Voluntary Project

SyArm Mk1

A simple robotic arm

By

**Samuel Nösslböck**

Started

April 2022

# Credits

At this point I personally want to express my thanks to all people who made this thing possible. As a lot of time and effort of this project was spent during school, I want to thank all teachers who supported me actively or did not force me to fully participate in their lessons.

Special thanks belong to my teacher for machine elements Mr. Dipl.-Ing. Manuel Leibetseder, Mr. Dipl.-Ing. Gottfried Preuer and Mr. Dipl.-Ing. Peter Rachinger. First mentioned even answered calls on the weekend, even though he had way more important things to take care about and always had good advice for most of my problems.

Second, I want to mention my incredible friends, they always listen to my personal problems coming along with such a project. Some of them even tried to listen to some details and always showed interest.

Again, I want to point out a few people: First Tobias Niedermayr, a very talented young man who is way smarter he wants to admit he is. He always showed a high interest in my projects and gave me the motivation the continue when my motivation was more than down.

Second there is my long year best friend Fabio Muratore. He just knows how to make me laugh, have a good time and really get my head out of my projects from time to time, which is a very important thing as I learned in the process. Finally, my dear friend Laura Taubinger, who always cared for my health and reminded me when to take breaks.

# Abstract

Many manufacturing tasks require a series of complex work or transport processes, which can be difficult to build a machine for. Robotic arms are a very flexible type of robots that can perform a lot of those complex tasks and can be re-programmed for every new application needed.

The SyArm is a simple and compact version of such a robot, not built for heavy loads or accuracy, but for speed and the basic concept. Its materials are cheap, simple and were mainly chosen by the factor of how easy it is to acquire them.

The whole project can be seen as a kind of learning process, that is the reason why you will find version numbers of three and above. Many of the hand-drawn sketches have very only concepts drawn onto them.

Contents

[Credits 2](#_Toc119760111)

[Abstract 3](#_Toc119760112)

[1. Introduction 6](#_Toc119760113)

[1.1. Motivation 6](#_Toc119760114)

[1.2. Goals 6](#_Toc119760115)

[2. Basics 7](#_Toc119760116)

[2.1. Robotic Arms 7](#_Toc119760117)

[2.2. CNC Machines 7](#_Toc119760118)

[3. Construction 8](#_Toc119760119)

[3.1. Base 8](#_Toc119760120)

[3.2. Arm 8](#_Toc119760121)

[3.2.1. First Segment 8](#_Toc119760122)

[3.2.2. Second Segment 8](#_Toc119760123)

[3.3. Axial Bearing 8](#_Toc119760124)

[3.4. Tools 8](#_Toc119760125)

[4. Electronics 9](#_Toc119760126)

[4.1. Controller 9](#_Toc119760127)

[4.2. Measurements 9](#_Toc119760128)

[4.3. Tool supplies 9](#_Toc119760129)

[5. Software 10](#_Toc119760130)

[5.1. Stepper library 10](#_Toc119760131)

[5.1.1. Acceleration curves 10](#_Toc119760132)

[5.1.2. G-Code interpreter 10](#_Toc119760133)

[5.2. SyArm library 10](#_Toc119760134)

[5.2.1. Modelling 10](#_Toc119760135)

[5.2.2. Movements and actions 10](#_Toc119760136)

[5.3. Service 10](#_Toc119760137)

[6. Conclusion 11](#_Toc119760138)

[7. Appendix 12](#_Toc119760139)

[7.1. Technical drawings 12](#_Toc119760140)

[7.2. Articles and other self-made references 12](#_Toc119760141)

[7.3. Sketches and drawings 12](#_Toc119760142)

[8. Sources and references 13](#_Toc119760143)

# Introduction

## Motivation

Robots always fascinated me in many ways: The way they are built, the way the software is made or simply the enormous tasks they can complete nowadays. As I had almost no lessons in school attending robotics and other related topics, so I have decided to learn it myself by building one.

I personally believe automation and especially robots are one of the most effective ways to fight poverty. When used right, they can get people out of jobs with miserable working conditions and accelerate technical development especially in low-wage countries.

Building this robot was kind of a first step towards helping people with the field of research I like.

## Goals

The goal is to build a fully functional robotic arm (see section “Construction”) that can equip multiple different tools for tasks like drawing, lifting things around and so on. Other things a high value was put on are the stability against oscillations that will cause an inaccuracy in all movements and paths.

Flexibility is the overall term this construction was designed after, in terms of hardware and software as well. The software has been packed into libraries and separated from this specific robot as much as possible, so it can be used of other types of robots too.

## Structure

The naming and versioning system with “Mk X” makes it easy to differentiate complete reworks of the same idea from each other with its basic principle being the same as for other versioning systems.

*Version: X.Y.Z => Mk X, Version Y.Z*

# Basics

## Robotic Arms

## CNC Machines

# Construction

The whole construction is separated into five main parts listed in the following chapters.

## Base

## Arm

### First Segment

### Second Segment

## Axial Bearing

## Tools

# Electronics

## Controller

## Measurements

## Tool supplies

# Software

## Stepper library

### Acceleration curves

### G-Code interpreter

## SyArm library

### Modelling

The main purpose of modelling is calculating loads and inertias for the motor in order to calculate the right acceleration curves. If the model results in giving incorrect values to the controls this could lead to motors moving too slow, not in the right direction or not at all. On the other hand, it could lead to a low efficiency due to the motors not using their full potential.

**Inertia**

To keep things simple, every part of the robot can be seen as a rod, represented by a certain mass, a positional vector and a representing vector

### Movements and actions

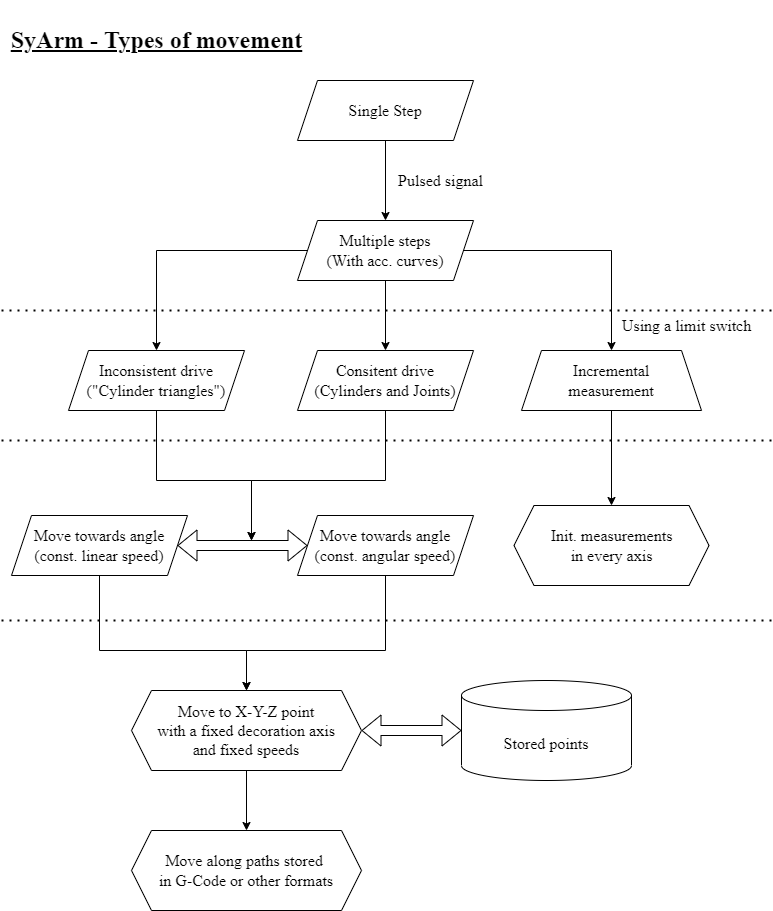
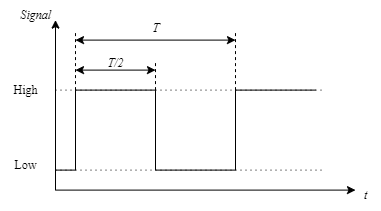
The movements of the robot can be divided into different layers and kinds:

Figure 5.2.2.1

This flowchart shows the general library structure, which features the shown actions and movements that get more and more complex the farther one goes to the bottom. Each action will now be described in detail.



**Single step**

A single step with a single motor, defined by the steptime . The control keeps the signal up for half the desired steptime, therefore it’s low the other half of said time.

The direction is changed by inverting the directional pin of the control, not by modifying the step pulse itself.

Figure 5.2.2.2

**Multiple steps (with curves)**

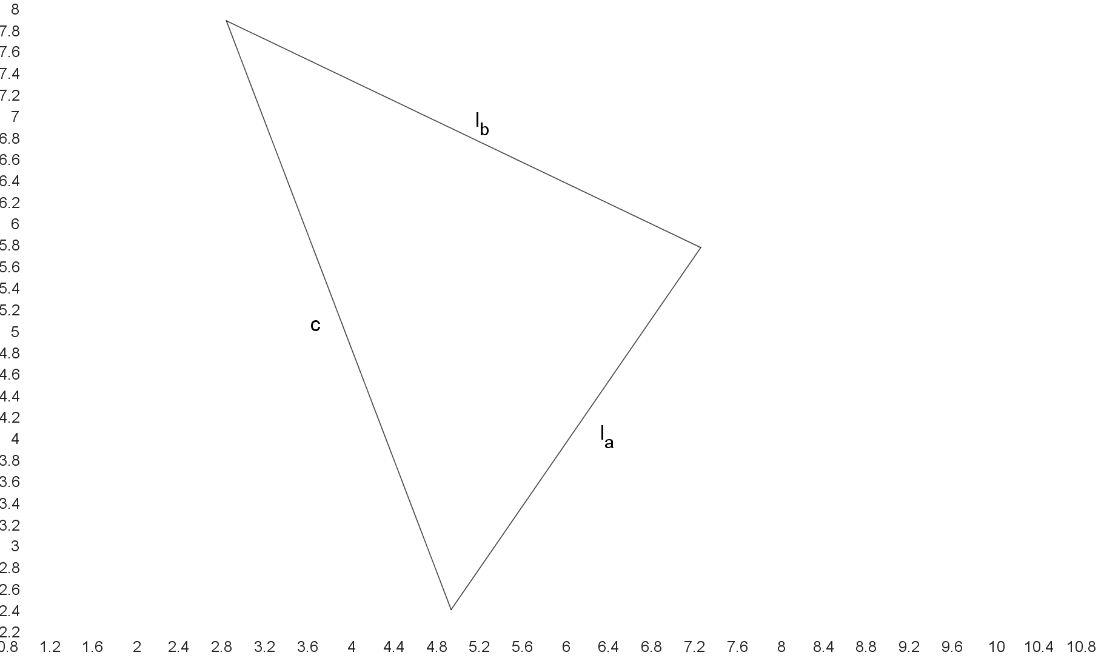
Moves a stepper motor multiple steps into the defined direction with proper acceleration and deceleration considering the inertia of the construction and motor torque. A full explanation of acceleration curves and different methods can be found in the appendix[[1]](#footnote-1).

The two relevant functions are then defining a series, that look like the following:

The motors then accelerate until the certain speed is reached or half the steps are travelled, after this distance, the control will mirror the curve at half the steps travelled.

**Consistent drive (Used for cylinders or geared joints)**

The consistent drive mode converts an angle into the right direction and steps that need to be travelled, using a predefined gear ratio or spindle pitch.



**Inconsistent drive (Used for “cylinder triangles”)**

The inconsistent drive is used to keep a certain omega up when a cylinder is built in a triangular construction like the one displayed on the right. The law of cosines gives an angle or the cylinder length if either one of the variables is given.

Considering the change in the cylinder extent is non-linear proportional to the angle and depending on the time the following differential emerges:

When now a constant angular velocity is desired, the angular acceleration ends up at being zero, which makes the second derivative simpler.

With the formular displayed above the velocity correction of the cylinder can be calculated, which leads to a consistent angular velocity.

**Incremental measurement**

The robot should always know it’s exact position, which is why a combination of stepper motors, and an initial incremental measurement system are used. When running this measurement operation, the motors move into the defined direction until the controls recognize a HIGH signal on the defined measurement input pin (one for each motor). The positional value is then set to another predefined value, the initial position value.

Additionally, the controls set a limit in the direction on the initial position value, as it would overextend and damage the construction.

**Moving to a specific angle (with or without constant speed)**

The certain movements then are made by taking the current position and calculating the difference to the desired angle .

**Moving to a specific X, Y and Z coordinate point**

To move to a specific position in space expressed by a X, Y and Z coordinate, the control must convert such a point into angles for the motors to move to.

The base angle can be determined with ease as it is the only angle that performs a rotation around the Z-Axis. Looking at the robot from top down can be used to calculate the base angle as following:

The base angle is shifted by ninety degrees as the base arm is orientated along the Y-axis in order to perform correct rotation operations around the X-axis. This base angle can now be used to rotate the point onto the Y-Z plane.

Now the decoration vector can be subtracted without applying the rotation matrix to it. In order to finally simplify the point into a triangle that is fully defined, the base vector must be subtracted from the rotated point as well.

This point now stands for the distance the first two arm segments must reach. Their exact angles are calculated as following:

As displayed in the formula, can be expressed as the difference between the decoration angle and the angle left by the first two segments when moving to the required point .

## Service

# Conclusion

# Appendix

The whole project can be found on a public GitHub repository, either search the name (“SamuelNoesslboeck/SyArm\_Mk1”) or the QR-Code displayed on the right.

## Technical drawings

## Articles and other self-made references

## Sketches and drawings

# Sources, references and figures

**Images and Figures**

5.2.2.1. Library, movements and actions structure (self-made)  
5.2.2.2. Steptime and pulses diagram (self-made)

**Used software libraries (cargo rust)**

See the cargo.toml files for used versions!

* glam, vector calculation library
* gpio, raspberry pi IO library

1. Article “Stepper Motors” [↑](#footnote-ref-1)