

**Czech Technical University in Prague**

Department of Electromagnetic Field



**BACHELOR THESIS**

# **Two Axis Positioning 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 18, 2024

---

Title: Two Axis Positioning 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: Dvojosá polohovací platforma 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

<b>Introduction</b>	<b>1</b>
<b>1 Design parameters</b>	<b>2</b>
1.1 Physical capabilities . . . . .	2
1.2 Software requirements . . . . .	2
<b>2 Hardware construction</b>	<b>3</b>
<b>3 Software realization</b>	<b>4</b>
3.1 Communication layer . . . . .	4
3.2 Application layer . . . . .	4
3.3 HAL Layer . . . . .	5
<b>Conclusion</b>	<b>6</b>

---

# Introduction

# 1. Design parameters

First of all it is necessary to outline basic requirements we have for the platform. We need to Accommodate SiRad Easy<sup>®</sup> r4 from Radar company and provide control over its tilt and rotation in horizontal plane. Transmission of the data from the radar ought not to be impeded and platform itself must provide sufficient data about its movement. As the radar data processing will be handled in MATLAB the platform needs to provide interface that is accessible from MATLAB, serial line comes up as the most straightforward option.

## 1.1 Physical capabilities

Main design limitations on physical pattern stem from radiation pattern of radar used as we want to prevent reflection from the structure.

## 1.2 Software requirements

NOTE: only basic functionality we want – programming, spindle, limits and such, point out solution needs to be rather generic

TODO -> move to realization Main problem on software side is to provide smooth excitation of stepper motors so that the motion is continuous and with predictable speed. Given simple equation

$$t_{\text{delay}}(s) = \frac{60}{N_{\text{steps}} \cdot s}, \quad (1.1)$$

where  $s$  is speed in RMP,  $N_{\text{steps}}$  number of steps of the stepper and  $t_{\text{delay}}$  time between steps the time we can calculate that even for relatively low speed of 60 RPM delay between steps is only 5ms, take into account we need to switch state two times we get 2500 $\mu$ s between changes. This makes software generation of control signal hard to do.

In order to provide reliable and timely control over platform's movement executing commands in real time as they come wouldn't suffice. Thus it is needed to queue general commands and for more complex movement have a way to pre-programm them into the platform and let it do the execution itself.

## 2. Hardware construction

## 3. Software realization

In order to achieve maximal efficiency in processing commands and most importantly maintain accurate driving of stepper motors workflow of the program is split into three diactinic layers, as showed on diagram 3.1. With increasing layer the amount of abstraction in command decreases and processing gets easier. There are two main benefits of this approach:

- HAL layer function can be made really simple and thus fast.
- Synchronization with queues enables the precompute many commands ahead
  - thus executing next move is as fast as possible.

TODO: point out while two layer architecture wasn't possible here - wouldn't support programming interface, would require super long buffer for serial communication

### 3.1 Communication layer

Communication layer handles incoming data on serial line, reading from which is realized efficiently 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 data are processed response is send to a user as Communication Layer doesn't check whole pipeline of commands receiving SUCCESS response doesn't mean command will be valid in context it was sent.

### 3.2 Application layer

Application layers does two main things - keeps track of current device position and schedules commands to be sent to steppers. As part of scheduling layer calculates end positions of future commands this enables it to enforce user define limits on movement.



### 3. Software realization

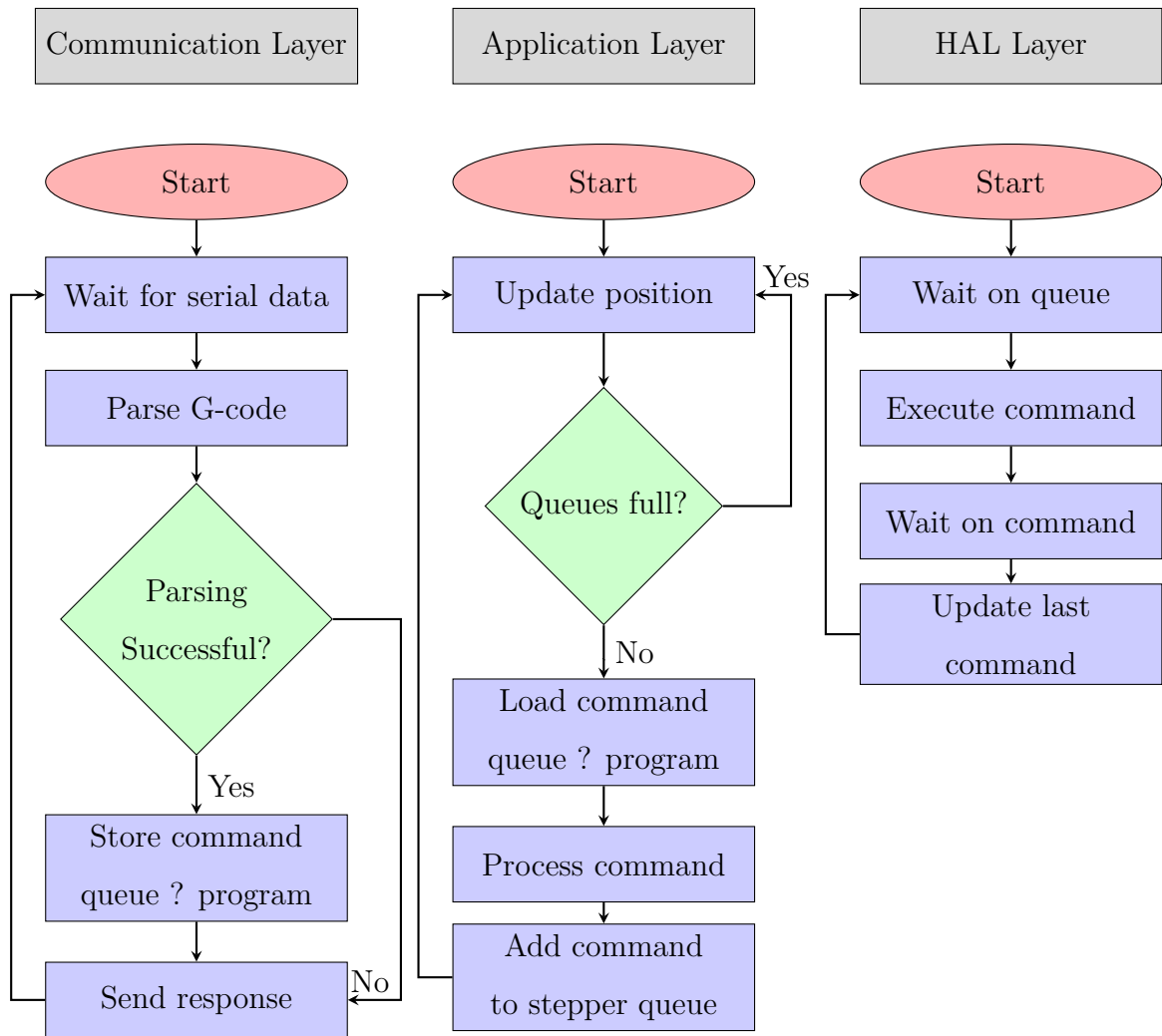


Figure 3.1: Programm diagram

### 3.3 HAL Layer

Final layer handles basic driving of steppers and provides Application layer with all necessary for position calculation. In its loop we simply wait on next command in queue than setup its execution after which we wait for stepper or both steppers to finish moving.

Main problem arises with generation of accurate PWM signal and having ability to stop generation upon given number of steps wer taken.

---

# Conclusions

---

# List of Figures

3.1 Programm diagram . . . . .	5
--------------------------------	---

---

# List of Tables