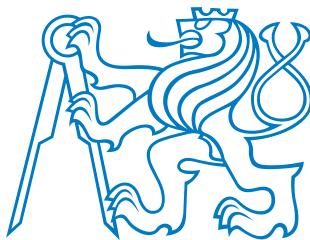


Czech Technical University in Prague

Department of Electromagnetic Field



BACHELOR THESIS

**Two Axis Rotary Platform for Radar
System**

Supervisor of the bachelor thesis: Ing. Viktor Adler, Ph.D

Study programme: Elektronika a komunikace

Prague December 2024

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací a Rámcovými pravidly používání umělé inteligence na ČVUT pro studijní a pedagogické účely v Bc a NM studiu.

V Praze, November 23, 2024

Title: Two Axis Rotary Platform for Radar System

Author: Havránek Kryštof

Department: Department of Electromagnetic Field

Supervisor: Ing. Viktor Adler, Ph.D, Department of Electromagnetic Field

Abstract:

Keywords: Radar platform, Two Axis Platform, Positioning Platform, ESP32, GCode interpreter, Indexing Table

Název práce: Dvojosa rotační plafoma pro radar

Autor: Havránek Kryštof

Katedra: Katedra elektromagnetického pole

Vedoucí bakalářské práce: Ing. Viktor Adler, Ph.D, Katedra elektromagnetického pole

Abstrakt:

Klíčová slova: klíčová slova, klíčové fráze

Contents

Introduction	1
1 Design parameters	2
1.1 Physical capabilities	2
1.2 Software requirements	2
2 Hardware construction	3
2.1 Electronics	3
3 Software realization	4
3.1 Communication layer	5
3.2 Application layer	5
3.3 HAL Layer	5
3.3.1 Performance of HAL Layer	6
Conclusion	8

Introduction

While two axis rotary platforms are fairly common in machining they are often costly and their operational parameters are rarely suitable for radar applications. Aside from laboratory indexing tables, which are rather slow, most platforms aren't designed with communication interface that would facilitate such usecase. Timely execution of movements is not guaranteed and most solutions don't support frequent uplink about current position.

Following thesis is focused on designing and building a two axis rotary platform with such usecase in mind. This mainly entails very timely and deterministic movement of the steppers in order for the radar processing to be able to correctly counteract movement in post processing. However other functionality is also needed in order to limit the burden on the user.

First chapter of the thesis outlines basic design requirements for the platform according to radar parameters. Follow on focuses on Hardware side of the project be it electronic parts used or mechanical design. Last chapter is dedicated to the software handling control of the platform.

1. Design parameters

To begin, it is essential to outline the fundamental requirements for the platform. These stem from the physical capabilities of the radar system and requirements for the software.

1.1 Physical capabilities

The primary constraints on the physical design arise from the radar's radiation pattern (24 and 122 GHz headers), as it is crucial to minimize strong reflections from the platform's structure. According to the 122 GHz Transceiver datasheet, the angular width at -3dB is approximately $\pm 30^\circ$ in both the E-plane and H-plane [1]. With the addition of a radar dome, this value decreases to $\pm 4^\circ$ [2].

The radiation pattern for the 24 GHz microstrip patch antenna is not explicitly provided by the manufacturer. However, based on similar designs, a conservative estimate of $\pm 15^\circ$ is adopted, informed by [3] and [4]. Considering these values, we set a reasonably forgiving clearance limit of $\pm 45^\circ$ in front of the radar.

Due to the radar system's relatively low polling rate (1 Mbit/s), high-speed movement is unnecessary. The manufacturer specifies a maximum update frequency of 50 Hz, equating to a new update every 20 ms [2]. Using the following equation:

$$t_{\text{angle}} = \frac{60}{360 \cdot N_{\text{RMP}}} \cdot \alpha, \quad (1.1)$$

where t_{angle} is time between spend on traveling angle of α in seconds and N_{RMP} is number of rotations per minute, we can calculate that even for low RPM of 60 an angle of 8 degrees platforma traveles (angular width of main lobe for 122 GHz radar) in 10ms.

1.2 Software requirements

Given its widespread adoption as an industry standard for controlling multi-axis machines, G-code is a natural choice for the platform's control software. Beyond the basic functionality typically offered by G-code interpreters, the platform must support additional features to reduce the user's manual control burden. These features include the ability to define movement limits and preprogram sequences of movements for autonomous execution by the platform.

For uplink communication, the platform must provide real-time information about its current position and speed. This data allows the user to make mathematical corrections to the radar's gathered data.

2. Hardware construction

To enable continuous rotational motion, the use of a slip ring is indispensable. Given the radar system's relatively low transmission speed (maximum 1 Mbit/s), a standard contact slip ring suffices – a model with a USB 2.0 interface and eight additional lines was selected to meet the system's needs. The rest of the structure is 3D printed from PLA, since mechanical stresses on the platform are minimal.



(a) 3D render

(b) Photo

Figure 2.1: Form of the final assembly

2.1 Electronics

Electronic side of the project is rather simple given that only control of two stepper motors and having ability to home their position is needed. The system is managed by an ESP32 microcontroller. Since the project does not demand advanced capabilities, a basic ESP32 model is sufficient.

Given the low load on the stepper motors and the platform's inability to accumulate significant momentum, a simple stepper driver without feedback control is adequate. For this purpose, the A4988 stepper driver given its low cost, microstepping capabilities and basic current control[5] was selected..

To implement homing, two potential solutions were considered: Hall effect sensors and optical gates. While Hall effect sensors offer the advantage of angle sensing, allowing correction of any positional drift during operation, they require precise alignment. Since if the orthogonal Hall effect sensor is not perfectly placed in the axis of rotation, calibration becomes necessary [6].

For simplicity and ease of integration, optical gates were selected to enable homing. This decision eliminates the need for complex calibration while providing reliable functionality.

3. Software realization

To maximize efficiency in processing commands and ensure accurate stepper motor control, the program workflow is divided into three distinct layers, as illustrated in Figure 3.1.

The commonly used two-component architecture—where one component handles communication/command parsing and the other manages execution—was deemed unsuitable for this use case. Such an approach would complicate integration of programming interface and require just-in-time processing of commands, which could lead to performance issues.

In the chosen architecture, the degree of abstraction decreases with each successive layer, simplifying processing at each step. This design allows the final layer to operate with maximum efficiency, where transitioning from one command to the next is primarily limited by the inertia of the stepper motors and not by the software.

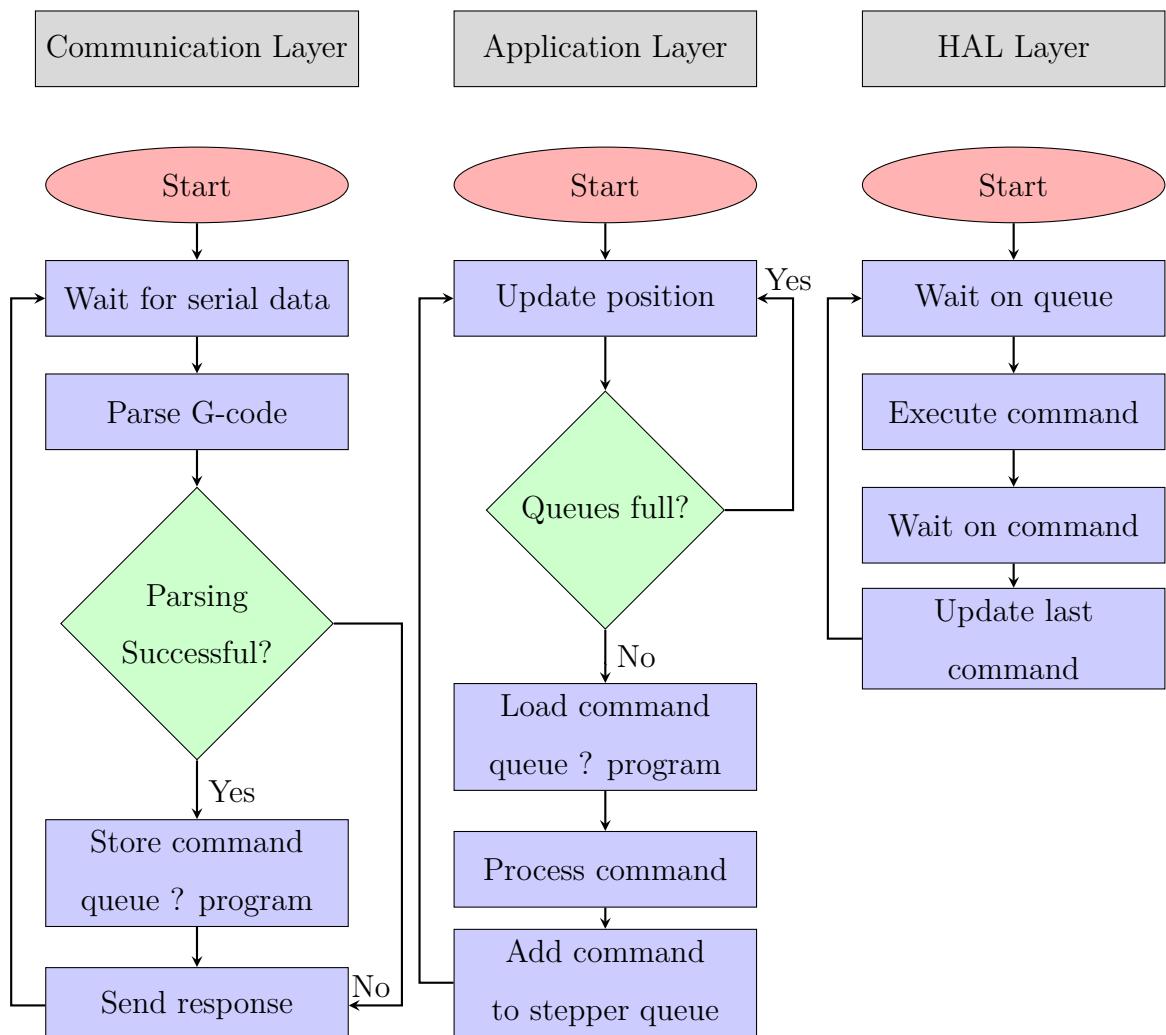


Figure 3.1: Programm diagram

3. Software realization

3.1 Communication layer

The communication layer manages incoming data over the serial line, with efficient handling facilitated with the aid of RTOS queues. Upon receiving data the text string is parsed and either push to queue (if we are declaring a programm) or added to programm declaration.

Immediately after parsing response is send to the user confirming whether the command was parsed correctly or not. However as communication layer does not have a context of all previous commands it is possible that command will be parsed correctly but its execution will fail in the application layer.

3.2 Application layer

The application layer performs two primary functions: tracking the current device position and scheduling commands to be sent to the stepper motors. Aside from tracking current position program also keeps track of the end position of the last scheduled command. Thanks to this the application layer can handle calculations necessary for absolute positioning and enforce movement limits.

A significant departure from standard G-code interpreters is the platform's handling of single-axis move commands. If a move command targets only one axis, the other axis remains free to read the next command and begin execution. If this behavior is undesirable, the user must issue commands for both axes. In relative positioning mode, a zero value will result in no motion; in absolute positioning mode, the command must specify the current position for no movement to occur.

A key departure from standard G-code interpreters is how the platform handles single-axis move commands. When a move command targets only one axis, the other axis remains free to read the next command and begin execution. If this behavior is undesirable, the user must issue commands for both axes. In relative positioning mode, a zero value results in no motion; in absolute positioning mode, the command must specify the current position to prevent movement.

This behavior is a necessary side effect of the spindle regime, which typically cannot be toggled on or off dynamically. Another consequence is the requirement for separate positioning modes for each axis. Continuous rotation prevents the calculation of a move's end position, making it impossible to pre-schedule absolute positioning commands.

3.3 HAL Layer

The final layer manages stepper motor control and provides the application layer with essential data for position calculations. In its loop, the program waits for the next command in the stepper queue. Upon receiving a command, it sets up execution, waits for one or both steppers to complete their movement, and then proceeds to the next command. Since limit and absolute positioning calculations are handled in the application layer whole routine remains highly efficient.

The main challenge lies in generating precise PWM signals and stopping signal

3. Software realization

generation after a specific number of steps. Using the equation:

$$t_{\text{delay}}(s) = \frac{60}{2 \cdot N_{\text{steps}} \cdot s}, \quad (3.1)$$

where s is speed in RPM, N_{steps} is the number of steps, and t_{delay} is the time between steps, we calculate that even at 30 RPM, the delay between output changes is 5 ms per step. With microstepping at a 2:1 ratio, this reduces to 2.5 ms – faster than lowest sleep interval and without sleep watchdog will trigger. Therefore, signal generation must leverage specialized microcontroller peripherals.

The ESP32 platform offers two options: Remote Controlled Transceiver (RMT) and Motor Control Pulse Width Modulation (MCPWM) combined with Pulse Counter (PCNT). While RMT allows smooth PWM frequency adjustments, it has several drawbacks. These include: generating a specific number of pulses is supported only on newer ESP32 models [7], synchronization is restricted to its proprietary API, and there is no straightforward way to track progress during a move [8].

For these reasons, MCPWM and PCNT were chosen. MCPWM handles pulse generation, while PCNT counts steps, enabling easy synchronization, continuous rotation, and a robust API for step tracking [9]. The only limitation is the PCNT’s 15-bit counter, which caps the maximum steps per move at 32,767.

3.3.1 Performance of HAL Layer

Table 3.1 illustrates the stability of PWM generation by the MCPWM module at various speeds. Measurements were conducted using a Saleae Logic Pro 16 logic analyzer, with no microstepping enabled.

The results show that frequency deviation is minimal, though the generated speed is consistently marginally faster than the target, and the error increases slightly with higher speeds. Nevertheless, for 24,000 steps at 120 RPM, the relative error in time taken would be only $\epsilon = -0.004\%$, demonstrating excellent accuracy.

Table 3.1: Stability of PWM generation

RPM	f_{desired} (Hz)	f_{low} (Hz)	f_{high} (Hz)	f_{avg} (Hz)
10	33.334	33.334	33.334	33.334
30	100	100	100.003	100.002
60	200	200	200.01	200.004
120	400	400	400.02	400.007

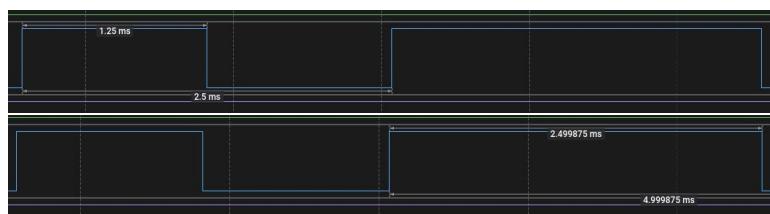


Figure 3.2: Moment of change between commands (120RPM \Rightarrow 60RPM)

3. Software realization

An attempt was made to measure the speed of switching between commands, as shown in Figure 3.2. The results indicate that the delay between commands is imperceptible, with similar outcomes observed for other command combinations.

This demonstrates the efficiency of the HAL layer in managing stepper motor control and transitioning seamlessly between commands. As long as the stepper queues are supplied with commands in advance, the platform can operate without noticeable interruptions. Most importantly, the platform's timely and predictable behavior ensures that mathematical corrections to the radar data can be applied accurately.

Conclusions

Bibliography

- [1] INDIE SEMICONDUCTORS GMBH. *TRX_120_001: Datasheet* [online]. 2021. Version 1.6 [visited on 2024-11-19]. Available from: https://siliconradar.com/datasheets/Datasheet%5C_TRX%5C_120%5C_001%5C_V1.6.pdf.
- [2] INDIE SEMICONDUCTORS GMBH. *SiRad Easy® r4, SiRad Easy® & SiRad Simple®: User Guide* [online]. 2021. Version 2.5 [visited on 2024-11-19]. Available from: https://downloads.ffe.indiesemi.com/datasheets/User_Guide_Easy_Simple_V2.5.pdf.
- [3] JIN, Hao; ZHU, Lijia; LIU, Xu; YANG, Guangli. Design of a Microstrip Antenna Array with Low Side-Lobe for 24GHz Radar Sensors. In: *2018 International Conference on Microwave and Millimeter Wave Technology (ICMWT)* [online]. IEEE, 2018, pp. 1–3 [visited on 2024-11-19]. ISBN 978-1-5386-2416-6. Available from DOI: [10.1109/ICMWT.2018.8563259](https://doi.org/10.1109/ICMWT.2018.8563259).
- [4] LAN, Shengchang; XU, Yang; CHU, Hongjun; QIU, Jinghui; HE, Zonglong; DENISOV, Alexander. A novel 24GHz microstrip array module design for bioradarars. In: *2015 International Symposium on Antennas and Propagation (ISAP)*. IEEE, 2015, pp. 1–3. ISBN 978-4-8855-2302-1. Available also from: <https://ieeexplore.ieee.org/document/7447491>.
- [5] *A4988 Datasheet* [online]. [visited on 2024-11-23]. Available from: <https://www.allegromicro.com/~/media/Files/Datasheets/A4988-Datasheet.ashx>.
- [6] CHEN, Xiao; HE, Xiran; ZHANG, Ying; GAO, Bo. Angle Measurement Method Based on Cross-Orthogonal Hall Sensor. In: *2023 IEEE 16th International Conference on Electronic Measurement & Instruments (ICEMI)* [online]. IEEE, 2023-8-9, pp. 153–158 [visited on 2024-11-23]. ISBN 979-8-3503-2714-4. Available from DOI: [10.1109/ICEMI59194.2023.10270587](https://doi.org/10.1109/ICEMI59194.2023.10270587).
- [7] *RMT Based Stepper Motor Smooth Controller* [online]. [visited on 2024-10-03]. Available from: github.com/espressif/esp-idf/tree/master/examples/peripherals/rmt/stepper_motor.
- [8] *Remote Control Transceiver (RMT)* [online]. [visited on 2024-10-03]. Available from: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/rmt.html>.
- [9] *Pulse Counter (PCNT)* [online]. [visited on 2024-11-09]. Available from: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/peripherals/pcnt.html>.

List of Figures

2.1	Form of the final assembly	3
3.1	Programm diagram	4
3.2	Moment of change between commands with 120RPM and 60RPM	6

List of Tables

3.1 Stability of PWM generation	6
---	---