TECHNISCHE UNIVERSITÄT BERLIN

Faculty V

Institut für Maschinenkonstruktion und Systemtechnik - Agromechatronik

Module:

MARS
Mobiles Arbeitsroboter System
Mobile Working Robotic Systems

Final Report – Group 6

Mechanical tool for in row cultivation

Supervisor:

Prof. Dr. Ing. Weltzien Cornelia

Project of:

Andrea Soffietti – 468384
Etamar Bareket – 475790
Mirco Pozzoli – 468360
Tommaso Vicariotto – 476171

Summer Semester 2021-2022

Declaration of Authorship

We Andrea Soffietti, Etamar Bareket, Mirco Pozzoli and Tommaso Vicariotto, hereby declare that in this document only our original work is included. Further, we have acknowledged all sources used and we have cited these in the bibliography section. We as a group reckon that no other persons were involved in preparing the report.

PLACE and DATE: Berlin, 20/07/2022

Andrea Soffietti

Etamar Bareket

Mirco Pozzoli

Tommaso Vicariotto

Table of Contents

List of Figures

List of Tables

1. Introduction

This final report concludes the module “Mobile Working Robotic Systems” in particular, it has the goal to explain how the group worked during the summer semester 2021/2022 regarding the assigned project: Mechanical tool for in row cultivation.

1.1. Scenario & Problem Analysis

The application of the system concerns agriculture, and in particular the cultivation of carrots. The working environment will be characterized by small distances between carrots and crops (2 to 5 cm) and the impossibility to foresee their exact position. This makes it a particularly challenging situation: high precision in the machine tool operation will therefore be required. No errors are allowed concerning the destruction of the crop. As a testing scenario, a lightly weeded field has been considered, with carrots in a very early growth stage, approximately a 2^{nd} leave stage, that corresponds to leaf a radius of 2 cm.

Such farming environments are characterized by a dense presence of weeds, which can damage the crop, in this case, the carrot harvest. Weeds compete with the crop plant for light, nutrients, water, space, and other growth requirements and reduce the crop yield.

Consequently, the use of a machine that can distinguish healthy plants from weeds and destroy the latter has become a primary urgency in the world of agriculture as it is moving forward to more biological farming and trying to avoid the use of pesticides or other chemicals. The characteristics sought in the construction of an autonomous machine for this purpose are good reliability, precision, robustness, and speed. The usage of an automized tool able to destroy weeds would significantly increase the efficiency of cultivation, and would also have an important ecological impact, making it easier to achieve organic cultivation. Indeed, nowadays, excessive, and inefficient fertilizers are mainly used for weed removal, resulting in a non-negligible environmental impact due to nutrient losses to the environment and possible contamination of water resources through consumption and eutrophication of freshwater systems and coastal areas.

1.2. Aims of the project & methodology

The objective of the project is to design, build and test an autonomous mechatronic system, to be implemented on a mobile robot, with the ultimate goal of destroying in-row weeds from the soil while preserving crops. It is asked to remove at least 80% of the weeds present on the cultivating row, and completely preserve the crops while maintaining a constant robot speed of approximately 1 km/h. To reach these results and ensure the good reliability of our cutting tool, a minimum distance between weed and crops of 3 cm has been assumed.

In addition, this project aims to replace the extensive use of pesticides, i.e. introduce more environmentally friendly and organic cultivation, taking into account also the well-being of the workers by introducing autonomous vehicles. The final realization of the project requires the integration of a system capable of detecting weeds, employing image detection analysis, and providing as output the coordinates of each weed detected in the row. The latter system is under development by group number three; a collaboration between groups is therefore required.

1.3. Requirements

The main requirements of the automated system concern robustness and reliability, precision in the destruction of weeds, and complete crop preservation with a minimum fulfilment of ± 2 cm. The complete and detailed requirement list is reported in Table 1-1. Besides the fundamental requirements needed for the correct functioning of the system, easily accessible maintenance, velocity of work, compliance with ISO regulations during the design of the mechanical parts, and resistance to environmental conditions (such as rain and prolonged exposure to the sun) are key aspects that have been considered. In addition, good flexibility and adaptability to different working scenarios are required: the ideal working environment will be a Brandenburg crop field, typically characterized by medium to light (sandy) soil, but the overall aim is to make the system reliable and adaptable also for other types of terrain. Lastly, the system must ensure safety towards possible workers nearby, a reasonable setup time (estimated at no more than 5 minutes), adaptability to the existent tool changing system (also considering a maximum supportable load of 20 N per direction), an optimal weight balance, and a power supply voltage of 18 V maximum. Regarding the project's process requirements, the group aims to meet all deadlines and rules for the use of the laboratory and available machinery and to be within the budget of 30/50 € for parts purchasing and the complete implementation of the system.

Table 1-1 Requirement list

TU Berlin Chair Agro Mechatronics Prof. Dr.-Ing. C. Weltzien	Requirements List Modul: Mobile Working Robotic Systems **Mechanical tool for in row cultivation: Weed removal**	**Order No.:** SoSe-2022 (Group No.6) **Team Members**: Andrea Soffietti 0468384 Etamar Bareket 0475790 Mirco Pozzoli 0468360 Tommaso Vicariotto 0476171

No.	Date:	Origin	D/R (F/W)	**Requirements**	Specification – Data Minimum Fulfilment	STANDARD Fulfilment	Ideal Fulfilment	Unit
D01	28.04.2022	G	D	Robustness and reliability of the design				
D02	28.04.2022	G	D	Tool working precision		+/- 2		cm
D03	28.04.2022	G	D	Weed plants destroyed	80%			/
D04	28.04.2022	G	D	Protect crop	100%			/
D05	28.04.2022	G	R	High working efficiency and speed	1			km/h
D06	28.04.2022	A-E	R	Design must allow easy maintenance				
D07	28.04.2022	G	R	Maintenance schedule				
D08	28.04.2022	G	R	Safe design - Not harmful for workers and environment				
D09	28.04.2022	G	R	Respect standards regulations (ISO)				
D10	28.04.2022	M-T	R	Resistance to environmental aggressions				
D11	28.04.2022	G	D	Self-operated system				
D12	28.04.2022	G	R	Flexibility & Adaptability to different crops/weeds				
D13	16.05.2022	A-E	D	Maximum load on all the directions	20			N
D14	16.05.2022	G	R	Set up time	<5			min
D15	16.05.2022	M-T	D	Power supply voltage	≤18			V
D16	16.05.2022	A-E	R	Weight balancing				
				Process Requirements				
P01	28.04.2022	G	D/F	Meet all deadlines and due dates				
P02	28.04.2022	G	D/F	Keep to the house rules				
P03	28.04.2022	G	D/F	Respectful and careful handling of facilities, infrastructure, tools in student working room				
P04	28.04.2022	G	D/F	Meet the available purchasing budget	50	30		€

Type of Requirements:
D = Demands (F-Forderungen), R = Request (W-Wunsch, "nice-to-have");
Short name members:
G = Group; A = Andrea Soffietti; E = Etamar Bareket; M = Mirco Pozzoli; T = Tommaso Vicariotto

Replaces version:	28/04/2022	**Version:**	**2**
From:	//	**Date:**	**18.05.2022**
First version:	28/04/2022		
Responsible:	A; E; M; T	**Page**	**1 of 1**

2. State of the art and research in industry solutions

The literature review phase was the first step conducted by the group, it mainly provided information on the state of the art in weed-removing systems used at large scales, including common issues and information gained during the experiments in the field.

Following a crop emergence in the early 2000s, Peruzzi et al implemented an automatic system to be used in the largest organic carrot cultivation in Italy [1]. In particular, it consisted of an 11-tine precision hoe equipped with spring implements: torsion weeders and vibrating tines. This enabled the farmers to achieve a weed reduction of between 65 and 90%, as well as a significant reduction in the labour required for hand weeding.

Another automated tool used for row cultivation is the cycloid hoe [2]. It is based on a cylindrical rotor that functions as an actuator and some vertical tines that scratch the ground. Specifically, the cycloid movement has been adapted to the tractor speed and the rotational speed of the tool. While weed recognition and precision of the mechanism are ensured by a height profile sensor, an area allocation sensor, and a soil-plant sensor.

Finally, van Evert et al developed an autonomous robot, featuring a vertical rod weeder, with a vertical single blade powered by a high-speed hydrostatic motor, and laterally moved through a rail [3]. Moreover, the blade has a cylindrical cover, aimed at ensuring that the loose soil forms a mound on top of the hole. The weeds are detected through a camera and image processing, and the robot can move with precision in the field, following a predefined path, thanks to GPS navigation.

This research phase was important to understand the level of technology employed in automated weed removal operations in the context of organic cultivation. However, the concept to be implemented in the project differed considerably in terms of size, cost, and complexity, compared to the tools studied in the literature review. Therefore, none of them was chosen. Nevertheless, this theoretical basis helped in the first phase of devising possible concepts.

Indeed, this phase allowed the group to have good input to design something not tested yet in the field.

2.1. Responsibilities

The group decided to work together for the initial phase and then split into the two main sides of the project to optimize the working schedule. Therefore, the group has been divided into the following two sub-groups:

- **Mechanical Part**
  Responsible: Andrea Soffietti and Etamar Bareket
  Tasks:
  CAD drawing, concept realization (printing, manual work), assembly of the mechanism, and dynamics analysis.

- **Electrical Part**
  Responsible: Mirco Pozzoli and Tommaso Vicariotto
  Tasks:
  Arduino code, sensors, and motors connection, physical implementation, wirings, testing phases, and optimization of the robot functions.

2.2. Project time schedule

The project time plan has been organized in a Gantt Chart, where also the subdivision of the tasks has been reported. The main steps after the concept design were the mechanical analysis together with a selection of the required electrical and mechanical parts, CAD drawing of the final model, production and purchase of all necessary components, implementation of the Arduino's software, mechanical and electrical assembly, and finally off-and-on-field testing.

The initial working schedule is reported in Figure 2.1.

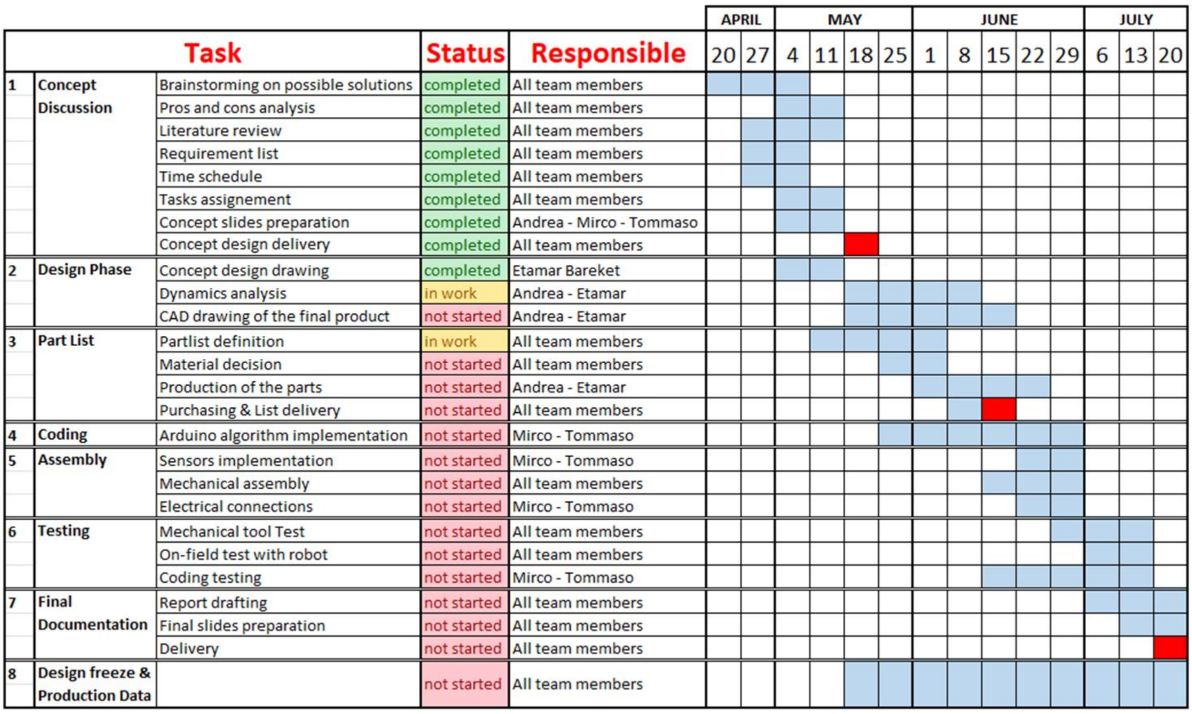


Figure 2.1 Initial Gantt chart updated on 18/05/2022

The group managed to optimally follow the schedule originally thought. In particular, the mechanical tool test has been started two weeks earlier than expected, also thanks to the absence of delivery times of the frame parts. Instead, the mechanical assembly has been realized a week later, due to an issue related to the non-efficient working of the laboratory's servomotor, which had to be replaced with a new one. Also, the production of the support to connect the ultrasonic sensor to the rake shield has been realized a week later than planned. Finally, the on-field testing has been delayed by one week due to problems related to the robot itself. It was not possible to test the tool with the robot in motion during the predefined, due to a shortcut on the robot circuit. During the time before this problem was fixed, it was only possible to test the mechanism of the tool in stationary conditions. The results were however satisfactory and confirmed the reliability of the automated tool.

The final working schedule is reported in Figure 2.2.

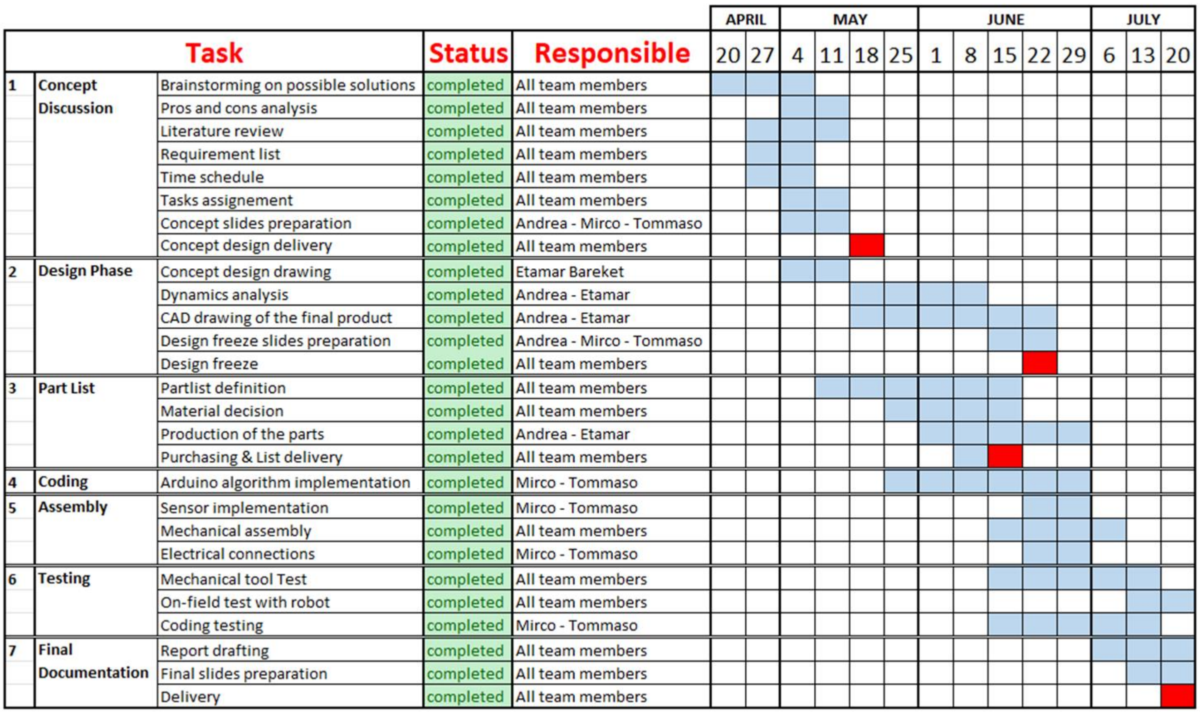


Figure 2.2 Final Gantt chart updated on 20/07/2022

3. Solutions Finding

A solution-finding brainstorming has been the first step after the schedule of the work, all team members agreed on the idea to design a new system, different from the already existing ones, to find an innovative solution for the target issue, which meet the project constraints, that may introduce new advantages in the agriculture field. After a constructive discussion among group members, three main possible systems have been agreed upon by all team members and sketched down for deeper examination of the methods suggested.

3.1. Concept 1

The first solution makes use of a moving arm, fixed to a cantilevered beam connected to the robot, with a T-shape end. The idea is that while the robot is moving along the row, the beam is moved with an alternate trajectory to hit the weed with its tip. The concept would be easy to implement but leads to several disadvantages: slow system (as the arm has to cross the entire row for each weed plant), alternate movement not precise, and high risk of tool hitting the crop.

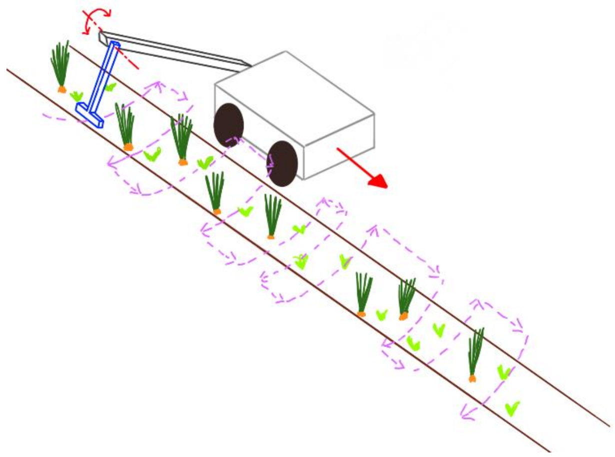


Figure 3.1 Concept 1 - Operation draft

3.2. Concept 2

The second solution, meaning the one the group decides to propose for the final implementation, consists of a rotating rake perpendicular to the soil fixed on a cantilevered arm which can be lifted and lowered to easily avoid the crop. The continuous movement leads to a good working speed, involving good precision. On the other hand, since the system cannot move the rake longitudinally, carrot sidewards weed cannot be removed.

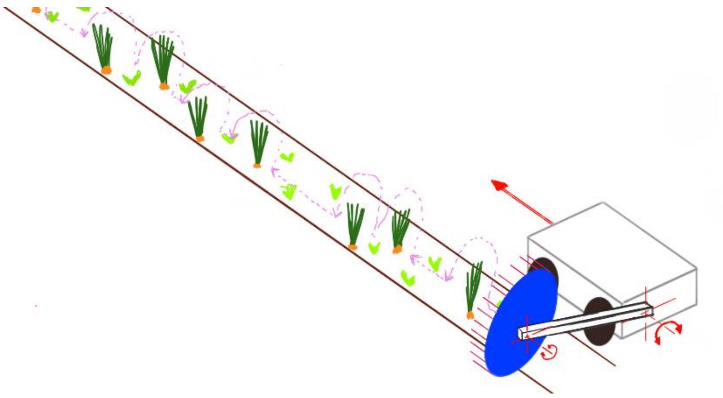


Figure 3.2 Concept 2 - Operation draft

3.3. Concept 3

The third solution consists of a 2-degree of freedom arm with a rotating rake. This concept would allow removing all the weeds with high precision, in all the possible positions, due to the flexibility of movement of the working tool. The drawbacks are the complexity of the system, which requires multiple sensors and mechanical devices, and the slow speed needed to cover the full in-row area.

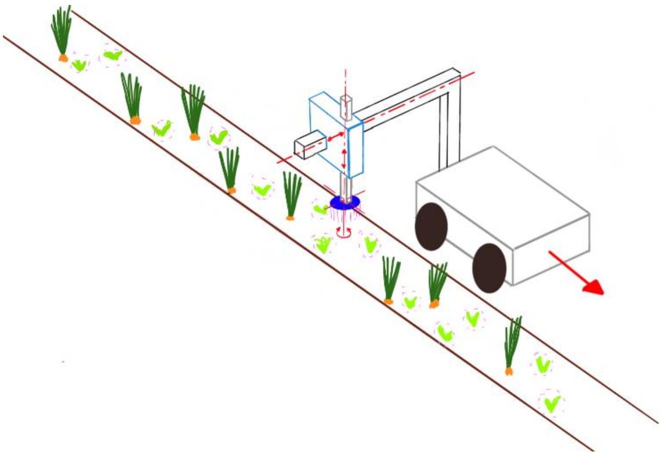


Figure 3.3 Concept 3 - Operation draft

4. Final Concept

To select the optimal concept to meet the project goals a pros and cons analysis has been performed with a discussion between the team members. Concept number 2 turns out to be the most promising one, thanks to a good trade-off between implementation cost, precision, and target achievement.

After an in-depth analysis of the proposed concept in Figure 3.2, taking into account the working constraints, the initial design of the rake has been substituted with a very thin rake wheel, which allows it to properly rotate in the small distance between the crops and by that achieving much higher precision. The rake is rotated by a DC motor, at the right distance from the soil. For this purpose, an ultrasonic sensor constantly evaluates the distance between the rake's centre of rotation and the ground. It was designed to be placed on the shield of the rack, which has a dual function: crop/workers protection and attachment location for the sensor.

The cutting wheel is connected via a shaft to an overhanging beam, which will be lifted and lowered by a servomotor, depending on the distance measured by the sensor, to avoid the crop. In this process, the rotation of the rack will be stopped to preserve the integrity of the leaves in case of contact.

All the above-described parts are fixed to a metal plate, which is then connected to the robot's tool changing system. The most critical constraint for the mechanical design is the maximum allowed force (20N per direction) of the tool changing system; a backup plan involves additionally fixing the plate directly to the robot frame through bolts and spacers.

The idea is to implement this cutting system on both sides of the robot, increasing its efficiency in removing weeds and, at the same time, balancing the robot due to the cantilevered mechanism.

An additional safety switcher is placed on the upper lifting limit of each arm to guarantee the safe functioning of the system. Finally, all the electronic devices will be controlled by an Arduino microcontroller, the system will have as input the location coordinates of the weeds, coming from the image detection system, plus the distance to the ground.

A code able to control the lifting and lowering operation of the main beams and to control the cutting wheel rotation accordingly to the sensor output has been written. To check the correct functioning of the algorithm and the overall system, visual feedback to the operators has been foreseen through LEDs. The first attempt CAD drawing concept has been reported in Figure 4.1.

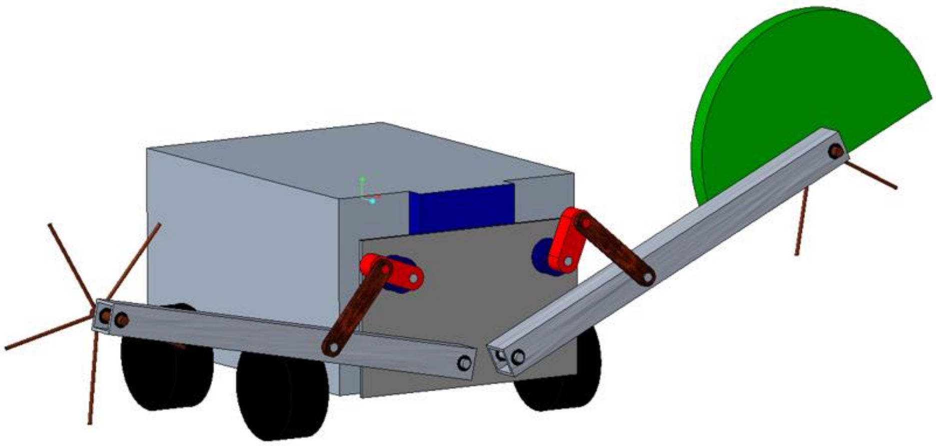


Figure 4.1 Selected concept - Initial representation

4.1. Prototype functioning

As initially introduced, the main purpose of the proposed automated system is the weed removal between carrot crops with extreme precision; the mechanism will be mechanically integrated with the robot, which will move in between the rows in the cultivated field.

The arms are connected at one end to the robot through a metal plate, while at the other tip there is a rotating rake capable of removing weed, thanks to metal flexible rods. For this purpose, two actuators for each arm are needed: a servo motor capable of raising and lowering the arm according to the presence of weeds, and a DC motor capable of rotating the rakes for weed removal. In particular, the servomotor will be placed on the metallic plate, and small leverage allows the arms to be lifted.

The decision to use a servo motor instead of a normal DC motor is due to the need of controlling the angle of rotation with quite big torque. In addition, for each arm, our system is requiring two shafts and a total of four bearings. Both bearing pairs are positioned with appositely designed and 3D printed supports; this solution turned out to be the lightest, the cheapest, and the one requiring less space at the tip of the arm. Since the arm is lowered inside the crop rows, it is important to reduce as much as possible the dimension of the devices placed at the end of the arm since they may enter in contact with the leaves.

3D printed spacers, sealing rings, and nuts are then allowed a proper mounting of the bearings. A final key component of our project is the ultrasonic sensor, which will measure the distance to the ground and serve as input for Arduino code. It will be positioned, by means of customized support, on a shield that will cover the rake and thus have the dual function of the sensor housing and safety protection.

4.2. Design Process

Being a low-budget concept, the group had to make many trade-offs to stay within the budget and try to use as many parts from the lab as possible. The overall accuracy of the system was neglected to achieve the real-life prototype realization and to prove the concept's capability.

Initially, the group started with a proper mechanical design, which would guarantee reliability and ease of maintenance for the real scale purpose. Once found out the budget limitation for the concept realization and the only few machining processes available in the workshop, the group decided to change the design perspective. Therefore, the design has been modified following the part found in the lab; for example, the shafts have been replaced with threaded rods available in the lab that required only a few and easy machining operations.

By trial and error, considering alternative items from the lab stock, the group managed to finally design the concept with only a small impact on the available budget. Once collected all the lab parts, their functions and size have been tested through dynamic analysis, CAD assembly, and other several tests, ensuring that their characteristics and performance were suitable for the project. In particular, to meet the budget limitation, the group decided to develop only one arm. This decision has been taken also because the realization of a single arm is enough to prove the feasibility of the mechanism and the second arm would function exactly in the same way.

The mounting plate has been designed to allow the connection of both arms; therefore, as a further step, since the reliability of the tool has been proved and the design of the arm has been optimized, it would be possible to easily manufacture a new arm as a copy of the one already assembled.

The initially designed shafts have been then replaced with a threaded rod from the lab, as already mentioned the ideal design done at the beginning would have required expensive machining operations since not directly available in the workshop. The group is aware that using a threaded rod as a spinning shaft in direct contact with bearings is not an ideal solution, but the trade-off is good enough for the realization of a prototype whose goal is just to prove the correct functioning of the tool and not work on the field.

The DC Motor has been placed on the inner side of the arm for several reasons. First of all, it allows to lighten the load at the over-hanging side of the arm reducing the torque required by the servo motor. Moreover, being on the left side of the leverage point of the arm helps to balance the weight on the tip of the arm itself. Second, it reduces the space occupied at the tip of the arm, lowering the risk of interference with the carrot's leaves. Finally, by placing the motor in the described position and by using a belt to transfer the power to the rake, it is possible to increase the safety of the mechanism. If the rake's rods get stuck in the terrain, for example, stopped by an obstacle, or if for any reason it is not able to rotate, the belt will slip over the pulley without damaging the motor and all the remaining system.

The belt was chosen from the lab's parts; since there was only one available the group decided to design the concept based on its dimensions. Both pulleys were designed and 3D printed to match the available belt and motor/shaft from the lab; also the length of the arm has been defined according to the final wheelbase of the transmission. In particular, since it is possible to easily control the speed of rotation of the rake through the selected step-down module, the pulleys were designed with the same outer diameter, thus creating a transmission ratio of 1:1. Furthermore, a mechanism on the arm was realized to tighten the belt with a bolt that moves the outer bearing support, to adjust the wheelbase between the DC motor and the shaft that receives the power.

Given the relatively high number of parts that make up the device, the total weight exceeds the maximum load of 20 N per direction that the tool-changing system can withstand. Therefore, as mentioned above, in addition to the tool changing system it has been foreseen to connect the metal plate directly to the frame of the robot with four additional screws by realizing holes at different heights. A more stable and fixed mounting connection has been in this way realized.

Concerning the actual removal of weed, a 3D printed circular structure, mounted on the outer shaft, was used as a hub for rake's rods that will scratch the ground. The rake's rods are subjected to stress and flexion during their operation. This solution allows fast and easy maintenance or replacement of the rakes in case of breakage or deformation. For the real-scale implementation, interchangeable rods made from different materials can be foreseen; in this way, according to the specific types of soil, the scratching rods can be changed to match the best working conditions for each application. In addition, the number of rods can be decided depending on the required speed of rotation and other factors that are evaluated by field condition parameters like the speed of the robot or the specific offset between crops.

The same conservative principles have also been used for the electronics of the system: cheap, reliable, already available in the lab, and easy-to-control sensors and motors were used. The only part ordered was the servomotor, as the one available in the workshop had been previously modified for other purposes, making it no longer possible to control the angular position. Instead, the servomotor available from ATB stock was extremely complex the control and it requires a specific driver board.

In particular, was bought the cheap and easy-to-find MG995 servomotor, after having compared its torque with the one required to lift the arm, previously calculated in the kinematic analysis, and having physically tested the same motor in the lab. The kinematic study has been carried out also to find the optimal position of the servo, to reduce the required torque to lift the arm; in the final design, it was positioned as close as possible to the external edge of the plate.

The arm rotates around a shaft (threaded rod) fixed to the plate and is lifted and lowered through an articulated quadrilateral mechanism; this solution allows to charge most of the load directly to the plate to avoid the need for a huge torque by the servo.

4.3. Selected Microcontroller

Arduino UNO (Figure 4.2) has been selected as a microcontroller for the system, has enough pins for the target applications, and is well known to be an optimal and robust board for mechatronics projects like the one here presented. It can be easily connected with all the sensors and switches needed and can be easily programmed using the Arduino IDE provided by the producer.

In Table 4-1 all the technical specifications have been reported.

Figure 4.2 Arduino UNO [4]

MICROCONTROLLER	ATmega328P	**FLASH MEMORY**	32 KB (0.5KB used by bootloader)
OPERATING VOLTAGE	5V	**SRAM**	2 KB
INPUT VOLTAGE	7-12V	**EEPROM**	1 KB
INPUT VOLTAGE (LIMIT)	6-20V	**CLOCK SPEED**	16 MHz
DIGITAL I/O PINS	14 (6 provide PWM output)	**LED_BUILTIN**	13
ANALOG INPUT PINS	6	**LENGTH**	68.6 mm
DC CURRENT PER I/O PIN	20 mA	**WIDTH**	53.4 mm
DC CURRENT FOR 3.3V PIN	50 mA	**WEIGHT**	25 g

Table 4-1 Arduino Uno Rev3 - Technical specifications [4]

4.4. Selected Sensors and Components

As previously written, the selection of sensors and components has been based on the ones available in the workshop.

For the target application, a distance sensor was needed to continuously evaluate the distance between the tip of the arm and the soil. The ultrasonic sensor HC-SR04 (Figure 4.3) has been selected among the others available (ex. Infrared sensor) due to the quite good accuracy and the ranging distance which met the requirement of the device.

Figure 4.3 HC-SR04 Ultrasonic sensor for Arduino [5]

The HC-SR04 uses non-contact ultrasound sonar to measure the distance from an object. It is equipped with two ultrasonic transmitters, a receiver, and a control circuit. High-frequency ultrasonic waves are emitted and are going to bounce off any nearby solid objects. The receiver evaluates the return echo which is then processed by the control circuit to calculate the time difference between the signal being transmitted and received. This time is converted to measure the distance between the sensor itself and the reflecting object.

In Table 4-2 all the technical specifications have been reported.

POWER SUPPLY	DC 5V	**WORKING CURRENT**	15 mA
RESOLUTION	0.3 cm	**WORKING FREQUENCY**	40 Hz
RANGING DISTANCE	2 - 400 Cm	**MEASURING ANGLE**	15 °
TRIGGER INPUT PULSE WIDTH	10 µs	**DIMENSIONS**	45 x 20 x 15 mm

Table 4-2 HC-SR04 - Technical specifications [5]

The safety limit for the arm lifting operation is guaranteed by a micro limit switch (Figure 4.4). This device is needed to avoid critical situations in which the arm for whatever reason (ex. Inclined terrain, big inertia force, etc..) is lifted over the safety limit. If pressed, the microcontroller receives an alert input: first, it will stop all the operations, and then it will send a command to the servo to bring the arm to the home position.

In Table 4-3 all the technical specifications have been reported.

Figure 4.4 Momentary Hinge Roller Lever Micro Limit Switch [6]

CURRENT RATING	5 A	**OPERATING VOLTAGE**	250 V
TERMINAL	SPDT	**CONTACT TYPE**	1NO + 1NC
MATERIAL	PLASTIC and METAL	**DIMENSIONS**	20 x 6.5 x 13.5 mm

Table 4-3 Momentary Hinge Roller Lever Micro Limit Switch – Technical specifications [6]

To power, the DC motor that is putting the rake in rotation the group opted for a very simple and robust circuit. It is controlled through a step-down converter (Figure 4.5) and a single-channel Relay (Figure 4.6). The first allows to regulate the supply voltage to the motor (by rotating the brass screw visible in Figure 4.5), and therefore regulates the spinning rotation, while the second directly turns on and off the motor depending on the output sent by the microcontroller.

In Table 4-4 and Table 4-5 the respective technical specifications have been reported.

Figure 4.5 LM2596 DC-DC Module Step Down Converter Voltage Regulator [7]

INPUT VOLTAGE	3.2 – 40 V	**OUTPUT VOLTAGE**	1.25 – 35 V
CONVERSION EFFICIENCY	92%	**OUTPUT CURRENT**	3 A max
OUTPUT RIPPLE	30 mV	**SWITCHING FREQUENCY**	65 kHz
OPERATING TEMPERATURE	-45° - +85°	**DIMENSIONS**	43 x 21 x 14 mm

Table 4-4 LM2596 – Technical Specifications [7]

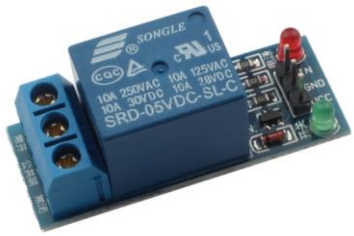


Figure 4.6 5V Single Channel Relay Module Board [8]

INPUT VOLTAGE	5 V	**SWITCHING VOLTAGE**	250V AC / 30 V DC
INPUT CURRENT	5 – 10 A	**SWITCHING CURRENT**	10 A
CONTROL SIGNAL	TTL level	**DIMENSIONS**	43 x 17 x 17 mm

Table 4-5 5V Single Channel Relay Module Board – Technical specification [8]

The power supply is coming from the robot; therefore, it has been decided to place a general power switch in between the battery and microcontroller. In this way, by pushing the button it is possible to power the whole mechanical system. A switch with a green LED (similar to the one in Figure 4.7) has been selected from the ones available in the lab to give feedback to the user about the presence of the power supply in the system.

In Table 4-6 the technical specifications have been reported.

Figure 4.7 Mini Switch with Green LED [9]

VOLTAGE	12 V DC	**SUPPLY CURRENT**	20 A

Table 4-6 Mini Switch with Green LED – Technical specifications [9]

Finally, three LEDs (green, yellow, and red - Figure 4.8) have been selected to give visual feedback to the user about the correct, or not, functioning of the machine tool. They are placed in apposite LEDs holders and mounted on the metal plate on the back of the robot.

In Table 4-7 the technical specifications have been reported.

The LEDs supply a voltage of 2.1V and have an output voltage from Arduino of 5V, one resistor for each LED has been used (Figure 4.9). Resistors are needed to keep the LEDs from drawing too much current. In particular 220Ω resistors have been selected.

In Table 4-8 the technical specifications have been reported.

Figure 4.8 5mm LED [10]

SUPPLY VOLTAGE	2.1 V DC	**SUPPLY CURRENT**	20 mA
LUMINOUS INTENSITY	20 mcd	**BEAM ANGLE**	60°
DIAMETER	5 mm	**COLOURS**	Red, yellow, green

Table 4-7 5mm LED – Technical specifications [10]

Figure 4.9 Resistors

RESISTANCE	220 Ω	**POWER**	0.25 W

Table 4-8 Resistors – Technical specifications

4.5. Selected Actuators

Considering one single arm, two actuators are needed. Due to the high torque required to lift the arm and the need to control the angular position of it a servo motor resulted to be the optimal choice to control the arm movement.

In the workshop a servomotor MG995 was available and the whole design of the system was done considering its use. However, during the test phase of all the electronic devices, the group found out that the servo has been modified and the encoder needed to evaluate the angular position of the shaft had been removed; furthermore, the latest resulted in a shaft able to rotate 360°. Therefore, it turned out to be not suitable for the specific application.

Nevertheless, it has been used to test if the provided torque was enough to lift the arm. After the test validation, the group decided to order a new servomotor MG995 (Figure 4.10).

In Table 4-9 the technical specifications have been reported.

Figure 4.10 MG995 SERVO – WHADDA [11]

SUPPLY VOLTAGE	4.8 – 7.2 V DC	**STALL TORQUE**	83.4 – 98.1 N*Cm
OPERATING SPEED	0.2 s/60° - 0.16 s/60°	**ANGLE RANGE**	0 – 120 °
DEAD BAND WIDTH	5 us	**TEMPERATURE RANGE**	0 – 55 °C
WEIGHT	55 g	**DIMENSIONS**	20 x 54 x 47.2 mm

Table 4-9 MG995 SERVO – Technical specification [11]

For what concerns the rotation of the rake, the DC motor "Makeblock 180 Optical Encoder Motor" (Figure 4.11) present in the lab has been selected. The choice has been done based on its easy control and the already present attachment solution realized with threaded inserts.

Figure 4.11 Makeblock 180 Optical Encoder Motor [12]

In Table 4-10 the technical specifications have been reported. The wide supply voltage range allows a fine regulation of the rotation speed using the step-down module.

SUPPLY VOLTAGE	7.4 V max	**REDUCTION RATIO**	39.6
NO LOAD CURRENT	240 Ma	**LOAD CURRENT**	<750 mA
NO LOAD SPEED	350 RPM	**LOAD SPEED**	14000 RPM
LOAD TORQUE	800 g*cm	**BREAK-OUT TORQUE**	5 Kg*cm
ENCODER ACCURACY	360	**DIMENSIONS**	65.3 x 48.1 x 24 mm

Table 4-10 Makeblock 180 Optical Encoder Motor – Technical specifications [12]

5. Hardware realization

As previously explained, the realization of the prototype is based on the already available parts from the workshop. In addition to those, other parts have been designed and printed to connect the available parts and realize the assembly. The final CAD model designed in Inventor is shown in Figure 5.1.

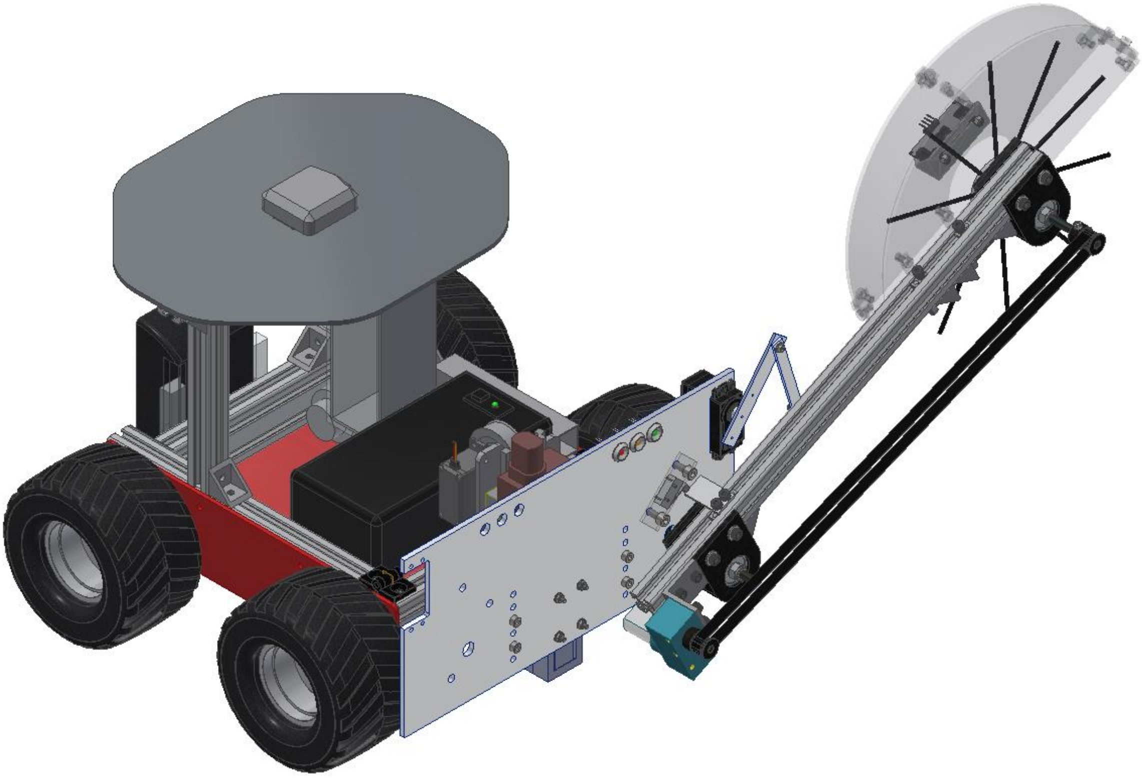


Figure 5.1 CAD model on Inventor

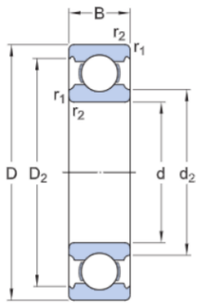
As already mentioned, the two shafts and the rake are made with threaded roads that the group found in the lab and only had to cut to the right length Figure 5.2. This allowed the group to skip expensive machining operations for the realization of proper mechanical shafts.

Figure 5.2 Threated roads in the lab

The arm is made with a 20x20 mm profile found in the lab and cut to fit the planned design; in particular to fit the wheelbase between the two pulleys of the belt transmission.

Figure 5.3 20x20mm profile in the lab

Only one size of bearing, suitable for the purpose, was available in 4 pieces in the lab. They are not coming with any code, but the dimensions are compatible with the W 636 from SKF Figure 5.4.



DIMENSIONS

d	6 mm	Bore diameter
D	22 mm	Outside diameter
B	7 mm	Width
d_2	≈ 10.5 mm	Recess diameter
D_2	≈ 19.03 mm	Recess diameter
$r_{1,2}$	min. 0.3 mm	Chamfer dimension

Figure 5.4 SKF W636

All the concept is realized around these 4 bearings, the other parts were designed/chosen to fit these components.

In addition to the one above described, other components taken from the lab are standard connection parts such as screws, nuts, washers, T-slotted framing fasteners, nylon insert locknuts, retaining rings, thread inserts, and nails.

Via FDM 3D printer, after a proper mechanical design, two bearing supports, two side covers, two spacers to fix the inner rings of the bearings, the rake hub, and two pulleys have been printed.

In Figure 5.5 are shown the three stages of a 3D printed part: CAD design on Inventor, sliced on Cura, and finally, the printed bearing support.

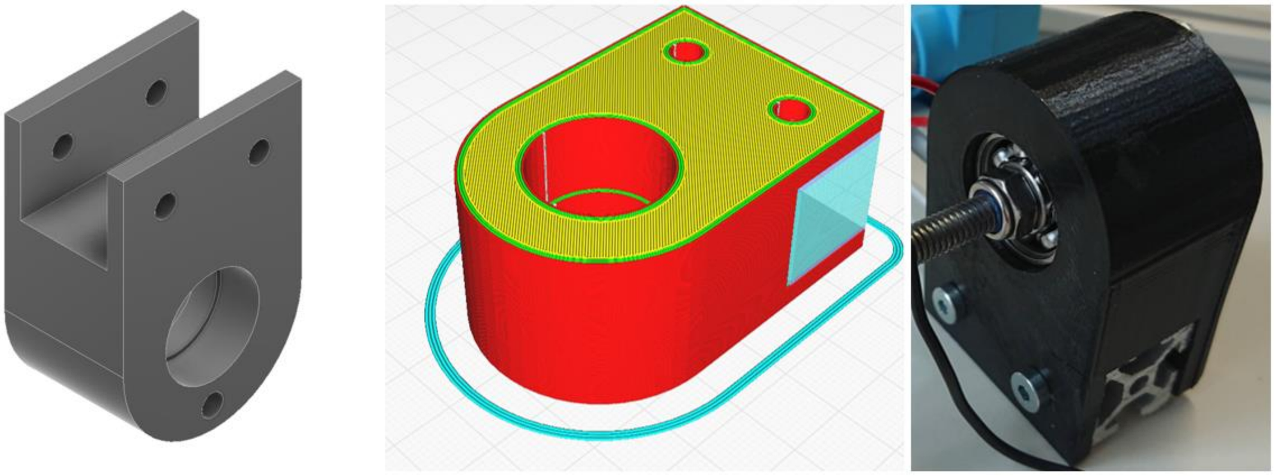


Figure 5.5 3D printed part realization

The material used is PETG and the considered costs for printing were:

- 0,05€ per gram
- 0,02€ per minute
- 2,00€ per run.

A metal plate, which allows connecting the mechanism to the robot, has been realized via CNC machining. The raw material selected was an aluminium plate (300x240x4mm) found in the lab. Aluminium has been chosen for its low weight, relatively high strength, and resistance to environmental aggression; the original design was involving a 5mm thick plate, but the material was not available in the workshop. FEM simulations have been realized to validate the concept with a 4mm thickness plate, the positive results allowed to validate the concept, and subsequently, the plate has been machined, the technical draw of the final component is shown in Figure 5.6. The plate is the only component that is in common with the two arms, it already has the required holes on the left side for the second arm attachment. In total it has:

- Ø3,18 x 8 holes for servomotors.
- Ø8,50 x 6 holes for LEDs holders.
- Ø4,50 x 12 holes for extra attachment to the frame, only 6 holes can be used at the same time. Overall, 12 holes have been foreseen to allow the mounting of the plate at different levels depending on the specific application. The offset between each hole is 10mm to meet the position of the T-slotted fasteners that are placed, in the robot's frame at a 20mm offset distance.
- Ø6,00 x 2 holes for the shafts.
- Ø3,40 x 4 holes for connection to the tool changing.
- Ø5,50 x 4 holes for safety switch connection.
- Ø10,00 x 2 holes for cable management, all the cables on the moving arm are passing through this hole.

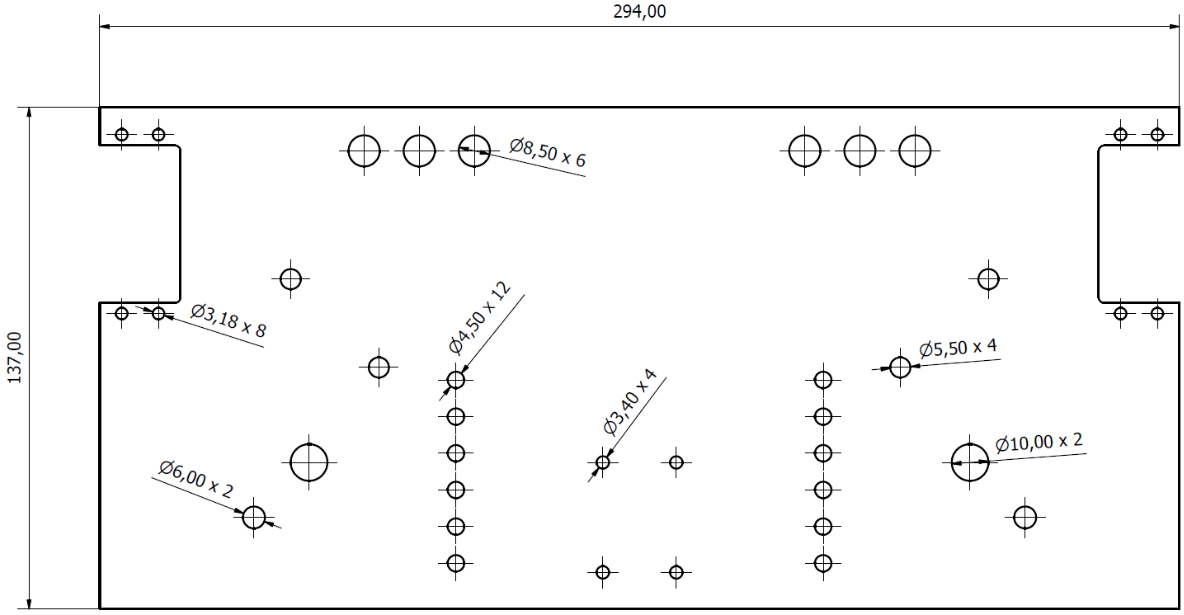


Figure 5.6 Plate – technical drawing

The first attempt of the g-code was sent to the CNC responsible. Given the simplicity of the part, the responsible preferred to recreate the code by himself instead of checking ours.

The considered costs for CNC machining were:

- 0,20€ per minute
- 1,00€ per piece
- 4,00€ per run

The 3D CAD drawings of the entire concept can be found in the ”*CAD_Pack_and_Go”* folder, the production data of the printed and machined parts instead can be found in the folder ”*Production_Data*”.

The arm is connected to the plate by an M6 threaded rod, used as a shaft, and fixed on the plate by 2 nuts. On the shaft 2 bearings are mounted separated on the internal ring by a spacer, they are then blocked by 2 nuts, on the external side, a retaining ring prevents the nut to unscrew, while on the internal side the nut is pushed against a spacer that determines the distance between arm and plate. On the external ring, one bearing is clamped between the bearing support and a cover while the other is free to move, this avoids over-constraining the model. The bearing support is then closed on the internal side with a cover that holds the external ring of the internal bearing Figure 5.7.

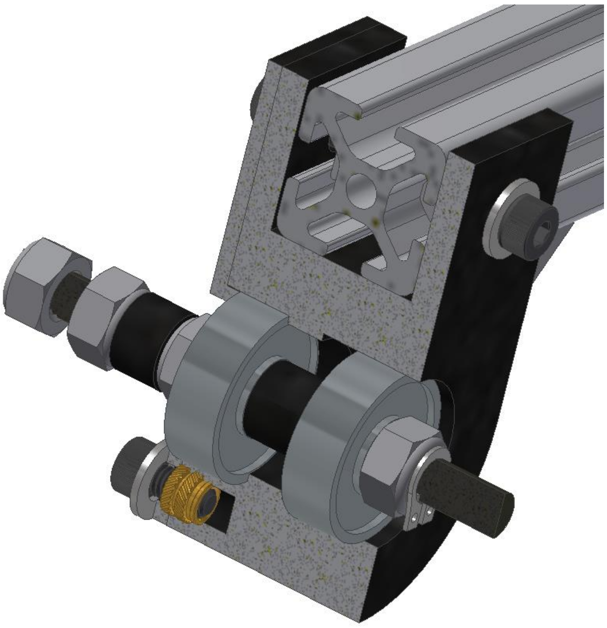


Figure 5.7 Plate shaft

On the other side of the arm, the rake is connected with the other two bearings inserted in the exact same bearing support with the same cover and spacer too. Moreover, the rake is clamped on the internal side thanks to the coupling between a nut: the rake's hub has a hexagonal slot that matches the M6 nut dimensions so to allow the rake and shaft to spin together. On that side, everything is locked with a retaining ring. On the external side, the pully is coupled to the shaft thanks to a locking bolt Figure 5.8.

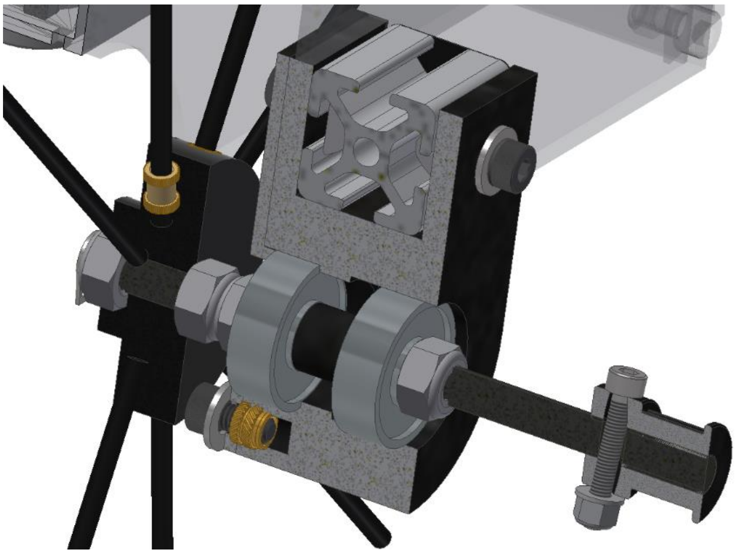


Figure 5.8 Rake shaft

The rake shield was realized completely by the group in Plexiglas, as it is light and relatively cheap material. It shouldn’t tolerate much force but still has to be able to stand field environmental conditions. Initially, the group considered to 3D print the shield, but afterwards opted for a Plexiglas solution due to the quite big size. The front and back of the shield were cut from a 2mm plat into a circle with a radius of 120 mm by a scroll saw. Then the circle was cut in half creating symmetry between both parts. To attach the back plate of the shield to the arm the bearing support shape had to be removed and at the centre of the front half circle, a circle with a radius of 30 mm was realized both by scroll saw. The curved part of the shield was made out of a 1mm Plexiglas which allows it to be bent; nevertheless, it is still strong enough to hold both front and back disks. It has been cut with a hand saw in the desired size. Before assembling the three parts hols were realized by a stand drill. For assembling the shield 4mm head bolts were used with a nylon-insert locknut to avoid the nut getting loose due to vibrations, 6 parts of a 90 deg corner angle were used and for the connection to the arm two bigger 90 deg corner angles were used.

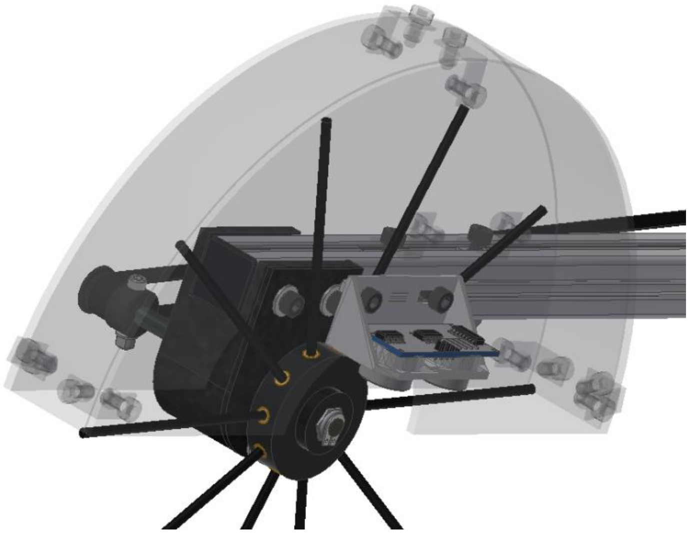


Figure 5.9 Rack shield

At first, the rake hub was designed to be made out of aluminium by using CNC machining. This would have led to expensive components out of budget, therefore a new design was established. To create the main rake hub a 3D printer was used as described before. In this way, it has been possible to print a hexagonal slot for the coupling with an M6 nut allowing the rake and the shaft to rotate together. The group decided to use threaded rods as teeth of the rake; in order to connect them to the main cylinder, holes have been foreseen in the design of the hub with the idea to glue the rods inside them. However, after critically analysing the solution, the group pointed out the impossibility to change the number and material of the rods without having to create another rake body for each set-up; let's consider for example a maintenance operation when only one rod has to be replaced. The new solution instead involves the use of threaded inserts found in the lab (Figure 5.10.), by using these components the ability to change rods and realize adjustments considering the actual working scenario is kept. To correctly place the inserts, a few processes were checked, and different sizes of holes were tested until the best method was found. Once printed, the holes in the hub were cleaned from the supports, and then a soldering iron was used to warm up the insert and place it in the correct position inside the hole in the hub.

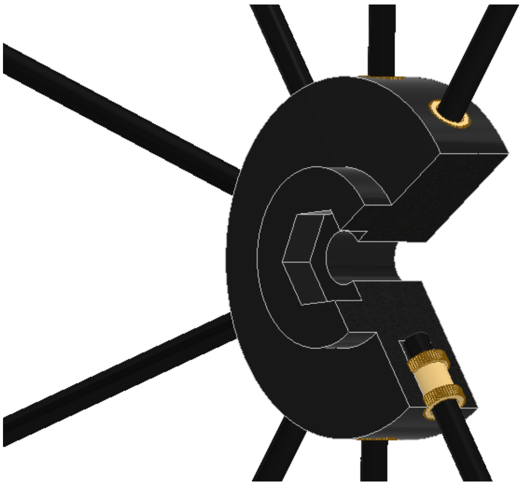


Figure 5.10 Rake hub 3/4 section with inserts and rods

After assembling the belt transmission, the need to tighten the belt arose. A belt tensioner was therefore designed again using parts present in the lab. As visible in Figure 5.11, it is made out of three 90° corner angles, one 6mm bolt, two locknuts, and three T-slotted fasteners. One 90° corner angle is free to move along the arm facing against the side of the bearing support; the other two 90° corner angles are fixed with an offset of 2cm in between. The need for two fixed elements arose during the test due to the misalignment of the bolt with respect to the tensioning direction. There is one locknut at the end of the bolt which pushes the moving 90° corner angle against the bearing support, while a normal nut allows keeping the position by touching the fixed 90° corner angle. The resultant mechanism allows the regulation of the tension of the belt by screwing or unscrewing the bolt.

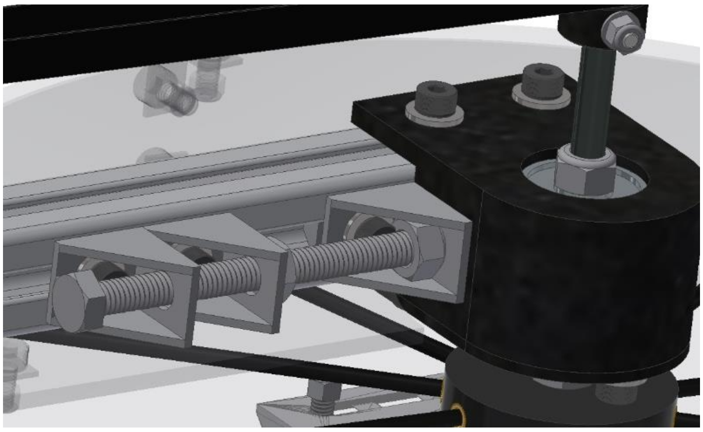


Figure 5.11 Belt tensioner

As explained before the shafts were built out of threaded rods as the trade-off is good enough for the realization of a prototype whose goal is just to prove the correct functioning of the tool and not work on the field. At first, the 6mm threaded rod was cut to the right length by a hand saw and then was ground at the ends by rasps to ensure a smooth pass of the nut after the damaging of the thread in the cutting area. To secure the belt pulley from losing up while spinning the design of a locking bolt was established. The locking bolt is perpendicular to the shaft and goes through the pulley and the shaft itself (Figure 5.12), to do that a 3mm hole had to be realized in the threaded rod. Firstly, the thread in correspondence with the area where the hole had to be placed was removed with a rasp to obtain a flat surface. Then, to ensure that the drill was going in the exact desired location, a centre point has been made using a centre puncher. Finally, the hole was drilled using a stand drill.

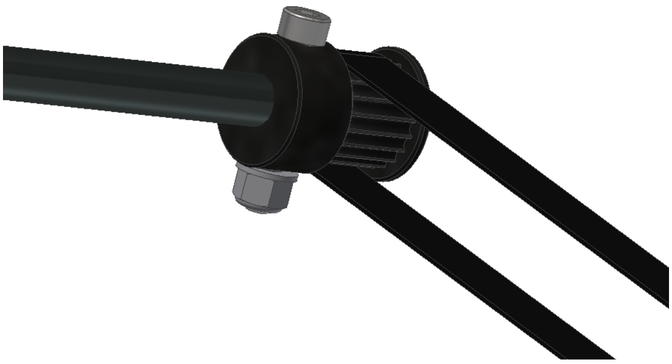


Figure 5.12 Locking bolt on rake shaft

The leverage arms have to tolerate the momentum and forces coming from the arm of the robot and the servo motor. Aluminium was chosen for the leverage arms for its light weight, strength, and environmental conditions tolerance. The raw material used was a 2mm thick and 15mm wide aluminium beam found in the lab stock. The leverage arm is connected to the servo motor through the servo horn connector which can be seen in Figure 5.13. The holes of the connector arm are very small and accurate, they required very high precision while drilling because any small free movement between the connector arm and the leverage arm might result in failure of the parts and not proper functioning of the whole system. To realize the holes the first step was again with the realization of a centre punch, this allowed to measure the distance between the centre of the holes and after to drill them. Once realized the holes, the leverage arm was cut using a hand saw to the right dimension. After being drilled and cut the leverage arm was sanded using a rasp to smoothen the edges for safety reasons. The second leverage arm was made in the same way just with only two holes instead of 4.

Figure 5.13 Servo horn

5.1. Electronics connections

As previously described, the proposed concept is based on the Arduino UNO micro-controller. The robot can supply different voltages as visible in Figure 5.14; for our system 12V supply and I²C bus are required.

	Pin	U (V)	I_{max} (A)
Data	1	SDA	
	2	SCL	
	3	5V	
	4	GND	
Power	1	5	5
	2	24	3
	3	GND	
	4	12	5
	5	18	6,3

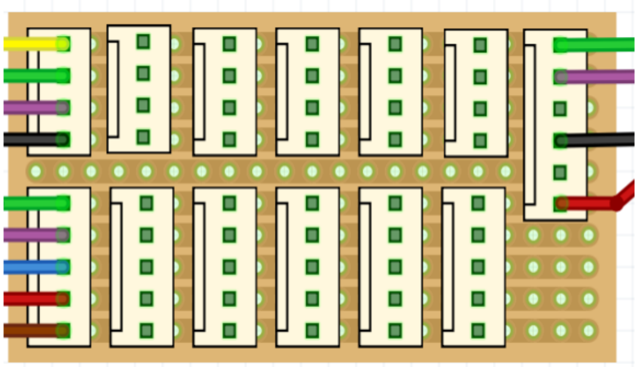


Figure 5.14 Robot connection interface

Arduino can be powered with a 12V voltage supply through two dedicated pins; in addition, the step-down voltage converter receives as input 12V and regulate the voltage supply to the DC motor used to rotate the rake; by that, it is possible to set the optimal rotating speed of the rake.

As the first step in the electronics implementation, the circuit has been cabled using a breadboard; this allowed to verify the correct functioning of all the devices and to make the modifications of the circuit easier to reach the final layout. In this initial phase, the power supply was taken from the connection between Arduino and the computer (5V) and a Lab power supply (12V).

In Figure 5.15 the testing circuit diagram has been reported, and for the sake of simplicity in Figure 5.16 the connection schematic of Arduino with all the devices has been represented. Both representations have been done using the Fritzing software [13].

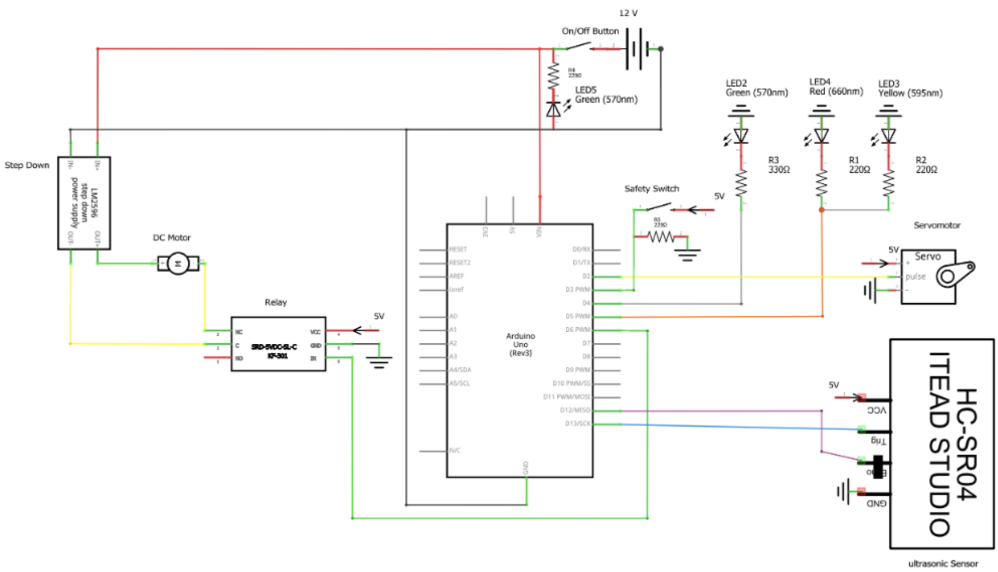


Figure 5.15 Testing phase - circuit diagram [13]

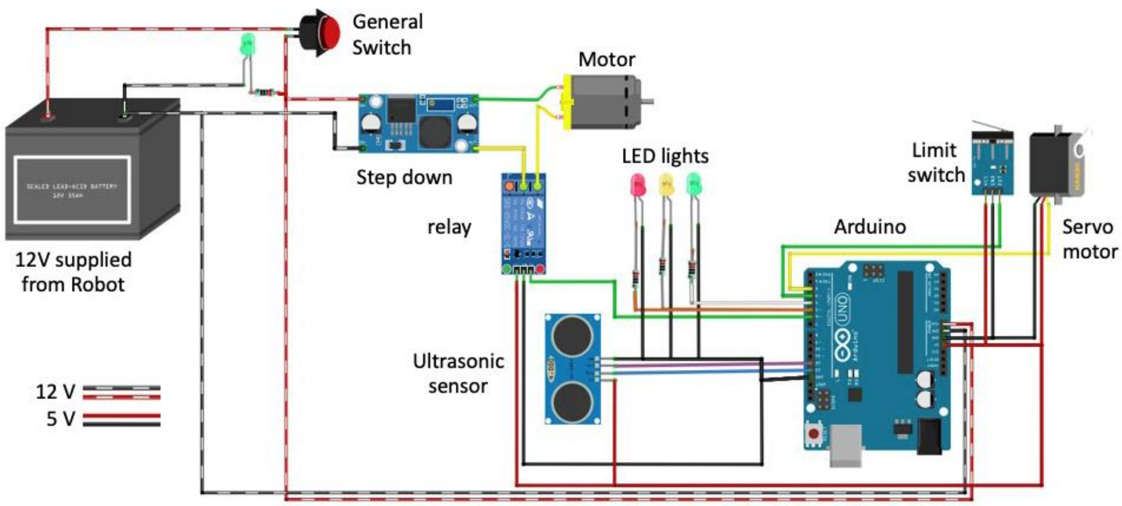


Figure 5.16 Testing phase - connection schematic [13]

Once proved the correct functioning of the system, after having implemented the first draft of the code, the group started to realize the final circuit by using the equipment available in the workshop.

All the components and sensors have been placed in the correct position, and a plastic electric box has been selected to safely store the device and wirings on the robot frame. Depending on the specific position of the part, electric cables have been cut with the proper length to guarantee easy accessibility and connection, without interfering with the moving part. For the same purpose heat shrink tubing and spiral wire, wrapping tubes have been used to avoid messy cables and to ensure a fixed positioning of the latest. In the metal plate, one hole for each side has been foreseen to reach the plastic box placed on the upper part of the robot.

For what concern the connection between components and cables, depending on the specific device, female/male Dupont connectors have been properly mounted on the wire's terminal or, as an alternative, a soldering connection has been realized.

Multiple rapid connections between parts of the circuit have been foreseen, using pin connectors, to allow easy maintenance, easy assembling operations, and fast connection of the mechanical device with the vehicle. Moreover, due to the presence of only one 5V pin and 2 GND pins on the Arduino board, and the need to connect more pins than the available pins, two "power strips" have been realized to provide 6 output female connectors starting from one single pin plugged into the board.

Finally, two rapid attachments have been realized to provide the connection, both for the power supply and the I^2C communication, between the robot interface and the circuit of the machine tool. In particular, through the I^2C BUS, the Arduino board can exchange information and command with the robot allowing the latter to move using one single algorithm uploaded into the microcontroller.

In Figure 5.17 and Figure 5.18, the Arduino final circuit representation and its electric schematic, designed through the Fritzing software [13], can be observed.

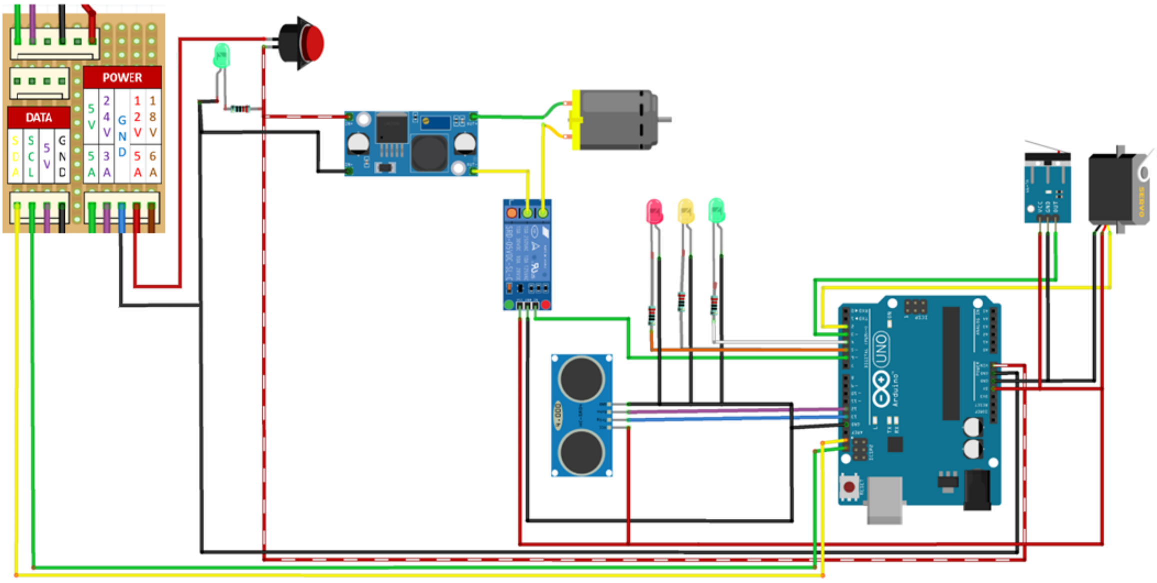


Figure 5.17 Final connection schematic [13]

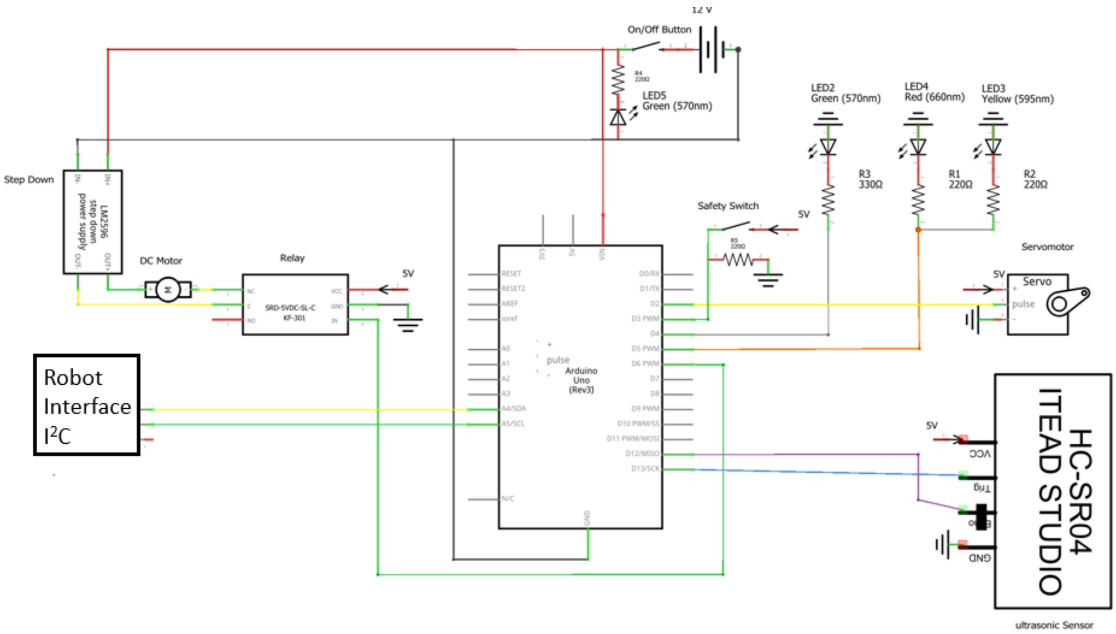


Figure 5.18 Final circuit diagram [13]

The result is visible in Figure 5.19, all the electrical components are placed inside the plastic box and the connection with the robot can be realized using the two connectors on the left side of the picture, one for the power supply and one for the BUS communication.

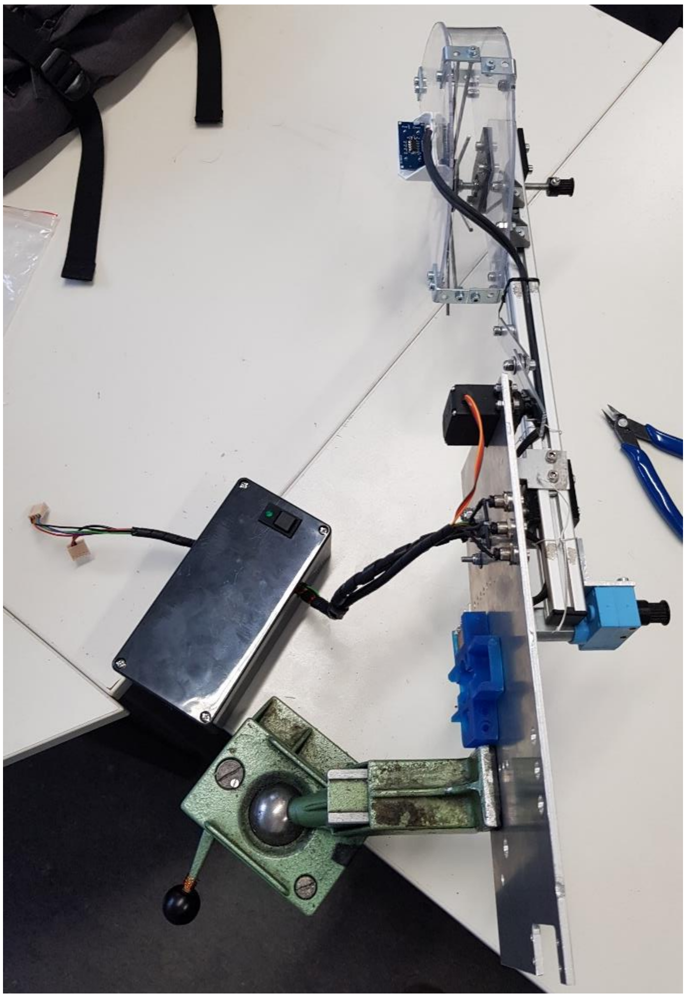
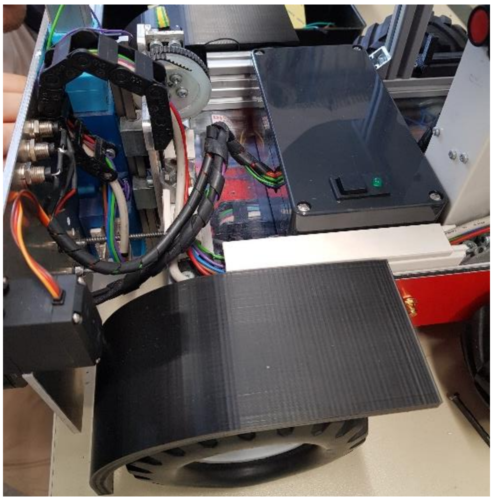


Figure 5.19 Overview of the mechanism and electrical connections



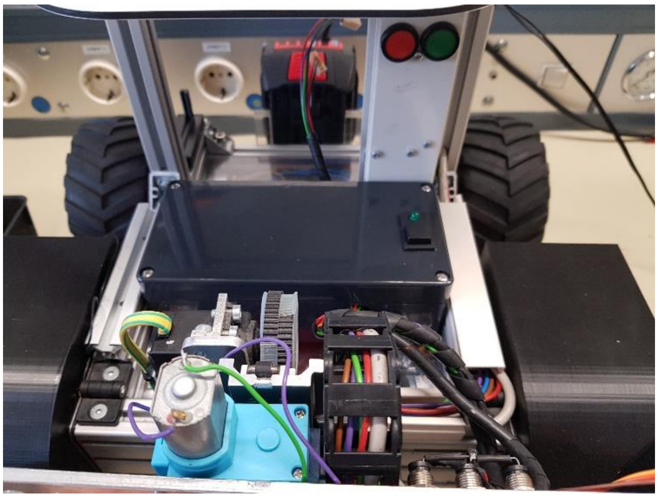


Figure 5.20 Position of the electric box onto the robot surface.

6. Software Logic of the Concept

In Figure 6.1, the software flow chart for the test phase is reported. The main difference with respect to the real device one is the condition that is checked to start the weed removal procedure. Since the weed detection system is not ready yet, the input for the system is a temporal function in which the weed removal procedure is repeated every 3s independently by the presence of the weed or crop.

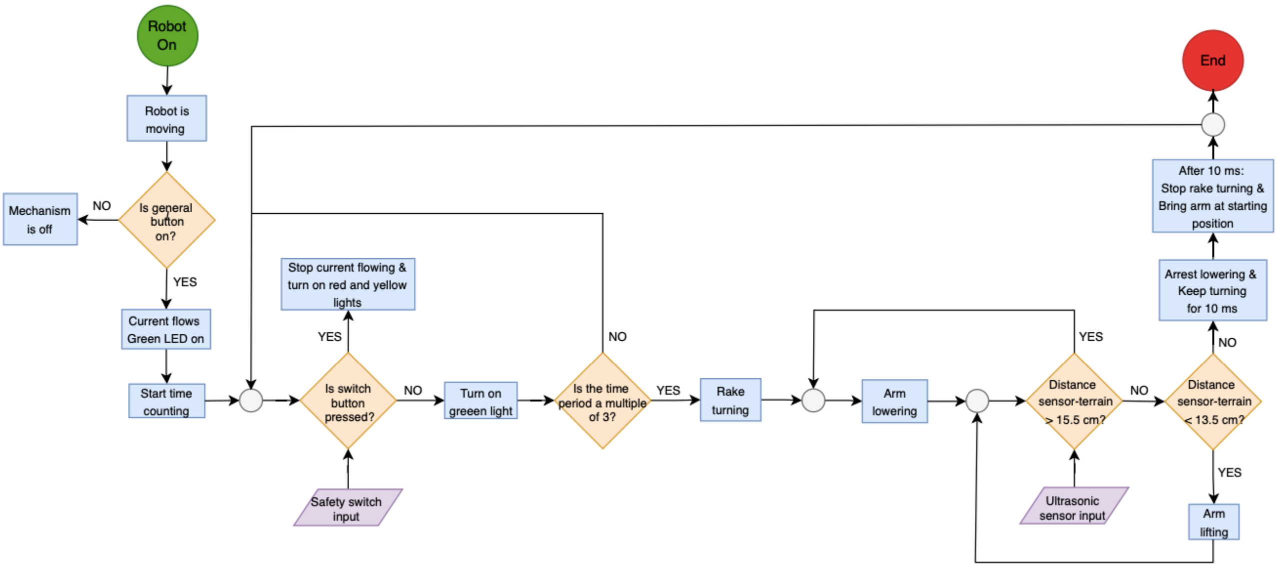


Figure 6.1 Testing phase - Software logic

In particular, the condition *Reminder of the division ≤20ms* is needed by the program due to the high running speed of the code. It is quite difficult for the software to catch exactly the time with zero as a reminder of the division therefore, after some tests, the interval of *±20ms* has been settled as the range of acceptable values.

Firstly, a general switch controls the power supply to the system; when this button is pressed, a first green LED is turned on representing the presence of current in the circuit. The power will therefore flow towards Arduino and the DC motor. As soon as the system is powered, both yellow and red LEDs will start flashing while all the devices are being checked, only once that process ends the green LED is turned on meaning that the system is ready to work, and the red and yellow LEDs are turned off. Subsequently, the controller has to check whether the safety switch is pressed or not. It has been strategically placed on the plate to verify that the arm does not exceed the upper limit imposed by the code for its rotation and that the operation of the mechanism is under control. If the system detects pressure on it, the flow of current from Arduino to all the actuators will cease, both yellow and red LEDs will turn on, and the command to go back to the servo home position will be sent. Until the initial position is not reached, the red and yellow LEDs will be on, and all the other actuators of the system are kept stopped. Whereas the correct functioning of the arm will be defined by a second green LED. This will be the first real input of the system. The yellow LED has been placed also for further implementation with the robot micro-controller; for example, one may think to turn it on while the robot is driving.

For the initial concept phase, the presence of weed has been represented by manual input from the keyboard of the computer connected to the microcontroller. Once all the circuit has been settled and placed on the robot, the above described time function has been coded in the program of Arduino since it was not possible to have a computer connected with it anymore.

Once the weed has been detected, the DC motor is turned on via the relay, the rake will start turning, and the servomotor will gradually lower the arm, thanks to a loop in Arduino code. At the same time, the ultrasonic sensor will constantly measure the distance between the tip of the arm and the ground. If the distance is greater than a specific value, the arm will continue to be lowered, instead, if it is less than a higher value, it will be lifted until it goes inside the predefined range. After on-field testing, the optimal distance to the soil has been defined at 14.5 cm, and a range of ±1cm has been selected. On the other hand, when the distance is within the set values (13.5 and 15.5 centimetres), meaning the rake is at the right working position, the arm will stop for a few seconds to proceed with the actual weed removal. Once the stop time has passed, the relay will stop the DC motor, ceasing the rotation of the rake, and the arm will be promptly lifted to the initial position, getting ready for the new weed detection.

In Figure 6.2 the ideal software flow chart has been reported. This would be implemented once the weed detection system is ready and merged with the work here described.

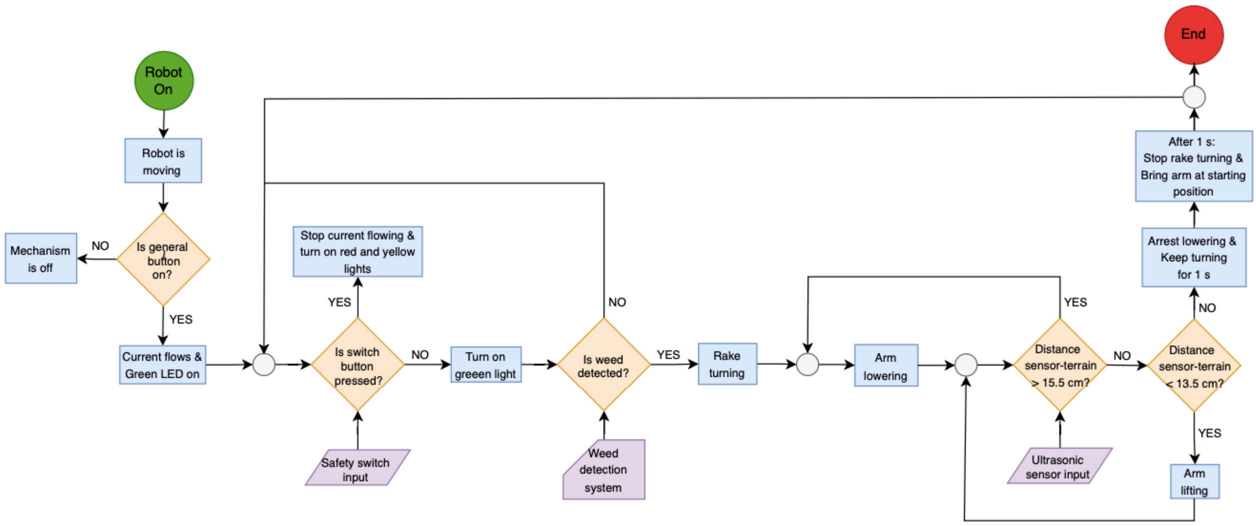


Figure 6.2 Ideal Software logic

6.1. Algorithm

The mechanism control algorithm can be found in *Appendix 1* section.

The complete control algorithm, including driving commands for the on-field testing phase, can be found in *Appendix - 2* sections.

The .ino file can be found in the folder "*Arduino_Code*".

7. Part list and costs

ELECTRICAL COMPONENTS							
ATB inventory No.	Item name	Qty	Description	Source	Link	Virtual price	Physical price
	Servomotor MG995 + accessories	1	Lifting arm mechanism	Ordered	CONRAD	-	€ 9.24
	DC Motor Makeblock 180	1	To spin the rake	Lab stock	Amazon	€ 12.99	
	Limit Switch / Safety switch	1	Safety purpose	Lab stock	Amazon	€ 0.36	
	Ultrasonic Sensor HC-SR04	1	Evaluate the distance between arm-soil	Lab stock	ECKSTEIN	€ 1.39	
	Jumper wires	//		Lab stock			
	Start switch	1	To initiate the system	Lab stock	Amazon	€ 0.78	
	Resistors	//	For LEDs	Lab stock	-		
	Breadboard	1	For testing purposes only - COST NOT CONSIDERED	Lab stock		€ 0.00	
	LED holder	3		Lab stock		€ 0.30	
	LED	3	Lab stock: Red, Yellow, and Green	Lab stock		€ 0.30	
	Plastic box 160x80x55	1	It contains the electrical components that are not fixed on the arm/plate	Lab stock	Amazon	€ 1.38	
	Cable tie	6		Lab stock	Amazon	€ 0.68	
	Electrical Wires		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
	Dupont Connector	20		Lab stock	Amazon	€ 0.35	
	Tape		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
	Spiral wire wrapping		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
	Heat shrink tube		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
	Soldering material (Tin)		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
	Hot glue		Difficult to estimate the exact quantity - COST NOT CONSIDERED	Lab stock	-	€ 0.00	
6	Arduino Uno microcontroller	1		ATB stock	Amazon	€ 10.00	
49	5V ONE Channel Relay Module Board	1	To control DC motor	ATB stock		€ 2.00	
53	LM2596 DC-DC Module Step Down Converter Voltage Regulator	1	To fix the DC motor speed	ATB stock		€ 3.00	
					TOT - ELECTRICAL COMPONENTS	€ 33.53	€ 9.24

Table 7-1 Part List: Electrical Components

MECHANICAL COMPONENTS						
Item name	**Qty**	**Description**	**Source**	**Link**	**Virtual price**	**Physical price**
Aluminum extruded profiles 20x20	1	39cm - 25€/m + 0,25€ piece	Lab stock		€ 1.91	
End Caps Plastic Black	2		Lab stock	Amazon	€ 0.53	
T-slot 2 Hole Corner Angle 90 Degree	4	triangular pieces	Lab stock		€ 6.00	
M4 x 8 Screws	40	shield 16x, bearings supports 10x, triangular parts 4x, motor support 4x, proximity sensor support 2x	Lab stock	Amazon	€ 1.20	
M4 x 16 Screws	2	safety switch 2x	Lab stock	Amazon	€ 0.06	
M4 Nuts	20	shield 16x, proximity sensor support 2x, safety switch 2x	Lab stock	Amazon	€ 0.60	
M4 Washers	22	bearings support 10x, triangular parts 4x, Frame attachment 8x	Lab stock	Amazon	€ 0.66	
M3 x 150 threaded rod	10	rake sticks 10x: 1.83 €/m	Lab stock	MCmaster	€ 2.75	
M3 x 20 Screws	1	Pulley rake 1x	Lab stock	Amazon	€ 0.03	
M3 Nuts	1	Pulley rake 1x	Lab stock	Amazon	€ 0.03	
M3 Washers	1	Pulley rake 1x	Lab stock	Amazon	€ 0.03	
T-Slotted Framing Fasteners	23	bearing supports 8x, triangular pieces 4x, motor support 2x, lifting mechanism 1x, shield 2x, frame attachment 6x	Lab stock	DOLD	€ 14.95	
M6 Nylon Insert Locknut	9	bearing supports 4x, aluminum plate 2x, rake 2x, belt tension 1x	Lab stock	MCmaster	€ 0.54	
M6 Nut	1	belt tension 1x	Lab stock	MCmaster	€ 0.05	
M6 threaded rod	2	110mm 1x, 70mm 1x	Lab stock	MCmaster	€ 2.64	
M2 x 14 screw	1	Pulley motor 1x	Lab stock	Amazon	€ 0.01	
Retaining Rings ⌀6	2	M6 threaded rods 2x	Lab stock	MCmaster	€ 0.14	
M3 thread insert	10	rake sticks 10x	Lab stock	Amazon	€ 0.50	
M4 thread insert	2	Bearing supports 2x	Lab stock	Amazon	€ 0.10	
Bearings 6mm int, 22mm ext., 7mm	4		Lab stock	Amazon	€ 5.40	
Aluminum bars	2	Lifting mechanism	Lab stock	MCmaster	€ 0.96	
M4 x 30 Screws	4	Extra screws for frame attachment	Lab stock	Amazon	€ 0.12	
Transmission Belt - A250615 HTD 753 3M	1	width 9 long 720	Lab stock	DOLD	€ 6.99	
Plexiglas Plate	1	1 disk D=120 t=2 + rectangle 60x360 t=1 + rectangle 50x20 t=2	Lab stock	MCmaster	€ 10.30	
Aluminum 90° rod	1	Motor support	Lab stock	MCmaster	€ 0.20	
90° inner corner connector	6	For Plexiglas shield - M4 holes	Lab stock	MCmaster	€ 0.51	
M6 x 70 bolt	1	Belt tensioner	Lab stock	MCmaster	€ 0.24	
Bearing support	2	FDM Printer	Printed		€ 15.96	
Bearing support Cover	2	FDM Printer	Printed		€ 7.44	
Spacer	2	FDM Printer	Printed		€ 4.26	
Rake	1	FDM Printer	Printed		€ 4.25	
Pulley motor	1	FDM Printer	Printed		€ 2.55	
Pulley rake	1	FDM Printer	Printed		€ 2.57	
Aluminum Plate	1	CNC	Lab stock		€ 29.36	
Nail	2	safety switch 2x	Lab stock	MCmaster	€ 0.02	
Ultrasonic sensor support	1	FCM Printer	Printed		€ 2.29	
				TOT - MECHANICAL COMPONENTS	**€ 126.14**	**€ 0.00**
				TOT - CONCEPT	**€ 159.67**	**€ 9.24**

Table 7-2 Part List: Mechanical Components

8. Testing phase

Not all the components were realized as initially planned in the CAD design, and some of them were realized only after an initial real-life test.

The safety switch was extremely tricky to design and realize, initially it was planned to be placed in contact with the bearings hub but after realizing the physical model, the group figured out that the sensor would have impacted the arm movement. By trial and error, the final position was found as well as the right curvature of the metallic plate that triggers the switch Figure 8.1.

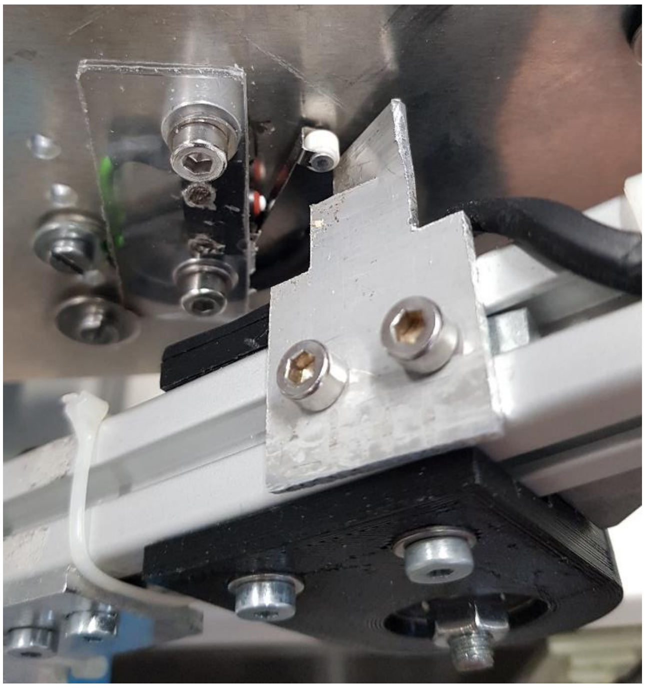


Figure 8.1 Safety switch

The lifting mechanism required some trial-and-error attempts to find the right length of the arms and the actual position of the centre of gravity of the mechanism.

Some of the extra holes on the plate done to attach it directly to the frame cannot be used because electronics wires exiting from the tool changing block them. This resulted in only four out of six screws available for the frame attachment, after some real-life tests the group did not notice any problem in this aspect. Moreover, in the last few weeks, the clamp of the tool changing system got broke and even in this situation, only four screws were enough to bear all the load.

The testing scenario visible in Figure 8.2 consist of a straightforward path (4 m) with on the right side a small dune made with the terrain found outside the lab. On the top were placed some leaves to simulate weed in the early stage.

Figure 8.2 Testing scenario

The first test was established to estimate the correct distance between the ultrasonic sensor and the soil to obtain a depth value able to sweep the weed.

On the code side our project does not have real input, the robot does not know when and where weed is detected. To overcome this problem a time sequence that controls the robot's behaviour was written. It consists of arm up for *"x"* seconds and arm down for *"y"* seconds, the original idea was to set *"x"* and *"y"* parameters and the velocity of the robot to match the *"x"* time with the time in which the robot would have been in the carrots sector and the *"y"* time with the time in which the robot would have been in the weed sector. In the future *"x"* and *"y"* times could be set by sensors that scan the environment, this might be done by implementing the work done by the Group 4 - Weed detection system. However, after the non-availability of the robot for maintenance, the code on it changed, and it was not possible to decrease its velocity, the servomotor up and down movement is too slow to properly allow the robot to work. Now the velocity is set at 2km/h a double value with respect to the required one of 1km/h. Since the robot was developed with 2 arms the original idea was to set its speed at 0.5km/h, overall the results would have been the same: 2 rows at half the speed, versus one row at a normal speed. During field tests, this results in a too long *"y"* time for a real scenario Figure 8.3, however by considering a quarter of that time given by using a quarter of the speed (i.e. the initial supposed value) the value seems reasonable.

Figure 8.3 Swept distance - 15 cm

The last improvement done on the robot was related to the weight balance. In fact, during the testing phase the robot was initially following the right direction: the load was pushing on the right side resulting in the left wheels slipping. A weight was then placed on the left side of the plate to simulate the second arm, initially in a different position by trial and error, until the group was satisfied with the straight trajectory of the robot. Finally, a brick was placed on the front part to avoid wheelies Figure 8.4.

Figure 8.4 Final configuration with balancing loads

During the last weeks, a problem arose on the robot, the I^2C Bus module of the driver micro-controller was burned. Dr. Redmond solved the problem by replacing the original driver with an additional Arduino equipped with a four-channel motor driver shield. Due to this change, the output interface of the robot was not working anymore, therefore the already planned connector for the power supply and the BUS communications were no more suitable. The 18V battery of the robot was therefore supplied only to the motors and the onboard Arduino.

Initially, the new configuration supplied only 5V for external applications. However, the system proposed has been designed taking into account a supply of 12V. The reduced voltage was not enough for powering the DC motor for the rake to properly scratch the soil. An external 12V battery taken from the lab, after the realization of new connections with the already built system, solved the DC motor issue. However, during the first test ever made with Arduino not plugged in a PC, the servomotor showed a not expected behaviour. At the moment the group was not able to clearly understand the issue generated by the different voltage supplied, even Tobias Mach was surprised by the behaviour and did not manage to solve it. As a backup plan the group came out with a workaround to avoid the issue: power Arduino with 5V through a power bank and the DC motor with the 12V battery through the step-down module. The solution is represented in Figure 8.5 where the three power sources have been marked.

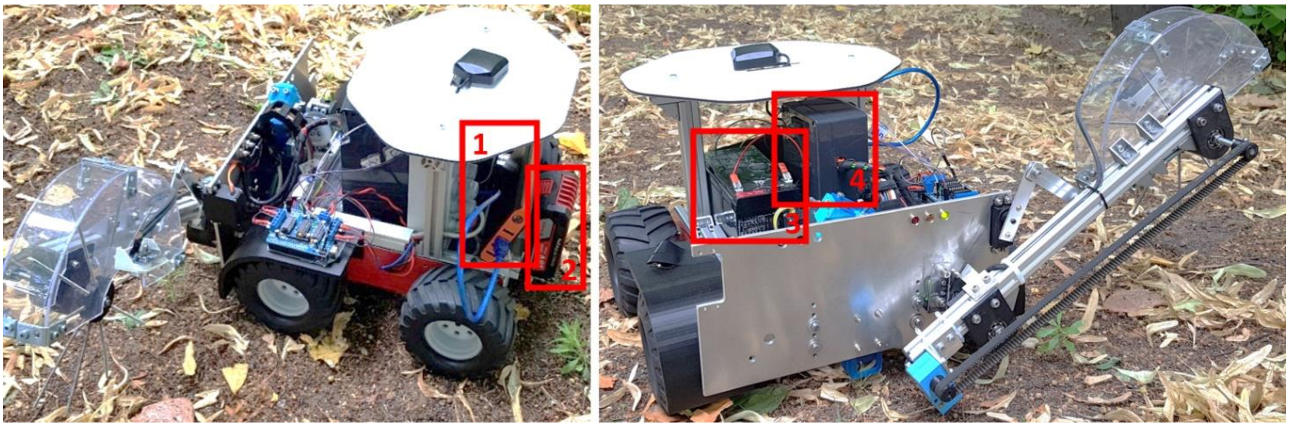


Figure 8.5 Backup configuration for test: (1) power bank, (2) 18V battery, (3) 12V battery, (4) electric box

After the on-field test, the responsible for the electrical part analyzed in depth the issue and found a solution. The problem was the Arduino's 5V voltage regulator which cannot supply enough stable current for the servo if it is not supplied by the USB cable; when the servomotor runs, the 5V was probably dropping. The output voltage from the step-down module used for the DC motor has been set to ~5.8V; this value leads to the optimal spinning speed for the rake. Therefore, the latter has been considered a good supply voltage for the servomotor, being its maximum input voltage of 6V. The new circuit has been realized and validated after several lab experiments. Moreover, the robot has been equipped with a 14.8V Li-Po battery by the lab technicians, including a step-down module to obtain a 12V supply for the external devices to mount on the robot. Thus, a new version of cable connection between the new step-down module and the presented circuit has been realized and the final system tested. Figure 8.6 is showing the final electrical connection interface on the robot; due to the new components on the robot the electrical box cannot fit properly anymore. In Figure 8.7 and Figure 8.8 are instead reported the new schematics of the circuit.

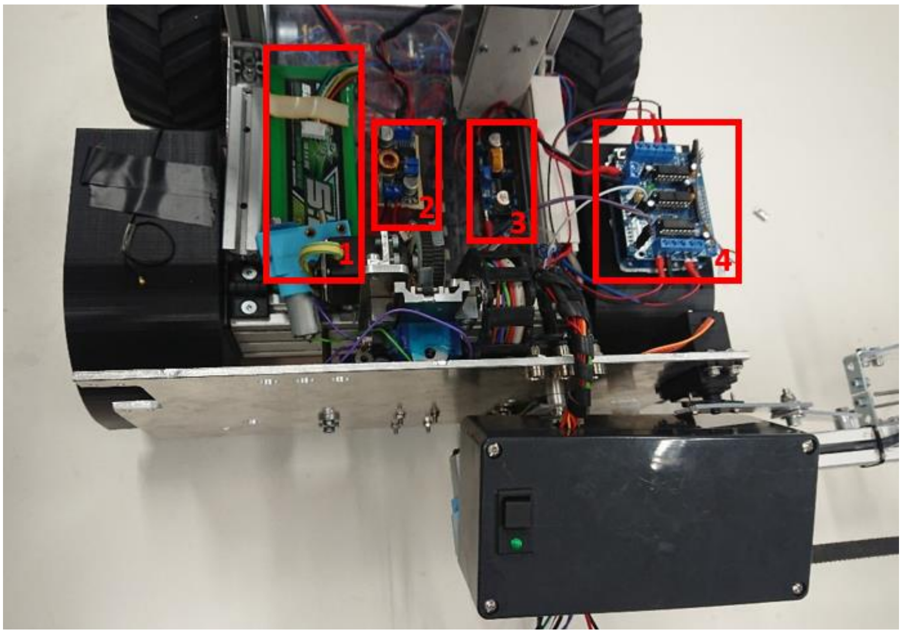


Figure 8.6 New electrical connection interface: (1) LiPo Battery, (2) Step Down for external devices, (3) Step Down for Arduino power supply, (4) Onboard Arduino with motor drivers

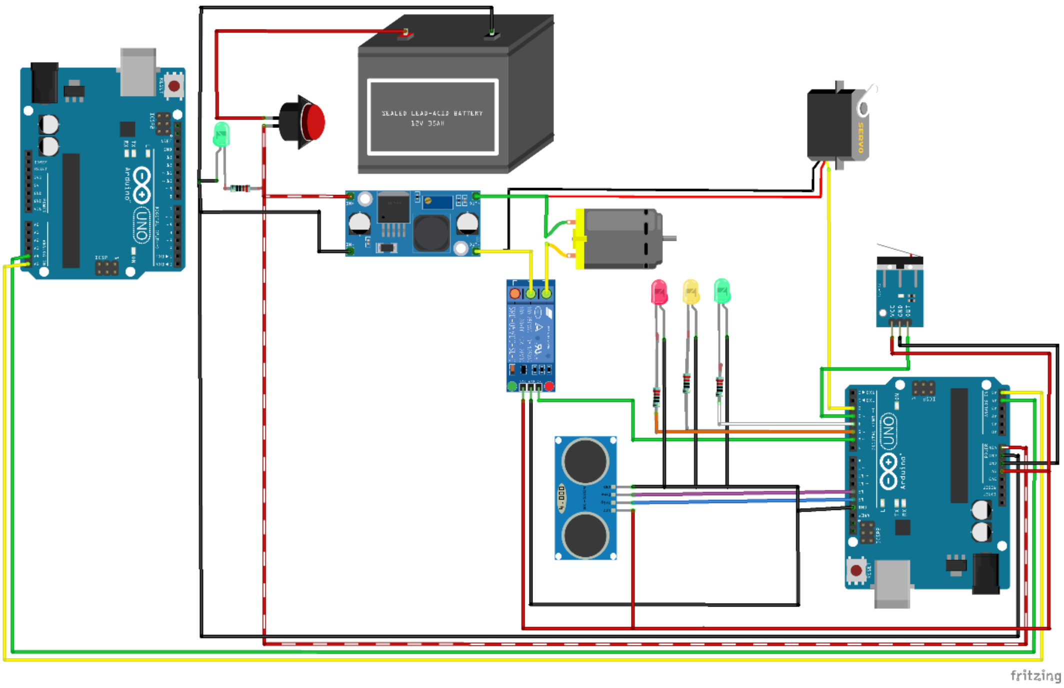


Figure 8.7 Final connection schematic [13]

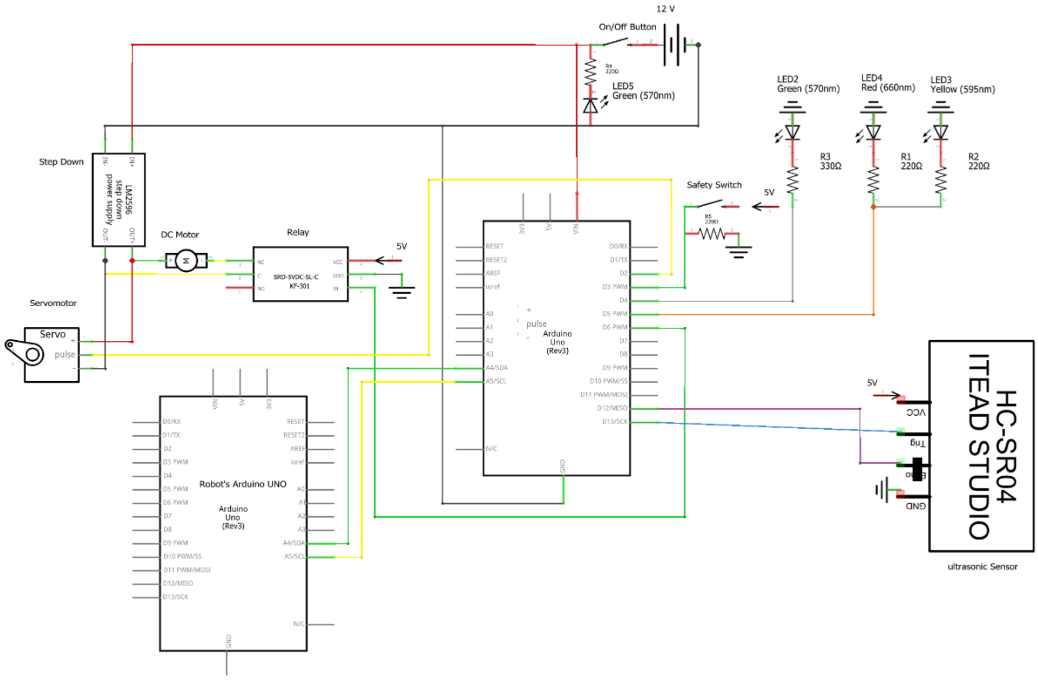


Figure 8.8 Final circuit diagram [13]

9. Conclusions and further developments

The developed concept met most of the targets identified at the beginning of the project, achieving extremely satisfactory results, and confirming the possibility of realizing an automatic tool dedicated to weed removal.

On-field tests validated the robustness and reliability of the design, ease of maintenance, and adaptability to different fields. Moreover, the project was completed respecting all deadlines and the 30-euro budget, taking advantage of the tool changing system and the power supply voltage available from the robot.

Once the robot is repaired, the presented mechanism will be ready to be mounted and connected directly with the original output interface.

As explained, the on-field test of the tool was carried out with the robot equipped with a temporary driver solution. The moving functions in the Arduino code and the interface connection with the tool's microcontroller have been changed due to the shortcut. Therefore, the group had to face a fixed speed of 2km/h, measured in the testing field, which is doubled value with respect to the original one. These conditions brought to a result of circa 15cm as scratching distance. However, the complete concept would involve the simultaneous use of two arms, therefore the group is expecting to set a speed of 0.5km/h. Hence, the achievable scratching precision of the concept would be 3.75cm, which results in the almost total achievement of the initial goal set at 3cm.

As further developments, to improve safety and system stability, a chassis that isolates the belt from external agents could be designed. Otherwise, the lifting arm could be entirely changed to fit the belt inside of it as happens in the already available applications on the market. The belt inside the arm not only would protect the belt from external dirt or the external workers from getting hurt, but also removes the overhanging pulley resulting in a more stable and compact product.

The only main initial target that has not been achieved is the capability of withstanding environmental aggressions: it has not been considered of major importance in the development of the concept since the group wanted first to show the correct functioning of the system. For the continuation of the project, the protection of electronic parts using sealing covers could be implemented.

Finally, as already written, at the moment the system does not have real input. When the weed detection system under development by Group 4 is ready, it will be possible to merge the codes in order to lower the arm exactly when a weed is detected. Moreover, when all the groups will finalize their project, a fully autonomous system can be achieved.

10. Bibliography

[1] G. F. R. B. Peruzzi, “Innovative strategies for on-farm weed management in organic carrot,” *Renewable Agriculture and Food System,* 2006.

[2] D. T. G. F. Kielhorn A, “Beikrautregulierung in Reihenkulturen,” *Sensorgesteuerte Querhacke im Mais Zeitschrift für Pflanzenkrankheiten und Pflanzenschutz,* 2000.

[3] S. P. V. e. a. van Evert, “A Robot to Detect and Control Broad-Leaved Dock (Rumex obtusifolius L.) in Grassland,” *Plant Research International,* 2010.

[4] “Arduio Uno Rev3,” [Online]. Available: https://store.arduino.cc/products/arduino-uno-rev3. [Accessed May 2022].

[5] Shawn, “Arduino Ultrasonic Sensor Overview and Tutorials,” [Online]. Available: https://www.seeedstudio.com/blog/2019/11/04/hc-sr04-features-arduino-raspberrypi-guide/. [Accessed May 2022].

[6] “Momentary Hinge Roller Lever Micro Limit Switch,” Amazon, [Online]. Available: https://www.amazon.de/-/en/HUAYAO-pieces-momentary-roller-changeover/dp/B08VDJMLSR/ref=sr_1_5?crid=26FP2Q14AIUYK&keywords=endschalter&qid=1655121393&s=diy&sprefix=limit+switch%2Cdiy%2C81&sr=1-5. [Accessed May 2022].

[7] “LM2596 DC-DC Module Step Down Converter Voltage Regulator,” Amazon, [Online]. Available: https://www.amazon.de/-/en/Converter-Regulator-Efficiency-1-25-35V-Adjustable/dp/B07F38DJLS. [Accessed May 2022].

[8] “Single channel 5V relay module,” [Online]. Available: https://www.mybotic.com.my/module/relay-module/single-channel-5v-relay-module. [Accessed May 2022].

[9] “Mini Rocker Switch, 6A Illuminated SPST Off-On with 12V Green LED,” Gamma Electronics, [Online]. Available: https://www.gamainc.com/product/23led-green/. [Accessed May 2022].

[10] “5MM LED,” Conrad, [Online]. Available: https://www.conrad.de/de/p/kingbright-ld-20-gb-5mm-led-bedrahtet-gelb-rund-5-mm-20-mcd-60-20-ma-2-1-v-184900.html. [Accessed May 2022].

[11] “SERVO – MG995,” WHADDA, [Online]. Available: https://whadda.com/product/servo-mg995-wpm603/. [Accessed May 2022].

[12] “Makeblock 180 Optical Encoder Motor,” Makeblock, [Online]. Available: https://www.amazon.de/-/en/Makeblock-180-Optical-Encoder-Motor/dp/B07ZKRV82J. [Accessed May 2022].

[13] “fritzing software,” [Online]. Available: https://fritzing.org/download/.

Appendix - 1

Appendix 1 - Arduino code – Only for mechanical device

```
#include "SR04.h"                         //Include Ultrasonic sensor library
#define TRIG_PIN 1                        //Set pin 12 as trig
#define ECHO_PIN 12                       //Set pin 11 as echo
#include <Servo.h>                        //Include servo library
  Servo Lifting_servo;                    //Declare servo object to control

const int Emergency_red_LED = 6;          //On pin 6 red LED is connected
const int Emergency_yellow_LED = 5;       //On pin 5 yellow LED is connected
const int Running_LED = 4;                //On pin 5 green LED is connected
const int Safety_Switch = 3;              //On pin 3 limit switch is connected
const int Relay = 7;                      //Relay on pin 7 activates the DC motor
const int Home_servo =90;                 //Declares the starting position of the servo
int angle = 0;                            //Variable needed to control the servo
int alert = 0;                            //To exit from the loop if safety switch pushed
unsigned long Running_Time;         //Total count of seconds since the board is powered on

//%%SETUP INPUT%%
const int scratching_time = 10;       //time rake stays on the soil scratching [ms]
const int Stop_distance = 14.5;         //cm distance to the soil to stop the arm lowering
unsigned long time_step = 3000;         //time step between each lowering operation [ms]

SR04 Ultrasonic_S = SR04(ECHO_PIN,TRIG_PIN);         //Define the ultrasonic sensor
long distance;                                       //Define distance read by the sensor

void setup()
  {
      Serial.begin(9600);                            //Initialize the serial communication
with the computer
      Lifting_servo.attach(2);                       //Set pin 2 as servo control
      pinMode(Emergency_red_LED, OUTPUT);            //Set pin as output
      pinMode(Emergency_yellow_LED, OUTPUT);         //Set pin as output
      pinMode(Running_LED, OUTPUT);                  //set pin as output
      pinMode(Safety_Switch, INPUT);                 //set pin as input
      pinMode(Relay, OUTPUT);                        //set pin as output


  }

void loop()
  {
     Running_Time = millis(); //Total count of ms since the power on of the board

     Lifting_servo.write(Home_servo);                       //Start from home position
     digitalWrite(Relay, HIGH);                             //Turn off the motor

     int Safety_Switch_status = digitalRead(Safety_Switch);      //Read the switch status

     if (Safety_Switch_status == LOW)                     //If emergency switch pushed
        {
          digitalWrite(Running_LED, LOW);                 //Turn off the green LED
          digitalWrite(Emergency_red_LED, HIGH);          //Turn on the red LED
          digitalWrite(Emergency_yellow_LED, HIGH);       //Turn on the yellow LED
          digitalWrite(Relay, HIGH);                      //Turn off the motor
          Lifting_servo.write(Home_servo);                //Return to home position
        }
     else if (Safety_Switch_status == HIGH)               //If emergency switch not pushed
        {
          digitalWrite(Emergency_red_LED, LOW);           //Turn off the red LED
          digitalWrite(Emergency_yellow_LED, LOW);        //Turn off the yellow LED
          digitalWrite(Running_LED, HIGH);                //Turn on the green LED
```

```
        Serial.print(Running_Time); //Print the time passed since the board is powered
        Serial.print("\n");

        if (Running_Time % time_step <=20)  //If a time frame circa equal to time step
                                               has passed
            {
                digitalWrite(Relay, LOW);               //Turn on the DC motor
                distance = Ultrasonic_S.Distance();     //Read distance from the sensor

                for (angle=Home_servo; (distance < Stop_distance-1 or distance >
Stop_distance+1) and (0<=angle<=180)and (alert==0);)  //When the distance arm-ground
                                                        needs to be correct
                  {
                    distance = Ultrasonic_S.Distance();      //Read distance from
ultrasonic sensor
                    Serial.print(distance);      //Print on the monitor the distance
                    Serial.println("cm");        //Print the measurement unit
                    if (distance < Stop_distance-1)   //If it's too close to the ground
                      {
                        Lifting_servo.write(angle);   //Servo is lifting the arm
                        angle++;
                      }
                    else if (distance > Stop_distance+1)       //If it's too far
                      {
                        Lifting_servo.write(angle);      //Servo is lowering the arm
                        angle--;
                      }

                    int Safety_Switch_status = digitalRead(Safety_Switch);
                                                 // Read the status of the switch

                    if (Safety_Switch_status == LOW)      //If emergency switch pushed
                      {
                       alert=1;
                      }
                  }
                if (alert==1)
                  {
                    digitalWrite(Running_LED, LOW);            //Turn off the green LED
                    digitalWrite(Emergency_red_LED, HIGH);     //Turn on the red LED
                    digitalWrite(Emergency_yellow_LED, HIGH); //Turn on the yellow LED
                    digitalWrite(Relay, HIGH);                 //Turn off the motor
                    Lifting_servo.write(Home_servo);           //Return to home position
                  }
                if (alert==0)
                  {
                    delay(scratching_time);      //Wait: the rake is scratching the soil
                    digitalWrite(Relay, HIGH);        //Turn off the motor
                    delay(5);
                    Lifting_servo.write(Home_servo); //Lift the arm to home position
                  }
            }
        alert=0;
        Serial.flush();
      }
  }
```

Appendix - 2

Appendix 2 - Complete Arduino code including driving commands

```
#include "SR04.h"                          //Include Ultrasonic sensor library
#define TRIG_PIN 1                         //Set pin 12 as trig
#define ECHO_PIN 12                        //Set pin 11 as echo
#include <Servo.h>                         //Include servo library
#define SLAVE_ADDR 9                      // Define Slave I2C Address
#define ANSWERSIZE 5                      // Define Slave answer size


  Servo Lifting_servo;                     //Declare servo object to control

const int Emergency_red_LED = 6;           //On pin 6 red LED is connected
const int Emergency_yellow_LED = 5;        //On pin 5 yellow LED is connected
const int Running_LED = 4;                 //On pin 5 green LED is connected
const int Safety_Switch = 3;               //On pin 3 limit switch is connected
const int Relay = 7;                       //Relay on pin 7 activates the DC motor
const int Home_servo =90;                  //Declares the starting position of the servo
int angle = 0;                             //Variable needed to control the servo
int alert = 0;                             //To exit from the loop if safety switch pushed
unsigned long Running_Time;          //Total count of seconds since the board is powered on

//%%SETUP INPUT%%
const int scratching_time = 10;        //time rake stays on the soil scratching [ms]
const int Stop_distance = 14.5;          //cm distance to the soil to stop the arm lowering
unsigned long time_step = 3000;          //time step between each lowering operation [ms]

SR04 Ultrasonic_S = SR04(ECHO_PIN,TRIG_PIN);         //Define the ultrasonic sensor
long distance;                                       //Define distance read by the sensor

void setup()
  {
      Serial.begin(9600);          //Initialize the serial communication with the computer
      Lifting_servo.attach(2);                       //Set pin 2 as servo control
      pinMode(Emergency_red_LED, OUTPUT);            //Set pin as output
      pinMode(Emergency_yellow_LED, OUTPUT);         //Set pin as output
      pinMode(Running_LED, OUTPUT);                  //set pin as output
      pinMode(Safety_Switch, INPUT);                 //set pin as input
      pinMode(Relay, OUTPUT);                        //set pin as output
     Wire.begin();                 //Initialize I2C communications as Master
      Serial.println("I WRITE DATA AS MASTER"); //Testing your serial
 }

void loop()
  {
     Running_Time = millis(); //Total count of ms since the power on of the board

     // Write 1 to the Arduino Robot, so it will go forward for 2 seconds
     x=2;
     Wire.beginTransmission(SLAVE_ADDR);
     Wire.write(x);
     Wire.endTransmission();
     Serial.println(x); //Printing the I2C command on the Arduino Serial, for testing

     Lifting_servo.write(Home_servo);                      //Start from home position
     digitalWrite(Relay, HIGH);                            //Turn off the motor

     int Safety_Switch_status = digitalRead(Safety_Switch);     //Read the switch status

     if (Safety_Switch_status == LOW)                    //If emergency switch pushed
        {
          digitalWrite(Running_LED, LOW);                //Turn off the green LED
          digitalWrite(Emergency_red_LED, HIGH);         //Turn on the red LED
          digitalWrite(Emergency_yellow_LED, HIGH);      //Turn on the yellow LED
```

```
          digitalWrite(Relay, HIGH);                    //Turn off the motor
          Lifting_servo.write(Home_servo);              //Return to home position
        }
      else if (Safety_Switch_status == HIGH)            //If emergency switch not pushed
        {
          digitalWrite(Emergency_red_LED, LOW);         //Turn off the red LED
          digitalWrite(Emergency_yellow_LED, LOW);      //Turn off the yellow LED
          digitalWrite(Running_LED, HIGH);              //Turn on the green LED

          Serial.print(Running_Time); //Print the time passed since the board is powered
          Serial.print("\n");

          if (Running_Time % time_step <=20)  //If a time frame circa equal to time step
                                                 has passed
              {
                  digitalWrite(Relay, LOW);              //Turn on the DC motor
                  distance = Ultrasonic_S.Distance();    //Read distance from the sensor

                  for (angle=Home_servo; (distance < Stop_distance-1 or distance >
Stop_distance+1) and (0<=angle<=180)and (alert==0);)  //When the distance arm-ground
                                                          needs to be correct
                    {
                      distance = Ultrasonic_S.Distance();     //Read distance from
ultrasonic sensor
                      Serial.print(distance);      //Print on the monitor the distance
                      Serial.println("cm");        //Print the measurement unit
                      if (distance < Stop_distance-1)   //If it’s too close to the ground
                        {
                          Lifting_servo.write(angle);   //Servo is lifting the arm
                          angle++;
                        }
                      else if (distance > Stop_distance+1)      //If it’s too far
                        {
                          Lifting_servo.write(angle);      //Servo is lowering the arm
                          angle--;
                        }

                      int Safety_Switch_status = digitalRead(Safety_Switch);
                                                 // Read the status of the switch

                      if (Safety_Switch_status == LOW)      //If emergency switch pushed
                        {
                         alert=1;
                        }
                    }
                  if (alert==1)
                    {
                      digitalWrite(Running_LED, LOW);           //Turn off the green LED
                      digitalWrite(Emergency_red_LED, HIGH);    //Turn on the red LED
                      digitalWrite(Emergency_yellow_LED, HIGH); //Turn on the yellow LED
                      digitalWrite(Relay, HIGH);                //Turn off the motor
                      Lifting_servo.write(Home_servo);          //Return to home position
                    }
                  if (alert==0)
                    {
                      delay(scratching_time);      //Wait: the rake is scratching the soil
                      digitalWrite(Relay, HIGH);        //Turn off the motor
                      delay(5);
                      Lifting_servo.write(Home_servo); //Lift the arm to home position
                    }
              }
          alert=0;
          Serial.flush();
        }
  }
```