with the circuit design phase and established visual models for evaluation and feedback.

March 25-31:

Activities:

- Reviewed the designed circuits with the lecturer and discussed progress with colleagues to ensure alignment with project goals. This date is not sure but the professor looked through our progress and gave us instructions.

Outcomes:

- Received feedback on the circuit designs, fostering collaboration and refining design elements.

April 1-10 :

Activities:

- Finalized component selections and initiated procurement by placing orders through suppliers like Mouser.
- Those components supposed to by from shops were also purchased.

Outcomes:

- Started the procurement process for components and sent the PCB design for manufacturing.

April 22-May 15:

Activities:

- After arriving the components (after the mid brake)Commenced soldering the PCB with the components, marking the beginning of the physical assembly phase.
- Soldering phase was done keenly.
- Most of the parts were smd components and others were through hole ones.we used a soldering station for this. Yet the testing part was not done due academic strikes.
- Meanwhile we started looking into the software cubeIDE that we are supposed to use for coding the MCU.

Outcomes:

- Started the assembly process, moving towards integration and testing stages to validate the designed stepper motor driver.
- Started software side moving towards coding and debugging.

References

1. **An Introduction to Stepper Motors - University of Arizona**:
    - URL: https://wp.optics.arizona.edu/optomech/wp-content/uploads/sites/53/2016/10/Tutorial_Xinda-Hu.pdf
2. **PLC AS A DRIVER FOR STEPPER MOTOR CONTROL - Laurean Bogdan University**:
    - URL: PLC AS A DRIVER FOR STEPPER MOTOR CONTROL (researchgate.net)
3. **Design and Realization of Stepping Motor Drive System - Springer**:
    - URL: https://link.springer.com/article/10.1007/s11277-022-09534-z
4. **Stepper motor driver, Stepper motor drive - All industrial manufacturers**:
    - URL: Stepper motor driver, Stepper motor drive - All industrial manufacturers (directindustry.com)
5. **Texas Instruments Closed Loop Stepper Design Resources**:
    - URL: https://www.ti.com/solution/closed-loop-stepper
6. **24V/36W BLDC Motor Driver Reference Design With Close-Loop Speed Control**:
    - URL: https://www.ti.com/lit/ug/tiduds5/tiduds5.pdf
7. **Closed Loop Stepper Motor Design With Encoder for Stall-Detection**:
    - URL: https://www.ti.com/lit/pdf/tiducy6
8. **Y Series Closed Loop Driver Manual**:
    - URL: https://www.omc-stepperonline.com/download/Y-series.pdf