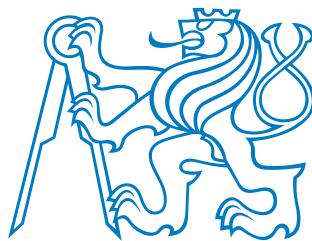


Czech Technical University in Prague

Department of Electromagnetic Field



BACHELOR THESIS

Surveillance FMCW Radar

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

Study programme: Elektronika a komunikace

Prague TODO 2024

I hereby declare that I have independently written the submitted work and that I have cited all sources of information used in accordance with the Methodological Guidelines on Adherence to Ethical Principles in the Preparation of University Theses and the Framework Rules for the Use of Artificial Intelligence at CTU for Study and Educational Purposes in Bachelor's and Master's Studies.
In Prague, January 20, 2025

Title: Surveillance FMCW Radar

Author: Havránek Kryštof

Department: Department of Electromagnetic Field

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

Abstract: TODO

Keywords: TODO

Název práce: Přehledový FMCW radar

Autor: Havránek Kryštof

Katedra: Katedra elektromagnetického pole

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

Abstrakt: TODO

Klíčová slova: TODO

Contents

Introduction	1
1 Basics of FMCW Radar	2
2 SiRad Easy[®]	3
2.1 24 GHz Header Simulation	3
3 Rotary plafrom	4
3.1 Platform Design Parameters	4
3.1.1 Physical Capabilities	4
3.1.2 Software Requirements	4
3.2 Platform Construction	5
3.2.1 Platform Electronics	5
3.3 Platform Software Realization	6
3.3.1 Communication layer	6
3.3.2 Application layer	6
3.3.3 HAL Layer	7
Conclusion	10

Introduction

TODO

1. Basics of FMCW Radar

TODO: Consider some semi-introductory chapter before this one. Also the name "Basics of FMCW Radar" is not very good.

2. SiRad Easy[®]

TODO: Consider whether to have it as a chapter or just section under chapter dealing with FMCW technology itself.

2.1 24 GHz Header Simulation

3. Rotary plafrom

Following chapter outlines design process and operation of a rotary platform specifically designed for SiRad Easy[®]radar system.

3.1 Platform 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, SiRad Easy evalutions kit, and requirements for the software.

3.1.1 Physical Capabilities

TODO: Should be more tied to the radar system - especially in regard to the radiation pattern of 24 GHz header.

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

One can also see that significant cost cutting can be done in the area of precision of the platform. As the radar system has relatively low angular resolution the sub degree control of commercial solutions is not needed. This allows basic 200 steps NEMA17 stepper motors be used with optional microstepping to increase smoothness of movement.

In terms of speed, due to the radar system's relatively low polling rate (1 Mbit/s [2]), 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:

$$t_{\text{angle}} = \frac{60}{360 \cdot N_{\text{RMP}}} \cdot \alpha, \quad (3.1)$$

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 the platform travels (angular width of main lobe for 122 GHz radar) in 10 ms.

3.1.2 Software Requirements

Given its widespread adoption as an industry standard for controlling multi-axis machines, G-code over serial is a natural choice for the platform's communication format. Beyond the basic functionality typically offered by G-code interpreters,

3. Rotary plaffrom

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.

3.2 Platform Construction

TODO: Add more pictures, list out dimensions, little bit more info about 3D print.

To enable continuous rotational motion, the use of a slip ring is indispensable. Given the radar system's relatively low transmission speed, 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 3.1: Form of the final assembly

In stark departure from commercial solutions [5, 6] due to need to accommodate a large slipring first axis controls rotation and second one tilt. This arrangement also leads to simpler calculation of speed vector, needed for radar processing, by reducing the interdependence of movements between each axis.

3.2.1 Platform Electronics

TODO: total rewrite of this section, drop things about Hall effect, outlay a basic schema of what components are present and how are they connected. Keep it rather short if possible

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.

3. Rotary plafrom

Given the low load on 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 was selected, due to its low cost, microstepping capabilities and basic current control [7].

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. If the orthogonal Hall effect sensor is not perfectly placed in the axis of rotation, non simply calibration becomes a necessity [8].

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

3.3 Platform Software Realization

NOTE: Basic structure of this chapter is fine.

To maximize efficiency in processing commands and ensure accurate stepper motor control, the program workflow is divided into three distinct layers, as illustrated by figure 3.2.

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 transition from one command to the next is primarily limited by the inertia of stepper motors and not by the software.

3.3.1 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 pushed to a queue or added to programm declaration, in case we are currently declaring program.

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

3.3.2 Application layer

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

3. Rotary plafrom

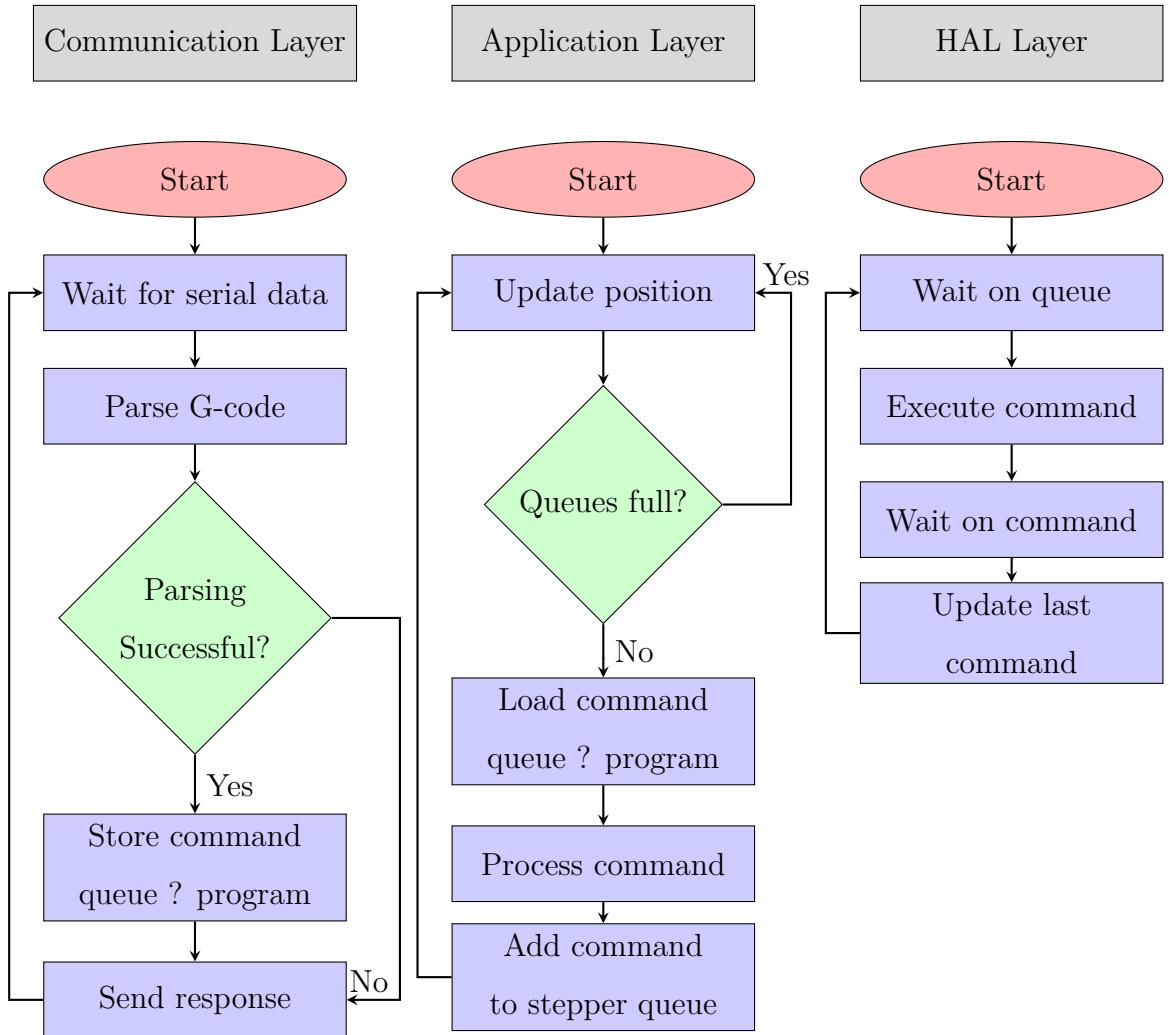


Figure 3.2: Programm diagram

A key departure from standard G-code interpreters, like [9], is how the platform handles single-axis move commands. When a move command targets only one axis, the other axis remains free to read next command and begin its 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 calculations of a move's end position, making it impossible to make calculation for absolute positioning commands – thus necessitating relative positioning. However it would be rather restrictive to force user to relative positioning on second axis, therefore the independent positioning settings.

3.3.3 HAL Layer

TODO: Drop primitive equation which is totally unnecessary and was only included to comply with the original assignment

3. Rotary plafrom

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 (Used to control stepper motors drivers.) and stopping signal generation after a specific number of steps. Using the equation:

$$t_{\text{delay}}(s) = \frac{60}{2 \cdot N_{\text{steps}} \cdot s}, \quad (3.2)$$

where s is speed in RPM, N_{steps} is the number of steps (Anywhere from 200 to 1600 depending on microstepping.), 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 on ESP32 and without sleeping the RTOS 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. Such as the fact that generating a specific number of pulses is supported only on newer ESP32 models [10], synchronization is restricted to its proprietary API, and there is no straightforward way to track progress during a move [11].

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 [12]. The only limitation is the PCNT's 15-bit counter, which caps the maximum steps per move at 32,767.

Performance of the HAL Layer

TODO: Including Measurement is not a bad idea however the table is totally redundant and doesn't provide much.

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, when measuring time of 24,000 steps at 120 RPM, the relative error in time duration (or speed) was only $\epsilon = -0.004\%$, demonstrating excellent accuracy.

An attempt was made to also measure the delay between switching commands, displayed in figure 3.3. The results indicate that the delay between commands is imperceptible. Similar outcomes were also 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 stepper

3. Rotary plafrom

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

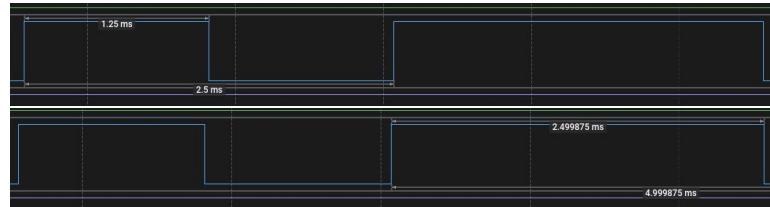


Figure 3.3: Moment of change between commands (120RPM \Rightarrow 60RPM)

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.

Conclusion

TODO

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_TRX_120_001_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 (ICMMT)* [online]. IEEE, 2018, pp. 1–3 [visited on 2024-11-19]. ISBN 978-1-5386-2416-6. Available from DOI: [10.1109/ICMMT.2018.8563259](https://doi.org/10.1109/ICMMT.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] *Two Axis Rotary Indexing Tables* [online]. [visited on 2024-11-24]. Available from: <https://www.carlhirschmann.us/en/Rotary-tables/Products/Two-Axis-Rotary-Indexing-Tables>.
- [6] *2-axis (XY) Rotation Stage Motorized XYZ Translation Stage* [online]. [visited on 2024-11-24]. Available from: https://www.standa.lt/products/catalog/custom_engineering?item=658.
- [7] *A4988 Datasheet* [online]. [visited on 2024-11-23]. Available from: <https://www.allegromicro.com/~media/Files/Datasheets/A4988-Datasheet.ashx>.
- [8] 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).
- [9] *GCode dictionary* [online]. [visited on 2024-11-24]. Available from: https://docs.duet3d.com/en/User_manual/Reference/Gcodes.
- [10] *RMT Based Stepper Motor Smooth Controller* [online]. [visited on 2024-10-03]. Available from: https://github.com/espressif/esp-idf/tree/master/examples/peripherals/rmt/stepper_motor.
- [11] *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>.
- [12] *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

3.1	Form of the final assembly	5
3.2	Programm diagram	7
3.3	Moment of change between commands with 120RPM and 60RPM	9

List of Tables

3.1 Stability of PWM generation	9
---	---