

# **Week 1**

## **Goal**

The general goal for the project is to make a robotic arm with a single joint operating in a polar plane. Such design involves two degrees of freedom:

1. for the rotation of the arm
2. for the length of the arm.

My proposal is making incremental improvements on the design in order to make the arm more modular each time. Rough scratch would be:

1. An arm that can move in the polar plane, with a simple API for moving to a specific point, moving along a line specified by two points, along a circle or desired radius etc.
2. Making it do a single thing - drawing on the paper is my idea for this step as it seems like a natural progression and should not be too hard to add on top of previous step.
3. Remaking the arm part into the one with a modular end and reimplementing the previous step so it fits the newly constructed arm.
4. From this point on we should be able to make whatever we want. We may want to integrate it with some computer vision software so it can do more complex tasks like grabbing objects, solving sudoku, playing chess etc.

## **Project details**

### **Parts**

First thing we need to decide on is the type of microcontroller to use. My proposal is to go with [Arduino Uno](#) - we are both familiar with using it, it is fairly cheap and there are a lot of resources online. On the contrary, going with the cheap parts means sacrificing on the precision and reliability.

Next thing is motors - I think the must here are servo motors. There are a lot of them on the market in all price ranges; we need to do some more research on the specific models to use taking into account the precision, price, power they can produce and compatibility with the chosen microcontroller and other parts. It is important to keep the weight of the arm part of the robot fairly light in order for the servo to keep up.

For the custom parts 3d-printed ones seem like the best bet. There may be printer available on the faculty, but the printing should not be that expensive if that's not the case. I am open for other options, but they do not seem viable once we get to step 3 since the parts will inevitably become more complex.

## **Software**

If we go with Arduino, Arduino software seems pretty good. Arduino basically uses C++ with its custom [arduino-cli](#) for compiling, testing and vendoring the software onto a chip.

C++ generally seems like a nice options since it has a broad library selection, like [OpenCV](#) for computer vision later.

We should also learn some 3d modeling in [Blender](#) or an alternative.

Other than that I am using [typst](#) to type these documents, and we have setup a [Github repository](#) for collaboration.

## **References for future research**

- [What Does Robotics Polarity Mean](#)
- [What is Polar Robot? Explanation, Applications & Advantages\)](#)
- [Polar coordinates robotic arm](#)