Creation of A Remote Controlled Water Turret

M. Bourmistrov[a], E.K.Endresen[a]

[a] *Institutt for fysikk, Norges Teknisk-Naturvitenskapelige Universitet, N-7491 Trondheim, Norway.*

Abstract

This project is a part of an overall project carried out in TFY4190 at NTNU to teach the participants about graphical programming and data communication. Our solution composed of building a remote controlled water turret. We discovered that our design was flawed, and could not rotate 360 degrees. The water turret did not work as planned on the showcase due to loose wiring.

1. Introduction

In the subject ”TFY4190 Instrumentering” we are given the opportunity to choose a project freely, as long as it meets some criteria. It has to have a purpose, and has to be controllable either through user input or onboard measurement systems. One can use any programming language to achieve these criteria. We choose to create a water turret controlled by user inputs through an XBox controller. Our aim is that the robot will be able to shoot water in all directions and lengths given that we use 2 motors to control the cannon in the xy- and yz-plane. The components required for this build will be made by us through 3D-printing.

2. Theory

The pump used to shoot water has limited strength. Therefore to shoot the water an adequate distance, we have to consider the continuity equation:

$$A_{in}v_{in} = A_{out}v_{out} \tag{1}$$

$$v_{out} = \frac{A_{in}v_{in}}{A_{out}} \tag{2}$$

As we can see, to increase the speed of the water we can reduce the area the water goes through. Therefore we use a nozzle.

The motor used to turn in the xy-plane is a stepper motor. Unlike other DC-motors, a stepper motor does not continuously rotate for an arbitrary number of times until the voltage supply is cut. A stepper motor works more or less like a digital input-output device, where the current hits a series of coils arranged in phases. This is to separate the rotation into smaller rotation for precise control.

To understand how we have to send signals to the stepper motor, one has to understand the insides of the motor RS, 2023. A stepper motor consists of 6 stator teeth, each with a coil. The rotator sylinder has only 4 rotor teeth. This is so that only 2 rotor teeth are aligned with the stator teeth at a time. By energizing the stator teeth at different intervals, one creates a magnetic field within the motor. Opposite stator teeth will pull the rotor teeth into place. When it rotates, one set of stator teeth turns of, while the other turns on. In this way you can turn the sylinder an accurate 30 degrees each time. However, through energizing 2 sets at once, you can make the rotation degree as low as 15 degrees.

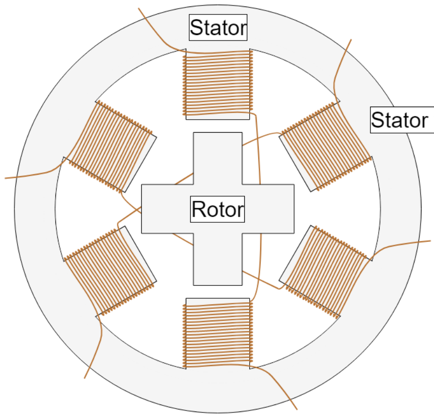


Figure 1: Sketch of inside of stepper motor MPS, n.d.

Yet, we have to take into consideration the induction created by the different coils. If we ignore this, the stepper motor can experience heating problems and malfunction. To address this, we use a motordriver (see Figure (4)). This chip lets us send digital signals to the stepper motor for the correct stator teeth while also allowing us to turn of the coils not in use on the others.

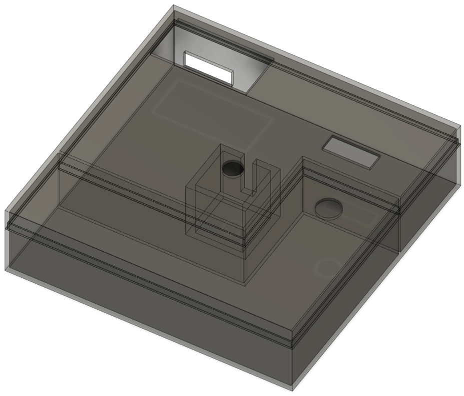
3. Method

Below is a table of the different components used to create the water turret (1).

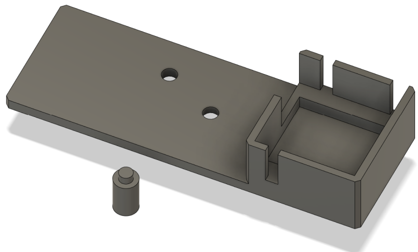
Table 1: A list of used assets for the project

Component / software
Pump
Big step motor (23F5317)
Small step motor
Malleable plastic tubing
Wires
Breadboard
Motordriver (SN754410)
Transistor (BC337)
9V Powersupply/Battery
Powersupply (USB A to USB C)
USB A to USB B cable
Micro HDMI to HDMI cable
Micro SD card
Raspberry Pi 4 Model B
Arduino UNO
Mouse
Keyboard
Screen
ABS plastic
Wiring box / water container (3D print)
Turret head plate (3D print)
Water nozzle (3D print)
Turret nozzle rotor axle and holder (3D print)
Fusion 360
Prusaslicer
Arduino IDE

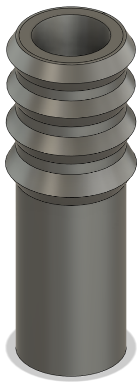
To make the Turret we needed components that were not so easy to make by standard means. Therefore a majority of the parts were designed and made via 3D printing, of which can be seen in Table (1) marked as "3D print". The parts were designed in Fusion 360, sliced on Prusaslicer and printed on a Voron 3D printer (see Figure (3)) using ABS plastic. This was to ensure that the components were waterproof. See Figure (2) (a), (b), (c), (d) for the design of the components in Fusion 360.



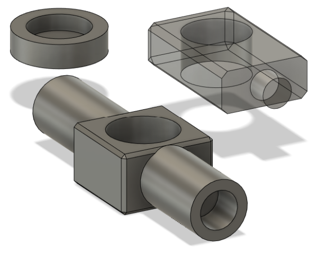
(a) The wiring box / water container for the turret. It is 70% opaque so you can see the inside of the design.



(b) The turret head plate of the turret. The small step motor fits in "the box" to the right of the design and the pole goes into the middle hole and the pole itself into the big step motor.



(c) The water Nozzle used in the project. The protrusions are there to make it sit tighter in the tubing.



(d) The turret nozzle rotor axle and its holder (top right, slightly not opaque). This goes into the small step motor to rotate the nozzle upwards.

Figure 2: The 3D designs in Fusion 360 for the water turret.

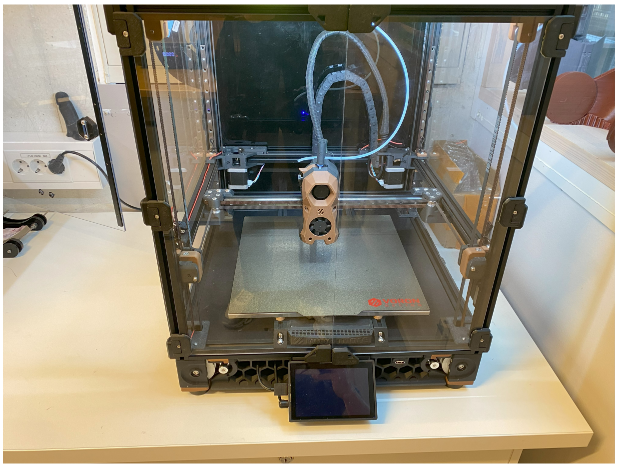


Figure 3: The 3D printer used in the project.

Both the bigger stepper motor and the smaller stepper motor is connected to the adruino's 5V source on the breadbord, in parallel with the motordriver chip. The input signal to the smaller motor comes directly from the adruino. The motordriver is essential to give inputs to the bigger stepper motor. See Figure (4). . Both V_{cc1} and V_{cc2} require

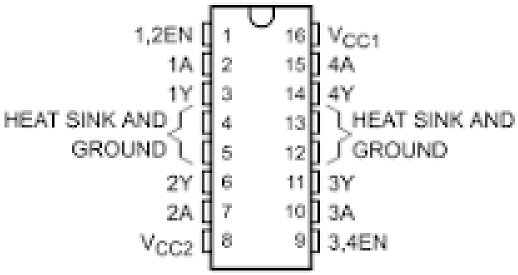


Figure 4: Sketch of motordriver Texas Instruments, n.d.

5V. 1A to 4A is where we send our signals, while 1Y-4Y is what we connect to the stepper motor. In our case, 1Y and 4Y are connected to the blue and pink wires on the stepper motor respectively. The blue striped wire and pink striped wire is connected to 2Y and 3Y respectively. 1,2EN and 3,4EN are the enable driver channels, which require a high input for the motordriver to give output in either 1-2Y or 3-4Y. In the programming we give these high and low inputs accordning to which stator teeth have to be on or off. To pump the water from our watertank to our nozzle we used a 9V pump. To turn on and off the pump we had to use a transistor, see Figure (5) (F). This way, the base would receive signals from the adruino and let the 9V current from the power source pass. It should be noted that we had to construct the circuit as a low side switch, otherwise the current would not pass. This is because the base would otherwise have to have a higher voltage than the collector. The adruino produces a voltage of 5V, so a low side switch is the only option.

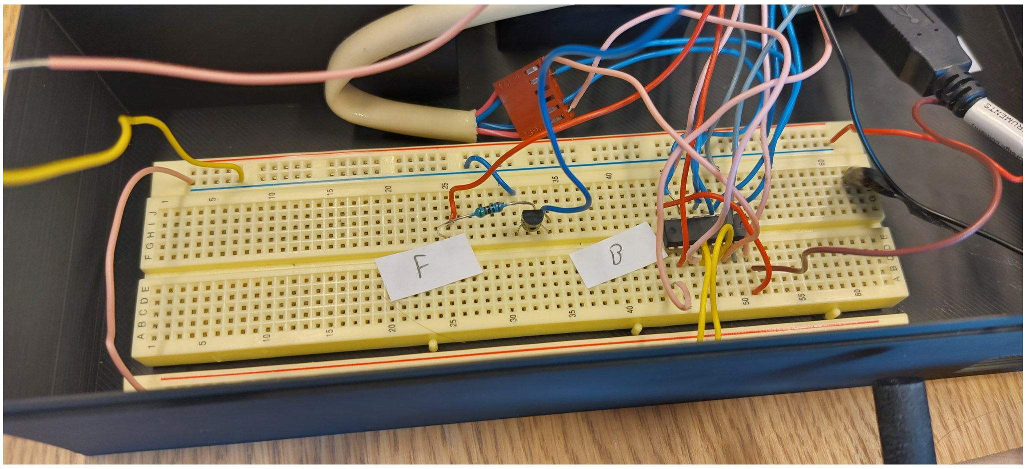


Figure 5: Low side switch

The raspberry pi 4 Model B and Arduino Uno serve as the brain for the Water Turret. The Raspberry Pi functions as a computer, waiting for inputs from the user. When it receives an (or multiple) inputs, it then relays the information with Serial to the Arduino through a Serial bus (USB A to USB B). The Arduino will then move the appropriate motors and/or activate the pump for shooting. The Raspberry Pi is powered through a power bank, and in turn powers the Arduino through the serial bus.

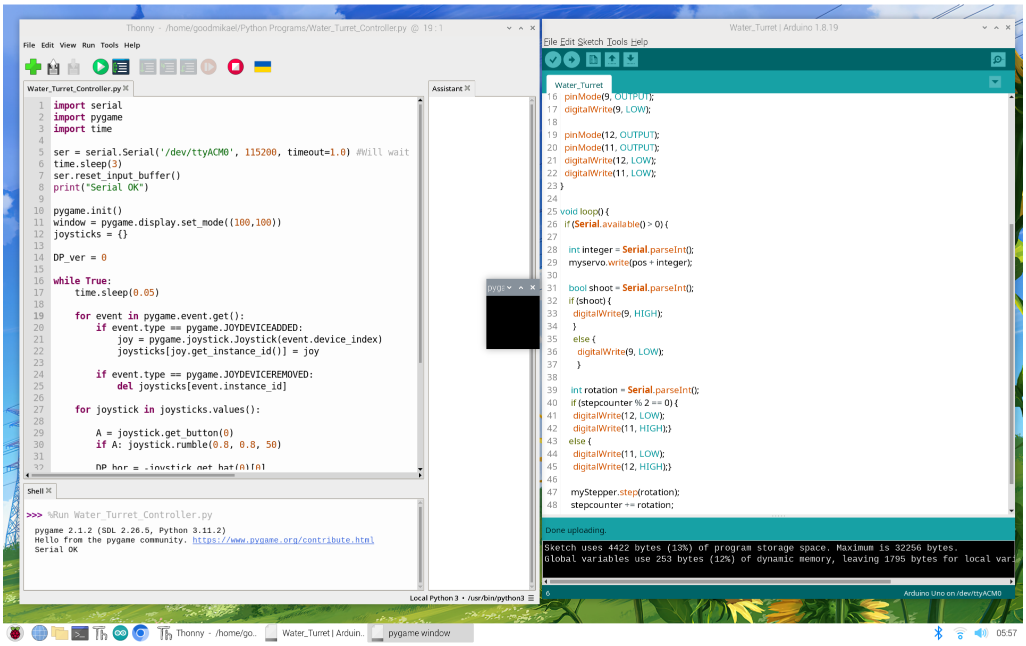


Figure 6: Raspberry Python to Arduino C/C++ communication through Serial. The middle window is a pygame window which constantly reads for inputs.

Inputs for the Water Turret come from a XBOX One controller we have connected via Bluetooth to the Raspberrry Pi. We used pygame, an open source python library, to constantly read inputs from the user by making a pop up window. This window will not be possible to exit via the controller, only stop the program. See Figure (6). While the program is constantly reading inputs, it stores the variables in a list, which is then sent over the Serial bus.

4. Result

Below is a picture of the final setup of the circuit. See picture 7.

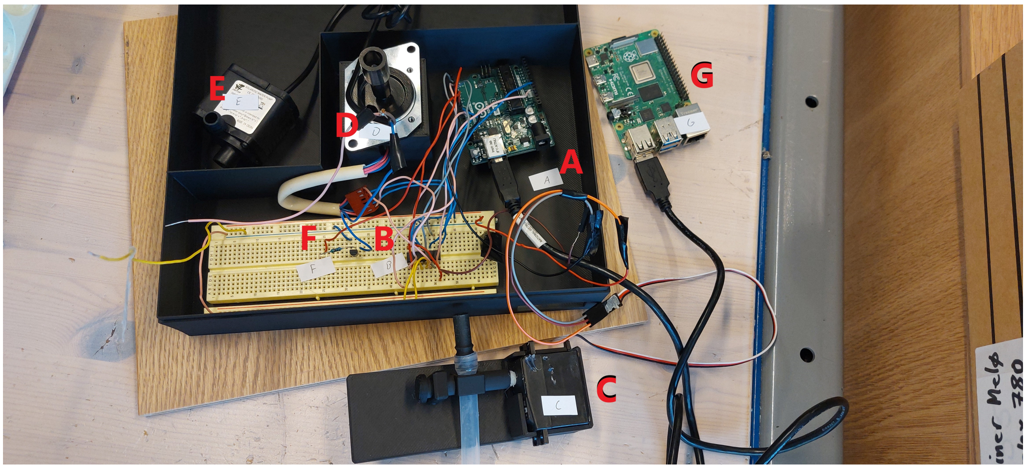


Figure 7: Whole setup

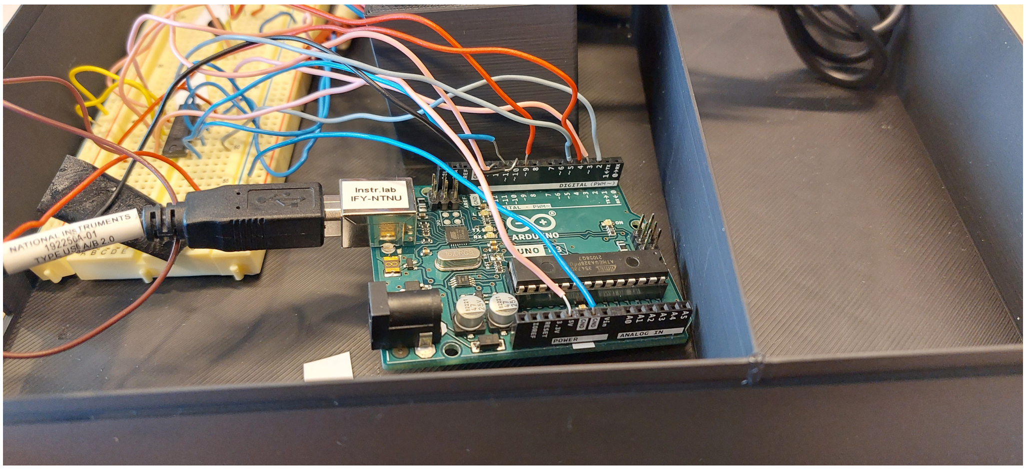


Figure 8: Outputs from adruino

Component labelled A is the adruino, and is its outputs are connected to the smaller motor (C), motordriver (B) for the bigger stepper motor (D), and the pump (E) through the transistor (F). All the signals come from the Raspberry Pi (G)

See picture 9 for the final product. Here all of the 3D printed components are in place.

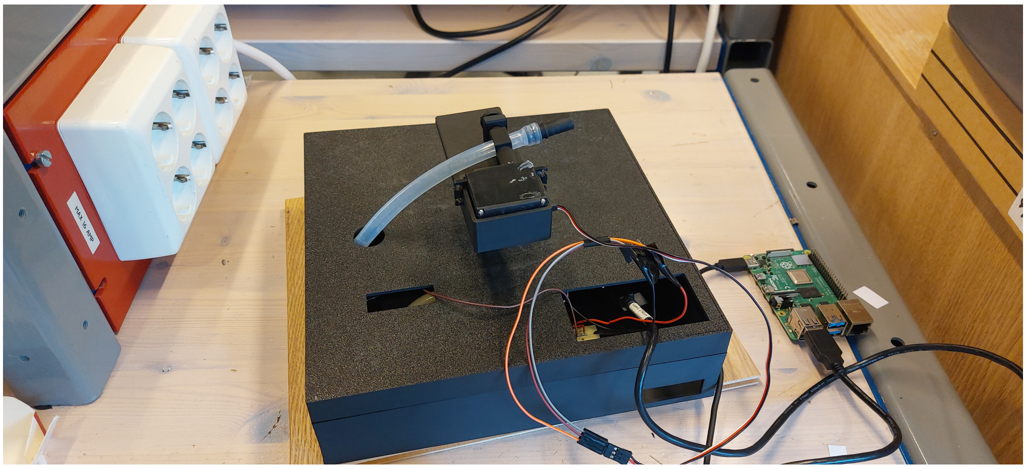


Figure 9: The 3D printed components

On the day of the showcase, the turret did not work as intended. It failed to rotate in the xy-plane, signaling an error with our bigger stepper motor. See picture 10 for the showcase setup.

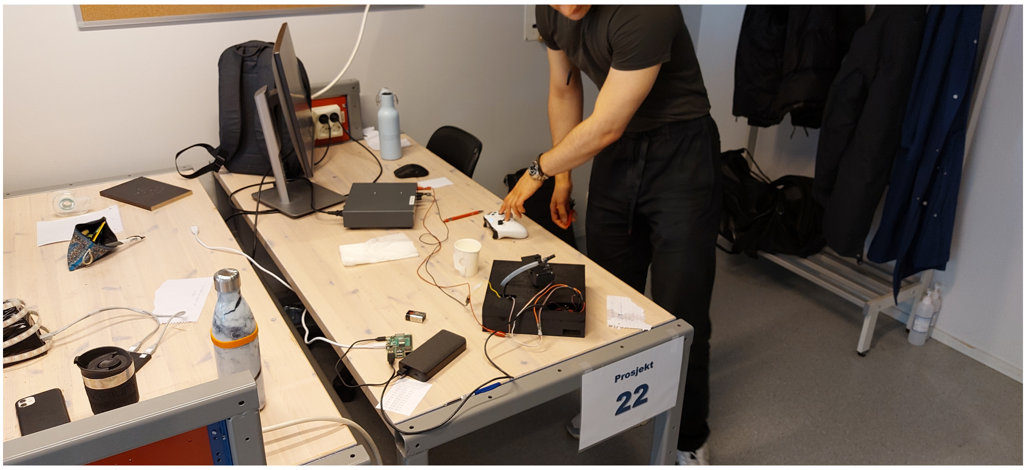


Figure 10: Showcasing of our waterturret on the final day

5. Discussion

The final showcase didn't go quite like planned. We believe the problem to stem from faulty wires in the breadboard. Loose cables could explain why the stepper motor didn't rotate as it should, since it seemed like only 1,2EN got input. This would cause the turret to turn back and forth, which seemed to be happening. A fix for this would simply be new wires, but could also be caused by our design. After the lid for the box is put on, it becomes very difficult to see and fix what goes on inside. Perhaps the wire came loose as a result from tugging on the wires when putting on the lid.

It should also be mentioned that our design has many flaws. One of them being that the rotation of the turret is limited. The reason for this is that the tubing and wiring comes from the box, and not the axle rotating the turret. Had we created a hollow axle, and used gears, we could have lead the tubing and wires through the axle and therefore rotated 360 degrees without being snagged by the box.

Another fault of our design is that the box has many holes open for water to come in. This could cause water damage to the circuit, which nearly happened when displaying the turret. It is also difficult to fill the box with water since the water hole is so small and poorly designed.

6. Conclusion

By using 3D printing and software like Adruino IDE and python we were able to build a watershooting turret. The original plan was for this turret to turn in the xy-plane and yz-plane, however due to loose wiring the nozzle could only move up or down. A fix for this would be to use new wires and make sure the connections are good, however one can not ignore that the design of our turret limits its capabilities.

Sources

MPS. (n.d.). *Stepper motors basics: Types, uses, and working principles*. Retrieved April 28, 2024, from https://www.monolithicpower.com/stepper-motors-basics-types-uses

RS. (2023, August 29). *The complete guide to stepper motors*. Retrieved April 28, 2024, from https://uk.rs-online.com/web/content/discovery/ideas-%20and-advice/stepper-motors-guide

Texas Instruments. (n.d.). *Sn754410 quadruple half-h driver datasheet*. Retrieved April 28, 2024, from https://www.ti.com/lit/ds/symlink/sn754410.pdf