11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

Midterm Report

11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

Concept of Operations

REVISION 0 – Rough Draft/1st Final Draft
12 September 2024

Concept of Operations
for
11: Multi-Axis Cobot For Factory Automation

Team <42>

Approved by:

____
Project Leader Date

____
Prof. Kalafatis Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/12/2024	All Members	All Members	Draft Release - Revision 0
1	9/26/2024	All Members	All Members	Changed MCU selection in Section 3.6

Table of Contents

List of Tables

List of Figures

1. Executive Summary

In many manufacturing industries, safety and efficiency are critical to ensure a smooth operation within a fabrication facility (fabs) or working environment. Fabs have heavy machinery and moving parts, making it difficult for humans to access certain places. To aid the workforce, collaborative robots (cobots) are deployed within factories allowing human control from safe distances. Cobots range in size, and application, and help streamline processes within manufacturing. This solution will protect human lives and reduce inefficiencies in labor by eliminating the process of entering a hostile or hazardous work environment.

This project aims to develop a multi-axis collaborative robot (cobot) that replaces manual human intervention with remote control, reducing the risk of injury and fatigue while improving efficiency in factory settings. The cobot will assist with light tasks such as lifting small loads (1-2 lbs), transporting items between locations, and providing precise movement control through a wireless app. Key software tools for the design and development include Altium, CCStudio, AutoCAD, and VSCode (with Swift). The cobot's hardware will feature a C2000x MCU and multiple B161x motor drivers to ensure reliable operation.

Versatility is a big focus of the cobot's design, and thus we are including the ability to lift a small object or lightweight box using the pincher arm. The cobot will be battery-powered with rechargeable batteries, with a voltage supply range of 24-48V. Additionally, a wireless app will be developed for precise control of the cobot's movements. To maximize its range of motion, the cobot will contain 5 axes of rotation supporting both 180° and 360°. Internally, the cobot will be driven through TI's best-in-class C2000™ MCUs, power ICs, and motor drivers, ensuring precise motor control.

2. Introduction

Factories and commercial industries use cobots to perform tasks that are unsafe and require numerous movements that would be considered strenuous for humans. Thus, the goal of this cobot is to make the production process safer and more efficient. For this project, we are developing a multi-axis cobot that will facilitate the movement of different loads across a factory or commercial setting. The cobot will be wirelessly connected using a mobile device, battery-powered, and support loads of 1-2 lbs. Considering that many warehouses have safety regulations and guidelines to follow, this cobot is designed with many of these in mind. Thus, this project's main focus is to develop an operational cobot that could be placed in a factory setting to complete tasks safely and in a time-oriented manner.

2.1. Background

Collaborative robots, also known as cobots, are a crucial component in factory settings where safety is at the utmost priority and manual labor is considered too dangerous. By having the cobot function through remote control, we can increase the safety and efficiency of workers in these industrial settings, where small tasks could be performed by these cobots rather than human workers. This project aims to develop a multi-axial cobot that can lift a load of 1-2 lbs through a wireless connection.

2.2. Overview

1. Designing a block diagram and flow chart including the key operating specifications.
2. Develop the motor driver, microprocessor, power supply, and battery management control with safety systems, and respective PCBs for each.
3. Develop a user interface to control the cobot in the form of a wireless app.
4. Develop the physical design of the cobot, including the machine and any 3D-printed parts.

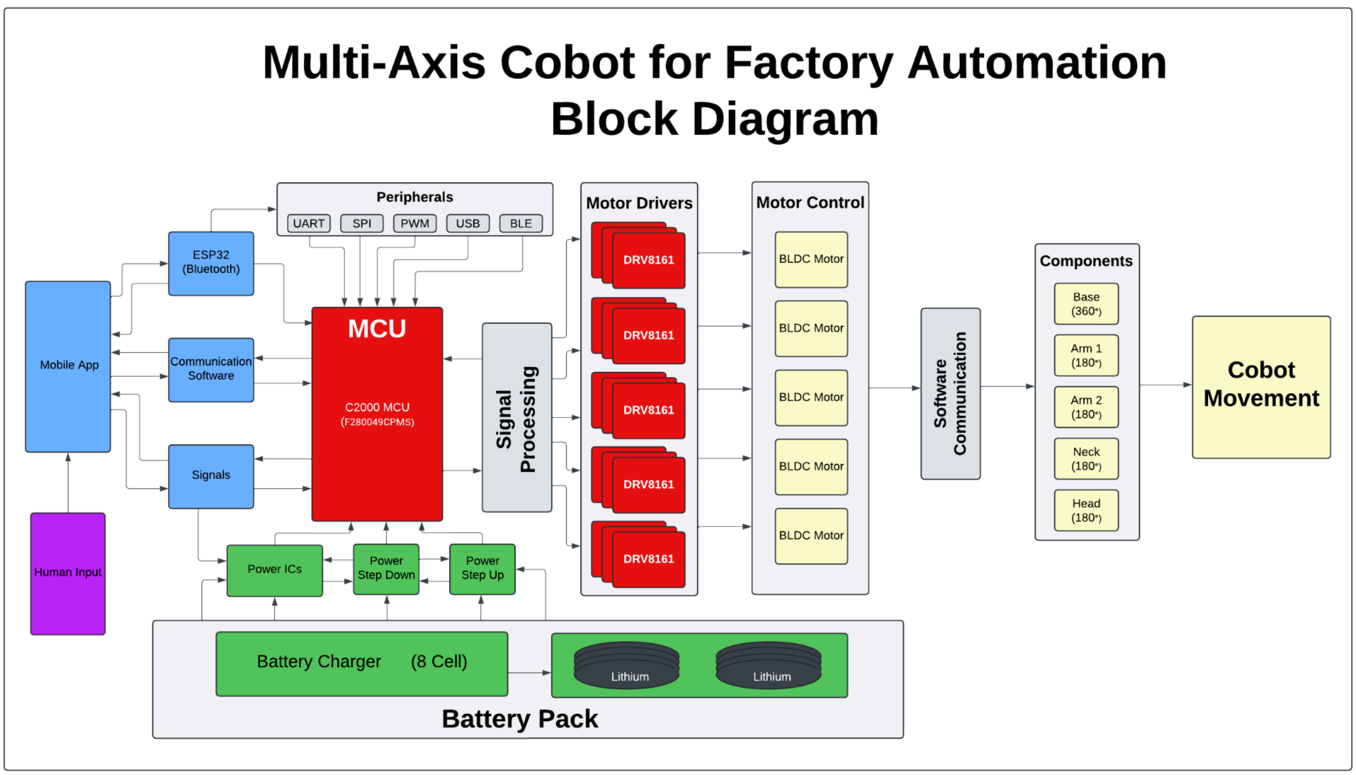

Figure 1. Multi-Axis Cobot Block Diagram

Segment Piece	Degree of Rotation
Base	360°
Arm 1	180°
Arm 2	180°
Neck	360° or 180°
Head	180°

Table 1. Cobot Segment Degrees of Rotation

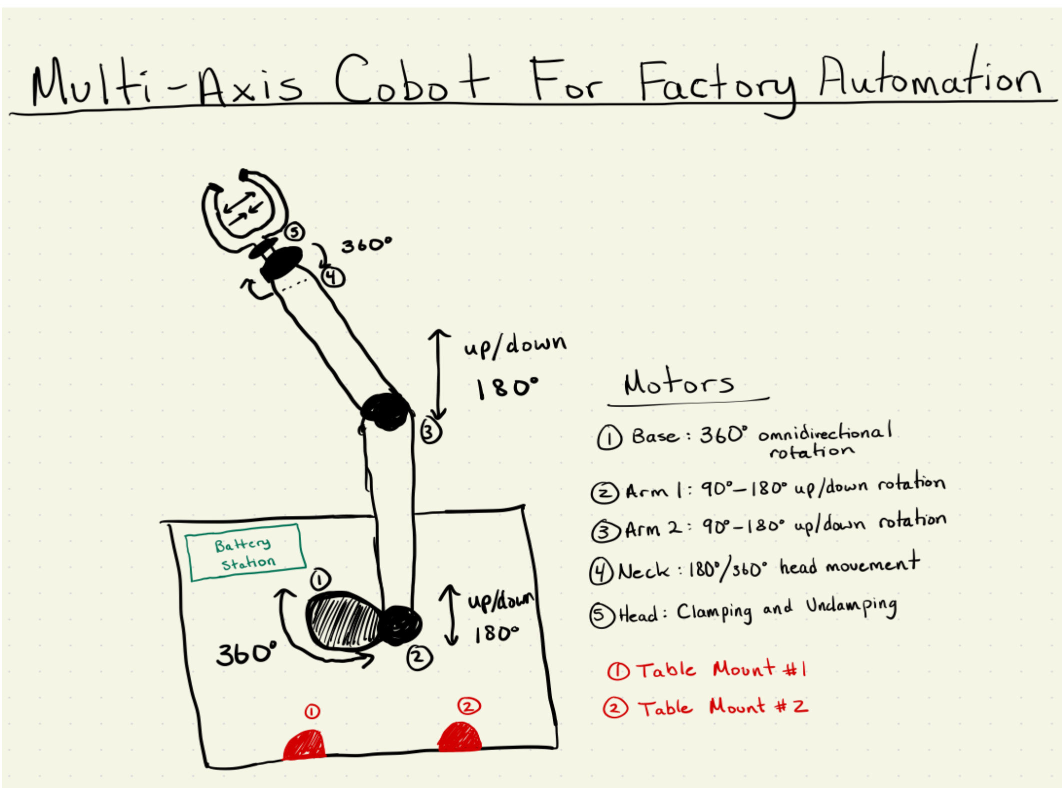

Figure 2. Cobot Prototype Diagram

2.3. Referenced Documents and Standards

Project Outline: Project Description

MCU Documentation

1. C2000 MCU Documentation: F280049CPZS

2. C2000 Evaluation Module: LAUNCHXL-F280049C - EVM (C2000 Evaluation Module)

Motor Driver Documentation
DRV816x Motor Driver Documentation: DRV816x 100V Half-Bridge Smart Gate Driver
DRV8161 Evaluation Module: DRV8161 EVM

Wireless Control Documentation
ESP 32 Documentation**:** ESP 32 Technical Reference Manual
Swift Programming Course: Swift Programming Fundamentals

Motor Documentation - TBD
Battery Documentation - TBD
Physical Design Documentation - TBD
Standards Documentation - TBD

3. Operating Concept

3.1. Scope

This project involves the design and fabrication of a human-controlled, multi-axis collaborative robot (cobot) for use in a variety of manufacturing and commercial environments. The cobot will assist employees by improving efficiency in both strenuous tasks, such as moving boxes, and precise tasks, like handling small loads and assembling items. Operated remotely and equipped with a pincher-style head, it is designed to enhance productivity in diverse settings, performing repetitive and labor-intensive duties that benefit from human oversight and precision.

3.2. Operational Description and Constraints

The cobot will work wirelessly in commercial industries as well as factory settings performing tasks such as moving boxes and small loads, and the operator will use an app to control the cobot. The claw attachment will be used for precise tasks such as moving small objects and loading boxes, as well as moving small boxes between locations. It will move on multiple axes, ranging from 180 degrees to 360 degrees of rotation at each joint, and will be powered by a rechargeable battery.

One constraint of this project is the weight constraints on the load the cobot can lift. The design is only specified to operate in lifting loads up to 2 pounds. The radius of its reach is another constraint, as it will only be able to reach 2 feet from its base. Design constraints will also need to be accounted for. There is a limited budget of $400, and a limited time frame to complete the design.

This cobot has applications in many industries, including the semiconductor, automotive, medical industries. This cobot could be modified to lift and move wafers in a semiconductor clean room setting. Being able to lift boxes would aid employees in the automotive field when building cars, and would be able to precisely move and place lighter metal parts. In the medical industry, the cobot would be able to use its precision for moving small tools such as tweezers or scalpels, assisting doctors during surgery.

Product	**Quantity**	**Cost**
EVM Board (MCU)	2	$78.00
Microprocessor Unit	2	$17.89
Bluetooth Driver	1	$19.99
EVM Board (Motor Drivers)	1	$229.00
Motor Driver Chip	5	$5.66
Anti Static Wrist Strap	2	$13.98
Robotic Arm Pincher	1	$26.99
SUM:		**$391.51**

Table 2. Cost Analysis (To be updated)

3.3. System Description

The system will be split into four subsystems: power and battery management, MCU and processing design, wireless connectivity and interface development, and motor driving system design. The physical construction of the cobot will be divided amongst team members equally in a 25% share.

The power and battery management system covers the design of the power PCB which will deliver power to the motor drivers, motors, MCU, Bluetooth systems, and other electrical components. On the battery management side, this system will cover the selection of a battery and the design of a battery charger. This system will also account for possible faults and implement safeguards to protect the system.

The MCU and processing subsystem are responsible for the MCU PCB design, ESP32 Bluetooth integration, and communication between the MCU and motor driver. This subsystem will use Altium to design schematics and build physical hardware using the C2000 MCU. Furthermore, the MCU/Processing & Motor Driving subsystems will work closely to align on signal communication between the MCU and motor driver.

The wireless connectivity and interface system consists of a mobile application and a Bluetooth connection that allows communication between the cobot and the device sending in the inputs. The mobile application will be developed using Swift and will be compatible with any iOS device, including iPad and iPhone. It will allow the user to control all movements seamlessly and wirelessly. The Bluetooth driver, an ESP-32, will connect directly to the MCU and allow the C2000 to take the inputs from the mobile device and convert them into outputs performed by the cobot. Since we are using a Bluetooth interface, the multi-device connection is compatible and allows any device to connect.

The motor driving design system will create the motor and motor driver PCB. This system will also create the signals needed to effectively operate the motors, which will be Brushless DC (BLDC) motors. This system will achieve the operation of two motors, a 360° and 180° motor controlled by the 8161x TI Motor Drivers. The motor drivers will read a signal from the C2000 MCU and spin/rotate as needed based on the signal. This will all be driven through the B161EVM before integration.

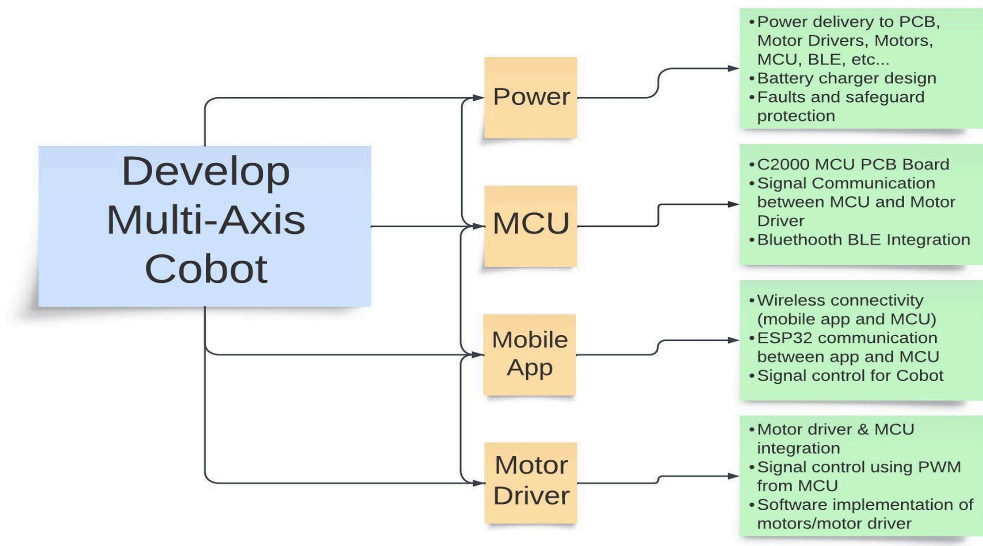

Figure 3. Subsystem Flow Chart

3.4. Modes of Operations

The cobot operates in three distinct modes. In the first mode, it uses its pincher to handle and pick up smaller objects in areas where employees may be present, though their presence is not required as the cobot is remotely controlled. The second mode involves lifting, moving, and placing down objects or boxes, as well as tasks such as using tools or adding items to boxes. The third mode is the charging state, where the cobot recharges its power.

3.5. Users

The target user for this cobot is a factory employee who performs redundant, strenuous tasks such as moving boxes from one place to another, transferring small objects, working with small tools, and loading boxes. These employees will benefit from using the cobot as they will be able to stay outside of dangerous areas to operate the cobot, removing the need to follow protocols to enter. Users will also be able to operate this cobot in areas amongst other employees to complete tasks like moving small loads and boxes.

Training will be required to use this cobot. Basic UI training will need to be completed to understand the functionality of the app, as well as training on using the Bluetooth features. Safety training will also be required to ensure standards are met for using a cobot in a factory setting. Understanding the cobot's thresholds, such as weight limits, is another important piece of training that will be needed before the operator can begin using the cobot. If the operator plans to use the cobot in areas with other employees, additional training regarding the safety of operation amongst humans may be required to ensure safety on the factory floor.

3.6. Support

Operators will be given a "crash course" on using the mobile application. In-person training on safety standards and operating conditions will also be provided, which will include weight management, operating radius, axis movement, claw operation, and other operating constraints. Operation in areas with other employees will also require additional training. Training on how to repair any damage to the equipment will be provided, as well.

4. Scenario(s)

4.1. Transporting and Using Small Objects or Tools

In this scenario, the cobot will be used in an environment where it will have the ability to operate in freedom alongside other human workers, with the goal of moving a small, lightweight object from one location to another. As the cobot is human operated, the primary objective is to transport the item without damaging or breaking it. The human operator must also be aware of the surroundings within the working environment in the case of external human intervention.

The cobot will be equipped with a set of pinchers or clamps with rubber pads attached to the end. It will be able to handle these objects with precision and full range of motion as it lifts the item up. The items the cobot can handle range from 1-2 pounds, such as light tools, food items, loose parts, and small plastic pieces.

The cobot has a hard time picking up flat items using its clamps, so the items would already be upright and ready to grab by the operator. Once picked up, the operator would be able to freely move the object with 360° of rotation and place it down or use the item at a secondary location, within the range of the cobot. This could be as easy as moving an object from one place to another to placing small objects inside bigger boxes.

From an operational point of view, a human operator will control the robot via remote control, responsible for all aspects of its movement and functionality. This includes tasks such as placing and charging the battery, as well as ensuring precise control during the cobot's movements. As the cobot is not autonomous, it requires a skilled human operator to handle the objects with the necessary precision and accuracy they need, alongside constant monitoring of the surroundings around the cobot, as to not injure or interfere with other workers.

4.2. Unpredictable Factory Environment

In this scenario, the cobot will be used in an environment where working conditions are dangerous for humans, such as hazardous and high-risk spaces. The cobot is human-operated, allowing for the pickup and placement of small, lightweight objects in areas where humans cannot operate safely.

The cobot will be equipped with pinchers or clamps that have rubber pads attached to the ends, allowing it to handle items with precision and care. These objects, weighing between 1-2 pounds, could include small objects that are potentially hazardous to humans. The cobot would struggle with flat items on a surface, so ideally the items are positioned upright already.

Once the object is picked up, the cobot can be rotated 360°, giving the operator full control to transport the item to another location or use it within the workspace. Tasks may range from moving items from one place to another, placing them in larger containers, or completing small assembly tasks. The human operator is responsible for controlling all aspects of the cobot’s movement via remote control, ensuring precise handling of objects and monitoring the surroundings to avoid accidents or collisions.

The cobot is not autonomous and relies entirely on a skilled human operator who ensures that tasks are performed safely and efficiently. This operator also manages the cobot's battery charging, positioning, and any other maintenance required to keep it operational. The use of the cobot in this environment significantly reduces risks for human workers while maintaining the efficiency needed for object handling tasks in hazardous conditions.

5. Analysis

5.1. Summary of Proposed Improvements

The multi-axis collaborative robot (cobot) introduces significant enhancements in safety, efficiency, and cost-effectiveness within warehouse and factory environments. By integrating a wireless user interface, the system minimizes direct human involvement, prioritizing worker safety and reducing potential hazards. The cobot’s ability to operate with five degrees of rotation enables it to handle and move objects weighing up to 2 lbs with high precision and efficiency. This capability streamlines repetitive tasks, reduces manual labor, and enhances overall operational productivity. Additionally, the cobot’s design contributes to cost savings by optimizing task performance and reducing the need for manual intervention in potentially risky environments.

5.2. Disadvantages and Limitations

Some limitations of our design include a carrying capacity of 1-2 lbs and an operational range of 2 feet from the base of the cobot. Additionally, due to the size and shape of our pincher mechanism, objects that are flat with little height off the ground will not be supported by the cobot. Since we are catering to industries which may have limited factory space, the range of motion must be precise in order to prevent damage to warehouse equipment and the cobot. One disadvantage of our design is the power supplied to the cobot. Since we are not connecting this machine to direct power but rather connecting a battery, the usage of the cobot will be limited to the duration of the 8-cell battery and the extra rechargeable batteries. To utilize the machine after a full day of use, the batteries must be recharged and plugged back into the machine.

5.3. Alternatives

Some alternate solutions we proposed include a physical controller and a combined cobot head that incorporates a hook and a pinching mechanism. With the physical controller, the benefit is that it would be easier to use, however, the drawbacks are lengthy enough to conclude that it is an inefficient proposition. With a dedicated controller with physical analog inputs, there would be a level of management of the signal interference that is avoided with the mobile application. Additionally, the mobile application allows many devices to communicate with the cobot, as well as being able to connect to different cobots. The final drawback of the physical controller is that if the device gets damaged, replacing or fixing it would be very costly. With a mobile application, releasing a software patch to fix any bugs lowers cost as well as time to resume functionality. The other alternative would be creating one cobot head with the hook and pincher attached. The issue with this solution is that it is not as professional in design, and does not display the versatility of the cobot in a consumer-friendly manner. Additionally, the hook is capable of damaging delicate parts that a dedicated rubber-headed pincher is capable of safely transporting.

5.4. Impact

The multi-axis collaborative robot (cobot) is designed to enhance operational efficiency and safety within manufacturing and commercial environments. By allowing human operators to control the cobot remotely, it reduces the physical strain and risk associated with repetitive or demanding tasks, such as moving objects and loading items. This shift not only minimizes the risk of worker fatigue and injury but also optimizes task precision and productivity.

Moreover, the cobot contributes to a safer work environment by performing tasks that could otherwise expose employees to hazardous conditions. Its presence in factory and warehouse settings helps maintain a safer operational environment by enabling workers to manage and control the cobot from a distance. This approach supports overall improvements in workplace ergonomics and efficiency, making it a valuable asset for modern manufacturing and commercial operations.

11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

Functional System Requirements

REVISION – 0
24 September 2024

Functional System Requirements

for

11: Multi-Axis Cobot For Factory Automation

Prepared by:

Author Date

Approved by:

Project Leader Date

John Lusher, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/24/2024	All Members	All Members	Draft Release - Revision 0

Table of Contents

List of Tables

List of Figures

1. Introduction

1.1. Purpose and Scope

This specification defines the technical requirements for the development items and support subsystems delivered to the client for the project. Figure 1 shows a representative integration of the project in the proposed CONOPS. The verification requirements for the project are contained in a separate Execution and Validation Plan.

Warehouse and factory settings are dangerous environments in which to work. Workers are subject to injury, burnout, and fatigue with repetitive tasks. Instead of exposing humans to these risks, we aim to provide a collaborative robot, or "cobot", solution to improve warehouse safety and efficiency. Our cobot shall be able to move objects (1-2 lbs) via human control from one location to another. A cobot solution will perform better than a human in tedious repetitive workloads as there is a reduced chance of error. Furthermore, cobots minimize the amount of risk humans are exposed to in a factory setting. Placing a cobot in dangerous environments eliminates the chance of a human worker getting struck by heavy machinery, falling objects, or other loose objects. Given that the cobot is wirelessly controlled, human intervention is still required, however, our solution will minimize the required manpower needed to achieve a task. We hope to minimize risk within warehouse operations and improve human worker longevity.

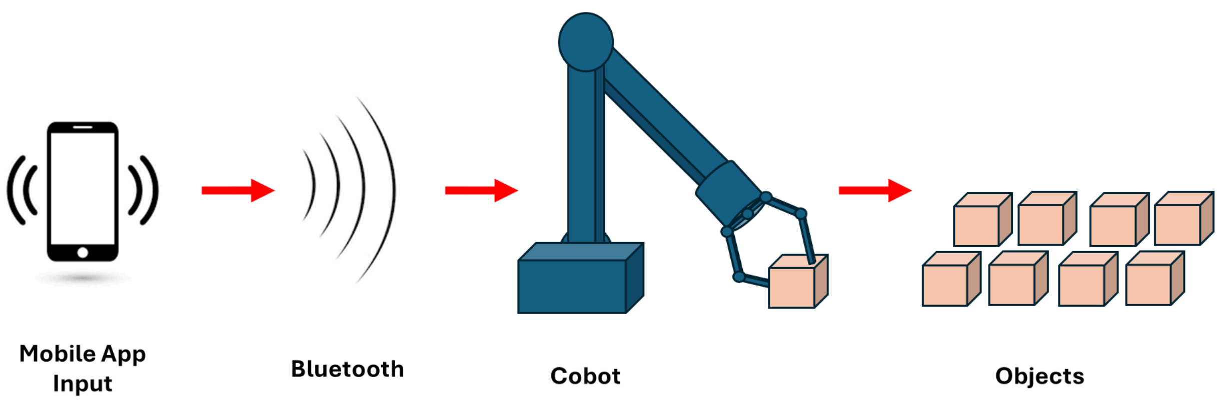

Figure 1. Conceptual Image of Multi-Axis Cobot

1.2. Responsibility and Change Authority

Every member is equally responsible for ensuring the requirements are met and will do so by reviewing deliverables when fellow team members signify their tasks have been completed. Changes proposed by any team members must be discussed with each other and the client, Joshua Maize. Responsibility will be further broken down at the subsystem level, which is outlined below in Table 1.

Subsystem	**Responsibility**
Power and Battery Management	Emily Hamsa
Microcontroller	Adrian Guzman
Wireless Connectivity	Jaishil Shah
Motors and Motor Drivers	Ethan Woods
Physical Design	Full Team Effort

Table 1. Subsystem Leads

2. Applicable and Reference Documents

2.1. Applicable Documents

The following documents, of the exact issue and revision shown, form a part of this specification to the extent specified herein:

Document Number	**Revision/Release Date**	**Document Title**
IEEE 802.15.1.	June 2005	IEEE Standard for Information technology

Table 2. Applicable Documents

2.2. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	Revision/Release Date	Document Title
SLVSGZ1A	July 2024	DRV816x 100V Half-Bridge Smart Gate Driver with Integrated Protection and Current Sense Amplifier
SPRS945G	January 2023	TMS320F28004x Real-Time Microcontrollers
ESP-32	September 2024	ESP32 Series

Table 3. Reference Documents

2.3. Order of Precedence

In the event of a conflict between the text of this specification and an applicable document cited herein, the text of this specification takes precedence without any exceptions.

All specifications, standards, exhibits, drawings, or other documents that are invoked as “applicable” in this specification are incorporated as cited. All documents that are referred to within an applicable report are considered to be for guidance and information only, except ICDs that have their relevant documents considered to be incorporated as cited.

3. Requirements

This section defines the minimum requirements that the development items must meet. The requirements and constraints that apply to performance, design, interoperability, reliability, etc., of the system, are covered.

3.1. System Definition

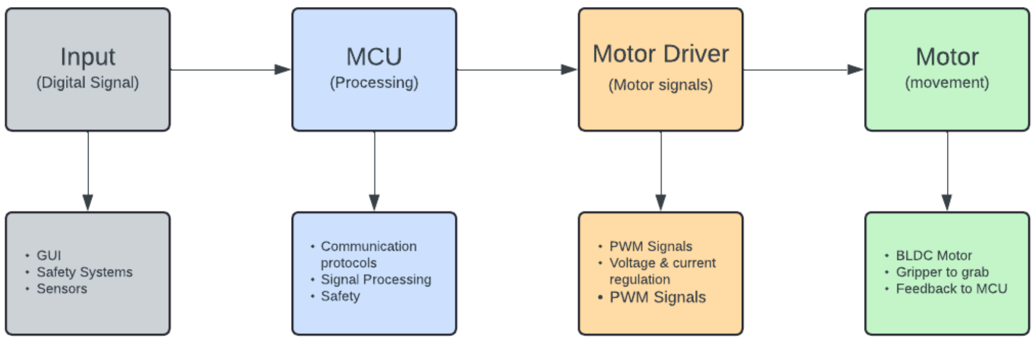

Figure 2. Multi-Axis Cobot Flowchart

The multi-axis collaborative robot used for factory automation is a solution to dangerous working environments for factory employees and the repetitive fatigue of monotonous tasks. This cobot will be operated via a wireless mobile app and will be powered by a rechargeable battery, allowing users flexibility to place the robot in a desired location and perform specific tasks. There will be four subsystems: power and battery management, microcontroller design, motor and motor driver design, and mobile application development.

When in use, the operator will be able to drive the motion of the cobot using the mobile application, which will be used on an iOS device. Signals sent from the app will be delivered to the cobot via Bluetooth, and we will use an ESP-32 for this functionality. The microcontroller will convert these signals into directions for the motor drivers, which will in turn deliver instructions to the motors. Using software communication, the motors will move each correlated component of the cobot. The cobot will be powered with a battery, and through the use of step-down converters, power will be delivered at the appropriate voltage and current rating to each component in the cobot. When charging, the battery management system will recharge the battery, and will include multi-cell balancing components to ensure each cell is charged to the maximum and not overloaded.

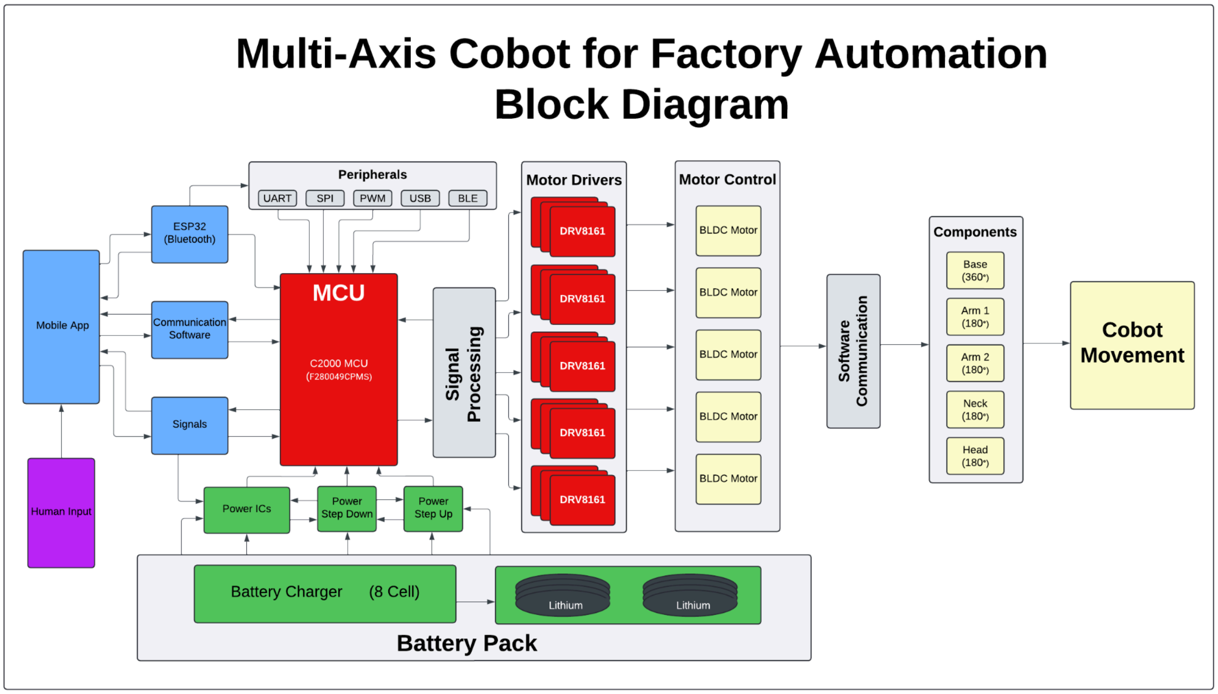

Figure 3. Multi-Axis Cobot Block Diagram

The cobot will also be able to send signals back to the mobile app to indicate any system errors. These errors include object detection, physical errors to motor drivers, and connection errors. Object detection will send warnings back to the user if there is an object interfering with the cobot's movement, such as a wall. Any physical damage or operation errors pertaining to the motor drivers will be sent to the user as well, indicating that servicing the cobot may be needed. If the mobile application is unable to connect to the cobot, error messages will be sent to the user as well.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1 C2000 EVM Wakeup

Launch EVM and connection is successful.

Rationale: In order to program MCU, understand the EVM by using example code on CCS and other TI-provided documents.

3.2.1.2 MCU & Motor Driver

MCU is correctly created to connect with motor drivers in the schematic.

Rationale: For motor motion, MCU and motor drivers have to work together to create signals with PWM modules.

3.2.1.3.1 Schematic & PCB Creation for MCU/Connectivity Subsystem

Schematic and PCB footprinting integrating MCU, motor drivers, and ESP32.

Rationale: Schematic and PCB design has no errors in validation. To ensure proper design of PCB the design rule check (DRC) shall come back with zero errors.

3.2.1.3.2 Schematic & PCB Creation for Motor Driver Subsystem

Creating schematics for PCB boards for individual motors, 5 in total.

Rationale: To ensure proper motor control, the MCU and motor driver subsystem need to be aligned on the proper signals required for motor motion.

3.2.1.4 Precise Motor Control

Through encoders, motors will move a few degrees at a time.

Rationale: Motors need to operate with different degrees of rotation. To support the different movements, various measurements of angles each motor can support will be performed.

3.2.1.5 All Motor Movement

All motors can move through communication with the MCU and motor drivers.

Rationale: Motor functionality is an integral part of ensuring the cobot moves accurately. The motors will need to be tested various times to ensure accurate representation of output with given input via wireless app.

3.2.1.6 Schematic & PCB Creation for Power Management

Power management system PCB schematics and footprinting are fully designed in Altium.

Rationale: Power will be imputed at 48V, not all components in the cobot will be at 48V. It is important that the power management subsystem can step down to different voltages that are required at different parts of the process. This will need a PCB with different power rails.

3.2.2. Physical Characteristics

3.2.2.1 Mounting

When the system is lifting maximum payload, it stays mounted on its surface without tipping.

Rationale: The cobot will only be able to support a specific load range. More than is allowed will make the cobot fall over. To find this range, the cobot will be tested by attaching weights to the arms of the cobot, measuring the maximum payload before the system tips over.

3.2.2.2 System Precision

Physical movement and load capacity functionality can move objects precisely and accurately.

Rationale: The cobot should not move an object differently than is intended as it will lead to damage to the item and or cobot. Ensure that the system axes can move at a minimum of 2 degrees, and move objects 2 inches on the table.

3.2.2.3 Structure Integrity

All components and joints are structurally sound.

Rationale: So that the cobot does not collapse when under load, the maximum payload needs to be tested. The cobot should support lifting a maximum payload, the physical design can support both the weight of itself and the load.

3.2.3. Electrical Requirements

3.2.3.1. Inputs

3.2.3.1.1 Input Voltage

The battery will supply a constant 48V to the system at the input

Rationale: The battery shall give 48V to the system. By using a multimeter, test to ensure the system holds at 48 volts while the battery is connected. This is required to ensure no voltage drop-off that could impact the functionality of the cobot.

3.2.3.1.2 Battery Sizing

Choose a battery size that will adequately power the cobot based on the load list.

Rationale: In order to power the cobot properly, battery sizing using formulas and a load list to determine appropriate voltage and amp-hour sizing is required.

3.2.3.1.3 Voltage Step-Down

Power PCB will step-down voltage to the subsequent system requirements

Rationale: Different systems of the cobot will require different input and output voltages. By using a multimeter, test each voltage rail and ensure it is holding the correct voltage.

3.2.3.1.4 MCU & Bluetooth

MCU schematic is created with ESP-32 for Bluetooth connectivity

Rationale: In order for Bluetooth to work via the wireless app, the mobile app needs to be able to communicate with the MCU. The MCU and wireless subsystems shall align on requirements, all necessary signals are accounted for and communication between Bluetooth and MCU can be displayed.

3.2.3.1.5 Motor Movement

Motor Drivers can send signals to move the Motor through the MCU

Rationale: For accurate and rapid motion of the motor, the MCU and motor drivers need to exchange signals. To achieve this, the two subsystems must align on required signals for control, and display movement through MCU control with motor drivers and motor.

3.2.3.1.6 GUI Operational

User Interface (application) is deployed to mobile device

Rationale: To control the cobot, the wireless GUI has to be operational and send out signals via Bluetooth. Once the connection is stable, input lag is minimal, and outputs are properly received/translated.

3.2.3.1.7 Button Functionality

The button will respond to human input and send digital output

Rationale: The cobot will be controlled via a wireless app. The app needs to be able to accept human input so that signals can be sent to the cobot and output can be visualized. This will lead to human interaction that works without fault.

3.2.3.2. Outputs

3.2.3.2.1 LED Program for MCU/Connectivity System

After PCB assembly, flash the MCU and perform the LED light blinking program to ensure it is working.

Rationale: To demonstrate understanding of C2000 controls, software, and signals the C2000 EVM will be used. The CCS GUI with TI-provided software will help develop code that functions on the EVM. Once the EVM development is successful, this code can be replicated onto the created PCB for the MCU subsystem. Programming the MCU is an important part of achieving cobot functionality. In the end, an LED blinking on the board will confirm functionality.

3.2.3.2.2 Wireless Connectivity and Connection to MCU

Mobile application connects to the ESP-32 ad C2000 MCU.

Rationale: The mobile app is the main form of communication with the cobot. It is important to not lose connection and therefore we must test that the connection does not drop out more than 1 time in 5 minutes while also connecting within 30 seconds.

3.2.3.2.3 MCU to Motor Driver Output

MCU is correctly created to connect to the motor drivers in the schematic.

Rationale: To ensure proper communication between the motor driver and MCU, the schematic on Altium must be created with zero DRC errors. Furthermore, the two subsystems must Align to ensure signals are being sent properly to show motor movements.

3.2.4. Environmental Characteristics

3.2.4.1 Warehouse and Factory Safety Considerations

Ensure the surface that the cobot is stationed on is able to support at least 30 pounds so that it can sufficiently hold the cobot's weight, and that the surroundings are clear of items that could potentially be damaged by the cobot's movement.

Rationale: The cobot should not damage the environment it is placed in. Test the weight limit of the surface using weights that total 30 pounds and use a ruler to measure the radius of movement of the cobot, checking if there are damageable items in its path.

3.2.4.1 Delicate Material (Handle with Caution)

The cobot is able to lift and handle delicate items without causing them any physical damage.

> *Rationale: The cobot shall not break anything it picks up. If objects are broken when being lifted, the purpose of the cobot is defeated. Cobot shall pick up a delicate item, such as a cracker or an object that has surfaces that could be easily punctured, and move from one location to another. Ensure there is no physical damage to the delicate item.*

3.2.5. Failure Propagation

3.2.5.1 Object Detection in Factory Setting

If part of the cobot's arm, head, or other components is turning or moving into a surface, such as a wall, that is restricting the movement of the cobot, an error message will be sent to the user's mobile application.

> *Rationale: In a factory setting, many objects are moving around. The cobot should not damage any other objects and will send warning messages to the user if movement can not be continued. Using the controller, turn the cobot into a wall so that it is unable to continue turning, and ensure an error message is sent to the user through the mobile application.*

3.2.5.2 Physical Errors with Motor Drivers

If there is physical damage detected to the motor drivers that is inhibiting the cobot to function correctly, an error will be sent to the user's mobile application.

> *Rationale: To work the user of potential faults, the motor driver will give the user a warning if an error has been detected. Such errors can include physical damage, object obstruction, or low battery.*

3.2.5.3 Connection Errors

If the mobile app is not able to connect to the cobot, or the mobile app loses connection to the cobot, an error message will be sent to the user's mobile application.

> *Rationale: Connection should be constant from wireless app to cobot. If connection drops, danger can occur and humans no longer have control of the cobot. To prevent this, error messages will be sent to the user via mobile app.*

3.2.5.4 C2000 Error Detection System Functionality

Any errors that the cobot or MCU detects will send a signal to the ESP-32 for the mobile application to receive.

> *Rationale: Signal detection using the MCU and ESP-32 is implemented to give human warning. The error message will display on the mobile application so that the user is made aware and the error or faulty use can be corrected.*

4. Support Requirements

4.1 iOS Device Compatibility

The device is required to run the latest version of iOS to run this application. (9/24/24: iOS 18.0) Make sure that bluetooth is on and other devices are disconnected prior to device setup.

Rationale: The mobile app will be placed on the app store for user download. The user must have the latest iOS versions to comply with mobile applications. It is recommended to have the battery on the phone at full charge so that the connection will not drop if the phone dies.

4.2 Surface Preparation

Before placement of the device onto the workstation, make sure to clear the area and make sure the clamp is secure onto the surface.

Rationale: The device should be placed on a clean flat surface. If this is not done, risk of short circuit or other damage can be caused. Ensure that all liquids, hazardous material, and sharp or dangerous objects are either relocated or placed in an area that is outside the range of the cobot.

4.3 Device Warranty and Replacement

This device contains parts manufactured by Texas Instruments and individually designed components.

Rationale: As the sponsor of the project, Texas Instruments devices should go into this cobot. To replace TI parts, directly contact Texas Instruments. In order to replace any individually designed parts, contact the Multi-Axis Cobot Design Team.

4.4 Training Requirement

Please ensure all technicians complete training and have been given warehouse/factory instruction on what the machine is being utilized for.

Rationale: In order to access the device connection and control on the mobile application, the user must complete the training and watch the safety demonstration.

4.5 Electrical Safety

Make sure all batteries, wires, and electrical components are secure and not in the range of the cobot's movement.

Rationale: The Multi-Axial Cobot Design Team is not responsible for any damage, harm, or injury caused in the workplace. Please understand that all responsibility falls upon the user.

4.6 Human Safety

Do not operate the device if humans are within 5 feet of the cobot.

Rationale: As previously stated, any harm caused in the workplace due to the cobot falls on the user of the device. Ensure all technicians understand their responsibility and are aware of the preventative measures.

Appendix A: Acronyms and Abbreviations

TI	Texas Instruments
IEEE	Institute of Electrical and Electronics Engineers
iOS	iPhone Operating System
CCS	Code Composer Studio
LED	Light-Emitting Diode
PCB	Printed Circuit Board
DRC	Design Rule Check
MCU	Micro-Controller
ESP-32	Espressif32 Bluetooth Driver
C2000	(TI) Microcontroller
GUI	Graphical User Interface
DRC	Design Rule Checking
EVM	Evaluation Module
PWM	Pulse Width Modulation
BLDC	Brushless Direct Current
ICD	Interface Control Document
UART	Universal Asynchronous Receiver Transmitter
SPI	Serial Peripheral Interface
USB	Universal Serial Bus
BLE	Bluetooth Low Energy

Appendix B: Definition of Terms

11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

INTERFACE CONTROL DOCUMENT

REVISION – Draft
25 January 2018

Interface Control Document

for

11: Multi-Axis Cobot For Factory Automation

Prepared by:

Author Date

Approved by:

Project Leader Date

John Lusher II, P.E. Date

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
0	9/26/2024	All Members	All Members	Draft Release - Revision 0

Table of Contents

List of Tables

List of Figures

1. Overview

This document provides an overview of the integration between the different subsystems. The four subsystems are split into power, MCU/processing, wireless, and motor driving. The power subsystem will deliver accurate power to components, MCU/processing will ensure signals are sent from MCU, wireless will communicate with the cobot via Bluetooth, and motor driving will create motor movement. In this document, you will find the necessary system requirements to achieve this integration for the cobot. This document includes necessary references and definitions as well as physical, thermal, and electrical interfaces.

2. References and Definitions

2.1. References

Document Number	Revision/Release Date	Document Title
IEEE 802.15	June 2005	IEEE Standard for Information Technology
TMS320F28004x	January 2023	TMS320F28004x Real-Time Microcontrollers
DRV816x	July 2024	DRV816x 100V Half-Bridge Smart Gate Driver with Integrated Protection and Current Sense Amplifier
ESP-32	September 2024	ESP-32 Series

Table 1: References

2.2. Definitions

Cobot	Collaborative Robot
MCU	Microcontroller Unit
mA	Milliamp
mW	Milliwatt
in	Inch
lb	Pound
PCB	Printed Circuit Board
PMS	Power Management System
TBD	To Be Determined
V	Volt

3. Physical Interface

3.1. Weight

3.1.1. Weight of Cobot Apparatus

This section covers the weight of physical components and the design of the cobot. Encompassed in this section are the motor driving subsystem, MCU and processing subsystem, power management subsystem, and wireless communication subsystem. While we have not confirmed exact weights, we roughly estimate the cobot apparatus will weigh 10 pounds, not including the physical design and base.

Component	Weight	Number of Items	Total Weight
Motors	~ .4lb - 2.6lb	5	TBD
Motor Drivers	~ 0.22lb	15	TBD
Pincher	0.121lb	1	0.121lb
ESP-32	0.121lb	1	0.121lb
Physical Design and Base	TBD	1	TBD
Associated PCB's	TBD	TBD	TBD

Table 2: Weight of Cobot Components

3.1.2. Weight of Battery Management

The battery management system is weighed separately because, in the final design, the battery and its charging system will be removable. While we have not confirmed exact weights, we roughly estimate the battery management system will weigh 4 pounds.

Component	Weight	Number of Items	Total Weight
Battery	~4lb	1	~4lb
Associated PCB's	TBD	TBD	TBD

Table 3: Weight of Battery Management Components

3.2. Dimensions

3.2.1. Dimension of Motor Driving Subsystem

Each motor will have different specifications. For each motor to function, it will have a PCB with 3 DRV8161 Motor Drivers. We estimate the PCB alongside each motor will be at least 1.18 inches by 1.18 inches.

Component	**Length**	**Width**	**Height**
Motors	~ 1.57 - 3.15in	~ 1.18 - 3.15in	~ 1.57 - 3.94in
Motor Drivers	~ 0.2in	~ 0.12in	TBD
Associated PCB’s	~ 1.18in	~ 1.18in	TBD

Table 4: Motor Driving Subsystem Dimensions

3.2.2. Dimension of MCU/Processing Subsystem

The MCU subsystem will consist of a PCB that holds the F280049CPZS microcontroller which has dimensions 0.63 inches by 0.63 inches. Furthermore, with other ICs and passive components integrated into the PCB, we estimate it will be at least 3.94 inches by 3.94 inches.

Component	**Length**	**Width**	**Height**
F280049CPZS	~ 0.63in	~ 0.63in	TBD
Associated PCB’s	~ 3.94in	~ 3.94in	TBD

Table 5: MCU and Signals Processing Subsystem Dimensions

3.2.3. Dimension of Power/Battery Management Subsystem

The power and battery management subsystem envelops the battery, which we estimate will be 48 volts, alongside the PCBs that are needed to run the PMS. We estimate the battery will be around 10.63 inches by 3.15 inches by 2.76 inches and the PCBs needed will be around 3.34 inches by 3.34 inches.

Component	**Length**	**Width**	**Height**
Battery	~ 10.63in	~ 3.15in	~ 2.76in
Associated PCB’s	~ 3.34in	~ 3.34in	TBD

Table 6: Power and Battery Management Subsystem Dimensions

3.2.4. Dimension of Wireless Communication Subsystem

The wireless communication subsystem is built around an ESP-32 board. The ESP-32 typically comes in size 0.71 inches by 1.00 inches by 0.12 inches. Including the size of the ESP-32 board and associated ICs, we estimate the dimensions of the PCB will be 2.76 inches by 2.76 inches.

Component	**Length**	**Width**	**Height**
ESP-32	~ 0.71in	~ 1.00in	~ 0.12in
Associated PCB’s	~ 2.76in	~ 2.76in	TBD

Table 7: Wireless Communication Subsystem Dimensions

3.3. Mounting Locations

3.3.1. Motor Mounting

Each motor will need to be mounted and secured to the robotic arm apparatus. To prevent sag and weight issues, the mountings will have to be strong enough to support the weight of the motor under maximum load.

3.3.2. Cobot to Base Mounting

To ensure a strong center of gravity when lifting and moving a payload, the bottom (Base) of the cobot will be mounted to a baseboard or metal sheet. This allows for the cobot to have a designated location on a square surface to support the weight of the robotic arm with and without a load. By securing the cobot to a base plate, we can better control the lean and influence the movement of the cobot on its center of gravity.

3.3.3. Base to Table Mounting

To ensure the cobot does not tip over if its center of gravity changes, the base plate the cobot is mounted to will also be mounted to a table. The base plate can clamp onto the outside lip of the table it is placed on, which ideally means the table size is the size of the base plate. By clamping the base plate to the table, we eliminate the center of gravity changing due to the base plate lifting and leaning during movement.

4. Thermal Interface

Our project involves working with high voltage and current, which means cooling will be essential for the PCBs, motors, and other components to prevent overheating. Depending on the power draw and consumption, many of these parts will require adequate thermal management. Since the cobot is not designed for continuous operation over extended periods, a heatsink in the form of a metal heat dissipation plate, along with proper airflow, should be sufficient to maintain safe operating temperatures.

While a cold wall isn't necessary due to the intermittent operation of the system, heatsinks will still be required for most, if not all, of the motors in the robotic arm. These motors, subjected to high voltage and current, will generate heat more rapidly than lower-power alternatives. However, given the cobot's non-continuous, human-triggered operation, we believe that proper ventilation and airflow should be enough to manage any potential heat issues, even with the higher power motors.

In summary, the use of simple metal heat dissipation plates combined with adequate ventilation should address all thermal concerns without the need for more complex cooling solutions like a cold wall.

5. Electrical Interface

Provide details on the electrical interface. Examples are:

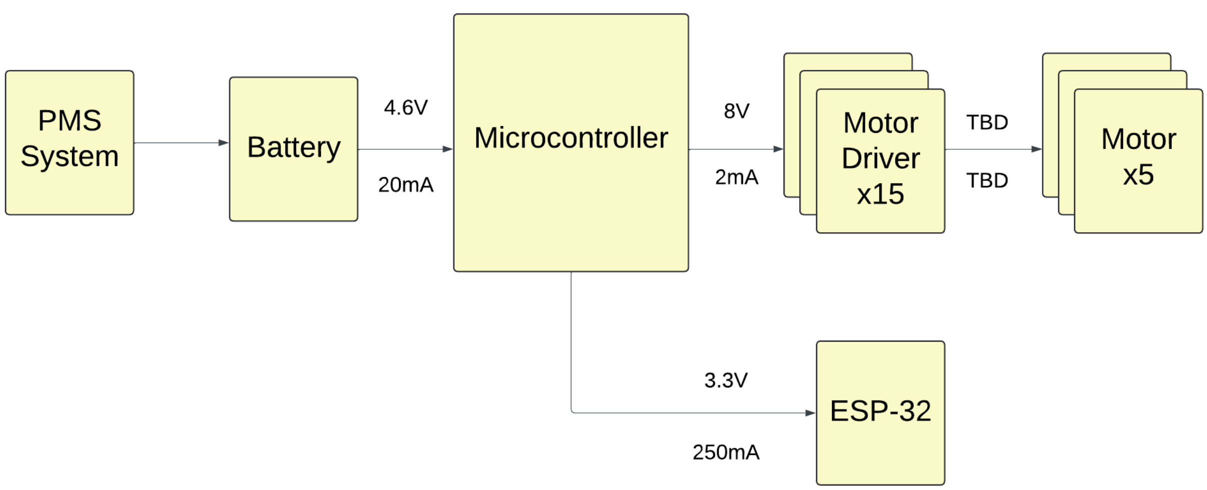

Figure 1: Electrical Interface Diagram

5.1. Primary Input Power

5.1.1. Primary Power Source for Cobot

All power for the cobot will be supplied by an external, rechargeable battery. The voltage rating for the battery is 48V, and the amp-hour specification is currently TBD as we are finalizing parts selection. Our battery will be connected to a battery management system, which will include regulators and buck converters to deliver power to all subsystems in the required amounts.

5.1.2. Battery Charging

Our team will also be developing a charging unit for the battery. Because we are using a large, multi-cell battery, this system will include safeguards and associated components to ensure when the battery is charging, cells are charged in a balanced way, and potential faults are detected.

5.1.3. Microcontroller

The cobot will be using the F280049CPZS microcontroller from the C2000 family at Texas Instruments. This MCU has a 32-bit CPU at 100 Mhz and includes 356 KB of on chip memory. There is no BLE module, hence an ESP-32 will be required externally.

5.2. Signal Interfaces

5.2.1. DRV8161 Pulse Width Modulation Signals

The communication between the motor drivers and motors will be facilitated through 3-phase Pulse Width Modulation (PWM), utilizing a C2000 MCU to generate the necessary control signals. This setup ensures precise and efficient control over the motor phases.

5.2.2. Digital Signals Between Mobile App and Cobot

The cobot will use an ESP-32 to connect the user’s mobile app to the cobot through Bluetooth. The ESP-32 for use with Bluetooth is compliant with Bluetooth v4.2 BR/EDR and Bluetooth LE specifications, and has +9 dBm transmitting power.

5.3. User Control Interface

Users will be able to operate the cobot using a wireless mobile application that will be developed using FlutterFlow. This mobile app will include dialogs and menus which will allow the user to complete training videos and modules, connect to the cobot via Bluetooth, and receive error messages from the cobot.

5.4. Voltage and Current Levels

5.4.1. Maximum Voltage and Current

Component	Voltage [V]	Current [mA]	Power [mW]
Motor Driver	20	2	40
MCU	3.3	20	92
ESP-32	3.6	250	900
Motors	TBD	TBD	TBD

Table 8. Maximum Voltage, Current, and Power Levels

The table above outlines the maximum ratings for voltage and current consumption of each component. Operating under maximum conditions will not be done frequently, however, it is important to take into consideration when designing safety and sizing our battery.

5.4.2. Typical Voltage and Current

Component	Voltage [V]	Current [mA]	Power [mW]
Motor Driver	8	2	16
MCU	3.3	20	66
ESP-32	3.3	250	825
Motors	TBD	TBD	TBD

Table 9. Typical Operating Voltage, Current, and Power

The table above shows a more accurate representation of the power needs of each component during operation. While most of the components are relatively low-power, we anticipate our motors will have a much larger power requirement.

6. Communications / Device Interface Protocols

6.1. Wireless Communications (Bluetooth)

6.1.1. Bluetooth

The cobot uses Bluetooth v4.2 for wireless communication between the mobile app and the cobot's microcontroller unit (MCU). Since the C2000 MCU does not have a built-in Bluetooth module, an ESP-32 is employed for this purpose. The ESP-32 complies with the IEEE 802.15.1 standard, which ensures low-power, short-range wireless communication.

6.2. Host Device

6.2.1. C2000 Microcontroller Unit (MCU)

The C2000 serves as the primary Host Device for the cobot, as it is responsible for real-time processing and motor control. The C2000 will receive signals from the ESP-32, which communicates with the user’s controller. Then, the C2000 will generate PWM signals to send to the Motor Drivers for precise Motor control.

6.3. Device Peripheral Interface

6.3.1. ESP-32 Connection with User Interface

The ESP-32 serves as the wireless communication interface between the User Interface and the C2000. Using Bluetooth, the ESP-32 transits signals and commands from the Interface to the C2000. The communication is handled through UART.

6.3.2. Motor Driver and Motor Communication

The communication between the C2000 MCU and the motor drivers is managed through 3-phase Pulse Width Modulation (PWM) signals. These PWM signals control the motor phases to regulate speed, torque, and positioning. The C2000 MCU outputs these PWM signals directly to the motor drivers, ensuring precise control.

6.3.3. Micro-USB

A Micro-USB port will be needed to interface between the computer and the MCU. The MCU offers I2C, CAN, SPI, UART, and FSI. The communication protocol to use for micro-USB is still to be determined.

11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

Execution Plan

Multi-Axis Cobot for Factory Automation Gantt Chart

Week of:

	Titles	8/27/24	9/3/24	9/10/24	9/17/24	9/24/24	10/1/24	10/8/24	10/15/24	10/22/24	10/29/24	11/5/24	11/12/24	11/19/24	11/26/24	12/3/24	Date Completed
Key:	**Defining Project Scope (8/27)**																8/27/24
Black - ALL	**Subsystem Division / Brainstorming (9/3)**													Not Started			9/3/24
Green - Adrian Guzman	**Creating a Budget Parts List (9/10)**													In Progress			9/12/24
Purple - Jaishil Shah	**Subsystem Software Initialization (9/10/24)**													Completed			9/17/24
Blue - Emily Hamsa	**CONOPS Report (9/15)**													Behind Sch.			9/12/24
Red - Ethan Woods	**Creating a (TI) Parts List (9/17)**																9/17/24
	Swift Compatability (9/17)																9/17/24
	Motor Selection Finalization (9/20)																
	Complete Subsystem Introduction Project (9/24)																
	Subsystem Introduction Project (Emily) (9/27)																
	Project Selection Presentation (9/23)																
	FSR, IDC, Milestones and Validation plan (9/26)																
	Parts Selection Finalization and Battery Sizing (9/27)																
	Competed PMS PCB Design and Ordering (10/11)																
	Mid-Term Presentation (10/14)																
	PCB Design with MCU (10/18)																
	Order PCB Board & Components (10/18)																
	Motor/Motor Driver Testing with EVM Board (10/18)																
	Swift Playground Course/ Coursera Course (10/20)																
	Subsystem Introduction Project (Jaishil) (10/20)																
	Subsystem Introduction Project (Ethan) (10/20)																
	PCB Assembly and Testing with EVM (10/30)																
	PCB Assembly and Test (11/01)																
	PCB Design and Ordering (11/4)																
	Mobile App Functionality (11/5)																
	Final Presentation (11/11)																
	LED Blink Functionality (11/18)																
	Bluetooth Integration with Mobile App (11/19)																
	Subsystem Demonstration (Jaishil) (11/26)																
	Subsystem Demonstration (Ethan) (11/26)																
	Subsystem Demonstration (Emily) (11/26)																
	Subsystem Demonstration (Adrian) (11/26)																
	Final Report (12/5)																

11: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

VALIDATION PLAN

Multi-Axis Cobot for Factory Automation Validation Plan

Status Key:	Paragraph Number	Test Name	System Criteria	Criteria for Success	Status	Responsible Engineer(s)
In Progress		**Joint Effort**				
Completed	3.2.2.1	Mounting	When system is lifting the maximum payload, it stays mounted to its surface without tipping.	Attatch weights to arm of Cobot, measure the maximum payload before system tips over.		Full Team
Late	3.2.1.1	C2000 EVM Wakeup	Launch EVM and connection is successful	EVM turns on, CCS software identifies device, simple command run.		Full Team
TBD	3.2.2.2	System Precision	Physical movement and load capacity functionality is able to move objects precisely and accurately.	System axes can move at minimum 2 degrees, and move objects 2 inches on the table.		Full Team
	3.2.2.3	Structure Integrity	All components and joints are structrually sound	When cobot is lifting maximum weight, the physical design is able to support the weight of both itself and the load.		Full Team
		Power Subsystem				
	3.2.1.6	Schematic & PCB Creation	Power management system PCB is fully designed in Altium	Completion of all schematics in Altium and is validated, checked for validation errors, and DRC has no errors.		Emily Hamsa
	3.2.3.1.2	Battery Sizing	Choose battery size that will adequetly power cobot based on load list	Completion of battery sizing using formulas and load list to determine appropriate voltage and amp-hour sizing		Emily Hamsa
	3.2.3.1.1	Input Voltage	The battery will supply a constant 48V to the system at the input	Using a multimeter, test to ensure system hold at 48 volts while battery is connected.		Emily Hamsa
	3.2.3.1.3	Voltage Step-Down	Power PCB will step-down voltage to the subsequent systems requirements	Using a multimeter, test each voltage rail and ensure it is holding correct voltage		Emily Hamsa
		MCU/Connectivity Subsystem				
	3.2.1.3.1	Schematic & PCB Creation	Schematic and PCB footprint integrating MCU, motor drivers, and ESP32	Individual sheets within altium are completed and reviewed for accuracy, there are no errors in validation, and DRC comes back with zero errors.		Adrian Guzman
	3.2.3.2.1	LED Program	After PCB assembly, flash the MCU and get LED program working	Use the C2000 EVM and CCS software to develop code that functions on EVM, transfer code to PCB board and visualize LED blinking		Adrian Guzman
	3.2.1.2	MCU & Motor Driver	Test MCU and motor driver communication	Align with motor driving subsystem to ensure signals are accounted for. Integrate and show motor movement		Adrian Guzman, Ethan Woods
	3.2.3.1.4	MCU & Bluetooth	Test MCU and bluetooth (ESP32) communication	Align with wireless subsystem requirements, all necessary signals are accounted for and communication between bluetooth can be displayed		Adrian Guzman, Jaishil Shah
		Motor Driver Subsystem				
	3.2.1.3.2	Schemating and PCB Creation	Creating Schematics and PCB boards for Individual Motors (5 Total)	All schematic sheets are created, validated, and DRC has zero errors.		Ethan Woods
	3.2.3.1.5	Motor Movement	Motor Drivers can send signals to move Motor through MCU	Align with MCU subsystem on requried signals for control, display movement through MCU control with motor drivers and motor.		Ethan Woods, Adrian Guzman
	3.2.1.4	Precise Motor Control	Through encoders, Motors will move a few degrees at a time	Perform angular measurements on motor when spinning, record data.		Ethan Woods
	3.2.1.5	All Motor Movement	All motors can move through communication with the MCU and Motor Drivers	Motor functionality is precise and all movements correspond with the intended output		Ethan Woods
		Wireless Subsystem				
	3.2.3.2.2	Wireless Connectivity	Mobile app connection to ESP-32 and C2000 MCU	Mobile app connection does not drop out more than 1 time in 5 minutes, and connects within 30 seconds.		Jaishil Shah, Adrian Guzman
	3.2.3.1.6	GUI Operational	User Interface (application) is deployed to mobile device	Connection is stable, input lag is minimal, and outputs are properly received/translated		Jaishi Shah
	3.2.5	Error Warnings	C2000 Error Detection System Functionality	C2000 is able to detect error with cobot motion / faulty inputs from the User Interface and send those signals back to the mobile device via. ESP-32		Jaishil Shah, Adrian Guzman
	3.2.3.1.7	Button Functionality	Button Input Functionality	Human interaction is received and works without fault		Jaishil Shah