Table of contents

Introduction of the robot

Backstory

In a very particular airport, near the middle of nowhere, there was a very bored luggage sorting employee. Days gone by, yet he never really felt satisfied with his job, with repetitive tasks and no real outcome in the world around him. One day, he had a vision – he would educate the bypassers with the poetry of none other than mr. William Shakespeare.

However, he was faced with a very specific limitation – he had nothing but his baggage-sorting conveyor belts to introduce the poetry. However, being a smart chap, he realized he can use the luggage as pixels, and being able to sort them meant he could arrange them in any letter he would need. This never-before-seen format – gracefully called by him Cargo Handling Unit Interface – could break him out of his mundane life, and potentially inspire others with the beautiful art.

The dedicated employee decided to input the poetry of Shakespeare into the CHUI letter by letter so that it could be displayed using luggage for the visitors to see. He created a program that would allow him to do that – and for some peculiar reasons he chose to implement it in Rust – and hooked up some wires to automate the process. As time went on, he realized his robot made thousands of curious travelers more

keen to read the words of the Poet. It did not matter that their baggage was on its way to a remote airport in Argentina.

We present the model of this successful invention below. We hope it will bring a strong sense of beauty (and lost luggage) to the lives of its users.

Function of the robot

Overview

The robot presented is able to sort the white/black discs (baggages) into one of three positions. This allows for creating a low resolution display, in which the pixels are represented by the discs. To illustrate, a simple diagram is provided [*Figure 1.*] General robot pictures presented in *Figure 2.*, *3.*, *4.*

Specifics

The discs are placed on the conveyor belt using a designated ramp. As soon as the disc is detected on the conveyor belt, the “entrance” to the conveyor is shut off, and the disc is forwarded on the belt. It arrives at the color sensor, which assesses whether it is black, white or unknown, this information being passed to the program. The robot then decides whether the disc is needed (in creating a desired pattern), and if so it is forwarded on the conveyor. If the disc is not needed or if the color is unknown, the disc is removed to a discard bin using a dedicated motor arm. The undiscarded disc is later dropped on a ramp, which uses two servos to steer a disc into the proper position. After the operation is completed, the conveyor belt is made available again.

The operation, with focus on every electrical component is also described in the robot operation section.

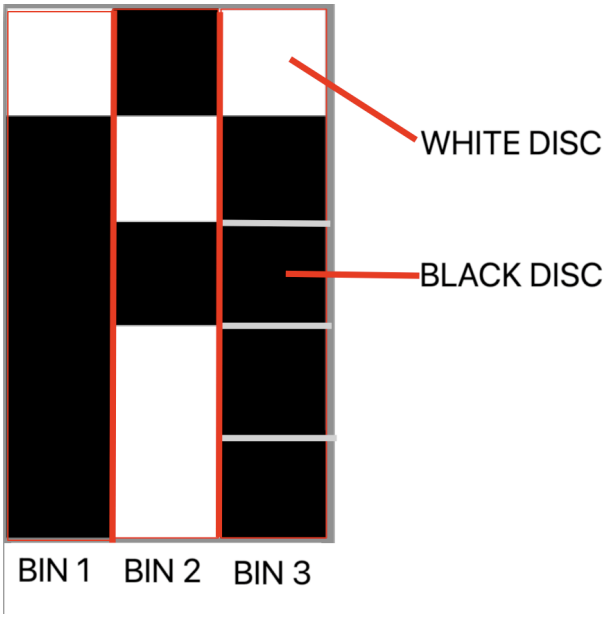


Figure 1. Low resolution display (3x5) showing letter "A" by using discs as pixels.

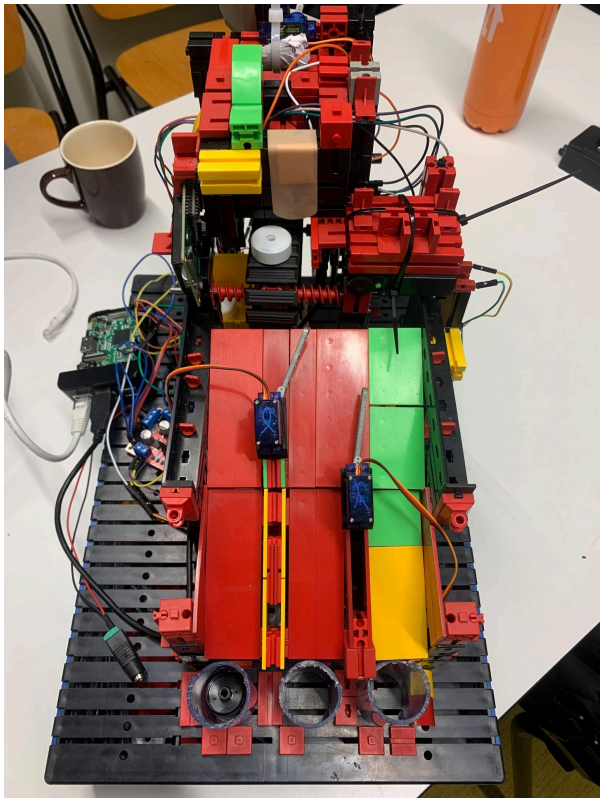


Figure 2. Robot

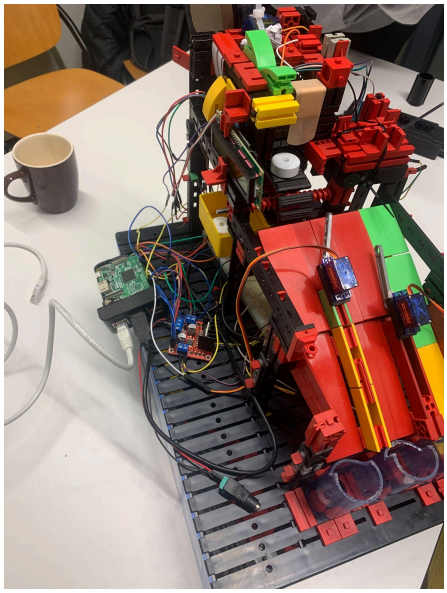


Figure 3. Robot

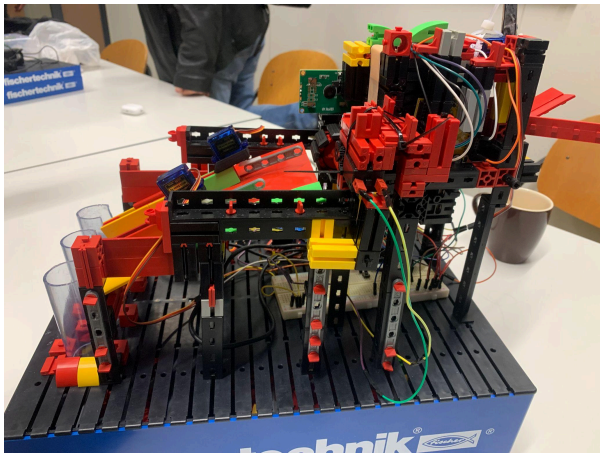
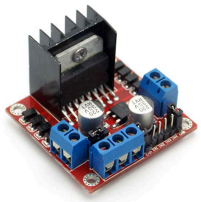
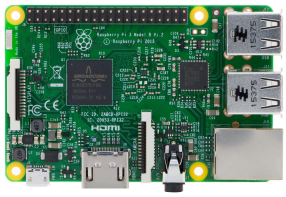
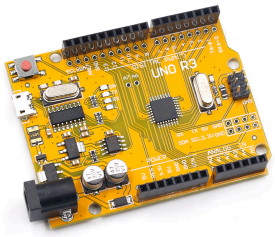
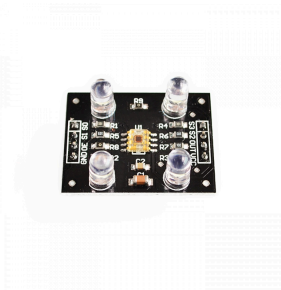
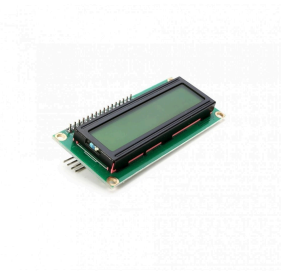


Figure 4. Robot

Technical overview

Electrical parts used

Picture of part	Part name	Function	# used
	Mini Servo SG90	The servos are used for “entrance” gate, sorting arm #1 and sorting arm #2.	3
	9V DC Motors (Fischer)	The motors are used to propel the conveyor belt and the discard arm.	2
	L298N Motor shield	Used to power the DC motors (conveyor and discard arm), is connected to power supply and Raspberry Pi.	1
	Raspberry Pi 3B	Used as the “brain”. All electrical components but the color sensor are connected to it. Connected to Arduino Uno with USB. For details consult the pinout diagram and table.	1

	Arduino UNO R3	Used together with Raspberry Pi to drive the color sensor. Chosen over RP to connect to the sensor as it was highly problematic to work with this particular sensor using RP (analog values problem). In the process, a simple serial connection protocol was developed to communicate the two boards. Consult the pinout diagram for details.	1
	TCS230 Color sensor	Used to detect color of disc on the conveyor belt. The values are read to the Arduino UNO.	1
	RCWL-1601 Ultrasonic Distance Sensor	Used to detect presence of a disc on the belt to signal to the program that something will need to be processed. Connected to Raspberry Pi.	1
	LCD Display with I2C backpack	Used to display errors in the operation of the robot. The I2C allows for a relatively easy connection and operation.	1

Connections

Pinout Diagram

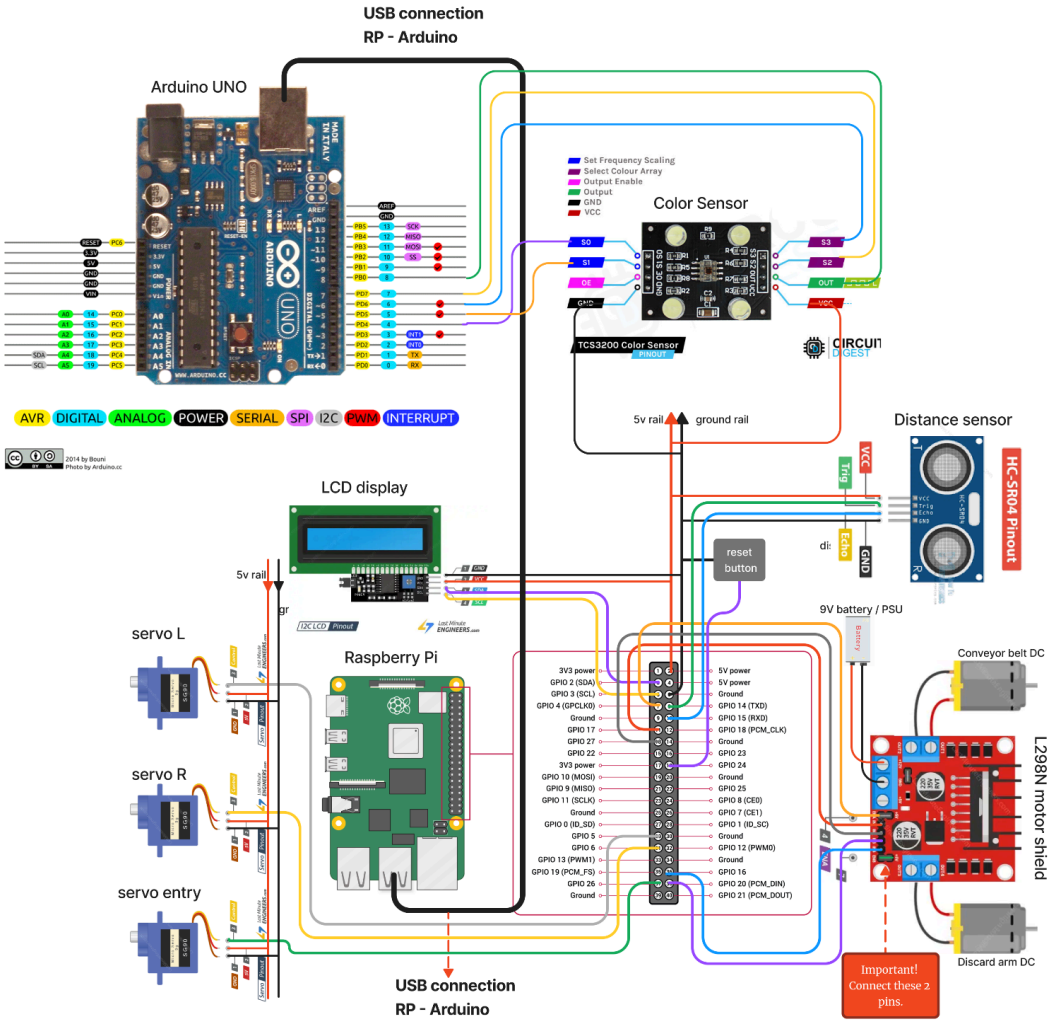


Figure 5. Pinout diagram (also available as separate pdf).

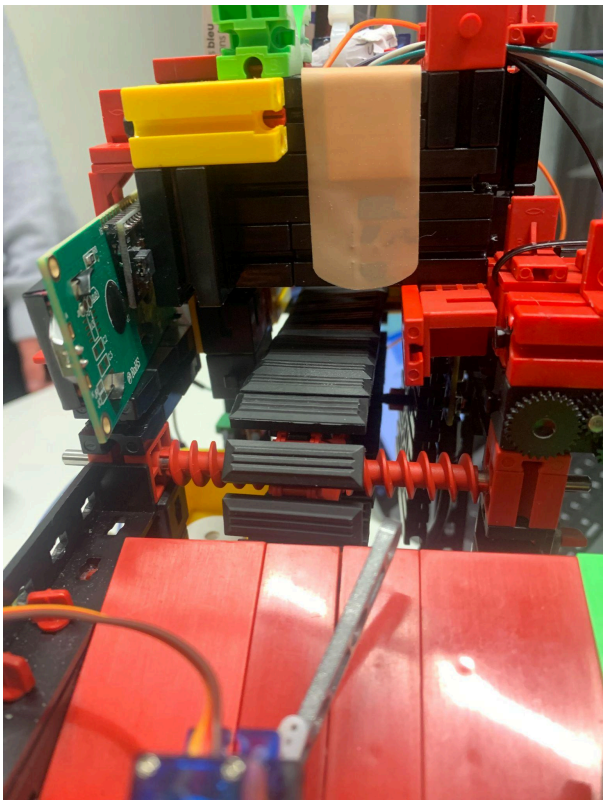
Pinout table

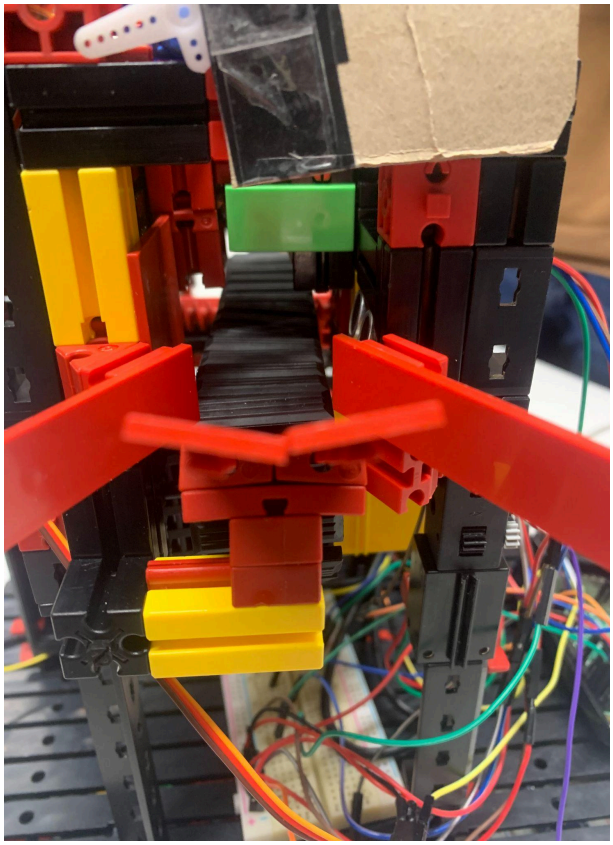
Pin on Unit	*Pin on Destination*
Unit: Servo Left Arm	**Dest: Raspberry Pi**
Pin: 5V	Pin: 2 (5V)
Pin: GND	Pin: 6 (GND)
Pin: Control	Pin: 29 (GPIO 5)
Unit: Servo Right Arm	**Dest: Raspberry Pi**
Pin: 5V	Pin: 2 (5V)
Pin: GND	Pin: 5 (GND)
Pin: Control	Pin: 31 (GPIO 6)
Unit: Servo Entrance	**Dest: Raspberry Pi**
Pin: 5V	Pin: 2 (5V)
Pin: GND	Pin: 6 (GND)
Pin: Control	Pin: 37 (GPIO 26)
Unit: L298N Motor Shield	**Dest: Misc.**
+12 V	“+” on 12V Power Supply
GND	“-” on 12V Power Supply
Pin: ENA	Pin: 7 (GPIO 4) on RP
Pin: IN1	Pin: 11 (GPIO 17) on RP
Pin: IN2	Pin: 13 (GPIO 27) on RP
Pin: IN3	Pin: 36 (GPIO 16) on RP
Pin: IN4	Pin: 38 (GPIO 20) on RP
Pin: ENB	+5V on Shield (see: the diagram)
OUT1	“+” on Conveyor Belt DC motor
OUT2	“-” on Conveyor Belt DC motor
OUT3	“-” on Discard Arm DC motor
OUT4	“+” on Discard Arm DC motor

Unit: Distance Sensor	**Dest: Raspberry Pi**
Pin: VCC	Pin: 2 (5V)
Pin: GND	Pin: 6 (GND)
Pin: Trig	Pin: 8 (GPIO 14)
Pin: Echo	Pin: 10 (GPIO 15)
Unit: Color Sensor	**Dest: Arduino UNO**
Pin: VCC	Pin: 5V
Pin: GND	Pin: GND
Pin: S0	Pin: 4
Pin: S1	Pin: 5
Pin: S2	Pin: 7
Pin: S3	Pin: 6
Pin: OUT	Pin: 8
Unit: LCD Display	**Dest: Raspberry Pi**
Pin: VCC	Pin: 2 (5V)
Pin: GND	Pin: 6 (GND)
Pin: SDA	Pin: 3 (GPIO 2)
Pin: SCL	Pin: 5 (GPIO 3)

Robot operation

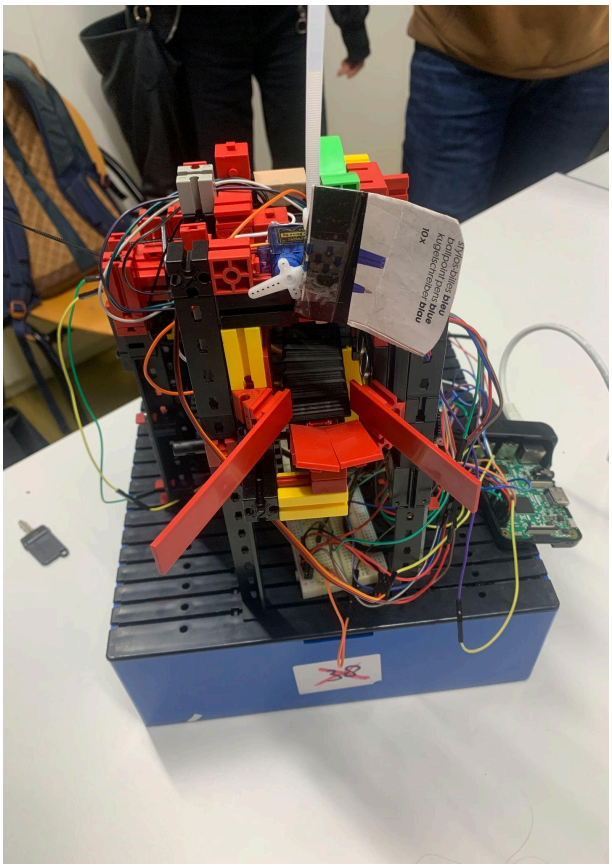
Conveyor belt

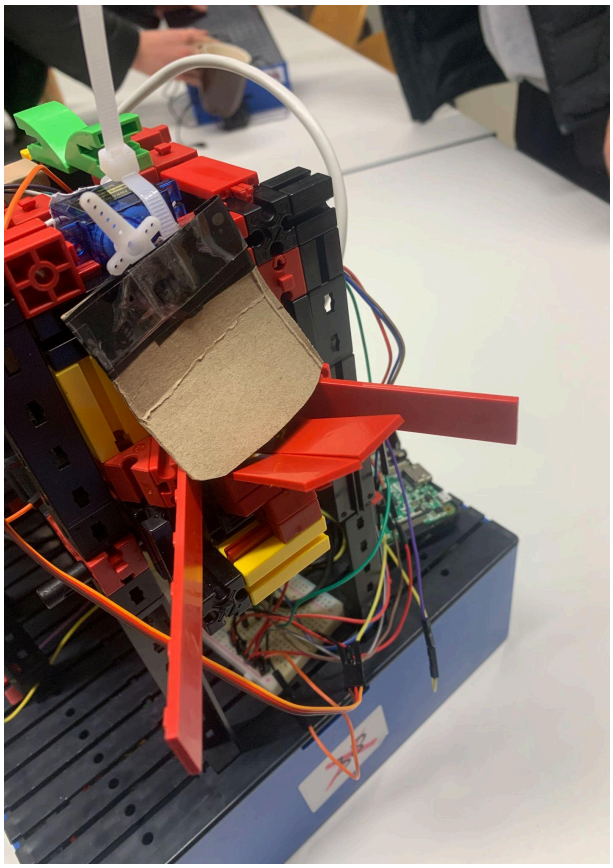




The belt is powered by a 9V DC motor provided with FischerTechnik elements. It is mounted on two gears, which help it move steadily on its path. The belt moves slowly to allow for proper processing in every step.

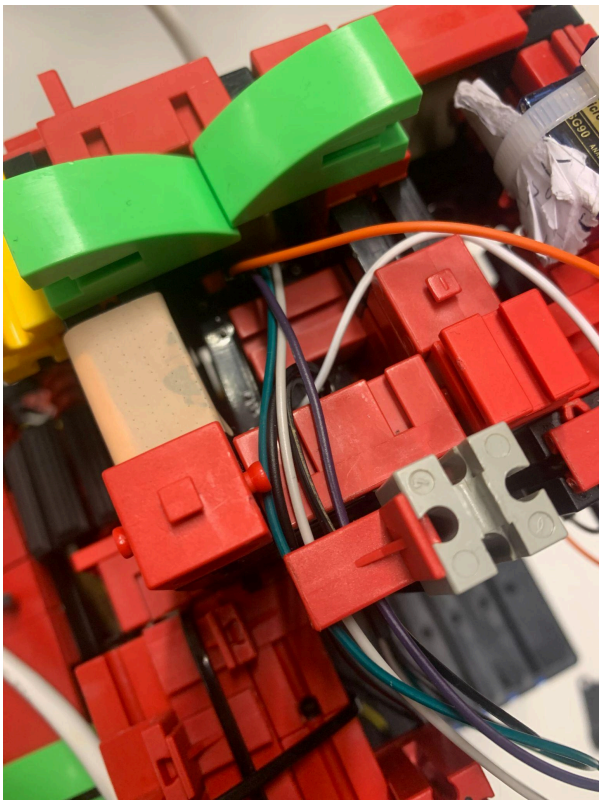
Entrance

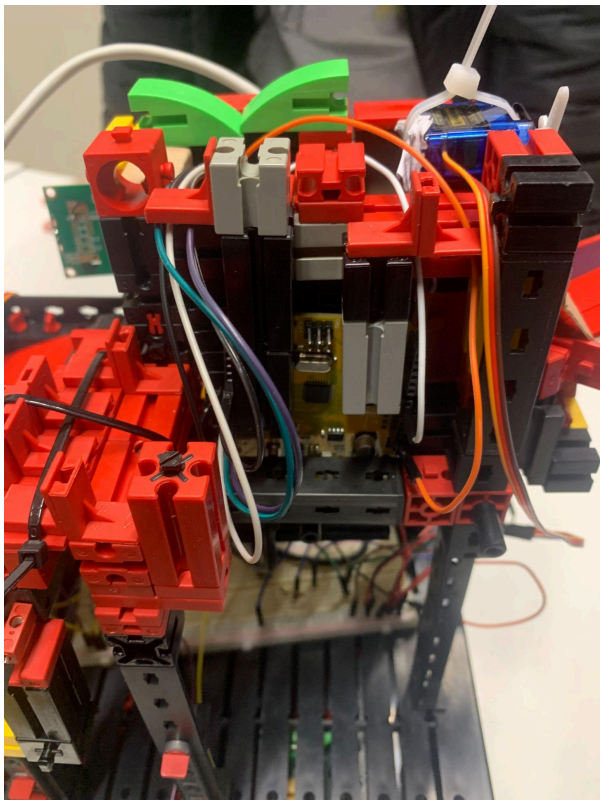




The entrance is a ramp on which the Operator puts discs, which allows for repeated correct positioning of the discs on the belt. The ramp is controlled by the entrance servo, which can block access to the conveyor belt if the processing of the previous disc is not done. The entrance is shut when a disc is detected on the conveyor belt by the distance sensor. The distance sensor uses a median filter, to discard random noise in the input signal.

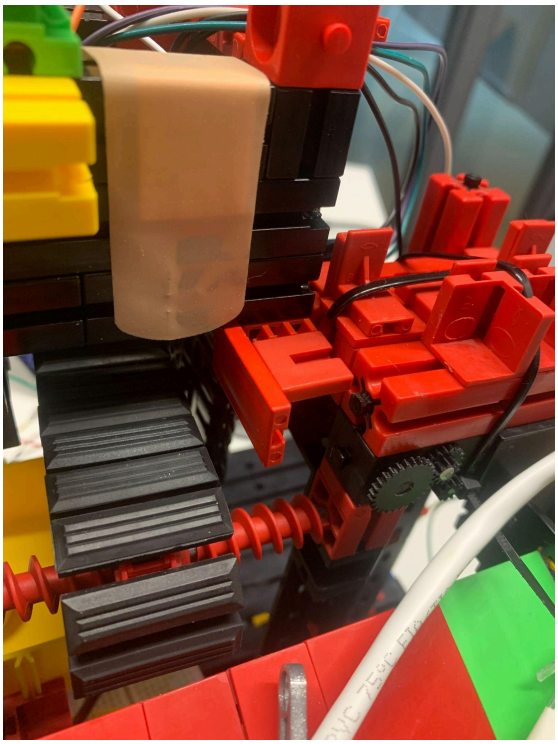
Color processing

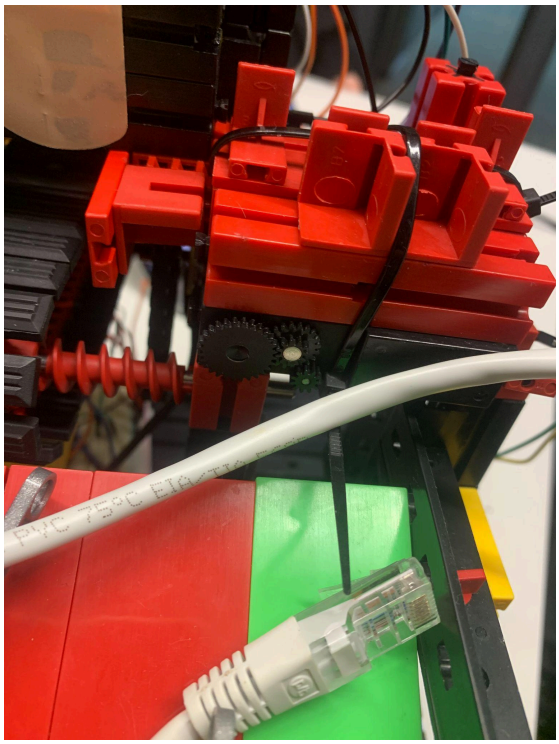




Color processing is done on the discs detected on the conveyor belt. The sensor takes a measurement, and if it is unsure then repeats it 5 times to confirm the values. The sensor is able to distinguish the color of the conveyor belt, the black/white discs, and values labeled as “other/unwanted”. The color sensor is surrounded by a cage of blocks, to limit the unwanted light coming from the environment.

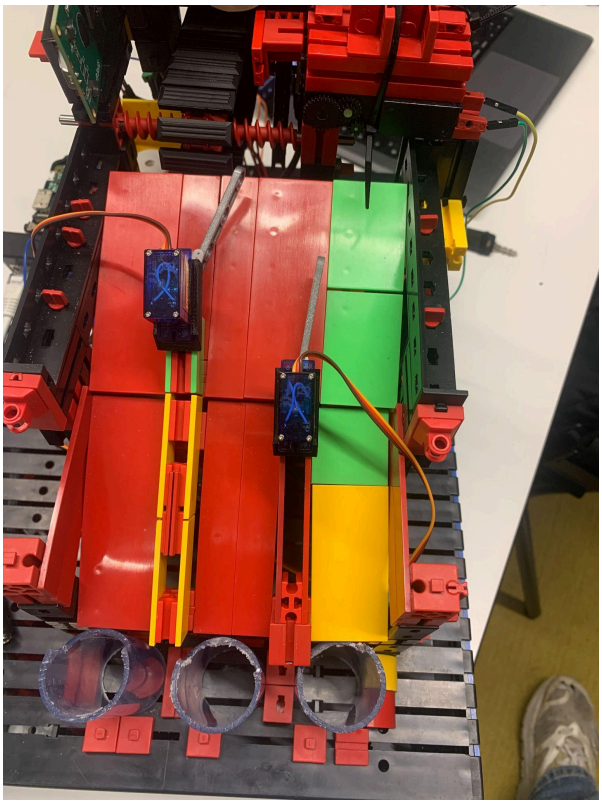
Discard arm

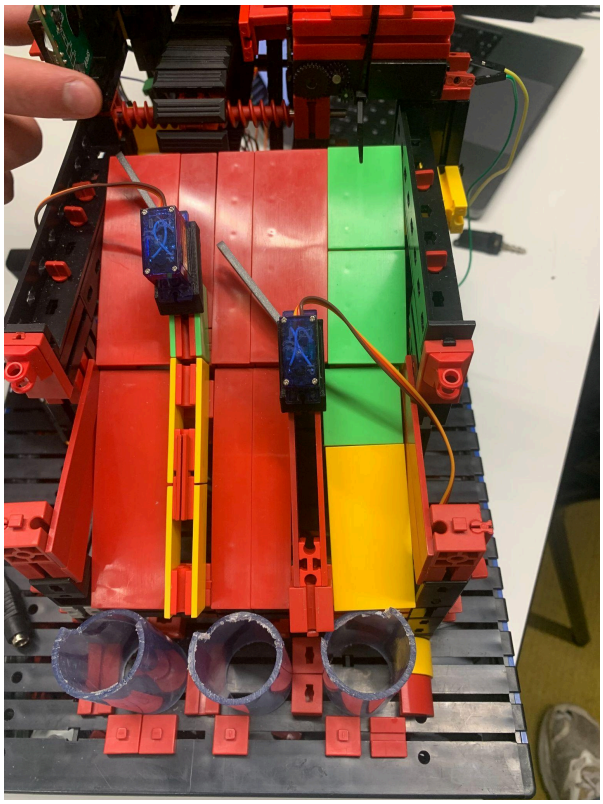




The discard arm is attached to the DC motor, which slides it onto the conveyor. It pushes off the unwanted discs into the discard bin. The arm is controlled from the raspberry pi and the motor shield. The arm pushing off the discs is guided by an inside rail, which prevents it from slipping and falling.

Sorting





The sorting is done using two servos. The discs first are sorted by servo L(eft), and then by R(ight), to be pushed into the appropriate one of 3 bins. Each arm moves between two positions, and it is implemented in the code to position them correctly when color is detected (and wanted). After 10s, both the arms move into the opposite position so as to potentially “shake off” the disc that could have gotten stuck on the ramp. When the discs are sorted they slide into the specifically prepared transparent tubes, which act as a makeshift display.