Department of Electronic & Telecommunication Engineering,
University of Moratuwa, Sri Lanka.

Pick and Place Robot Arm

Design Document

V 4.0

Discussion Group C
Group 8

Group Members:

210054F	Atapattu A.M.L.R.
210079K	Charles J.
210179R	Gammune D.J.T.
210285M	Kavishan G.T.

Submitted in partial fulfillment of the requirements for the module
EN2160 - Electronic Design Realization

2024/03/26

Contents

1 General **2**
- 1.1 Marketing Needs . . . 2
- 1.2 Existing Product Analysis . . . 2
- 1.3 User Profile . . . 2
  - 1.3.1 Need List . . . 2
- 1.4 Market Segment . . . 2
- 1.5 User Observation . . . 2
- 1.6 Specifications . . . 2

2 Electronics Design **3**
- 2.1 Conceptual Block Diagram . . . 3
- 2.2 Sub-Assembly Design . . . 3
- 2.3 Component Selection . . . 3
  - 2.3.1 Stepper Motors - NEMA 17 . . . 3
  - 2.3.2 Microcontroller - ATmega328P . . . 3
  - 2.3.3 Motor Driver - A4988 . . . 3
- 2.4 Layout Information . . . 4
  - 2.4.1 Component List . . . 4
- 2.5 Final Circuit Diagrams . . . 4
  - 2.5.1 PCB Design . . . 4
- 2.6 PCB Dimensions Finalized . . . 5
- 2.7 Bare PCB . . . 5
- 2.8 Soldered PCB . . . 5
- 2.9 Circuit Testing . . . 5
- 2.10 PCB Testing . . . 6
- 2.11 Industry Standards . . . 6

3 Industrial (Mechanical) Design **6**
- 3.1 Design of Mechanical Sub-assemblies 6
- 3.2 Specifications . . . 6
- 3.3 Selection Criteria for Various Parts 6
- 3.4 Rough Sketches . . . 6
  - 3.4.1 Horizontal and Vertical Conveyor Systems . . . 6
  - 3.4.2 Gripper . . . 6
- 3.5 3D Modeled Simulation . . . 7
  - 3.5.1 3D Printed Parts . . . 7
  - 3.5.2 Mechanical Gripper Mechanism . . . 7
- 3.6 Physically Built Parts . . . 8
  - 3.6.1 Enclosure . . . 8
  - 3.6.2 Conveyor Parts . . . 8
  - 3.6.3 Gripper . . . 8
- 3.7 System Integration . . . 9
  - 3.7.1 Mechanical System Integration 9
- 3.8 Reasons for Selection and Non-selection . . . 9
- 3.9 Standards Used . . . 9
- 3.10 Finish . . . 9
- 3.11 Material Used . . . 9
- 3.12 Final Drawings . . . 9

4 Mechanical Design **9**
- 4.1 Design of Mechanical Sub-assemblies 9
- 4.2 Specifications . . . 9
- 4.3 Selection Criteria for Various Parts 10
- 4.4 Design of Mechanical Sub-Assemblies 10
- 4.5 Specifications of Mechanical Sub-Assemblies . . . 10
- 4.6 Additional Components . . . 10
- 4.7 Related Calculations . . . 11
  - 4.7.1 Basic Parameters . . . 11
  - 4.7.2 RPM and Velocity . . . 11
  - 4.7.3 Acceleration . . . 11
  - 4.7.4 Distance Calculations . . . 11
  - 4.7.5 Velocity Time Graphs for Motions . . . 12

5 Software Structure **12**
- 5.1 Structure Diagram . . . 12
- 5.2 Rough Work . . . 12
- 5.3 Flowchart . . . 13

6 Code Base - Language: C++ **14**

A Log Entries **20**
- A.1 26 February - 3 March . . . 20
  - A.1.1 Research Phase . . . 20
- A.2 4 March - 10 March . . . 20
  - A.2.1 Conceptualization Phase . . 20
- A.3 11 March - 17 March . . . 20
  - A.3.1 Design Evaluation Phase . . 20
- A.4 18 March - 24 March . . . 20
  - A.4.1 Gripper Mechanism Selection 20
- A.5 25 March - 31 March . . . 20
  - A.5.1 Design Phase . . . 20
- A.6 1 April - 7 April . . . 21
  - A.6.1 Design Refinement . . . 21
- A.7 8 April - 14 April . . . 21
  - A.7.1 Design Finalizing . . . 21
- A.8 15 April - 21 April . . . 21
  - A.8.1 PCB Printing . . . 21
- A.9 22 April - 28 April . . . 21
  - A.9.1 Final Design Adjustments . 21
- A.10 29 April - 5 May . . . 21
  - A.10.1 Review and Testing . . . 21
- A.11 6 May - 12 May . . . 21
  - A.11.1 Controller Enclosure . . . 21
- A.12 13 May - 19 May . . . 21
  - A.12.1 Final Report Preparation . 21

B Declaration **22**

C Document Reviews **22**

1 General

1.1 Marketing Needs

The inspiration for an automated system to assemble electronic components such as rivets, washers, and transistors onto cylindrical heat sinks for a 3-phase H-bridge system using industrial-grade motors stemmed from discussions with Professor Jayasinghe.

1.2 Existing Product Analysis

Solutions from industry leaders like ABB, KUKA, FANUC, and Yaskawa Motoman were analyzed through YouTube videos, product brochures, and academic papers. Key features such as multi-axis motion control, gripper customization, and integrated vision systems were highlighted.

1.3 User Profile

The target users for the product are manufacturers and industrial facilities involved in electronics assembly and production.

1.3.1 Need List

a) **High Precision:** The robot arm should be capable of precisely picking up and placing electronic components with accuracy in the micrometer range to ensure proper assembly.

b) **Adaptability:** The system should be adaptable to handle a variety of electronic components of different shapes, sizes, and weights commonly used in electronics manufacturing.

c) **Speed and Efficiency:** The pick and place process should be efficient, with high-speed movements to optimize production throughput.

d) **Reliability:** The system should operate reliably over extended periods without frequent breakdowns, ensuring continuous production without interruptions.

e) **User-Friendly Interface:** The control interface should be intuitive and user-friendly, allowing operators to easily program and operate the robot arm without extensive training.

f) **Safety Features:** Incorporate safety features such as emergency stop mechanisms and protective enclosures to prevent accidents and ensure the safety of operators and nearby personnel.

g) **Integration with Existing Systems:** The robot arm should be compatible with existing production systems and workflows, allowing seamless integration into the manufacturing process.

h) **Cost-Effectiveness:** The overall cost of the system, including initial investment, maintenance, and operational costs, should be reasonable and cost-effective compared to the benefits it provides.

i) **Scalability:** The system should be scalable to accommodate future expansion and changes in production requirements, allowing for flexibility and adaptability to evolving needs.

j) **Documentation and Support:** Provide comprehensive documentation and support resources, including user manuals, troubleshooting guides, and technical support, to assist users in operating and maintaining the system effectively.

k) **Environmental Considerations:** Minimize environmental impact by designing energy-efficient components and using sustainable materials wherever possible.

l) **Compliance with Regulations:** Ensure compliance with relevant regulations and standards governing robotics and manufacturing processes to meet legal requirements and industry best practices.

1.4 Market Segment

The market segment for the product is the industrial automation and electronics manufacturing sector.

1.5 User Observation

We used internet to check the traditional hand picked method of the system and observed the targeting users.

1.6 Specifications

The specifications of the device are as follows:

- **Control Capabilities:** The device can control various connected devices and move in four directions: left, right, up, and down, using specified commands.
- **Automatic Functionality:** When the automatic function is enabled, the device operates flawlessly, executing predefined tasks with precision and efficiency.

- **Platform Compatibility:** The device is designed to be placed on a platform approximately 30cm high, providing flexibility in placement options and compatibility with various setups.

2 Electronics Design

2.1 Conceptual Block Diagram

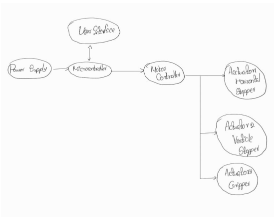


Figure 1: Electronic Block Diagram

2.2 Sub-Assembly Design

The system includes electronic sub-assemblies such as motor drivers, microcontrollers, and control circuits.

2.3 Component Selection

In electronic design, selecting components is pivotal for the success of any project, influencing functionality, efficiency, and reliability. This section details the key components chosen for our project, focusing on stepper motors, microcontrollers, motor drivers, and an emergency stop switch.

2.3.1 Stepper Motors - NEMA 17

- **Model Selected:** NEMA 17 (Nenna 17)
- **Reasons for Selection:**
  - **Specifications and Performance:** The NEMA 17 stepper motor was chosen for its robust performance characteristics, including a 1.8-degree step angle (200 steps per revolution), sufficient torque output at 12 volts, and high reliability under load conditions (25N axial, 65N push, 29N pull radial).
  - **Application Suitability:** While closed-loop systems with encoders were considered superior for precise control and for this project, **budget constraints** led us to opt for an open-loop system. Through detailed analysis, the NEMA 17 was found adequate for our project's positioning accuracy and speed control requirements.
- **Alternative Considerations:** Several alternative NEMA stepper motors were evaluated but deemed less suitable due to either lower torque, higher voltage requirements, or larger physical size compared to the NEMA 17.

Figure 2: NEMA 17 Stepper Motor

2.3.2 Microcontroller - ATmega328P

- **Reasons for Selection:**
  - **Direct Register-Level Programming:** The ATmega328P was chosen for its ability to facilitate precise control through low-level hardware manipulation, essential for optimizing performance in our application.
  - **Versatility and I/O Capability:** With 23 GPIO pins and PWM support, it offers flexibility for controlling multiple stepper motors and integrating additional peripherals.
  - **Operational Efficiency:** Operating from 1.8V to 5.5V, it adapts well to various power scenarios, ensuring robust power management and compatibility.
- **Comparison with Alternatives:** The ATmega328P was preferred over alternatives like the STM32F103C8T6 and PIC16F877A due to its balance of computational efficiency, ease of programming, and suitability for embedded systems.

2.3.3 Motor Driver - A4988

- **Reasons for Selection:**
  - **Wide Operating Range and Current Handling:** The A4988 motor driver, with an 8V to 35V range and capability to handle 1A continuous current per phase (2A peak), was selected for its versatility and reliability in driving stepper motors.

  - **Simplicity and Cost-effectiveness:** Known for its straightforward implementation and economical pricing, it meets our project's requirements without compromising essential features.
- **Comparison with Alternatives:** Alternatives such as DRV8825 and TB67S249-FTG were considered but were either overspecification for our needs or less cost-effective.

2.4 Layout Information

2.4.1 Component List

1. **ATmega328P Microcontroller**
   - **Model:** ATmega328P-PU
   - **Description:** An 8-bit AVR microcontroller with 23 GPIO pins, 32KB flash memory, and 2KB SRAM. Ideal for embedded applications requiring precise control.
2. **LM7805 Voltage Regulator**
   - **Model:** LM7805CT
   - **Description:** A linear voltage regulator that outputs a stable 5V from an input voltage between 7V and 35V. Provides up to 1.5A of current.
3. **LM7812 Voltage Regulator**
   - **Model:** LM7812CT
   - **Description:** A linear voltage regulator that outputs a stable 12V from an input voltage between 14V and 35V. Provides up to 1.5A of current.
4. **A4988 Stepper Motor Driver ICs (3x)**
   - **Model:** A4988
   - **Description:** A microstepping driver for controlling bipolar stepper motors, capable of up to 1/16th microstepping. Features adjustable current limiting and thermal protection.
5. **Crystal Oscillator (16MHz)**
   - **Model:** HC-49S 16MHz
   - **Description:** Provides a stable 16MHz clock signal for the microcontroller, ensuring accurate timing and system stability.
6. **Ceramic Capacitors (for Decoupling)**
   - **Model:** MLCC 0.1µF 50V
   - **Description:** Multilayer ceramic capacitors used for noise reduction and voltage stabilization in power and signal lines.
7. **Electrolytic Capacitors (for Input/Output Filtering)**
   - **Model:** 100µF 25V
   - **Description:** Used for filtering and smoothing power supply outputs to maintain stable voltage levels.
8. **Resistors**
   - **Model:** 1/4W 1k
   - **Description:** Standard 1k resistors used for current limiting, voltage division, and signal pull-up/pull-down applications.
9. **Diodes**
   - **Model:** 1N4007
   - **Description:** General-purpose diodes used for rectification and protection in circuits, capable of handling up to 1A of current.
10. **Programming Header**
    - **Model:** 2x3 1.27mm Pitch
    - **Description:** A compact header for connecting the microcontroller to an external programmer for code uploading and debugging.
11. **External Power Source (12V DC)**
    - **Model:** ALITOVE 24V 5A
    - **Description:** A reliable 24V DC power supply for powering the entire circuit.

2.5 Final Circuit Diagrams

2.5.1 PCB Design

Key Considerations:

- **Traces:** Wide power traces minimize voltage drop and support high currents. Signal traces are carefully routed for interference mitigation and signal integrity.
- **Component Placement:** Strategic positioning reduces trace lengths and optimizes signal paths. Motor drivers are placed near motor terminals to reduce noise.

2.6 PCB Dimensions Finalized

The dimenssions of the PCB is finalized to 77.34mm in length and 66.42mm in width.

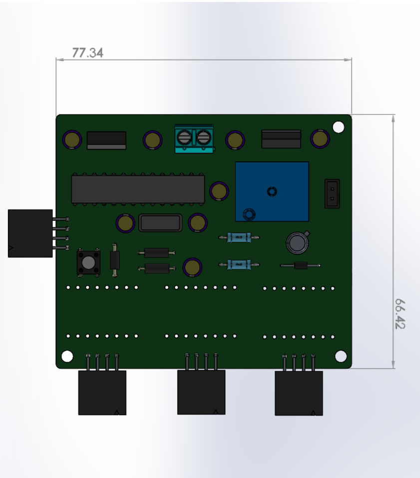
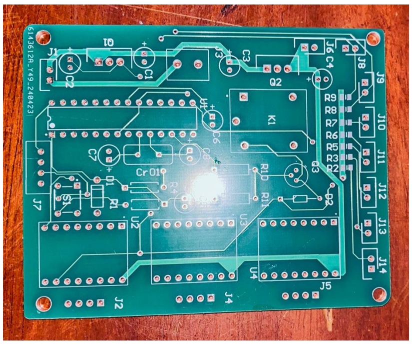
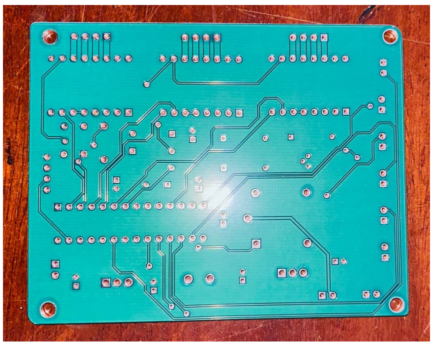


Figure 3: PCB with Dimenssions

2.7 Bare PCB



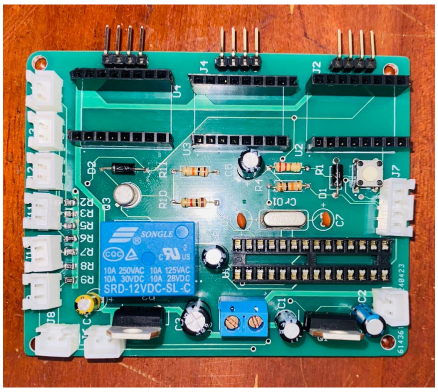
(a) Bare PCB Top



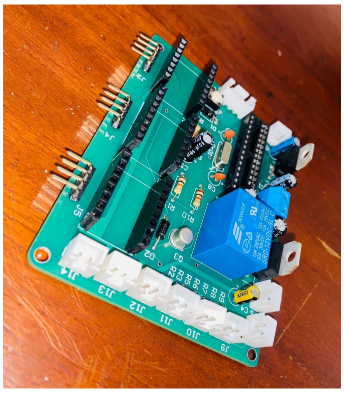
(b) Bare PCB Bottom

Figure 4: Bare PCB views

2.8 Soldered PCB



(a) Soldered PCB



(b) Soldered PCB

Figure 5: Soldered PCB views

2.9 Circuit Testing

During the testing phase, the following steps were undertaken to ensure the functionality and reliability of the circuits:

1. **Power Circuit Testing:** The power circuit was initially created, and power was supplied to the circuit. Voltages at different points in the circuit were measured to ensure proper functioning and stability.

2. **Microcontroller Circuit Testing:** Once the power circuit was verified, the microcontroller circuit part was connected. Voltages at various nodes of the microcontroller circuit were checked to confirm correct operation.

3. **Motor Driver, Relays, and Button JSTs Testing:** Subsequently, the motor drivers, relays, and button JSTs were connected to the circuit. Voltage testing was conducted to ensure that these components were receiving the appropriate voltage levels and functioning as expected.

2.10 PCB Testing

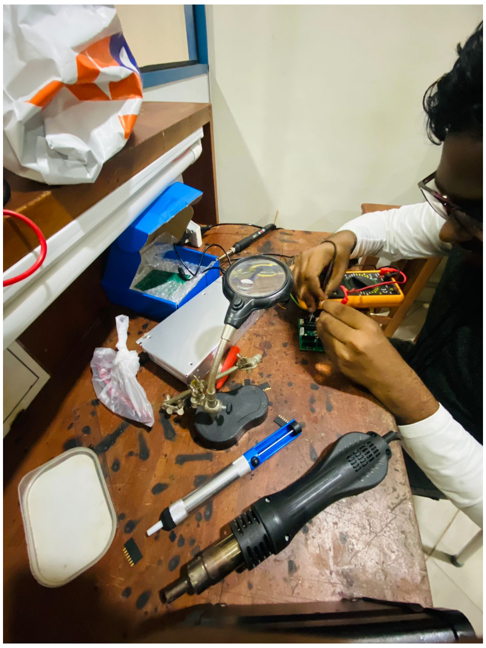


Figure 6: PCB Testing

2.11 Industry Standards

1. **WMMA/ANSI O1.1-3 -** Safety Requirements for CNC Machining Centers for the Woodworking Industry

2. **DIN 58768 -** Production in optical engineering - Input parameters for CNC machine tools

3. **API TR 7CR -** Cold Working Thread Roots with CNC Lathes for Rotary Shouldered Connections

4. **AGMA 91FTM2 -** CNC Technology and New Calculation Methods Permit Efficient System Independent Manufacturing of Spiral Bevel Gears

3 Industrial (Mechanical) Design

3.1 Design of Mechanical Sub-assemblies

The mechanical design includes a modular robot arm design, customized grippers for components, and a user-friendly programming interface. Safety measures like emergency stop systems and intuitive controls were prioritized. The design also considers the development of a slider system for efficient movement and stacking.

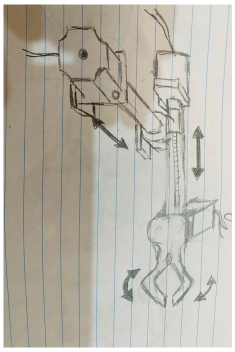


Figure 7: Mechanical Concept Design

3.2 Specifications

3.3 Selection Criteria for Various Parts

The selection of parts for the mechanical sub-assemblies is based on several criteria to ensure optimal performance and reliability of the system. Factors considered during part selection may include functionality, compatibility, durability, cost-effectiveness, availability, manufacturability, safety standards compliance, and material suitability.

3.4 Rough Sketches

3.4.1 Horizontal and Vertical Conveyor Systems

The horizontal conveyor system serves as the foundation for the linear movement of the robotic arm along the x-axis.

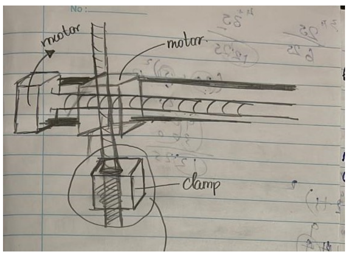


Figure 8: Proposed Device sketch

3.4.2 Gripper

- **Gripper Module:** The gripper module is designed to securely hold and release electronic components during the pick and place operation. It is attached to the vertical conveyor system to enable controlled vertical movement.

- **Actuation System:** The actuation system within the gripper module allows controlled opening and closing of the gripper, ensuring a secure hold on the components during transportation.

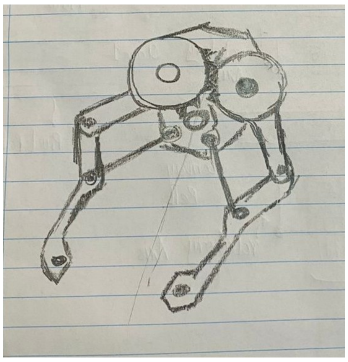


Figure 9: Mechanical Gripper Concept

3.5 3D Modeled Simulation

We created a Solid Works 3D Model for simulation of the Mechanical Concepts (using mates and etc.)

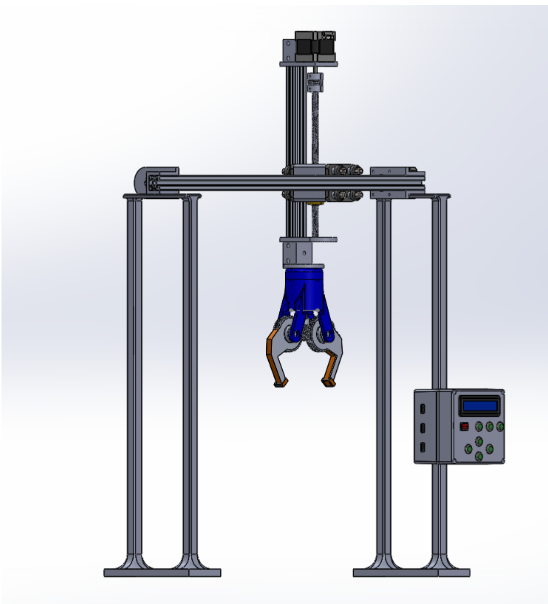
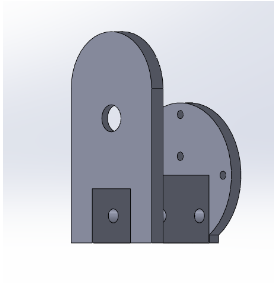


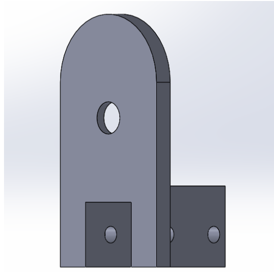
Figure 10: 3D Model

3.5.1 3D Printed Parts

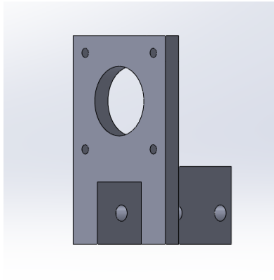
The connecting parts of the device will be created using 3D printing with 100% fill as it needs to be hard.



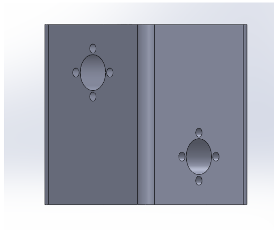
(a) Connector 1 - With Gripper Mount



(b) Connector 2



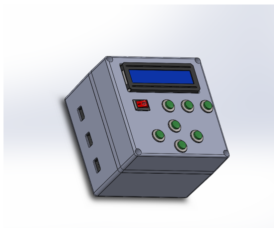
(c) Connector 3



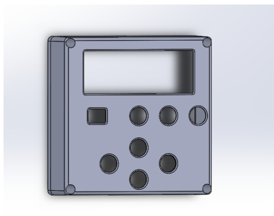
(d) Base Plate

Figure 11: 3D Printed Coupling Parts

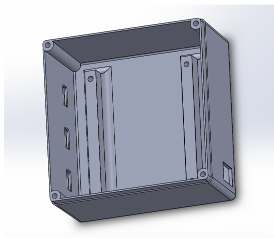
The for the enclosure we are using 3D printing to create the enclosure.



(a) Assembly of the Enclosure



(b) Top Part



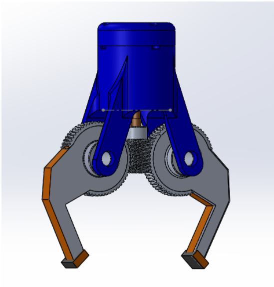
(c) Bottom Part

Figure 12: Enclosure Views

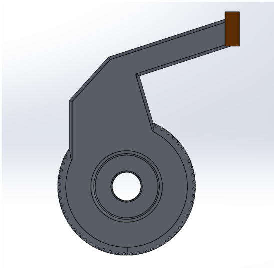
To create these plastic enclosures, 3D printing materials such as Nylon PA12 or Nylon PA11 are planned to be used.

3.5.2 Mechanical Gripper Mechanism

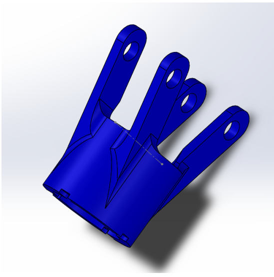
The gripper mechanism is connected to the vertical conveyor system and is responsible for picking up and placing H-Bridge components. It includes:



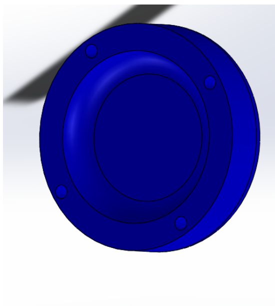
(a) Assembly of the Gripper



(b) Gripper Finger



(c) Stepper Housing 1



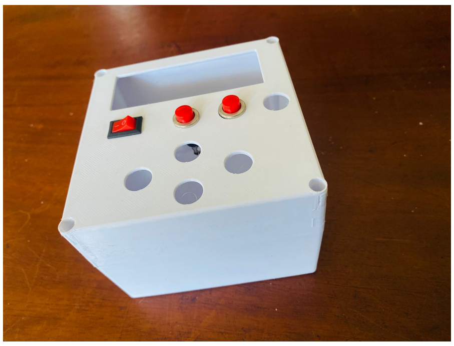
(d) Stepper Housing 2

Figure 13: Gripper 3D Model

3.6 Physically Built Parts

3.6.1 Enclosure

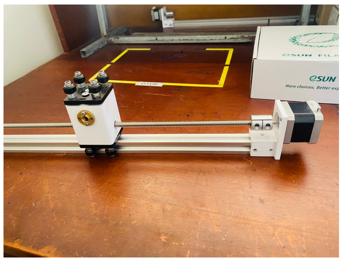
(a) Enclosure Bottom



(b) Enclosure Full

Figure 14: Enclosure Design views

3.6.2 Conveyor Parts

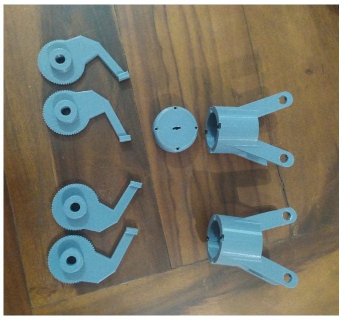


(a) Single Axis Conveyor Parts (View 1)

(b) Single Axis Conveyor Parts

Figure 15: Single Axis Conveyor Parts views

3.6.3 Gripper



(a) Gripper

(b) Gripper

Figure 16: Gripper views

3.7 System Integration

3.7.1 Mechanical System Integration

(a) Mechanical System Integration

(b) Mechanical System Integration

Figure 17: Mechanical System Integration views

3.8 Reasons for Selection and Non-selection

Reasons for the selection and non-selection of mechanical parts include factors such as performance, cost, availability, compatibility, manufacturability, and adherence to design specifications. Any changes made to the original design concept will also recorded.

3.9 Standards Used

Relevant industry standards for mechanical design and fabrication will be followed to ensure compliance with safety, quality, and regulatory requirements. Standards may include ISO, ANSI, ASTM, and specific industry guidelines for robotics and automation.

3.10 Finish

The finish of mechanical components is determined based on functional requirements, aesthetic considerations, and environmental factors. Common finishes include paint, powder coating, anodizing, plating, and surface treatments to enhance durability and appearance.

3.11 Material Used

Materials for mechanical components are selected based on their mechanical properties, compatibility with environmental conditions, manufacturability, and cost-effectiveness. Common materials include metals (e.g., aluminum, steel), plastics, composites, and alloys.

3.12 Final Drawings

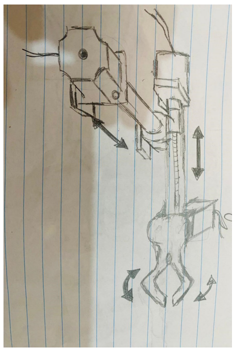


Figure 18: Mechanical Concept Design

4 Mechanical Design

4.1 Design of Mechanical Sub-assemblies

The mechanical design of the system as a whole involves the creation and integration of various sub-assemblies such as the robot arm, gripper mechanism, sliding system, and any other mechanical components necessary for the functioning of the system. Each sub-assembly will be designed to fulfill specific functions within the system, such as precise movement, component manipulation, and overall structural stability.

4.2 Specifications

Specifications for the mechanical sub-assemblies outline the desired characteristics and performance parameters that each component should meet. This includes factors such as dimensions, material properties, load-bearing capacity, precision, and compatibility with other system components. For example, specifications for the robot arm may include its length, payload capacity, range of motion, and speed, while specifications for the gripper mechanism may include its gripping force, opening/closing mechanism, and adaptability to different component shapes and sizes.

4.3 Selection Criteria for Various Parts

The selection of parts for the mechanical sub-assemblies is based on several criteria to ensure optimal performance and reliability of the system. Factors considered during part selection may include:

- **Functionality:** Parts should fulfill their intended functions within the system, such as providing movement, support, or gripping capabilities.

- **Compatibility:** Parts should be compatible with other components in terms of size, shape, mounting interfaces, and operational requirements.

- **Durability and Reliability:** Parts should be durable enough to withstand the operational stresses and environmental conditions they will be exposed to during use. They should also have a proven track record of reliability to minimize maintenance and downtime.

- **Cost-effectiveness:** Parts should be cost-effective in terms of initial procurement costs as well as long-term maintenance and replacement expenses.

- **Availability:** Parts should be readily available from reliable suppliers to ensure timely procurement and maintenance support.

- **Manufacturability:** Parts should be manufacturable using appropriate techniques and technologies, considering factors such as production lead times, scalability, and ease of assembly.

- **Safety:** Parts should meet safety standards and regulations to ensure the safety of operators and other personnel interacting with the system.

By adhering to these selection criteria, the mechanical design team can choose the most suitable parts and components to construct the mechanical sub-assemblies, thereby contributing to the overall functionality, efficiency, and longevity of the system.

4.4 Design of Mechanical Sub-Assemblies

1. **Robotic Arm Assembly:**
   - The robotic arm features a multi-axis design enabling precise movement in 2 directions. It incorporates specialized grippers for secure handling of components without deformation.
   - Designed with a frame that support high-speed operations and maintain stability.
2. **Gripper:**
   - The gripper utilizes a compliant mechanism to grab the the parts from the stacks and placing them.
3. **Base and Frame Structure:**
   - Provides structural support for all components, designed for stability and rigidity.
   - Constructed from high-strength aluminum alloy, balancing durability with lightweight properties.
   - A lifted frame that can endure the weight of the full device keeping it stabele.

4.5 Specifications of Mechanical Sub-Assemblies

1. **Robotic Arm:**
   - Payload Capacity: 1 kg
   - Material: 3D printed
   - Drive System: Stepper Motors
2. **Gripper:**
   - Grip Force: Adjustable, up to 10 N
   - Jaw Opening Range: 5 mm to 40 mm
   - Material: 3D printed
3. **Base and Frame Structure:**
   - Material: Extruded aluminum with stainless steel reinforcements
   - Dimensions: 600 mm x 500 mm x 800 mm (L x W x H)

4.6 Additional Components

- **T8 Lead Screw 1000mm ACME Thread:**
  - Length: 1000 mm
  - Type: ACME thread for efficient linear motion
  - Pitch : 2mm

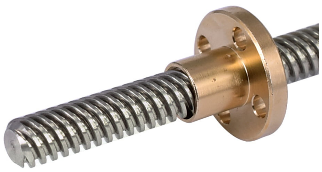


Figure 19: T8 Lead Screw

- **Aluminum Flexible Coupling 5×8mm:**
  - Compatible with 5 mm motor shafts and 8 mm lead screws, facilitating efficient power transmission.

Figure 20: Coupling

- **V Slot Aluminum Extrusion Profile:**
  - Offers versatile mounting options for linear motion components in mechanical assemblies.
  - Balances a high strength-to-weight ratio, ensuring structural integrity without unnecessary weight.
  - T-slot design simplifies assembly processes, enhancing efficiency during system setup.

Figure 21: V Slot

- **Openbuilds (Type A) V Gantry Plate Set:**
  - Designed specifically for compatibility with V Slot aluminum extrusions.
  - Ensures durability, stability, and precise alignment, crucial for reliable mechanical assemblies.
  - Modular design facilitates flexible configurations tailored to varying assembly needs.

Figure 22: Gantry Plate

4.7 Related Calculations

4.7.1 Basic Parameters

- Steps per revolution: 200
- Screw rod pitch: 2 mm
- Distance per step: $\frac{2\text{ mm}}{200} = 0.01$ mm

4.7.2 RPM and Velocity

- Set RPM: 1000
- Max velocity:
  2 mm × 1000 RPM = 33.33 mm/s

4.7.3 Acceleration

- Set Acceleration: 10 mm/s^2 (Acceptable for the product)
- Time to stop: $\frac{33.33\text{ mm/s} - 0\text{ mm/s}}{10\text{ mm/s}^2} = 3.33$ s

4.7.4 Distance Calculations

- Distance travelled with acceleration/deceleration:

$$\frac{1}{2} \times 10 \times 3.33 = 16.65 \text{ mm}$$

- Total Distance from acceleration & deceleration:

$$2 \times 16.65 \text{ mm} = 33.3 \text{ mm} =$$

  Min Distance between start position to first part

4.7.5 Velocity Time Graphs for Motions

- Distance : 100mm
- Time of Motion : 8.6s

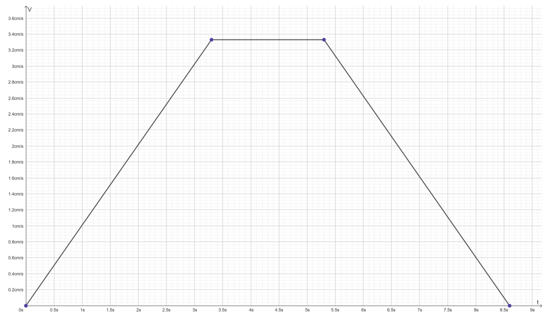


Figure 23: Item 1 - Velocity Time Graph (100mm)

- Distance : 250mm
- Time of Motion : 13.1s

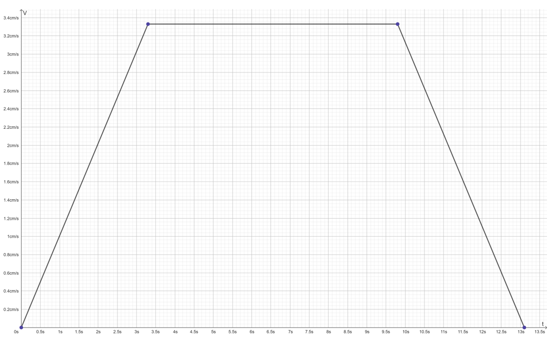


Figure 24: Item 2 - Velocity Time Graph (250mm)

- Distance : 400mm
- Time of Motion : 17.6s

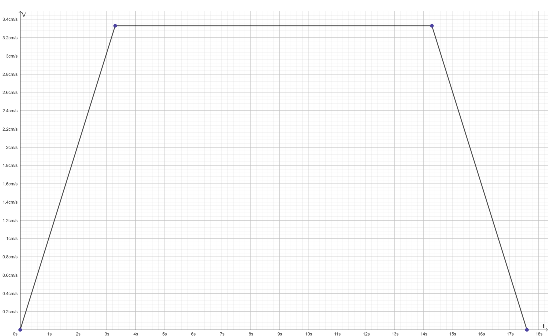


Figure 25: Item 3 - Velocity Time Graph (400mm)

5 Software Structure

5.1 Structure Diagram

The software could follow a hierarchical modular structure with the following components:

```
Main Program
 - Initialization Module
 - Pivot Handling Module
    - Grab Pivot
    - Place Pivot
    - Fault Handling
 - MOSFET Handling Module
    - Grab MOSFET
    - Place MOSFET
    - Fault Handling
 - Washer Handling Module
    - Grab Washer
    - Place Washer
    - Fault Handling
 - Step Control Module
 - User Interface Module
```

5.2 Rough Work

Based on the flowchart, some rough pseudocode or planning notes could be:

Algorithm 1 Assembly Process

Function `assemble()`**:**
  Initialize system
  while *True* **do**
    if *pivot not initialized* **then**
      Call Grab_Pivot()
    if *pivot not placed correctly* **then**
      Call Place_Pivot()
    if *MOSFET not initialized* **then**
      Call Grab_MOSFET()
    if *MOSFET not placed correctly* **then**
      Call Place_MOSFET()
    if *washer not initialized* **then**
      Call Grab_Washer()
    if *washer not placed correctly* **then**
    if *all parts placed* **then**
      Call Step_Control() ; `// Move to next step`
    if *stop button pressed* **then**
      return ; `// Exit loop`

5.3 Flowchart

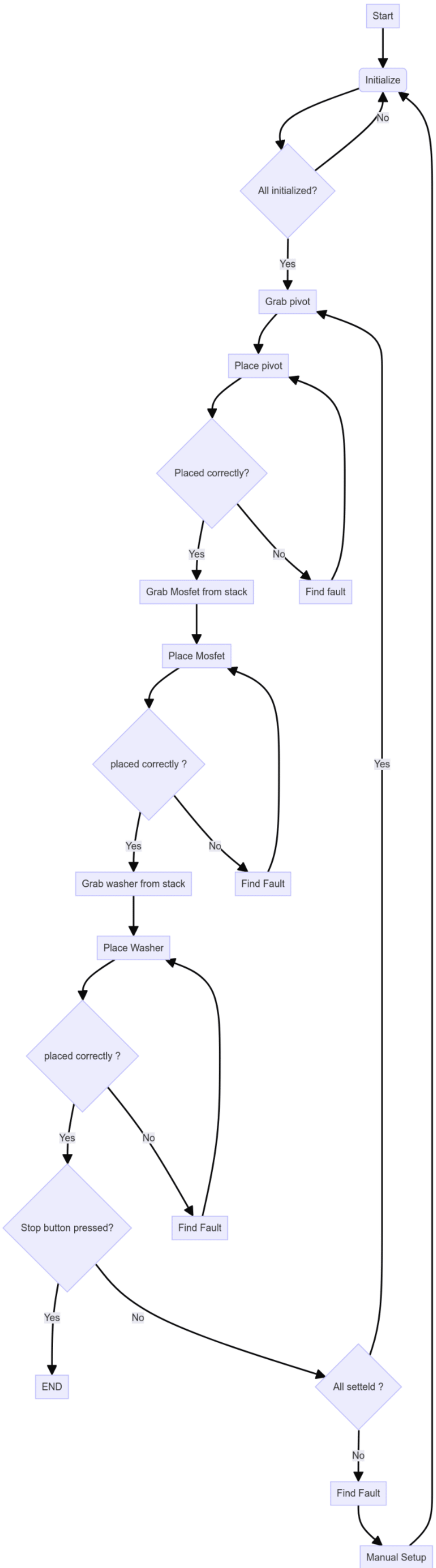


Figure 26: Software Flow Chart

6 Code Base - Language: C++

```cpp
#include <avr/io.h>
#include <util/delay.h>
#include <stdbool.h>
#include <avr/eeprom.h>

// Define the pins for the stepper motors
#define STEP_PIN1 PD0
#define DIR_PIN1  PD1
#define STEP_PIN2 PD2
#define DIR_PIN2  PD3
#define STEP_PIN3 PD4
#define DIR_PIN3  PD5

// Define the pins for the buttons
#define START_BUTTON_PIN PB0
#define STOP_BUTTON_PIN  PB1
#define EMERGENCY_STOP_BUTTON_PIN PB2
#define UP_BUTTON_PIN    PB3
#define DOWN_BUTTON_PIN PB4
#define LEFT_BUTTON_PIN PB5
#define RIGHT_BUTTON_PIN PB6
#define CALIBRATION_BUTTON_PIN PB7

// Define the I2C address for the LCD
#define LCD_I2C_ADDR 0x27

// Define the steps per millimeter for each stepper motor
const float stepsPerMmHorizontal = 100.0;
const float stepsPerMmVertical = 100.0;
const float stepsPerMmGripper = 100.0;

// Define variables for the current positions of the stepper motors
int16_t currentHorizontalPosition = 0;
int16_t currentVerticalPosition = 0;
int16_t currentGripperPosition = 0;

// Define helper macros for setting and clearing bits in a port
#define SET_BIT(port, pin)   ((port) |= (1 << (pin)))
#define CLEAR_BIT(port, pin) ((port) &= ~(1 << (pin)))

// Function to initialize the pins
void init_pins() {
    // Set the direction of stepper motor pins as output
    DDRD |= (1 << STEP_PIN1) | (1 << DIR_PIN1) | (1 << STEP_PIN2) | (1 << DIR_PIN2) | (1 << STEP_PIN3) | (1 << DIR_PIN3);
    // Set the direction of button pins as input
    DDRB &= ~((1 << START_BUTTON_PIN) | (1 << STOP_BUTTON_PIN) | (1 << EMERGENCY_STOP_BUTTON_PIN) | (1 << UP_BUTTON_PIN) | (1 << DOWN_BUTTON_PIN) | (1 << LEFT_BUTTON_PIN) | (1 << RIGHT_BUTTON_PIN) | (1 << CALIBRATION_BUTTON_PIN));
    // Enable internal pull-up resistors for button pins
    PORTB |= (1 << START_BUTTON_PIN) | (1 << STOP_BUTTON_PIN) | (1 << EMERGENCY_STOP_BUTTON_PIN) | (1 << UP_BUTTON_PIN) | (1 << DOWN_BUTTON_PIN) | (1 << LEFT_BUTTON_PIN) | (1 << RIGHT_BUTTON_PIN) | (1 << CALIBRATION_BUTTON_PIN);
}

// Function to initialize I2C communication
void i2c_init() {
    // Initialize I2C communication
```

```
    TWSR = 0x00;
    TWBR = 0x48;
    TWCR = (1 << TWEN);
}

// Function to start I2C communication
void i2c_start() {
    // Start I2C communication
    TWCR = (1 << TWSTA) | (1 << TWEN) | (1 << TWINT);
    while (!(TWCR & (1 << TWINT)));
}

// Function to stop I2C communication
void i2c_stop() {
    // Stop I2C communication
    TWCR = (1 << TWSTO) | (1 << TWEN) | (1 << TWINT);
}

// Function to write data over I2C
void i2c_write(uint8_t data) {
    // Write data over I2C
    TWDR = data;
    TWCR = (1 << TWEN) | (1 << TWINT);
    while (!(TWCR & (1 << TWINT)));
}

// Function to send command to LCD
void lcd_command(uint8_t cmd) {
    // Send command to LCD
    i2c_start();
    i2c_write(LCD_I2C_ADDR << 1);
    i2c_write(0x00);
    i2c_write(cmd);
    i2c_stop();
    _delay_ms(2);
}

// Function to send data to LCD
void lcd_data(uint8_t data) {
    // Send data to LCD
    i2c_start();
    i2c_write(LCD_I2C_ADDR << 1);
    i2c_write(0x40);
    i2c_write(data);
    i2c_stop();
    _delay_ms(2);
}

// Function to initialize the LCD
void lcd_init() {
    _delay_ms(50);

    // Initialize LCD
    lcd_command(0x38);
    lcd_command(0x0C);
    lcd_command(0x06);
    lcd_command(0x01);
    _delay_ms(2);
}

// Function to print a string on the LCD
```

```
void lcd_print(const char *str) {
    // Print string on LCD
    while (*str) {
        lcd_data(*str++);
    }
}

// Function to move a stepper motor
void move_stepper(volatile uint8_t *port, uint8_t stepPin, uint8_t dirPin, int16_t *currentPosition, float distance_mm, float stepsPerMm) {
    // Constants for acceleration and maximum velocity
    const float acceleration = 5.0;
    const float max_velocity = 33.33;
    const float distance_per_step = 0.01;

    // Calculate the total number of steps required to move the desired distance
    int total_steps = distance_mm * stepsPerMm;
    int current_steps = 0;

    // Calculate the acceleration time and distance
    float acc_time = max_velocity / acceleration;
    float acc_distance = 0.5 * acceleration * acc_time * acc_time;
    int acc_steps = acc_distance / distance_per_step;

    // Set the direction of the motor based on the distance
    if (distance_mm < 0) {
        CLEAR_BIT(*port, dirPin);
        distance_mm = -distance_mm;
    } else {
        SET_BIT(*port, dirPin);
    }

    // Move the motor step by step
    while (current_steps < total_steps) {
        float current_velocity = 0.0;
        if (current_steps < acc_steps) {
            current_velocity = acceleration * (current_steps * distance_per_step) / max_velocity;
        } else if (current_steps >= total_steps - acc_steps) {
            current_velocity = acceleration * ((total_steps - current_steps) * distance_per_step) / max_velocity;
        } else {
            current_velocity = max_velocity;
        }

        // Calculate the delay between steps based on the current velocity
        float step_delay = 1.0 / (stepsPerMm * current_velocity) * 1e6;

        // Step the motor
        SET_BIT(*port, stepPin);
        _delay_us(step_delay / 2);
        CLEAR_BIT(*port, stepPin);
        _delay_us(step_delay / 2);

        current_steps++;
    }

    // Update the current position of the motor
    *currentPosition += distance_mm * (stepsPerMm / 100.0);
}
```

```

// Function to move the horizontal stepper motor
void moveHorizontal(float distance_mm) {
    move_stepper(&PORTD, STEP_PIN1, DIR_PIN1, &currentHorizontalPosition,
       ↪ distance_mm, stepsPerMmHorizontal);
}

// Function to move the vertical stepper motor
void moveVertical(float distance_mm) {
    move_stepper(&PORTD, STEP_PIN2, DIR_PIN2, &currentVerticalPosition,
       ↪ distance_mm, stepsPerMmVertical);
}

// Function to move the gripper stepper motor
void moveGripper(float distance_mm) {
    move_stepper(&PORTD, STEP_PIN3, DIR_PIN3, &currentGripperPosition,
       ↪ distance_mm, stepsPerMmGripper);
}

// Function to perform a predefined sequence of movements
void performSequence() {
    moveHorizontal(100);
    moveVertical(100);
    moveGripper(10);
    moveVertical(50);
    moveHorizontal(-100);
    moveVertical(-100);
    moveGripper(-10);

    moveHorizontal(250);
    moveVertical(100);
    moveGripper(10);
    moveVertical(50);
    moveHorizontal(-250);
    moveVertical(-100);
    moveGripper(-10);

    moveHorizontal(400);
    moveVertical(100);
    moveGripper(10);
    moveVertical(50);
    moveHorizontal(-400);
    moveVertical(-100);
    moveGripper(-10);
}

// Function to check if a button is pressed
bool isButtonPressed(uint8_t pin) {
    if (!(PINB & (1 << pin))) {
        _delay_ms(50);
        if (!(PINB & (1 << pin))) {
            return true;
        }
    }
    return false;
}

// Function to restore the initial positions of the stepper motors
void restorePosition() {
    int16_t initialHorizontalPosition = eeprom_read_word(&
       ↪ eepromInitialHorizontalPosition);
```

```c
    int16_t initialVerticalPosition = eeprom_read_word(&eepromInitialVerticalPosition);
    int16_t initialGripperPosition = eeprom_read_word(&eepromInitialGripperPosition);

    moveHorizontal(initialHorizontalPosition / stepsPerMmHorizontal - currentHorizontalPosition / stepsPerMmHorizontal);
    moveVertical(initialVerticalPosition / stepsPerMmVertical - currentVerticalPosition / stepsPerMmVertical);
    moveGripper(initialGripperPosition / stepsPerMmGripper - currentGripperPosition / stepsPerMmGripper);

    currentHorizontalPosition = initialHorizontalPosition;
    currentVerticalPosition = initialVerticalPosition;
    currentGripperPosition = initialGripperPosition;
}

int main(void) {
    init_pins();
    i2c_init();
    lcd_init();
    lcd_print("System Ready");

    restorePosition();

    while (1) {
        if (isButtonPressed(START_BUTTON_PIN)) {
            running = true;
            emergencyStopped = false;
            calibrationMode = false;
            lcd_command(0x01);
            lcd_print("Running...");
        }

        if (isButtonPressed(STOP_BUTTON_PIN)) {
            running = false;
            lcd_command(0x01);
            lcd_print("Stopped");
        }

        if (isButtonPressed(EMERGENCY_STOP_BUTTON_PIN)) {
            running = false;
            emergencyStopped = true;
            lcd_command(0x01);
            lcd_print("Emergency Stop");
        }

        if (isButtonPressed(CALIBRATION_BUTTON_PIN)) {
            calibrationMode = !calibrationMode;
            if (calibrationMode) {
                lcd_command(0x01);
                lcd_print("Calibration Mode");
            } else {
                lcd_command(0x01);
                lcd_print("System Ready");
            }
        }

        if (calibrationMode) {
            if (isButtonPressed(UP_BUTTON_PIN)) {
                moveVertical(10);
```

```
            }
            if (isButtonPressed(DOWN_BUTTON_PIN)) {
                moveVertical(-10);
            }
            if (isButtonPressed(LEFT_BUTTON_PIN)) {
                moveHorizontal(-10);
            }
            if (isButtonPressed(RIGHT_BUTTON_PIN)) {
                moveHorizontal(10);
            }
        }

        if (running && !emergencyStopped) {
            performSequence();
        }
    }

    return 0;
}
```

References

[1] NEMA17 Data Sheet
https://components101.com/sites/default/files/component_datasheet/NEMA17.pdf

[2] ST-PM35-15-11C Stepper Data Sheet
https://cdn.sparkfun.com/datasheets/Robotics/ST-PM35-15-11C.pdf

[3] A4988 Stepper Motor Driver Data Sheet
https://www.alldatasheet.com/datasheet-pdf/pdf/338780/ALLEGRO/A4988.html

[4] Allegro A4988 Stepper Motor Driver:
https://www.allegromicro.com/~/media/Files/Datasheets/A4988-Datasheet.ashx

[5] Gripper Mechanism Reference:
https://grabcad.com/library/3-finger-gripper-2

[6] AVR Microcontroller lectures 2012 :
https://youtube.com/playlist?list=PLD7F7ED1F3505D8D5&feature=shared

[7] Port Manipulation for Atmel AVR MCUs :https://www.linkedin.com/pulse/port-manipulation-atmel-avr-mcus-practical-guide-yamil-garcia-00xte/.

A Log Entries

A.1 26 February - 3 March

A.1.1 Research Phase

- Conducted comprehensive research on existing pick and place robots, with a particular focus on products designed to handle small components such as nuts, washers, and MOSFET transistors. Emphasized high accuracy and efficiency in the research scope to ensure our design meets industry standards.

- Evaluated industry standards and the latest technologies utilized in similar projects. This included studying various automation techniques, gripper mechanisms, and robotic arm configurations to gather valuable insights.

- Decided to design a robot arm based on the gathered insights and specific project requirements, ensuring the design would be feasible and effective in practical applications.

A.2 4 March - 10 March

A.2.1 Conceptualization Phase

- Engaged in extensive brainstorming sessions to generate innovative ideas and methods for the pick and place robot. Encouraged team members to think creatively and consider unconventional solutions.

- Developed multiple conceptual designs, evaluating each based on criteria such as efficiency, scalability, cost, and technical complexity. Created preliminary sketches and diagrams to visualize these concepts.

- Analyzed and ranked each conceptual framework using a weighted scoring system, ultimately selecting the most promising designs for further development based on their potential to meet project goals.

A.3 11 March - 17 March

A.3.1 Design Evaluation Phase

- Conducted detailed evaluations of the selected conceptual designs. This involved assessing technical feasibility, potential scalability, and alignment with overall project objectives. Considered factors such as ease of manufacturing, integration with existing systems, and anticipated performance.

- Developed a structured assessment method to ensure objective evaluation of each design. This included creating a scoring matrix and criteria for judging the designs impartially.

- Selected the best conceptual design based on thorough analysis and consensus among team members, ensuring the chosen design would effectively meet the project's technical and operational requirements.

A.4 18 March - 24 March

A.4.1 Gripper Mechanism Selection

- Analyzed various gripper mechanisms for their efficiency, precision, and compatibility with the overall robot design. Considered different types of grippers, such as pneumatic, hydraulic, and electric grippers, assessing their suitability for handling small components.

- Finalized the most suitable gripper mechanism through collaborative discussions and technical assessments. Evaluated factors like gripping force, speed, and the ability to handle delicate components without causing damage.

- Developed a comprehensive project plan, outlining key milestones, tasks, and timelines. This plan served as a roadmap for the subsequent phases of the project.

- Defined design specifications, including technical requirements and performance benchmarks, to ensure the robot would meet the desired operational standards.

A.5 25 March - 31 March

A.5.1 Design Phase

- Commenced the design phase by creating a prototype of the pick and place robot using SolidWorks software. This included detailed modeling of the robot arm, gripper mechanism, and control systems.

- Visualized and validated key features and functionalities of the conceptual design through the prototype. Conducted simulations and stress tests to ensure the design could withstand operational demands.

- Iteratively refined the prototype based on feedback and feasibility assessments. Made adjustments to improve performance, address limitations, and enhance the overall design based on test results and team input.

A.6 1 April - 7 April

A.6.1 Design Refinement

- Continued iterative refinement of the robot's prototype. Focused on optimizing the design for efficiency, reliability, and ease of manufacturing.

- Incorporated feedback from initial reviews and conducted feasibility assessments to further enhance the design. This involved making minor modifications and adjustments based on practical considerations.

- Ensured the prototype met all specified performance requirements and addressed any identified limitations. Conducted additional testing and validation to confirm the design's effectiveness.

A.7 8 April - 14 April

A.7.1 Design Finalizing

- Finalized the PCB design and the SolidWorks designs. Ensured all components were correctly specified and that the design was ready for manufacturing.

- Gripper Mechanism Reference : `https://grabcad.com/library/3-finger-gripper-2`

A.8 15 April - 21 April

A.8.1 PCB Printing

- Submitted the PCB design for printing to JLCPCB, a renowned PCB manufacturing service. Provided all necessary design files and specifications to ensure accurate production.

- JLCPCB : `https://jlcpcb.com/`

A.9 22 April - 28 April

A.9.1 Final Design Adjustments

- Finalized the detailed design documentation for the robot, including schematics, assembly instructions, and performance specifications.

- Prepared for the upcoming evaluation by thoroughly reviewing all project elements to ensure completeness and accuracy. Ensured all documentation and prototypes were up-to-date and ready for assessment.

- Ensured that all design elements were aligned with stakeholder requirements and project goals. Made any final adjustments needed to meet these criteria.

A.10 29 April - 5 May

A.10.1 Review and Testing

- Created the physical prototype of the robot, assembling all components and integrating the PCB.

- Conducted internal reviews of the design documentation and prototype. Engaged team members in detailed discussions to identify any issues or improvements.

- Engaged in peer review sessions to gather additional feedback and make necessary adjustments. Sought input from colleagues and industry experts to ensure the design was robust and reliable.

- Tested the PCB designed for the robot to ensure it met all operational specifications. Conducted functional tests to verify the performance and reliability of the electronic components.

A.11 6 May - 12 May

A.11.1 Controller Enclosure

- Finalized the design for the controller enclosure using SolidWorks. Ensured the design provided adequate protection for the electronics while allowing easy access for maintenance.

- 3D printed the controller enclosure, verifying its fit and functionality. Made any necessary adjustments to ensure it met all design specifications.

A.12 13 May - 19 May

A.12.1 Final Report Preparation

- Compiled all project documentation, including detailed design reports, test results, and feedback from demonstrations. Ensured the documentation was comprehensive and clearly presented.

- Finalized the project report, ensuring it included detailed accounts of the design process, challenges faced, and solutions implemented. Prepared the report for submission, ensuring it met all required standards and formats.

B Declaration

We declare that the original code did not include any Arduino-type coding or utilize Arduino libraries.

Date	Index	Name with Initials	Signature
07/07/2023	210179R	Gammune D.J.P.	
07/07/2023	210285M	Kavishan G.P	
07/07/2023	210079K	Charles. J	
07/07/2023	210054F	Atapattu A.N. L.R	

C Document Reviews

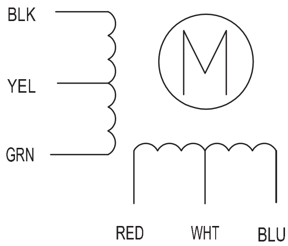
I have thoroughly reviewed the design documentation for the Pick and Place Robot Arm system. The document includes a hierarchical schematic and detailed SolidWorks designs. It is comprehensive, covering all necessary aspects with detailed codes, graphs, pictures, and appendices. I confirm that these comments reflect my unbiased critique of the provided documentation.

Date	Index	Name with Initials	Signature
07.07.2024	210625H	S.A.J.E. Surendra	
07.07.2024	210668P	H.V.P. Vidmal	
07.07.2024	210490L	K.P.K.A. Prabodha	
07.07.2024	210418C	J.N.P. Nayanthara	

HIGH TORQUE HYBRID STEPPING MOTOR SPECIFICATIONS

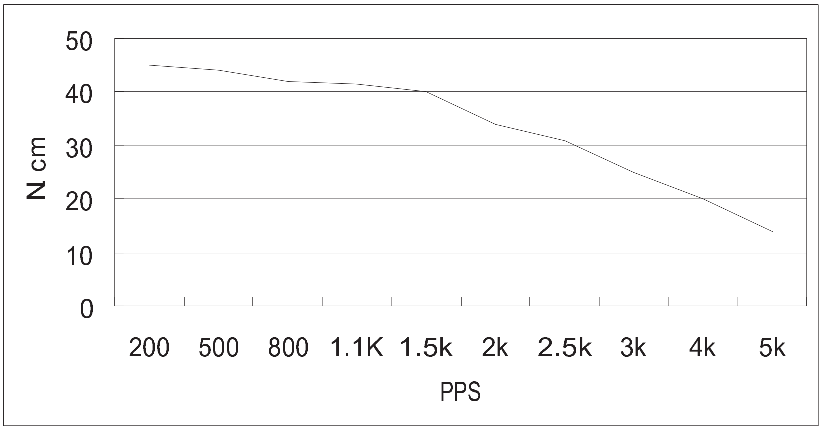
General specifications		Electrical specifications	
Step Angle (°)	1.8	Rated Voltage (V)	4
Temperature Rise (°C)	80 Max (rated current.2 phase on)	Rated Current (A)	1.2
Ambient temperature (°C)	-20～+50	Resistance Per Phase (±10%Ω)	3.3 (25°C)
Number of Phase	2	Inductance Per Phase (±20%mH)	2.8
Insulation Resistance	100MΩ, Min (500VDC)	Holding Torque (Kg.cm)	3.17
Insulation Class	Class B	Detent Torque (g.cm)	200
Max.radial force (N)	28 (20mm from the flange)	Rotor Inertia (g. cm²)	68
Max.axial force (N)	10	Weight (Kg)	0.365

● Wiring Diagram:

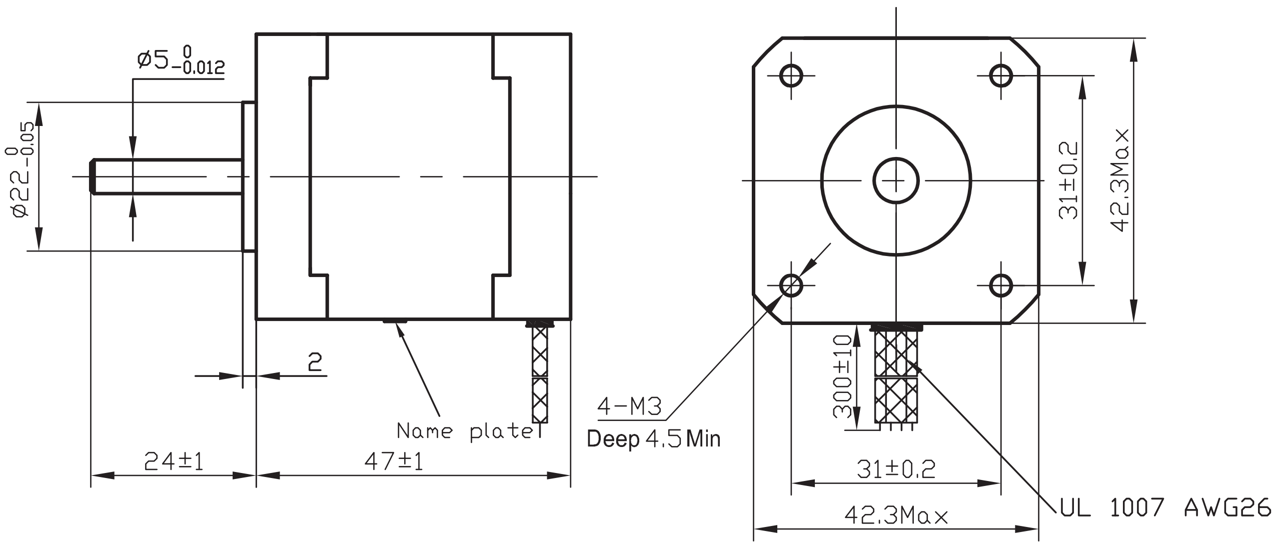


● Pull out torque curve:

VOLTAGE: 24VDC, CONSTANT CURRENT: 1.2A, HALF STEP



● Dimensions:
(unit=mm)



2					SY42STH47-1206A	TECHNICAL CONDITIONS
1						
REV	REVISIONS	DESCRIPTION	BY	DATE		
DRAW	任飞飞 2010.06.29				CHANGZHOU SONGYANG MACHINERY & ELECTRONICS NEW TECHNIC INSTITUTE	060047000
CHECK						
APPROVE						

A4988

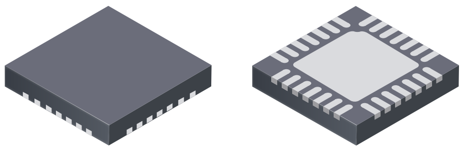
DMOS Microstepping Driver with Translator and Overcurrent Protection

FEATURES AND BENEFITS

- Low $R_{ds(on)}$ outputs
- Automatic current decay mode detection/selection
- Mixed and slow current decay modes
- Synchronous rectification for low power dissipation
- Internal UVLO
- Crossover-current protection
- 3.3 and 5 V compatible logic supply
- Thermal shutdown circuitry
- Short-to-ground protection
- Shorted load protection
- Five selectable step modes: full, $^{1}/_{2}$, $^{1}/_{4}$, $^{1}/_{8}$, and $^{1}/_{16}$

PACKAGE:

28-contact QFN
with exposed thermal pad
5 mm × 5 mm × 0.90 mm
(ET package)



Not to scale

DESCRIPTION

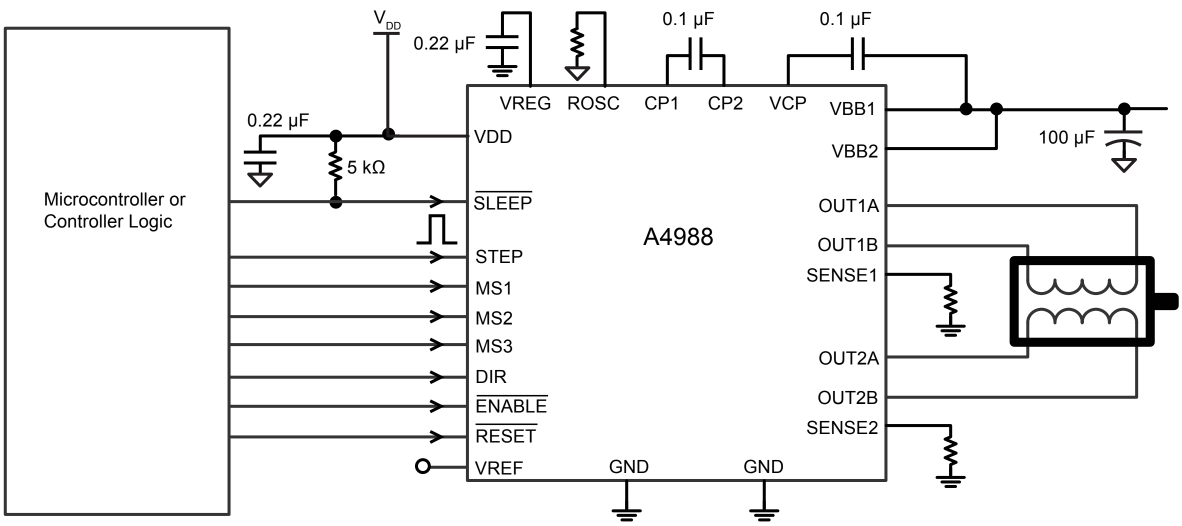
The A4988 is a complete microstepping motor driver with built-in translator for easy operation. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth-, and sixteenth-step modes, with an output drive capacity of up to 35 V and ±2 A. The A4988 includes a fixed off-time current regulator which has the ability to operate in slow or mixed decay modes.

The translator is the key to the easy implementation of the A4988. Simply inputting one pulse on the STEP input drives the motor one microstep. There are no phase sequence tables, high-frequency control lines, or complex interfaces to program. The A4988 interface is an ideal fit for applications where a complex microprocessor is unavailable or is overburdened.

During stepping operation, the chopping control in the A4988 automatically selects the current decay mode: slow or mixed. In mixed decay mode, the device is set initially to a fast decay for a proportion of the fixed off-time, then to a slow decay for the remainder of the off-time. Mixed decay current control results in reduced audible motor noise, increased step accuracy, and reduced power dissipation.

Continued on the next page…

TYPICAL APPLICATION DIAGRAM



4988-DS, Rev. 8
MCO-0000827
April 5, 2022

DESCRIPTION (continued)

Internal synchronous rectification control circuitry is provided to improve power dissipation during PWM operation. Internal circuit protection includes: thermal shutdown with hysteresis, undervoltage lockout (UVLO), and crossover-current protection. Special power-on sequencing is not required.

The A4988 is supplied in a surface-mount QFN package (ET), 5 mm × 5 mm, with a nominal overall package height of 0.90 mm and an exposed pad for enhanced thermal dissipation. It is lead (Pb) free (suffix –T), with 100% matte-tin-plated leadframes.

SELECTION GUIDE

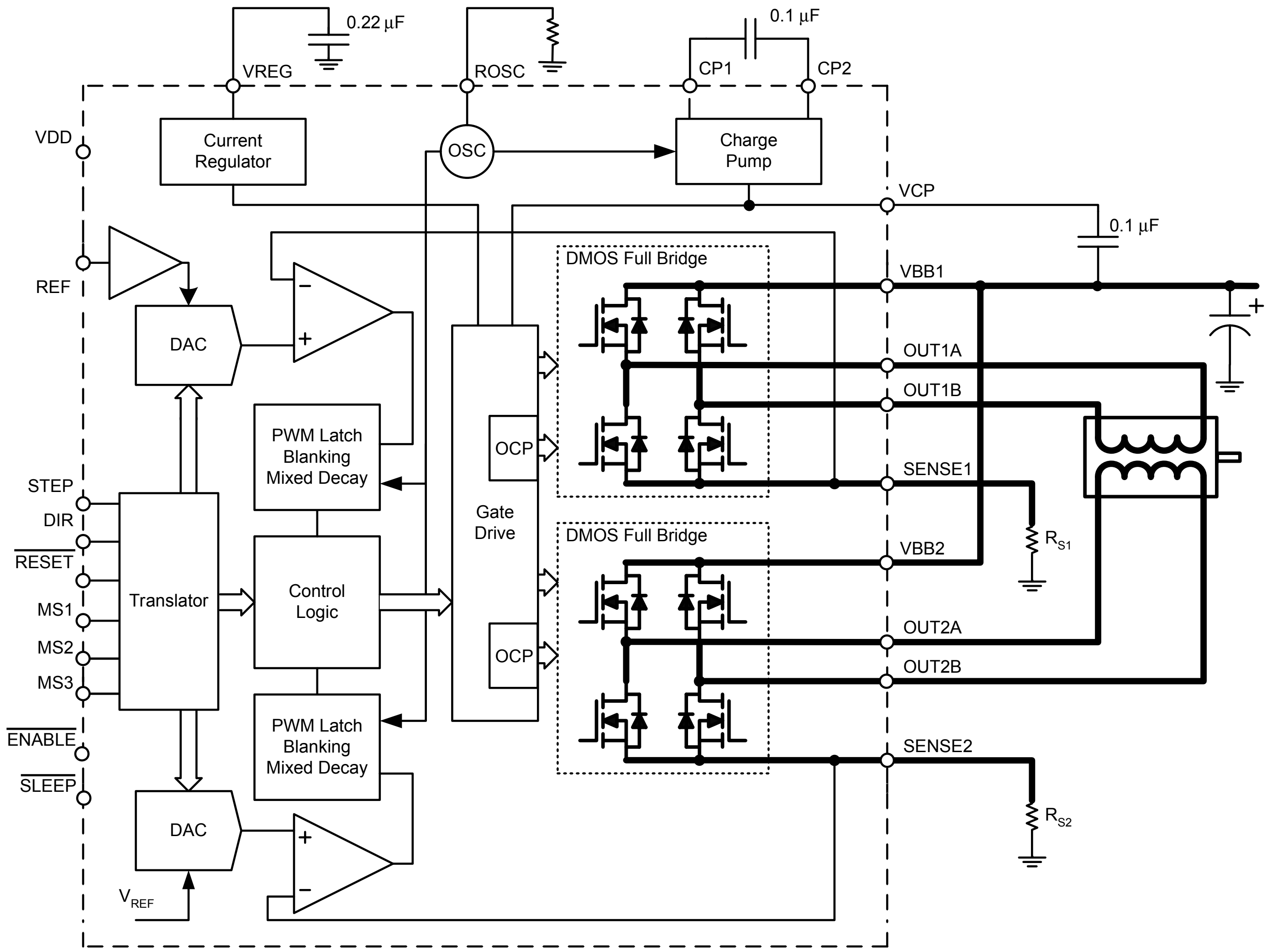
Part Number	Package	Packing
A4988SETTR-T	28-contact QFN with exposed thermal pad	1500 pieces per 7-in. reel

ABSOLUTE MAXIMUM RATINGS

Characteristic	Symbol	Notes	Rating	Units
Load Supply Voltage	V_{BB}		35	V
Output Current	I_{OUT}		±2	A
Logic Input Voltage	V_{IN}		–0.3 to 5.5	V
Logic Supply Voltage	V_{DD}		–0.3 to 5.5	V
Motor Outputs Voltage			–2.0 to 37	V
Sense Voltage	V_{SENSE}		–0.5 to 0.5	V
Reference Voltage	V_{REF}		5.5	V
Operating Ambient Temperature	T_A	Range S	–20 to 85	°C
Maximum Junction	$T_J(max)$		150	°C
Storage Temperature	T_{stg}		–55 to 150	°C

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

Functional Block Diagram



Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

ELECTRICAL CHARACTERISTICS [1]: Valid at T_A = 25°C, V_{BB} = 35 V (unless otherwise noted)

Characteristics	Symbol	Test Conditions	Min.	Typ. [2]	Max.	Units
OUTPUT DRIVERS						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
		During Sleep Mode	0	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3	–	5.5	V
Output On-Resistance	$R_{ds(on)}$	Source driver, I_{OUT} = –1.5 A	–	320	430	mΩ
		Sink driver, I_{OUT} = 1.5 A	–	320	430	mΩ
Body Diode Forward Voltage	V_F	Source diode, I_F = –1.5 A	–	–	1.2	V
		Sink diode, I_F = 1.5 A	–	–	1.2	V
Motor Supply Current	I_{BB}	f_{PWM} < 50 kHz	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
		Sleep Mode	–	–	10	μA
Logic Supply Current	I_{DD}	f_{PWM} < 50 kHz	–	–	8	mA
		Outputs off	–	–	5	mA
		Sleep Mode	–	–	10	μA
CONTROL LOGIC						
Logic Input Voltage	$V_{IN(1)}$		V_{DD} × 0.7	–	–	V
	$V_{IN(0)}$		–	–	V_{DD} × 0.3	V
Logic Input Current	$I_{IN(1)}$	V_{IN} = V_{DD} × 0.7	–20	<1.0	20	μA
	$I_{IN(0)}$	V_{IN} = V_{DD} × 0.3	–20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	kΩ
	R_{MS2}	MS2 pin	–	50	–	kΩ
	R_{MS3}	MS3 pin	–	100	–	kΩ
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = VDD or GND	20	30	40	μs
		R_{OSC} = 25 kΩ	23	30	37	μs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	μA
Current Trip-Level Error [3]	err_I	V_{REF} = 2 V, $\%I_{TripMAX}$ = 38.27%	–	–	±15	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 70.71%	–	–	±5	%
		V_{REF} = 2 V, $\%I_{TripMAX}$ = 100.00%	–	–	±5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
PROTECTION						
Overcurrent Protection Threshold [4]	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	°C
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	°C
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

[1] For input and output current specifications, negative current is defined as coming out of (sourcing) the specified device pin.

[2] Typical data are for initial design estimations only, and assume optimum manufacturing and application conditions. Performance may vary for individual units, within the specified maximum and minimum limits.

[3] $V_{ERR} = [(V_{REF}/8) – V_{SENSE}] / (V_{REF}/8)$.

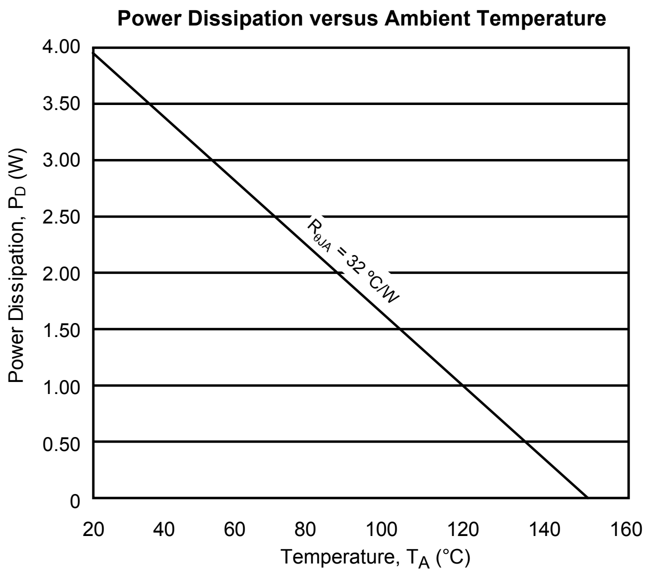
[4] Overcurrent protection (OCP) is tested at T_A = 25°C in a restricted range and guaranteed by characterization.

ALLEGRO microsystems
Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

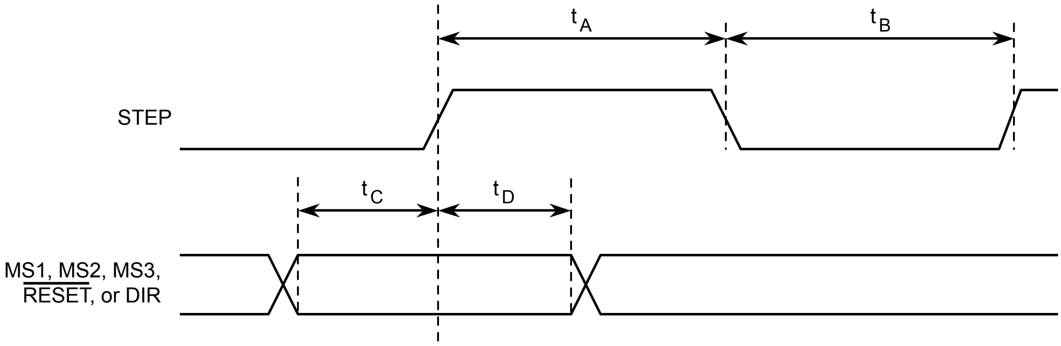
THERMAL CHARACTERISTICS

Characteristic	Symbol	Test Conditions*	Value	Units
Package Thermal Resistance	$R_{\theta JA}$	Four-layer PCB, based on JEDEC standard	32	°C/W

*Additional thermal information available on Allegro website.



Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com



Time Duration	Symbol	Typ.	Unit
STEP minimum, HIGH pulse width	t_A	1	μs
STEP minimum, LOW pulse width	t_B	1	μs
Setup time, input change to STEP	t_C	200	ns
Hold time, input change to STEP	t_D	200	ns

Figure 1: Logic Interface Timing Diagram

Table 1: Microstepping Resolution Truth Table

MS1	MS2	MS3	Microstep Resolution	Excitation Mode
L	L	L	Full Step	2 Phase
H	L	L	Half Step	1-2 Phase
L	H	L	Quarter Step	W1-2 Phase
H	H	L	Eighth Step	2W1-2 Phase
H	H	H	Sixteenth Step	4W1-2 Phase

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

FUNCTIONAL DESCRIPTION

Device Operation. The A4988 is a complete microstepping motor driver with a built-in translator for easy operation with minimal control lines. It is designed to operate bipolar stepper motors in full-, half-, quarter-, eighth, and sixteenth-step modes. The currents in each of the two output full-bridges and all of the N-channel DMOS FETs are regulated with fixed off-time PWM (pulse-width modulated) control circuitry. At each step, the current for each full-bridge is set by the value of its external current-sense resistor (R_{S1} and R_{S2}), a reference voltage (V_{REF}), and the output voltage of its DAC (which in turn is controlled by the output of the translator).

At power-on or reset, the translator sets the DACs and the phase current polarity to the initial Home state (shown in Figures 9 through 13), and the current regulator to Mixed decay mode for both phases. When a step command signal occurs on the STEP input, the translator automatically sequences the DACs to the next level and current polarity. (See Table 2 for the current-level sequence.) The microstep resolution is set by the combined effect of the MSx inputs, as shown in Table 1.

When stepping, if the new output levels of the DACs are lower than their previous output levels, then the decay mode for the active full-bridge is set to Mixed. If the new output levels of the DACs are higher than or equal to their previous levels, then the decay mode for the active full-bridge is set to Slow. This automatic current decay selection improves microstepping performance by reducing the distortion of the current waveform that results from the back EMF of the motor.

Microstep Select (MSx). The microstep resolution is set by the voltage on logic inputs MSx, as shown in Table 1. The MS1 and MS3 pins have a 100 kΩ pull-down resistance, and the MS2 pin has a 50 kΩ pull-down resistance. When changing the step mode, the change does not take effect until the next STEP rising edge.

If the step mode is changed without a translator reset, and absolute position must be maintained, it is important to change the step mode at a step position that is common to both step modes in order to avoid missing steps. When the device is powered down, or reset due to TSD or an overcurrent event, the translator is set to the home position which is by default common to all step modes.

Mixed Decay Operation. The bridge operates in Mixed decay mode, at power-on and reset, and during normal running according to the ROSC configuration and the step sequence, as shown in Figures 9 through 13. During Mixed decay mode, when the trip point is reached, the A4988 initially goes into a fast decay interval for 31.25% of the off-time, t_{OFF}. After that, it switches to slow decay for the remainder of t_{OFF}. A timing diagram for this feature appears on the next page.

Typically, mixed decay is only necessary when the current in the winding is going from a higher value to a lower value as determined by the state of the translator. For most loads, automatically selected mixed decay is convenient because it minimizes ripple when the current is rising and prevents missed steps when the current is falling. For some applications where microstepping at very low speeds is necessary, the lack of back EMF in the winding causes the current to increase in the load quickly, resulting in missed steps. This is shown in Figure 2. By pulling the ROSC pin to ground, mixed decay is set to be active 100% of the time, for both rising and falling currents, and prevents missed steps as shown in Figure 3. If this is not an issue, it is recommended that automatically selected mixed decay be used, because it will produce reduced ripple currents. Refer to the Fixed Off-Time section for details.

Low Current Microstepping. Intended for applications where the minimum on-time prevents the output current from regulating to the programmed current level at low current steps. To prevent this, the device can be set to operate in Mixed decay mode on both rising and falling portions of the current waveform. This feature is implemented by shorting the ROSC pin to ground. In this state, the off-time is internally set to 30 µs.

Reset Input ($\overline{\text{RESET}}$). The $\overline{\text{RESET}}$ input sets the translator to a predefined Home state (shown in Figures 9 through 13), and turns off all of the FET outputs. All STEP inputs are ignored until the $\overline{\text{RESET}}$ input is set to high.

Step Input (STEP). A low-to-high transition on the STEP input sequences the translator and advances the motor one increment. The translator controls the input to the DACs and the direc-

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

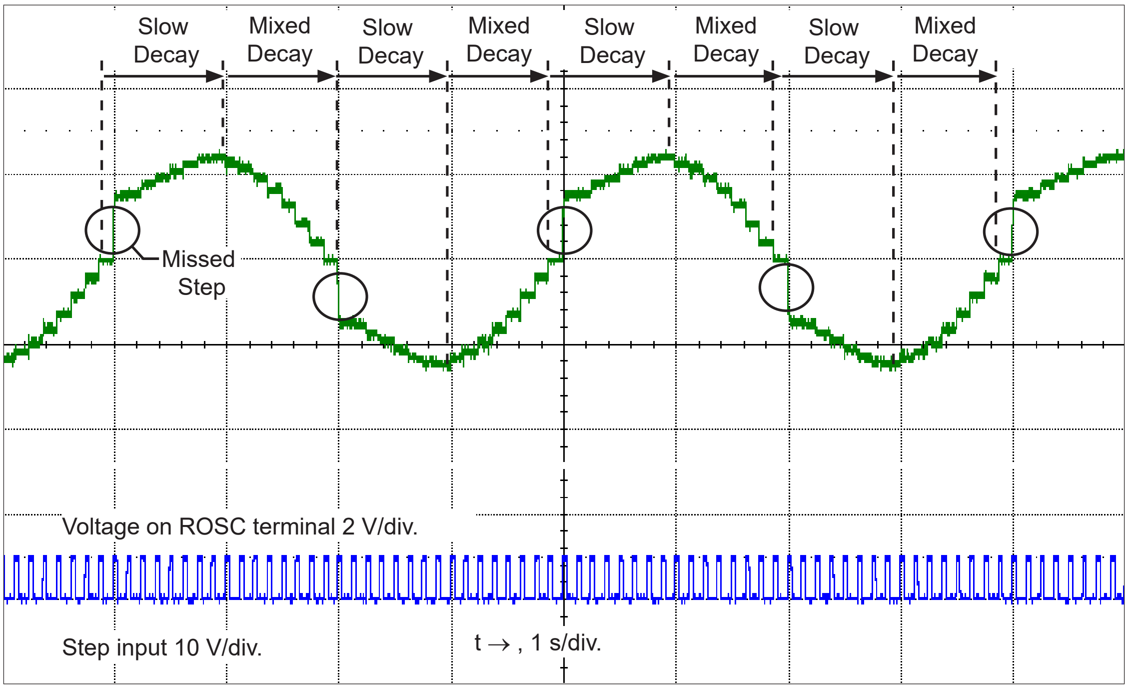


Figure 2: Missed Steps in Low-Speed Microstepping

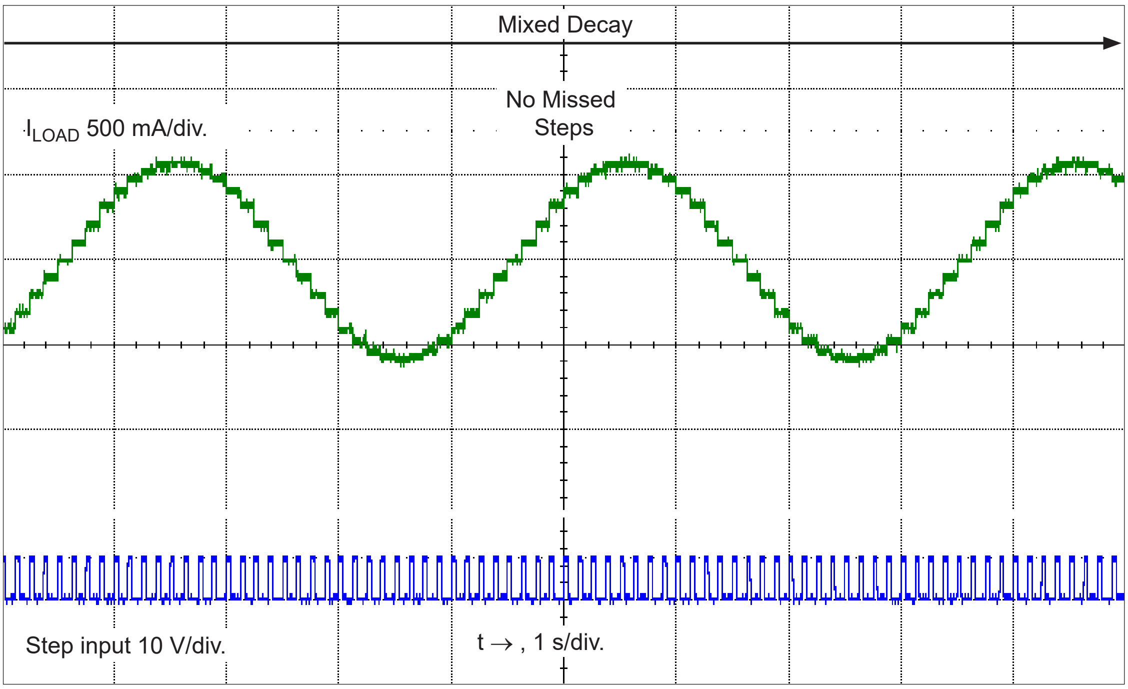


Figure 3: Continuous Stepping Using Automatically-Selected Mixed Stepping (ROSC pin grounded)

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

tion of current flow in each winding. The size of the increment is determined by the combined state of the MSx inputs.

Direction Input (DIR). This determines the direction of rotation of the motor. Changes to this input do not take effect until the next STEP rising edge.

Internal PWM Current Control. Each full-bridge is controlled by a fixed off-time PWM current control circuit that limits the load current to a desired value, I_{TRIP}. Initially, a diagonal pair of source and sink FET outputs are enabled and current flows through the motor winding and the current sense resistor, R_{Sx}. When the voltage across R_{Sx} equals the DAC output voltage, the current sense comparator resets the PWM latch. The latch then turns off the appropriate source driver and initiates a fixed off-time decay mode

The maximum value of current limiting is set by the selection of R_{Sx} and the voltage at the VREF pin. The transconductance function is approximated by the maximum value of current limiting, $I_{TripMAX}$ (A), which is set by

$$I_{TripMAX} = V_{REF} / (8 \times R_S)$$

where R_S is the resistance of the sense resistor (Ω) and V_{REF} is the input voltage on the REF pin (V).

The DAC output reduces the V_{REF} output to the current sense comparator in precise steps, such that

$$I_{trip} = (\%I_{TripMAX} / 100) \times I_{TripMAX}$$

(See Table 2 for $\%I_{TripMAX}$ at each step.)

It is critical that the maximum rating (0.5 V) on the SENSE1 and SENSE2 pins is not exceeded.

Fixed Off-Time. The internal PWM current control circuitry uses a one-shot circuit to control the duration of time that the DMOS FETs remain off. The off-time, t_{OFF}, is determined by the ROSC terminal. The ROSC terminal has three settings:

- ROSC tied to VDD — off-time internally set to 30 µs; decay mode is automatic Mixed, except when in full-step where decay mode is set to Slow.
- ROSC tied directly to ground — off-time internally set to 30 µs; current decay is set to Mixed for both increasing and decreasing currents for all step modes.
- ROSC through a resistor to ground — off-time is determined by the following formula; the decay mode is automatic Mixed for all step modes except full-step which is set to Slow.

$$t_{OFF} \approx R_{OSC} / 825$$

where t_{OFF} is in µs.

Blanking. This function blanks the output of the current sense comparators when the outputs are switched by the internal current control circuitry. The comparator outputs are blanked to prevent false overcurrent detection due to reverse recovery currents of the clamp diodes, and switching transients related to the capacitance of the load. The blank time, t_{BLANK} (µs), is approximately

$$t_{BLANK} \approx 1\ \mu s$$

Shorted Load and Short-to-Ground Protection. If the motor leads are shorted together, or if one of the leads is shorted to ground, the driver will protect itself by sensing the overcurrent event and disabling the driver that is shorted, protecting the device from damage. In the case of a short-to-ground, the device will remain disabled (latched) until the $\overline{\text{SLEEP}}$ input goes high or V_{DD} power is removed. A short-to-ground overcurrent event is shown in Figure 4.

When the two outputs are shorted together, the current path is through the sense resistor. After the blanking time (≈1 µs) expires, the sense resistor voltage is exceeding its trip value, due to the overcurrent condition that exists. This causes the driver to go into a fixed off-time cycle. After the fixed off-time expires, the driver turns on again and the process repeats. In this condition, the driver is completely protected against overcurrent events, but the short is repetitive with a period equal to the fixed off-time of the driver. This condition is shown in Figure 5.

During a shorted load event, it is normal to observe both a positive and negative current spike as shown in Figure 3, due to the direction change implemented by the Mixed decay feature. This is shown in Figure 6. In both instances, the overcurrent circuitry is protecting the driver and prevents damage to the device.

Charge Pump (CP1 and CP2). The charge pump is used to generate a gate supply greater than that of V_{BB} for driving the source-side FET gates. A 0.1 µF ceramic capacitor should be connected between CP1 and CP2. In addition, a 0.1 µF ceramic capacitor is required between VCP and VBB, to act as a reservoir for operating the high-side FET gates.

Capacitor values should be Class 2 dielectric ±15% maximum, or tolerance R, according to EIA (Electronic Industries Alliance) specifications.

V_{REG} (VREG). This internally generated voltage is used to operate the sink-side FET outputs. The nominal output voltage of the VREG terminal is 7 V. The VREG pin must be decoupled with a 0.22 µF ceramic capacitor to ground. V_{REG} is internally monitored. In the case of a fault condition, the FET outputs of the A4988 are disabled.

Capacitor values should be Class 2 dielectric ±15% maximum, or tolerance R, according to EIA (Electronic Industries Alliance) specifications.

Enable Input ($\overline{\text{ENABLE}}$). This input turns on or off all of the FET outputs. When set to a logic high, the outputs are disabled. When set to a logic low, the internal control enables the outputs as required. The translator inputs STEP, DIR, and MSx, as well as the internal sequencing logic, all remain active, independent of the $\overline{\text{ENABLE}}$ input state.

Shutdown. In the event of a fault, overtemperature (excess T_J) or an undervoltage (on VCP), the FET outputs of the A4988 are disabled until the fault condition is removed. At power-on, the UVLO (undervoltage lockout) circuit disables the FET outputs and resets the translator to the Home state.

Sleep Mode ($\overline{\text{SLEEP}}$). To minimize power consumption when the motor is not in use, this input disables much of the internal circuitry including the output FETs, current regulator, and charge pump. A logic low on the $\overline{\text{SLEEP}}$ pin puts the A4988 into Sleep mode. A logic high allows normal operation, as well as startup (at which time the A4988 drives the motor to the Home microstep position). When emerging from Sleep mode, in order to allow the charge pump to stabilize, provide a delay of 1 ms before issuing a Step command.

Mixed Decay Operation. The bridge operates in Mixed decay mode, depending on the step sequence, as shown in Figures 9 through 13. As the trip point is reached, the A4988 initially goes into a fast decay interval for 31.25% of the off-time, t_{OFF}. After that, it switches to slow decay for the remainder of t_{OFF}. A timing diagram for this feature appears in Figure 7.

Synchronous Rectification. When a PWM-off cycle is triggered by an internal fixed-off time cycle, load current recirculates according to the decay mode selected by the control logic. This synchronous rectification feature turns on the appropriate FETs during current decay, and effectively shorts out the body diodes with the low FET $R_{ds(on)}$. This reduces power dissipation significantly, and can eliminate the need for external Schottky diodes in many applications. Synchronous rectification turns off when the load current approaches zero (0 A), preventing reversal of the load current.

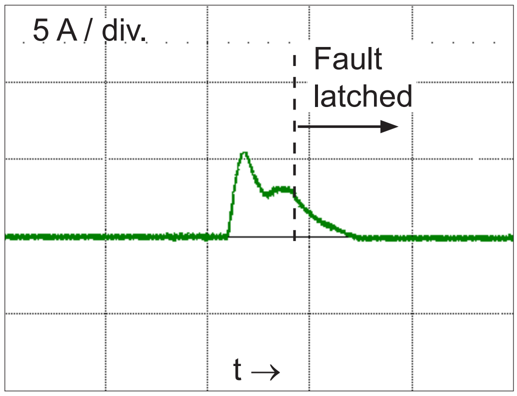


Figure 4: Short-to-Ground Event

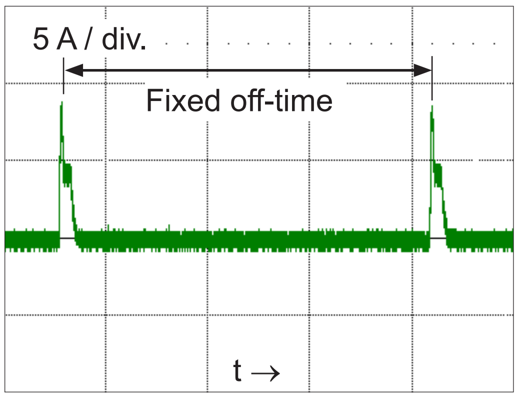


Figure 5. Shorted Load (OUTxA → OUTxB) in Slow Decay Mode

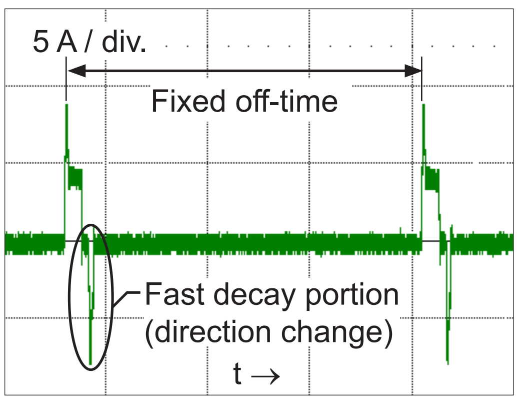
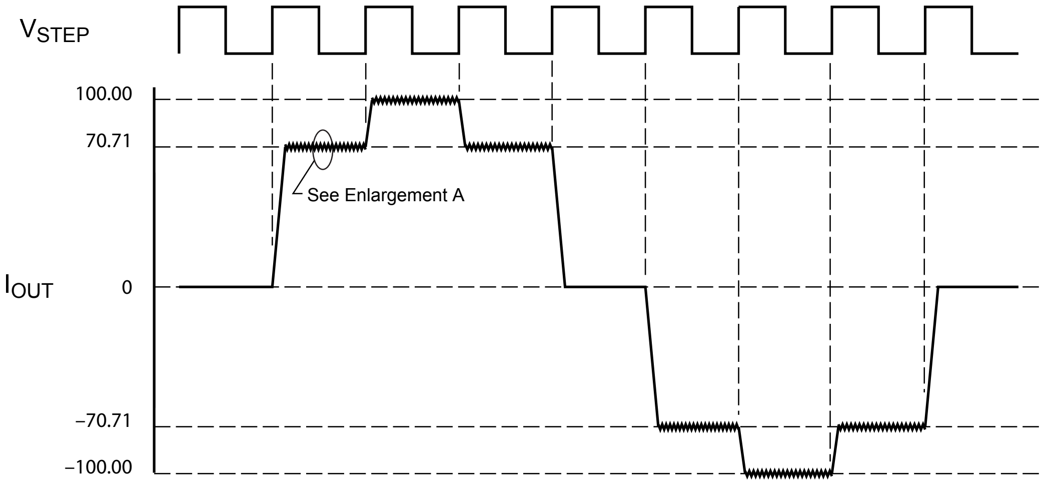
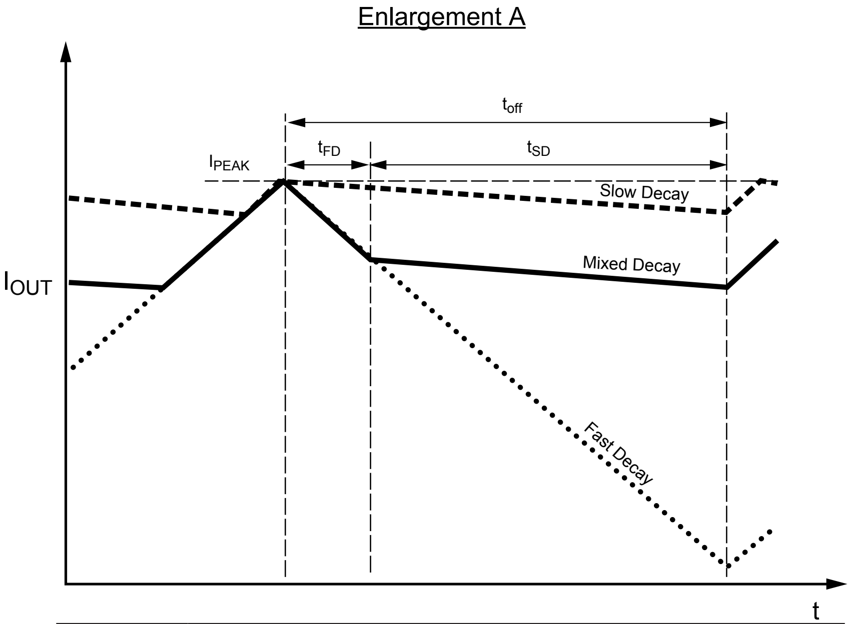


Figure 6: Shorted Load (OUTxA → OUTxB) in Mixed Decay Mode

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com





Symbol	Characteristic
t_{off}	Device fixed off-time
I_{PEAK}	Maximum output current
t_{SD}	Slow decay interval
t_{FD}	Fast decay interval
I_{OUT}	Device output current

Figure 7: Current Decay Modes Timing Chart

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

APPLICATION LAYOUT

Layout. The printed circuit board should use a heavy groundplane. For optimum electrical and thermal performance, the A4988 must be soldered directly onto the board. Pins 3 and 18 are internally fused, which provides a path for enhanced thermal dissipation. Theses pins should be soldered directly to an exposed surface on the PCB that connects to thermal vias are used to transfer heat to other layers of the PCB.

In order to minimize the effects of ground bounce and offset issues, it is important to have a low-impedance single-point ground, known as a *star ground*, located very close to the device. By making the connection between the pad and the ground plane directly under the A4988, that area becomes an ideal location for a star ground point. A low-impedance ground will prevent ground bounce during high-current operation and ensure that the supply voltage remains stable at the input terminal.

The two input capacitors should be placed in parallel, and as close to the device supply pins as possible. The ceramic capacitor (CIN1) should be closer to the pins than the bulk capacitor (CIN2). This is necessary because the ceramic capacitor will be responsible for delivering the high-frequency current components. The sense resistors, RSx, should have a very low-impedance path to ground, because they must carry a large current while supporting very accurate voltage measurements by the current sense comparators. Long ground traces will cause additional voltage drops, adversely affecting the ability of the comparators to accurately measure the current in the windings. The SENSEx pins have very short traces to the RSx resistors and very thick, low-impedance traces directly to the star ground under the device. If possible, there should be no other components on the sense circuits.

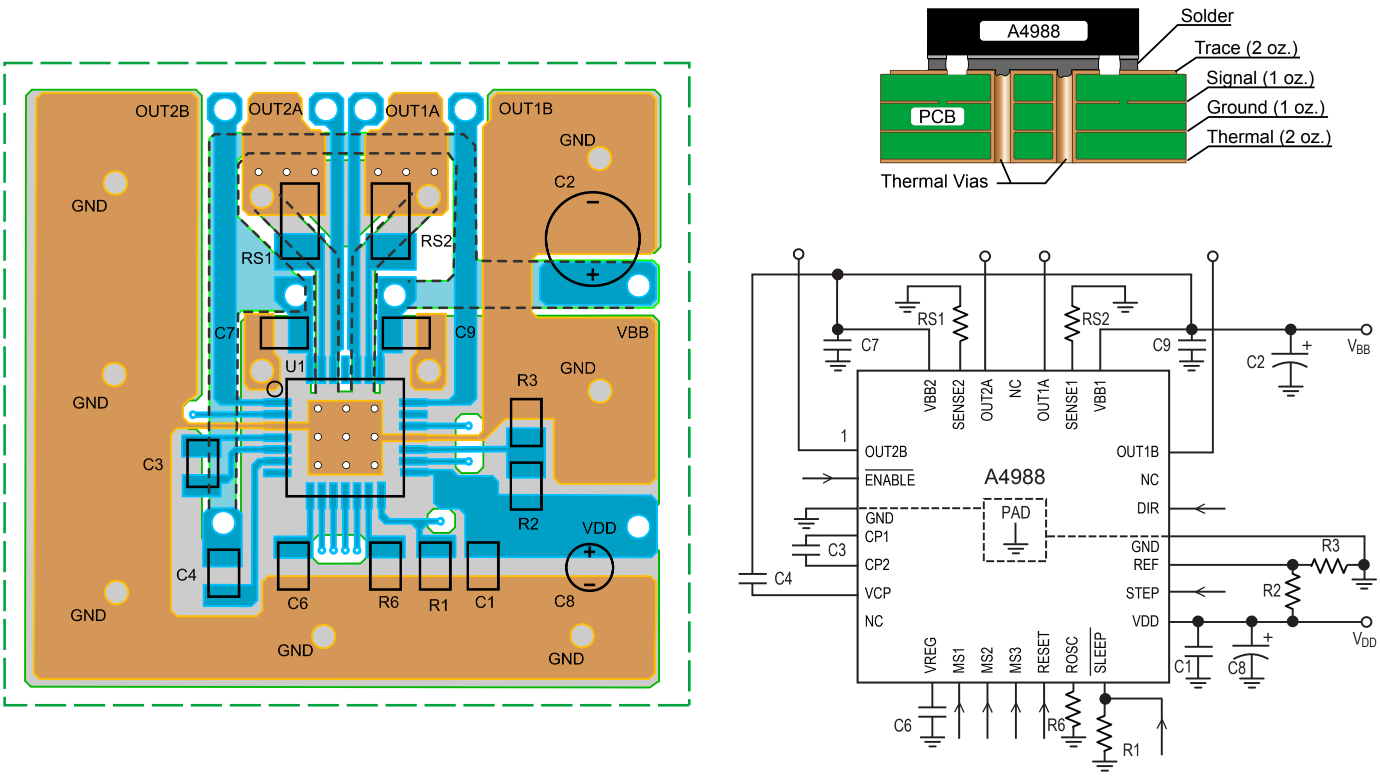
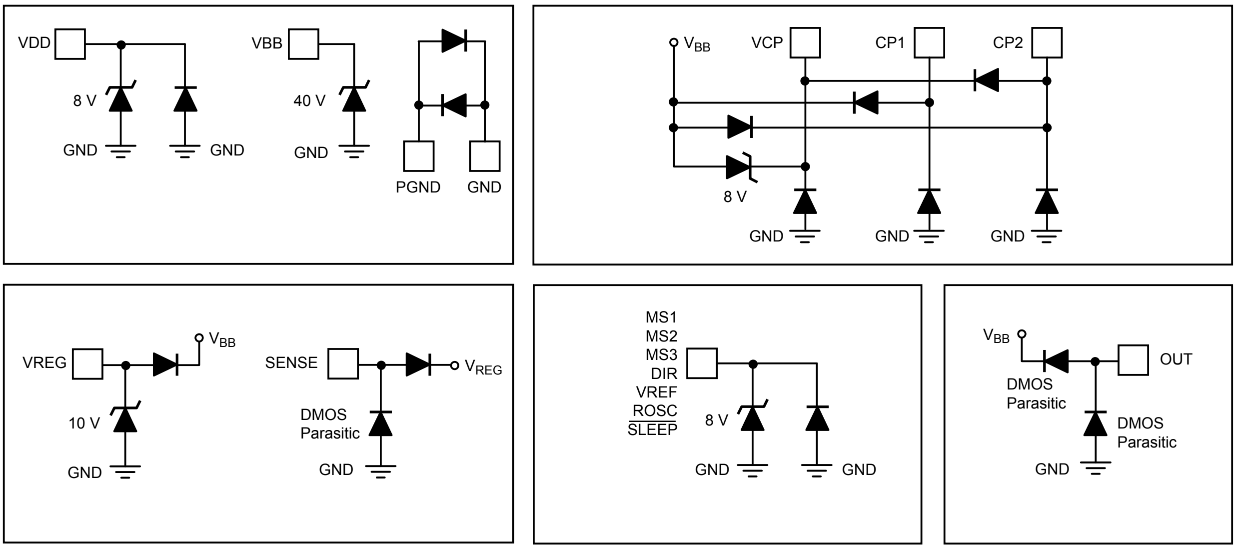


Figure 8: Typical Application and Circuit Layout

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

PIN CIRCUIT DIAGRAMS



Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

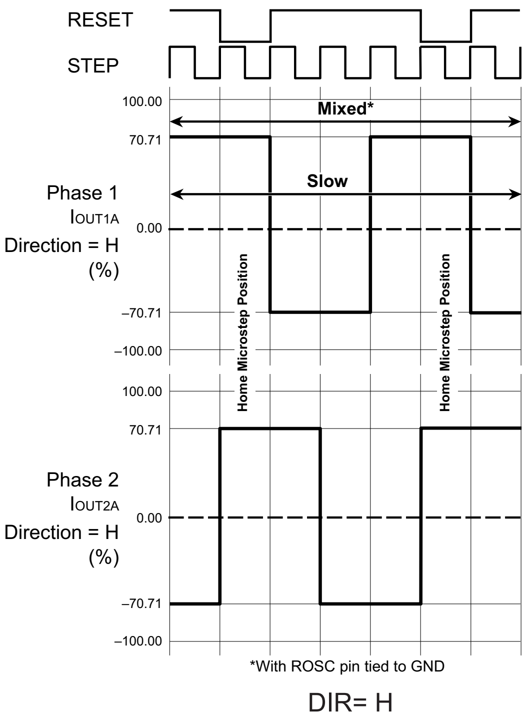


Figure 9: Decay Mode for Full-Step Increments

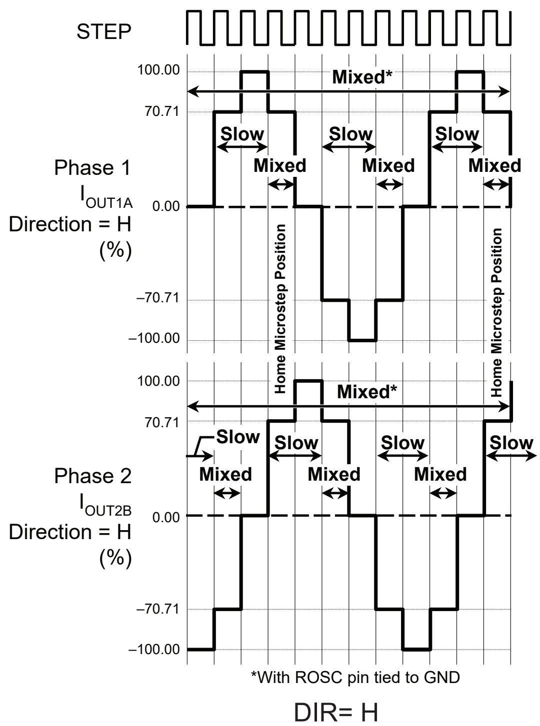


Figure 10: Decay Modes for Half-Step Increments

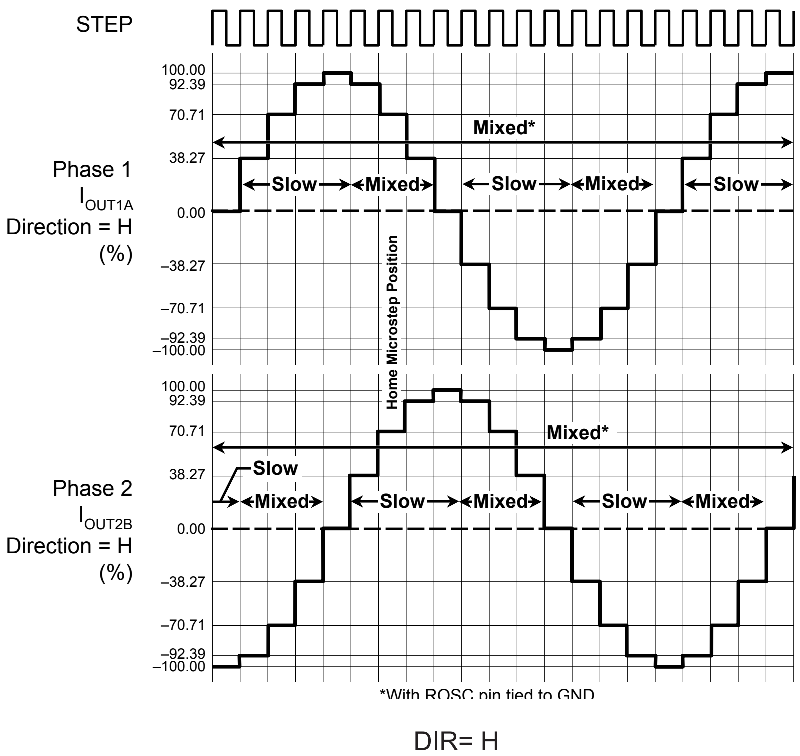


Figure 11: Decay Modes for Quarter-Step Increments

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

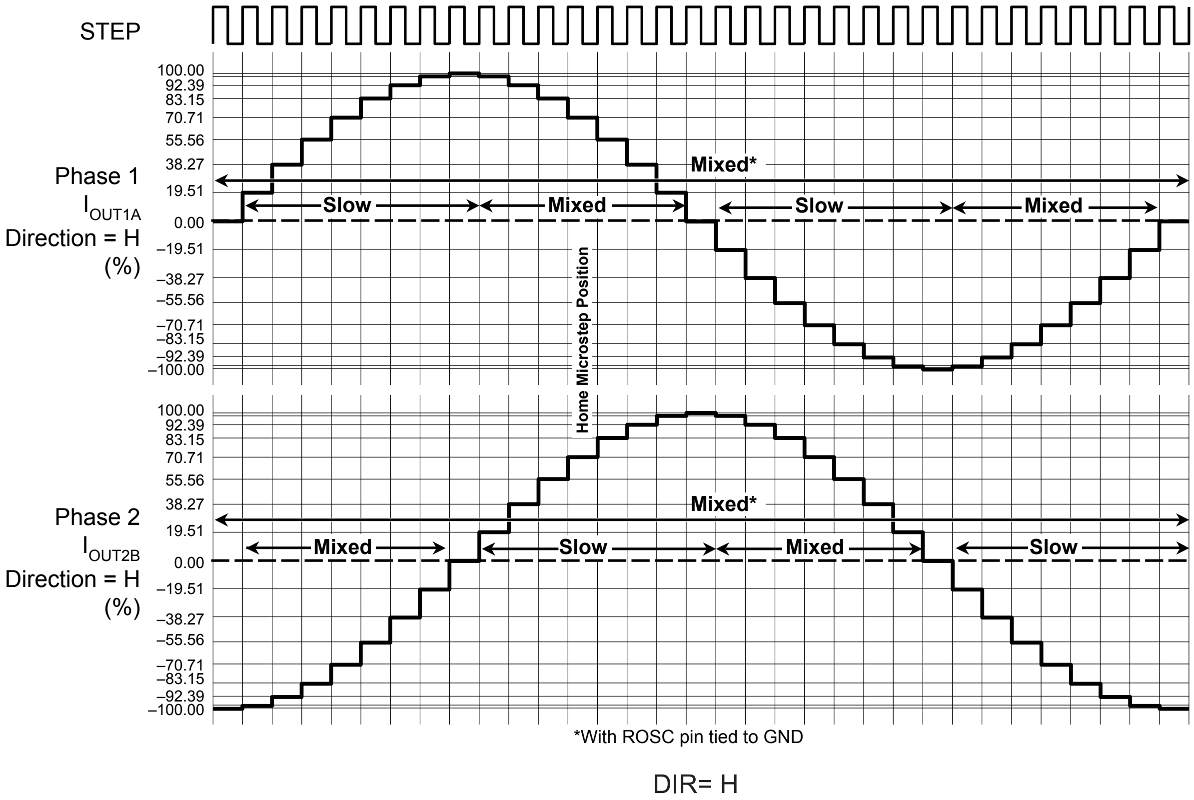
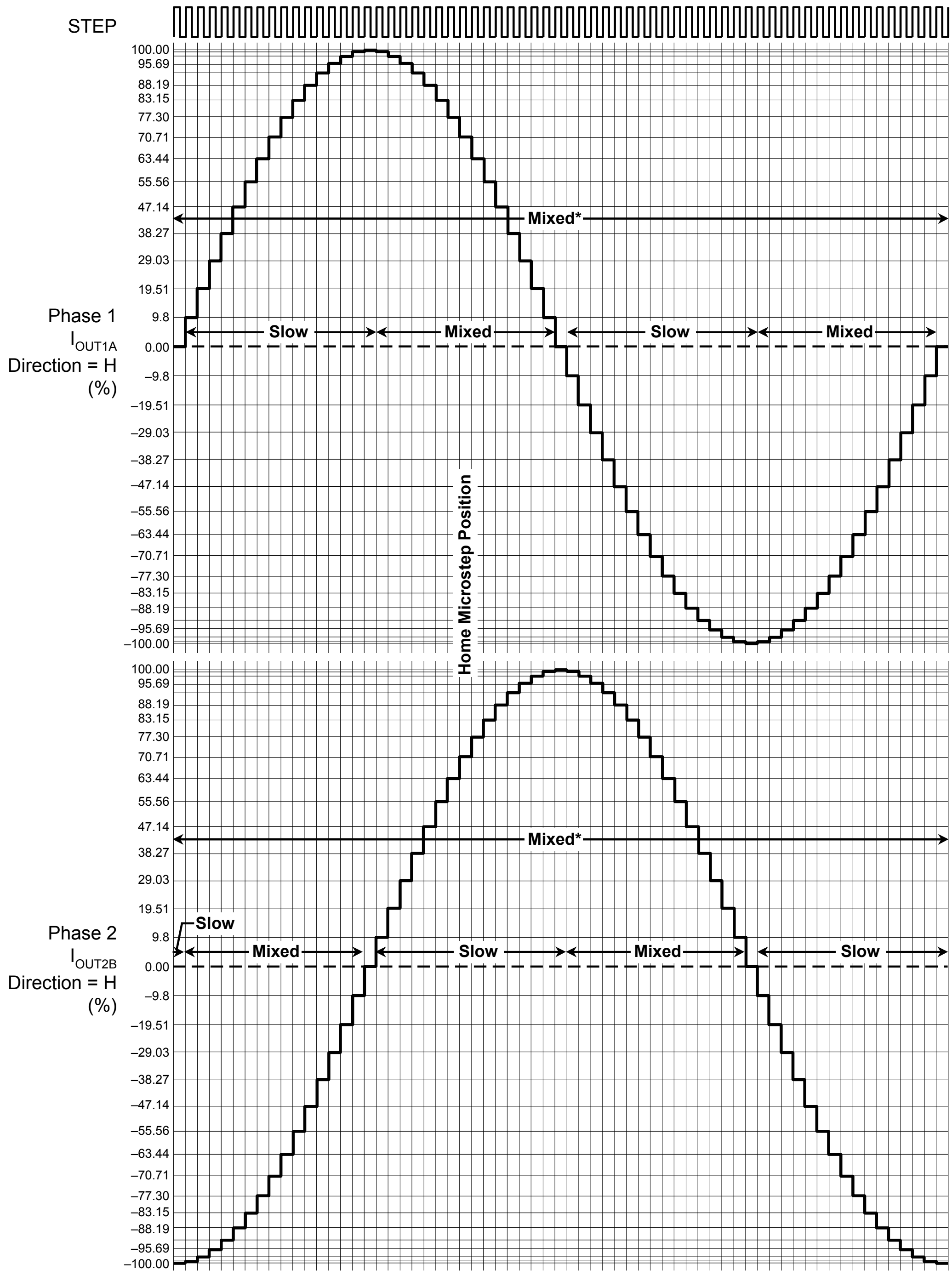


Figure 12: Decay Modes for Eighth-Step Increments



*With ROSC pin tied to GND
DIR=H

Figure 13: Decay Modes for Sixteenth-Step Increments

ALLEGRO microsystems

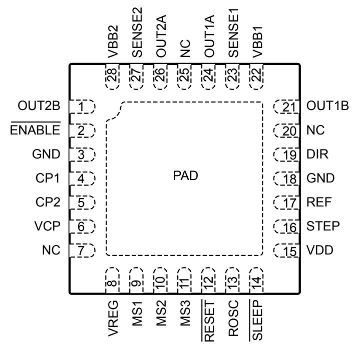
Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

Table 2: Step Sequencing Settings
Home microstep position at Step Angle 45°; DIR = H

Full Step #	Half Step #	1/4 Step #	1/8 Step #	1/16 Step #	Phase 1 Current [% $I_{tripMax}$] (%)	Phase 2 Current [% $I_{tripMax}$] (%)	Step Angle (°)
	1	1	1	1	100.00	0.00	0.0
				2	99.52	9.80	5.6
			2	3	98.08	19.51	11.3
				4	95.69	29.03	16.9
		2	3	5	92.39	38.27	22.5
				6	88.19	47.14	28.1
			4	7	83.15	55.56	33.8
				8	77.30	63.44	39.4
1	2	3	5	9	70.71	70.71	45.0
				10	63.44	77.30	50.6
			6	11	55.56	83.15	56.3
				12	47.14	88.19	61.9
		4	7	13	38.27	92.39	67.5
				14	29.03	95.69	73.1
			8	15	19.51	98.08	78.8
				16	9.80	99.52	84.4
	3	5	9	17	0.00	100.00	90.0
				18	–9.80	99.52	95.6
			10	19	–19.51	98.08	101.3
				20	–29.03	95.69	106.9
		6	11	21	–38.27	92.39	112.5
				22	–47.14	88.19	118.1
			12	23	–55.56	83.15	123.8
				24	–63.44	77.30	129.4
2	4	7	13	25	–70.71	70.71	135.0
				26	–77.30	63.44	140.6
			14	27	–83.15	55.56	146.3
				28	–88.19	47.14	151.9
		8	15	29	–92.39	38.27	157.5
				30	–95.69	29.03	163.1
			16	31	–98.08	19.51	168.8
				32	–99.52	9.80	174.4
	5	9	17	33	–100.00	0.00	180.0
				34	–99.52	–9.80	185.6
			18	35	–98.08	–19.51	191.3
				36	–95.69	–29.03	196.9
		10	19	37	–92.39	–38.27	202.5
				38	–88.19	–47.14	208.1
			20	39	–83.15	–55.56	213.8
				40	–77.30	–63.44	219.4
3	6	11	21	41	–70.71	–70.71	225.0
				42	–63.44	–77.30	230.6
			22	43	–55.56	–83.15	236.3
				44	–47.14	–88.19	241.9
		12	23	45	–38.27	–92.39	247.5
				46	–29.03	–95.69	253.1
			24	47	–19.51	–98.08	258.8
				48	–9.80	–99.52	264.4
	7	13	25	49	0.00	–100.00	270.0
				50	9.80	–99.52	275.6
			26	51	19.51	–98.08	281.3
				52	29.03	–95.69	286.9
		14	27	53	38.27	–92.39	292.5
				54	47.14	–88.19	298.1
			28	55	55.56	–83.15	303.8
				56	63.44	–77.30	309.4
4	8	15	29	57	70.71	–70.71	315.0
				58	77.30	–63.44	320.6
			30	59	83.15	–55.56	326.3
				60	88.19	–47.14	331.9
		16	31	61	92.39	–38.27	337.5
				62	95.69	–29.03	343.1
			32	63	98.08	–19.51	348.8
				64	99.52	–9.80	354.4

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

Pinout Diagram



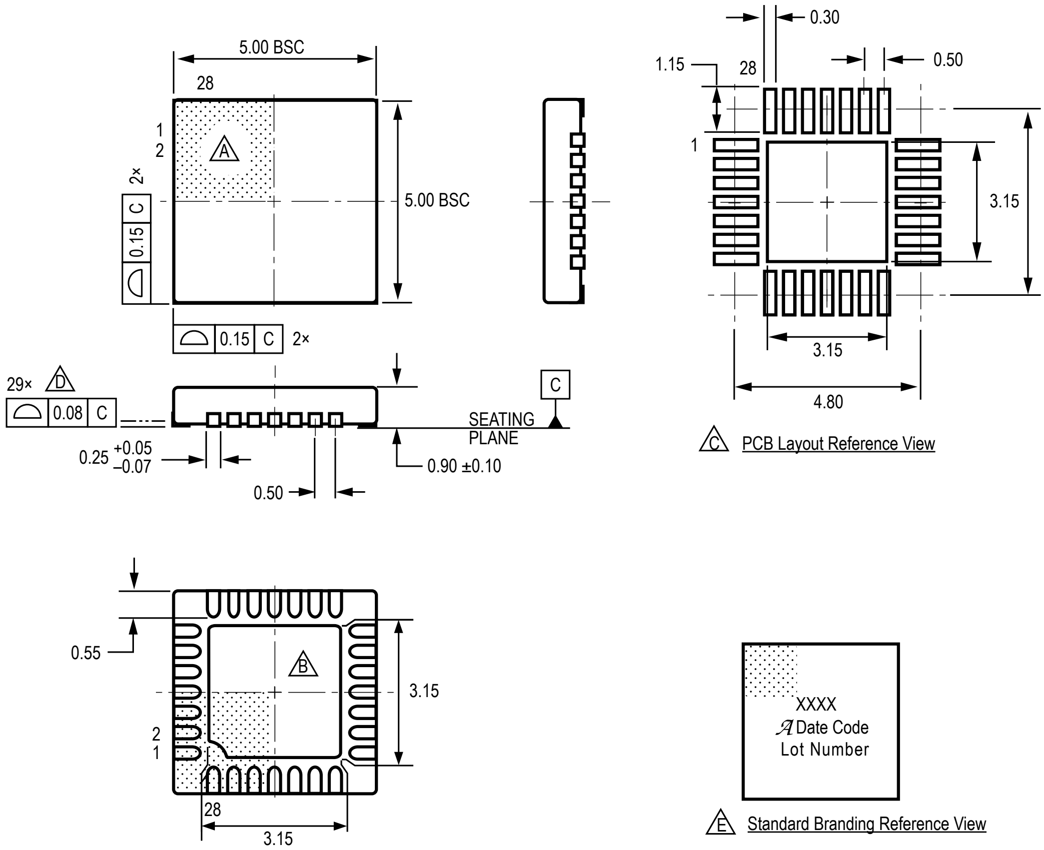
Terminal List Table

Name	Number	Description
CP1	4	Charge pump capacitor terminal
CP2	5	Charge pump capacitor terminal
VCP	6	Reservoir capacitor terminal
VREG	8	Regulator decoupling terminal
MS1	9	Logic input
MS2	10	Logic input
MS3	11	Logic input
$\overline{\text{RESET}}$	12	Logic input
ROSC	13	Timing set
$\overline{\text{SLEEP}}$	14	Logic input
VDD	15	Logic supply
STEP	16	Logic input
REF	17	G_m reference voltage input
GND	3, 18	Ground*
DIR	19	Logic input
OUT1B	21	DMOS Full Bridge 1 Output B
VBB1	22	Load supply
SENSE1	23	Sense resistor terminal for Bridge 1
OUT1A	24	DMOS Full Bridge 1 Output A
OUT2A	26	DMOS Full Bridge 2 Output A
SENSE2	27	Sense resistor terminal for Bridge 2
VBB2	28	Load supply
OUT2B	1	DMOS Full Bridge 2 Output B
$\overline{\text{ENABLE}}$	2	Logic input
NC	7, 20, 25	No connection
PAD	–	Exposed pad for enhanced thermal dissipation*

*The GND pins must be tied together externally by connecting to the PAD ground plane under the device.

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

ET Package, 28-Pin QFN with Exposed Thermal Pad



Line 1: Part Number
Line 2: Logo A, 4-Digit Date Code
Line 3: Characters 5, 6, 7, 8 of Assembly Lot Number

For Reference Only; not for tooling use
(reference DWG-0000378, Rev. 3)
Dimensions in millimeters
Exact case and lead configuration at supplier discretion within limits shown

A Terminal #1 mark area

B Exposed thermal pad (reference only, terminal #1 identifier appearance at supplier discretion)

C Reference land pattern layout (reference IPC7351 QFN50P500X500X100-29V1M);
All pads a minimum of 0.20 mm from all adjacent pads; adjust as necessary to meet application process requirements and PCB layout tolerances; when mounting on a multilayer PCB, thermal vias at the exposed thermal pad land can improve thermal dissipation (reference EIA/JEDEC Standard JESD51-5)

D Coplanarity includes exposed thermal pad and terminals

E Branding scale and appearance at supplier discretion

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com

Revision History

Number	Date	Description
4	January 27, 2012	Updated I_{OCPST}
5	May 7, 2014	Revised text on page 9; revised Figure 8 and Table 2
6	January 14, 2016	Updated VBB, IBB, and IDD in Electrical Characteristics table
7	April 9, 2020	Minor editorial updates
8	April 5, 2022	Updated package drawing (page 19)

For the latest version of this document, visit our website:
www.allegromicro.com

Allegro MicroSystems
955 Perimeter Road
Manchester, NH 03103-3353 U.S.A.
www.allegromicro.com