AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: GR Team

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Logo:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Table of Contents **2**

1. Project Title 3

2. Project Category 3

3. Project Description 3

4. Keywords 4

5. Target Audience 4

6. Project Objectives 4

7. Project Timeline 4

8. Resources Provided 5

9. Tools and Technologies Used 5

10. Step-by-Step Implementation Plan 5

11. Evaluation Criteria 9

12. Languages Available 9

13. Contributors 9

14. Feedback and Continuous Improvement 10

Submission Notes **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

1. Project Title:

Robotic Arm for Object Manipulation

2. Project Category:

AI & Robotics – STEM Maker Project

3. Project Description:

This project involves building and programming a **robotic arm controlled by Arduino and servo motors**. The robotic arm can move across multiple axes and perform simple manipulation tasks such as **picking and placing objects**.
Students assemble a mechanical robotic arm using **3D-printed parts, servo motors, and a microcontroller**, and then program it to control its movements. The project introduces learners to **robotics, automation, embedded programming, and mechanical design**.
Optionally, the system can be extended with **computer vision and AI-based object detection** to create a semi-autonomous robotic system.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Keywords:

robotics
servo motors
robot arm
automation
STEM education
AI robotics
maker learning

---

5. Target Audience:

- *Vocational education and training (VET) students*
- *Robotics and STEM clubs*
- *Beginner makers interested in robotics*

---

6. Project Objectives:

- Understand the basic principles of robotics and mechanical movement
- Learn how servo motors are controlled using Arduino
- Develop basic programming skills
- Gain experience in hardware assembly and electronics
- Explore how robotics can be combined with AI and computer vision

---

7. Project Timeline:

Introduction to robotics 1 hour

3D printing & preparation 2 hours

Mechanical assembly 2 hours

Electronics wiring 1 hour

Programming the robot 2 hours

Testing and improvements 1–2 hours

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Resources Provided:

- STL files for 3D printed parts
- Arduino control code
- Wiring diagram
- Assembly guide
- Demonstration video of the robotic arm

---

9. Tools and Technologies Used:

- Servo motors (4–6 units)
- Robotic gripper (claw)
- 3D printed robotic arm parts
- Power supply (battery or adapter)
- Jumper wires

---

10. Step-by-Step Implementation Plan:

Step 1: Story
Story
Story
Story
Story
The rough plan:

A glove that tracks the movements of the hand and then sends it via WiFi to a Raspberry Pi, which runs a program that then displays a visualization and transmits the movements of the hand in real time.

Accurate execution:

Our idea is to make it a glove that records the individual movements of the fingers, pitch and roll of the hand. This data is then sent to the Raspberry Pi/Laptop via the Wemos D1 Mini using a WiFi connection with TCP. In this connection, the Wemos D1 Mini is in AP mode i.e. it hosts its own WiFi network (as the school's network does not allow communication between the Raspberry Pi and Wemos). One advantage of this setup is that you can operate the glove where you want and are not tied to network connections. The collected data is processed by a Python script to send the correct signals to the individual servo motors of the arm to implement the movements. This involves converting the data from the gyro sensor directly on the Wemos D1 into position data, which is then sent to the Python script along with the movements of the individual fingers. This script then converts the position data into the positions of the individual servo motors, which are then controlled accordingly.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Division of labor:

I'm developing the hardware and the visualization in Unity.

My friend is developing software for the arm in Python, processing data from the gyroscope and interpreting it correctly, all other activities that have to do with the connection of glove and PC.

Progression:

We first considered how to capture finger movements so that we could use them as a reasonable control for gripping. An initially inexpensive solution for this appeared to be pressure sensors, which we can also manufacture ourselves and thus save ourselves the unnecessary expense of purchased sensor technology.

Next up was an arm to translate the movement of the hand. A small 3D printed robotic arm seemed practical for this, on which I can freely indulge my passion as a passionate 3D printer fan. The arm was small and quickly printed, but very limited in its range of motion. The arm consisted of two joints, which not only interfered with each other, making it difficult to calculate the optimal position of the servo motors to reach a point in space, but were also very limited in their radius.
https://www.instructables.com/EEZYbotARM/

After we, or rather the 3D printing Nerd, had finished the arm, we noticed the considerable flaw of the arm, which not only had a very limited range of motion due to its design, but was also very inaccurate and weak due to the use of cheap servos. We then decided on a version with larger servos and more freedom of movement (https://www.instructables.com/3D-Printed-ROBOTIC-ARM/). This was now a bit stronger and we additionally decided to develop a circuit board.

As a result, we had difficulty working together on our project due to the lockdown. We fixed this problem by setting up digital access to the Raspberry PI via a VNC client and a webcam, which enabled the programmer to control and see the arm from his home.

Parallel to the robot arm, we always worked on the glove for controlling the arm. Now that we knew the robot arm was working and we only had to work on the glove, the 3D printing Nerd designed PCBs (Printed Circuit Board) for the glove with Wemos, gyroscope sensor, multiplexer, pressure sensors and a header board for the Raspberry Pi to control the servo motors. We ordered these from JLCPCB.

For the Raspberry Pi and the battery, which we need because the Raspberry Pi does not provide enough amps, the 3d printing Nerd printed a holder. With this one, we made sure that you can use it again and again without breaking it. There are, for example, melted metal threads, so that the screws are always easy to remove, precisely adapted shapes of the battery, that you can simply press the battery box in and everything holds.

Then the boards arrived and there were minor problems with the layout of the paths on the board, resulting in a reorder. The board for the Wemos D1 Mini had the problem that the multiplexer in the designer tool of EasyEDA didn't have the same position for the pins (The multiplexer is the purple board on the picture). At first we tried to solder a PCB over which the multiplexer can be plugged in; but because the displacement was so big, there was not enough space on a breadboard to move everything around. Thus we decided to work around the problem with simple jumper wires that are soldered to the multiplexer.

When we connected the board for the robot, we realized that the design we had used so far was very poor. The arm was in a tilted position all the time, so the motor was very heavily loaded all the time. These motors were also not really high quality and only held a maximum of 10 kg. We saw this when the

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

lowest motor broke, because almost all the gears in the servo broke off. That made us think about designing our own robot arm with Inventor. In addition, the robot arm was not very stable and wobbled a lot when you used it. It also had a general tilt to the right side, which made controlling it a lot more difficult.

After 2 weeks of trying to build reliable pressure sensors ourselves and only got meager results, where the sensors could distinguish just three positions. We decided to order 5 pressure sensors for the glove. For the glove, we printed a holder for the circuit board that is adapted to the slight curvature of the back of the hand. Because of the circuit board, we only need a 10kOhm resistor in the circuit to read all the data from the flex sensors. A small disadvantage of having all flex sensors connected to one circuit is that you have a small error of 15Ohm per sensor that is bent. But this is only a small error, because our data range goes from 0 to 1024 i.e. if all but one finger is bent the error would only be 60 ohms or 5.8%.

The first hurdle for the new robot arm was to develop a stable base, since the original base was mounted directly on the servo. For this we then developed an axial bearing with softair balls. We had slight difficulties with the design, since it was generally the first ball bearing that we designed. The first idea was to drive the thrust bearing with a smaller gear, but there is a small error in thinking: you lose distance when you transfer power from a smaller to a larger gear, and since you have a limited radius with a servo, we ended up driving the thrust bearing directly with the servo. We also had problems with the yellow softair balls because their tolerances were too small, which caused the thrust bearing to get stuck all the time.

With this posture, it was important to us that the robot arm with the axis rests on two ball bearings and is in any case more massive than the original part.

With the original arm there were no fixed axes i.e. all axes were just plugged on top of each other without protecting the other parts from slipping out. Another advantage is that here we could put the cables directly through the body. Instead of relying on long threaded rods like the original, we directly developed solid parts that you 3D print and then just glue into a hole. These parts will never leave their position because the shape for each of the four holding points is a trapezoid.

Now that we had finished the design, we realized once again too late that the servo at the lowest point is probably much too weak to lift the whole arm. That's why we then decided to order a new, stronger servo with 20kg of power so that we don't destroy a second servo. The new servo even has an aluminum arm to transmit the force, but the screws didn't really hold, so we had to fix it with hot glue.

With our glove we still had the problem that we had no power, so we have a small powerbank holder designed which one makes tight with textile rubbers. This can be adjusted to the thickness of the upper arm and is also very easy to remove, because you have two small openers in the middle. In the power supply, we have but again replaced the power bank, because the old was no longer to charge.

The last step is just to turn the data from the glove into leasable signals. These are then sent to the Raspberry Pi, which interprets them and takes coordinates.

Step 2: Difficulties
The idea is that a position in space is calculated by the data from the gyro sensor. This position is then passed to the Raspberry, which then has to move the arm so that it is at the position in space. So the arm should correspond to the movement of the own hand. In order for the arm to really be in the position where it is supposed to be, it has to move every single joint correctly. Since the arm has four axes, but is in a 3D space, there are infinite possibilities to reach this position. Furthermore, there is the restriction of only 180° per axis, so the calculation of the optimal position for each axis was very complex. Therefore it is advantageous to implement an algorithm, which automatically finds a good optimum of the position. The whole thing is called Gradient Descent. The idea is to perform a curve analysis of the difference function, which represents the difference of the current position to the desired one. A minimum is then searched for. This minimum represents a minimum difference of the momentary position of the arm to desired. Therefore, if the difference function is at a minimum, the arm is close to

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

the optimal position.
The first robot arm was too small, had no force, was too inaccurate and had a limited range of motion. We solved this by building a larger arm with stronger servos with more accuracy.
The second robot arm definitely had more power, was more accurate and had much more room to move, but was still too wobbly for us.
Development of an axial bearing without prior knowledge
In the beginning we used the yellow softair balls, but this led to the fact that the bearing could not really be turned cleanly, because these balls have a lot of unevenness. We then got professional bullets with the same size in the softair range. Our final solution in the event that the new bullets would not have made a difference, to use Ballistol. It's a natural lubricant that doesn't break down plastic like WD-40. We didn't need it, though, because the new balls were running so well that we didn't want to change anything on them.
Another problem was that Markus forgot that transferring power from a small gear to a large gear reduces travel a lot. Solution was then simply to attach the servo directly to the thrust bearing.
The lowest servo destroyed itself, because the arm was too much for it and all the gears in the servo gearbox were torn off. So we bought a new stronger servo with 20 kg power and designed a new base to fit it.
With the glove there were problems, because the Wemos D1 Mini has only one analog pin but we wanted to read 5 different analog signals, that meant for us that we had to connect a multiplexer. This can switch between the different analog signals. This had the advantage that we had to solder only one resistor, because the signal of the sensors always comes out at the same place, even if we change the input.
At the battery box the connection had a loose contact, because Markus probably attached it only to 90% cleanly, whereby the servos got no more current. But we thought it was because of the board, went to Markus to get a new board and when we were back we noticed that it was because of the power supply.
Calculating the hand in space with only data from the gyroscope turned out to be more difficult than expected. The gyroscope sensor gives us data about the force acting on it at the moment and the movement in degrees around each axis. If we need to derive a position in space from this data, we have to integrate the force to get acceleration, and then integrate it again to get the position. As a result, measurement errors have a significant impact on the final position because they are integrated twice. The data from the gyroscope has an approximate measurement error of about 5%, which makes the whole approach of the calculation impossible. Even expensive gyroscopes are not able to determine an exact position from collected data. Therefore, the only solution to our problem is to make the control of the arm via gestures instead of mimicking the movement of the arm.

Step 3: What You Need
What You Need
What You Need
1* Raspberry Pi 3B+ or higher
3D Printer
thin Cables
2* SG 90 Servo
3* MG995 Servo
1* DS3218 RC
Soft Air Balls (But you need the high-quality balls, not the yellow ones.)
wood plate
Heat Inserts
A standard skatboard bearing
Device data sheets:

Wemos D1 Mini: https://docs.platformio.org/en/latest/boards/espressif8266/d1.html
Raspberry Pi:
https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2711/rpi_DATA_2711_1p0_preliminary.pdf
MG995 Servo: https://www.electronicoscaldas.com/datasheet/MG995_Tower-Pro.pdf

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

SG 90: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf
cjmcu-4051 8- Channel Analog Mulitplexer
https://assets.nexperia.com/documents/data-sheet/74HC_HCT4051.pdf
MPU 6050: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf
ZD 10 - 100 flexibler Biegesensor
DS3218 RC:
https://www.robotshop.com/media/files/pdf/servo-metal-gear-digital-servo-180-datasheet.pdf
Step 4: 3D Print
BearingBottom.stl

Alpha.stl icon
Alpha.stl
View in 3D
Download
bearingAxial.stl icon
bearingAxial.stl
View in 3D
Download
BearingMiddle.stl icon
BearingMiddle.stl
View in 3D
Download
BearingTop.stl icon
BearingTop.stl
View in 3D
Download
Beta.stl icon
Beta.stl
View in 3D
Download
View 9 more
Step 5: Assembly
Assembly
Assembly
Assembly
View 3 more
We start building

Base
Rotation
Gripper
Delta/Gamma
Beta
Alpha

Step 6: Base
Base
Base
Base
View 3 more
The base for the Raspberry Pi and battery is attached with 5 screws. Here are also 4 fusible threads attached that you can remove the Raspberry Pi without damaging the plastic. The cable of the battery is passed under the Raspberr Pi and then plugged into the power port of the board.

Step 7: Rotation

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Rotation
Rotation
Rotation
Rotation
View 9 more
The first step is to fix the mainbody to a wooden plate. This gives the arm its stability. This is what the three holes are for.Then one of the three MG995 servos is installed for the rotation. The servo is fixed in the middle of the mainbody, so that the pivot point is in the middle. Now the round black attachment that comes with the servo is pressed onto the axle. The BearingBottom is then attached to this with four M3 screws. We use a thrust bearing made of softair balls here to definitely secure the rotation. The second arm we had built hung directly on the servo axis and was not supported at all.

Now take the softair balls and place them carefully in the round. Please make sure that the softair bullets are good ones with a low tolerance range, otherwise the thrust bearing will grind. When the round is filled with the balls, you can fix BearingMiddle on the MainBody with M3 screws. You don't need to use as many screws as I did, that was rather overkill. On top of it comes another row of balls. Under the BearingTop plate there are recesses for M3 nuts. You can glue them in place with superglue or hot glue. Use the holes in the BearingTop part to secure it to the BearingBottom part. Here you should use at least 2cm long screws. Now the alpha part is fixed with three screws. Alpha is printed with support and has small recesses on the underside for the screw heads. Now two standard skate board ball bearings are inserted into Alpha.

Step 8: Gripper
Gripper
Gripper
Gripper
We used this Gripper design because it was pretty easy to assemble and simple.

https://www.thingiverse.com/thing:2415

Step 9: Alpha
Alpha
Alpha
Alpha
View 12 more
In the next step, we take the side panel of Gamma and attach another MG995 servo to it. For all these screws, we use washers so that the nuts do not loosen over time. We take the black part, that are sent with the servo, and put it on the servo.

Next we assemble Delta. Here, when 3D printing, I would give the tip that the part is placed on the bottom edge, e.g. with the PrusaSlicer. Then one of the two 90g servos is attached with two screws. Through the side with a hole is a 3 cm long screw with washer. This screw is then passed through the other side of Gamma and again fastened with a washer and two nuts pulled against each other. Next, the Gamma servo is attached to the Delta part. Now everything is a bit wobbly, but we fix that by attaching the GammaSpacer in the recesses in the side turns with superglue. Now you should wait only one minute until you are sure that the glue is dry.

Step 10: Beta
Beta
Beta
Beta
View 5 more
Now take the servoCasing and press 4 M3 screws into the recesses. Push a 3 cm screw through the hole and fasten the side with two nuts screwed together. Beta is glued together just like Gamma. Here, you have to be careful that the servo is pushed through the hole before gluing. This may feel wrong the first time, but the cable can withstand it. Now take Beta and put four screws in the lower holes and screw

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

them to the servo casing with a long and narrow screwdriver. Exactly on the nuts, which we had pressed in before. Now we just have to fix the servo. Through the holes, through which we have just fastened the lower screws, again screws and nuts are led and tightened. Now we use one of the black parts and two screws again to transmit the power of the servo. As a last step, we lead the cables through the torso.

Step 11: Gamma
Gamma
Gamma
Gamma
Gamma
View 9 more
Now take the bearingAxial and the Srewholder. Pass an M3 threaded rod through it and fasten two nuts with washers, screwed against each other, to the other end. The robot arm will rest on the bearingAxial part, so the threaded rod will not really move. Unscrew the servo for Alpha again a little without the nuts falling out. On the lower Beta part, we now drill two small holes into which we insert M2 screws to attach the metal arm of the 20 kg servo. I even used hot glue here because I wanted to be 100% sure the arm would hold. Now all that's left is to feed the cable of the 20 kg servo in through the side of Beta and out the bottom. Here I have the cables, then all together with one. Each of the cables has a label, so that I always know in an emergency to which servo which cable belongs.

Step 12: PCB
PCB
PCB
Either you wire the Raspberry Pi via cable or you order the board for the simplified wiring that I have designed. You can get the files for the PCBs here. I recommend to use terminal connectors instead of the standard header pins. It's pretty easy and makes your project look professional

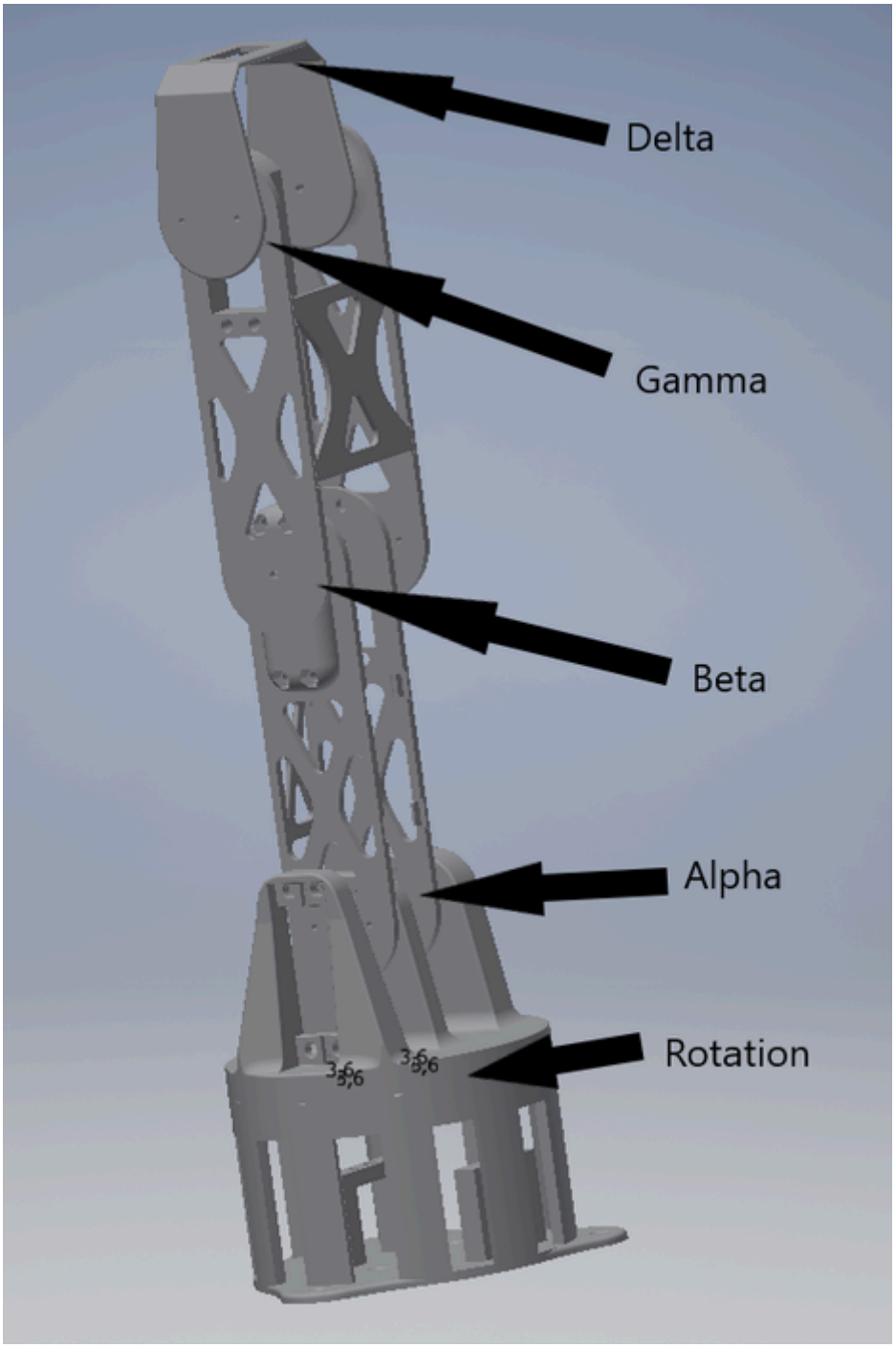
Step 13: Programming
You can find the code for the Raspberry Pi here. You have to download the code to the Raspberry Pi and then run it from the console. The arm Is programmed to follow a specific point in space, so you can move a point in a 3D Space and the arm will move to this point with the tip of the gripper and move all other servos accordingly.

Step 14: MPU Glove(Optional)
MPU Glove(Optional)
MPU Glove(Optional)
I'm working on the Instruction for this glove. You could use this glove to control the robot arm. If you really want to get the glove working before I publish the instruction:

the source code for a Unity example to show what the glove can measure
here you can get the PCB file.
To this step I also added the 3D file you need
PCBHolder.stl

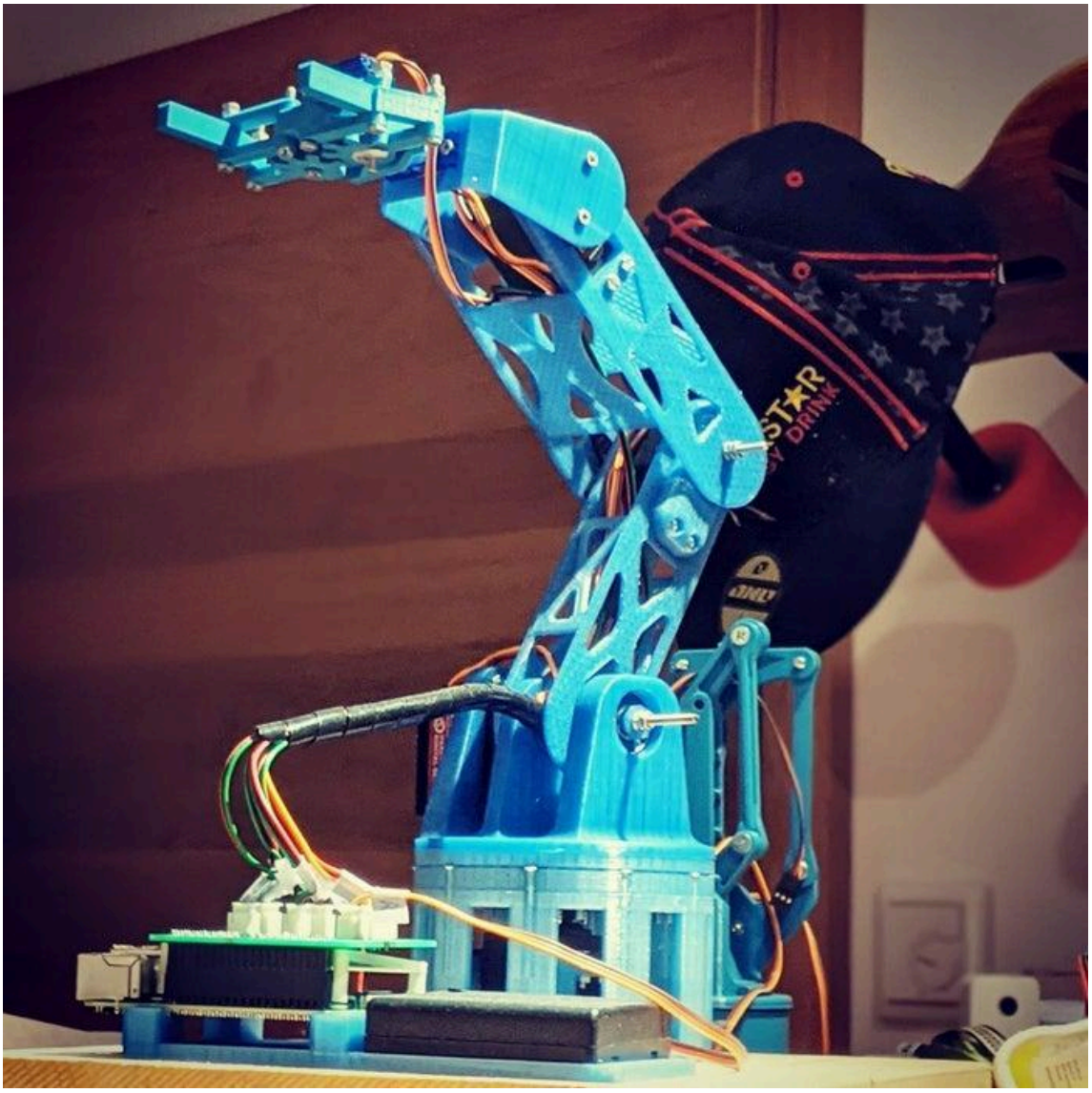
Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Co-funded by
the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



11. Evaluation Criteria:

Criterion	Description
Assembly accuracy	Correct mechanical assembly
Programming	Correct control code
Functionality	Robot performs movements successfully
Creativity	Additional features or improvements
Collaboration	Teamwork during the project

---

12. Languages Available:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

English

---

13. Contributors:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback and Continuous Improvement:

Teachers and students are encouraged to provide feedback regarding:

- clarity of instructions
- technical difficulty
- suggestions for improvement
- new AI extensions

Future versions may include:

- AI object detection
- mobile robot integration
- remote control via smartphone

.

Submission Notes:

Please ensure that all required documents and resources are attached to your submission. The AI-SAVES@ProjectHub review team will evaluate the submission based on the provided information. Once approved, the project will be published on the hub, and partners will be notified.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.