42: Multi-Axis Cobot For Factory Automation

Adrian Guzman, Emily Hamsa, Ethan Woods, Jaishil Shah

**Functional System Requirements**

REVISION – 2

27 February 2025

Functional System Requirements

for

42: 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 |
| **1** | 12/5/2024 | All Members | All Members | Updated error in team name, team 11 to team 42, updated characteristics section |
| **2** | 2/27/2025 | Emily Hamsa  Adrian Guzman | Emily Hamsa  Adrian Guzman | Updates to Validation Plan |

## Table of Contents

**Table of Contents III**

**List of Tables IV**

[Table 1 - Subsystem Leads 1](#_e2omcytb4h5f)

[Table 2 - Applicable Documents](#_gwuto6etz3on) 2

[Table 3 - Reference Documents](#_utr2x6bm9j4v)  3

**List of Figures IV**

[Figure 1 - Conceptual Image of Multi-Axis Cobot 1](#_1t3h5sf)

[Figure 2 - Multi-Axis Cobot Flowchart](#_fmmd1s1ure8n)4

Figure 3 - Multi-Axis Cobot Block Diagram 5

[**1. Introduction**](#_tyjcwt) **1**

[1.1. Purpose and Scope](#_3dy6vkm) 1

[1.2. Responsibility and Change Authority](#_4d34og8) 2

[**2. Applicable and Reference Documents**](#_2s8eyo1) **2**

[2.1. Applicable Documents](#_17dp8vu) 2

[2.2. Reference Documents](#_3rdcrjn) 3

[2.3. Order of Precedence](#_26in1rg) 3

[**3. Requirements**](#_35nkun2) **4**

[3.1. System Definition](#_1ksv4uv) 4

[3.2. Characteristics](#_2jxsxqh) 5

[3.2.1. Functional / Performance Requirements](#_z337ya)5

[3.2.2. Physical Characteristics](#_1y810tw) 6

[3.2.3. Electrical Requirements](#_2xcytpi) 7

[3.2.4. Environmental Characteristics](#_3whwml4) 8

[3.2.5. Failure Propagation](#_trzxr3tfiab2) 9

[**4. Support Requirements**](#_2bn6wsx) 10

[**Appendix A: Acronyms and Abbreviations**](#_qsh70q) **11**

[**Appendix B: Definition of Terms**](#_1pxezwc) **11**

**List of Tables**

[Table 1. Subsystem Leads 1](#_e2omcytb4h5f)

[Table 2. Applicable Documents](#_gwuto6etz3on) 2

[Table 3. Reference Documents](#_utr2x6bm9j4v)  3

**List of Figures**

[Figure 1. Conceptual Image of Multi-Axis Cobot 1](#_1t3h5sf)

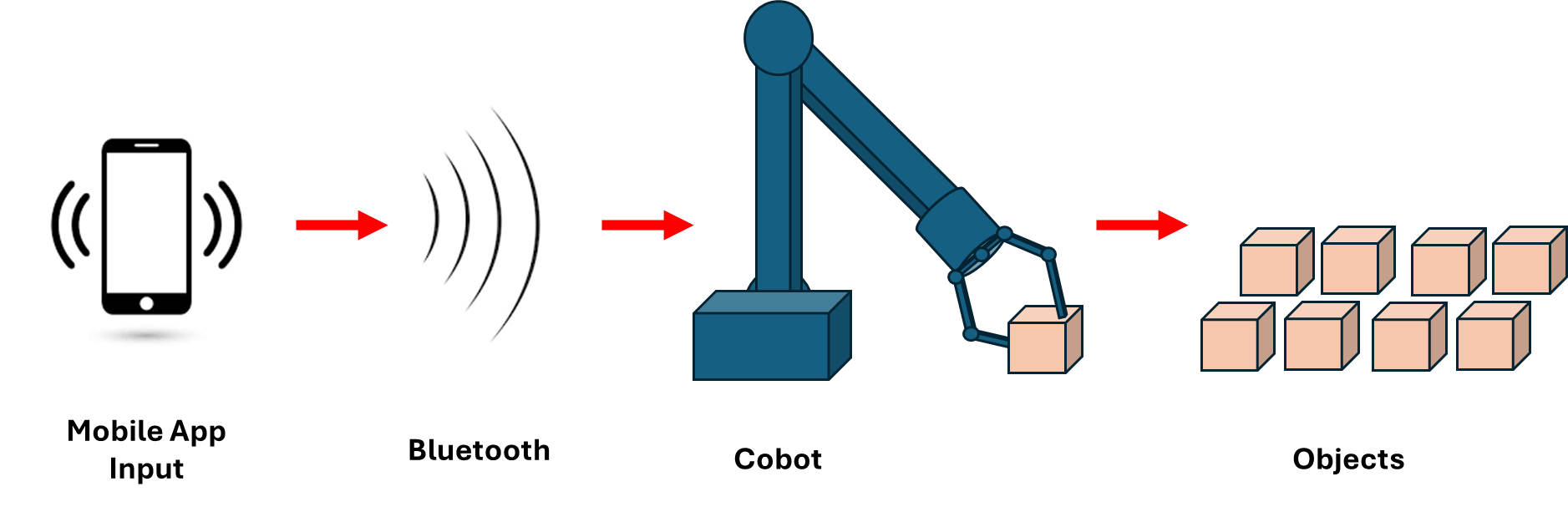
[Figure 2. Multi-Axis Cobot Flowchart 4](#_fmmd1s1ure8n)

# Introduction

## 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

## 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

# Applicable and Reference Documents

## 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

## 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

## 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.

# 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.

## 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.

## Characteristics

### 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.*

**3.2.1.2.1 BLDC Motor Movement**

The BLDC motor is able to spin bi-directionally. Furthermore, this movement is able to be controlled (Direction, Speed, etc...)

*Rationale: Develop software on CCS using TI provided GUI.c code to communicate between the MCU and motor drivers. Code will be flashed onto MCU to facilitate movement.*

#### 3.2.1.3 JTAG Programming of MCU

The MCU is able to be programmed through JTAG. This will allow flashing onto the chip.

*Rationale: The C2000 MCU needs to be capable of programming through JTAG. To enable this, signals such as TDI, TDO, TCK, TMS need to be properly routed.*

#### 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.*

### 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.*

### Electrical Requirements

#### Inputs

#### 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.*

#### 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.*

#### 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.*

#### 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.*

#### 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.*

#### 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.*

#### C2000 MCU Input from ESP-32

The C2000 is able to receive input from ESP-32 via toggling of high/low GPIO pins.

*Rationale: To control the cobot, the ESP-32 will need to send high/low signals to the C2000. The C2000 will then receive these signals to output the required PWM waveform.*

#### 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.*

#### Outputs

#### 3.2.3.2.1 LED Program

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 PWM Phase Output from Motor Drivers

The Motor Drivers can output Phase Signals which will be used to drive the Motors

*Rationale: To be able to actually move a 3-Phase BLDC Motor, there must be 3 separate PWM Phases, each generated by 1 Motor Driver. By showing we can output one Phase Signal from a motor driver, we can say we can theoretically output 3 as well.*

#### 3.2.3.2.4 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.*

### 3.2.3.2.5. Output DC Voltage

The power management and supply PCB will output a DC voltage with minimal noise.

*Rationale: To ensure the cobot components are working correctly, it is important that they are receiving the correct and expected voltage, which is a DC voltage value.*

#### 3.2.3.2.6 MCU Generates PWM Signals

MCU can generate PWM signals from GPIO Pins, these will be sent to the motor driver.

*Rationale: For motor motion, the MCU needs to be capable of generating PWM signals.*

#### 3.2.3.2.7 Handling Load

PCB is able to provide adequate adequate power to cobot when load is present

*Rationale: Using an E-Load, apply a load to the board that mimics the load required of the cobot and ensure that the board is still able to deliver a DC and accurate voltage.*

**3.2.4.2.8 Servo Motor Movement**

Four servo motors on cobot are able to move in their axis accurately

*Rationale: Develop software on CCS to move the servo motors using GPIO signals. Flash the software to the MCU and test the movement of all four motors*

### 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.*

### 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.*

# 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