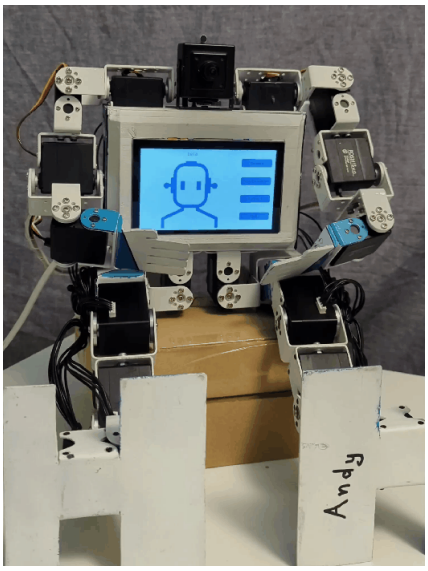
Humanoid robot BETA

Table of Contents

Humanoid robot BETA

BETA is an ambitious undertaking that seeks to develop a humanoid robot possessing a range of human-like capabilities, including the ability to walk, see, and communicate verbally.

Conceived in 2018, the project has undergone several transformations and upgrades over the years. Presently, it is being developed as part of the "Koło Naukowe Humanoid" student scientific association, where updates on its progress and new features are regularly shared.



Built with

Hardware components

- Raspberry Pi 4 Model B
- Raspberry Pi display
- Bus servos
- Pulse servos: RDS3235-180 and RDS3115-270
- USB Webcam

Programming language

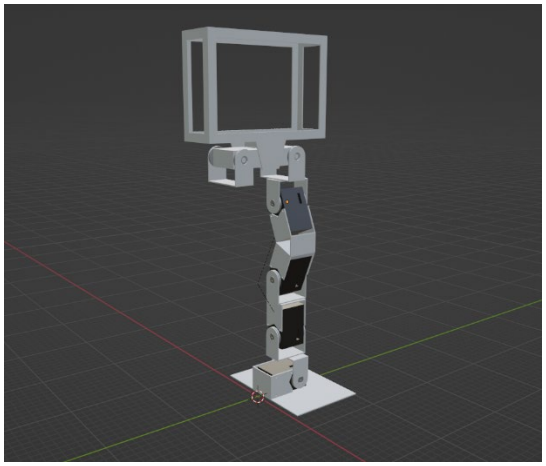
The programming language of choice for the project is Python. This decision was made because of its ease of use, large community, many libraries and frameworks and cross-platform compatilibity. Additionally, Python is a versatile language that can be used to develop a wide range of applications, including those that are used for the control of the robot's servos, the implementation of its vision system, and the development of its speech recognition capabilities.

Used libraries

- PySide2 - GUI toolkit used in Application
- Selenium - automation of browser used in Speech-to-text
- OpenCV - real-time computer vision used in Pose estimation
- OCServo - servo control app used in Servo control
- pigpio - control of GPIO to generate software PWM signal to control Pulse servos.

Construction

The framework of the robot is composed of a hybrid of components that are either 3D-printed, purchased, or self-fabricated. The design of these parts was created in Blender, a 3D modeling software, and subsequently constructed by our team.



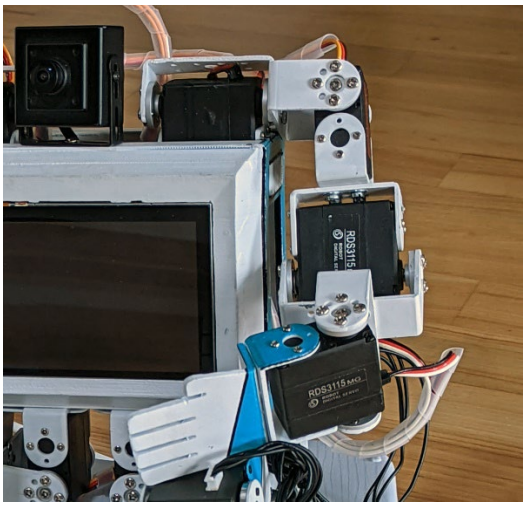
Additionally, a 3D printer was utilized to produce the robot's chest component.

Servo control

As previously stated, our project utilizes two distinct types of servos. Due to their differing control mechanisms, we have partitioned their descriptions into two distinct subcategories.

Pulse Servo

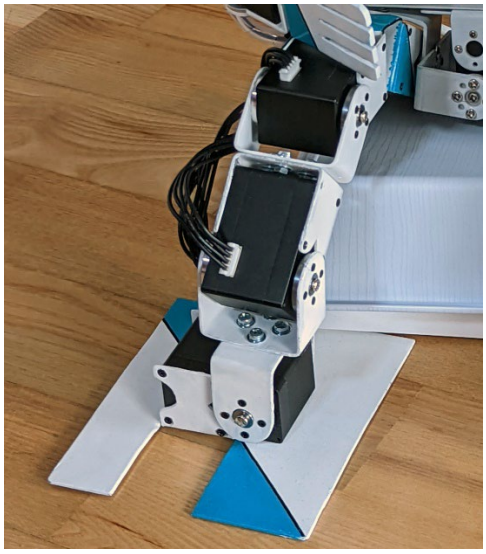
Using mentioned library we've created tool to control servos with single commands.
To affix the servos to the robot's framework, we employ brackets that are included in the kit alongside the servos. The Raspberry Pi's GPIO pins serve as the source of the PWM signal, which is generated by utilizing the software pigpio library. We have developed a tool utilizing the aforementioned library, which allows for the control of the servos through the use of single commands.



During the process of developing the robot, it became apparent that the existing servos were not ideally suited to our requirements, as they were prone to excessive play and struggled to maintain a fixed position. As a result, we made the decision to incorporate Bus servos in the leg joints, as these offer superior stability and control. By leveraging this approach, we can enhance the overall performance and reliability of the system, while enabling greater precision and accuracy in its movements.

Bus Servo

In regards to the management of the second group of servos, communication is established through the utilization of the UART protocol, as the servos in question incorporate internal electronics. This approach enables efficient transmission of commands and data, as well as simplifying the wiring and control of the servos. By interfacing with the internal electronics of the servos, the system can achieve greater precision and flexibility in its movements.

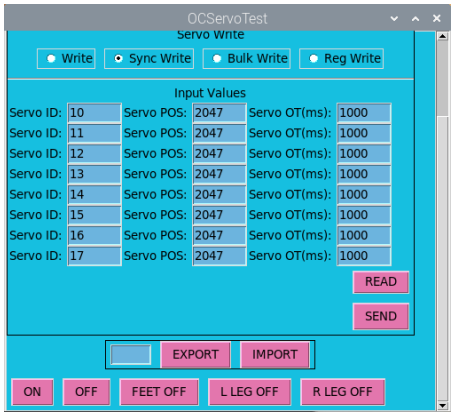


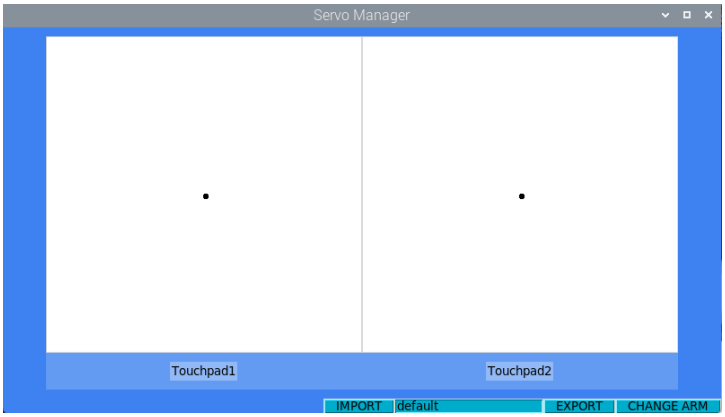
For more information about Bus servo control check out our Servo library.

Sequencial movement

To achieve smooth movement of our robot, we have developed a tool that enables the sequential control of servos. The primary objective of this tool is to facilitate the coordinated movement of servos to achieve seamless motion.

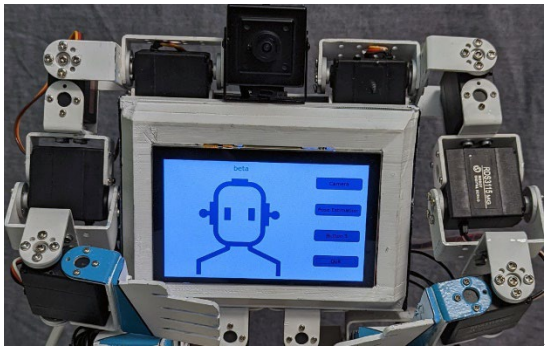
The program operates by utilizing text files, which contain data regarding the angles of each servo required for a given pose. By compiling a list of various poses, complex movements can be achieved through the sequential execution of poses. To create these text files, we have developed a custom tool that allows for the adjustment of the robot to a desired pose and the subsequent saving of this pose to a text file for later use.

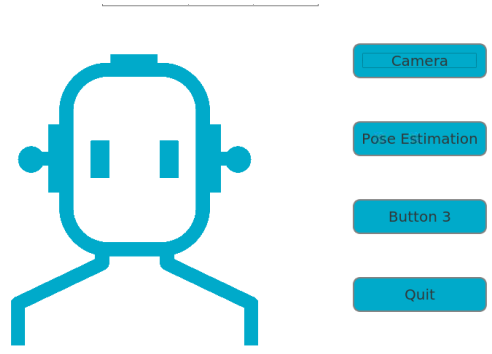




Application

The software application that has been developed utilizes the PySide2 library to create a graphical user interface (GUI). The primary objective of this application is to enable the user to manage and manipulate the behavior of a robot through a touch screen interface.





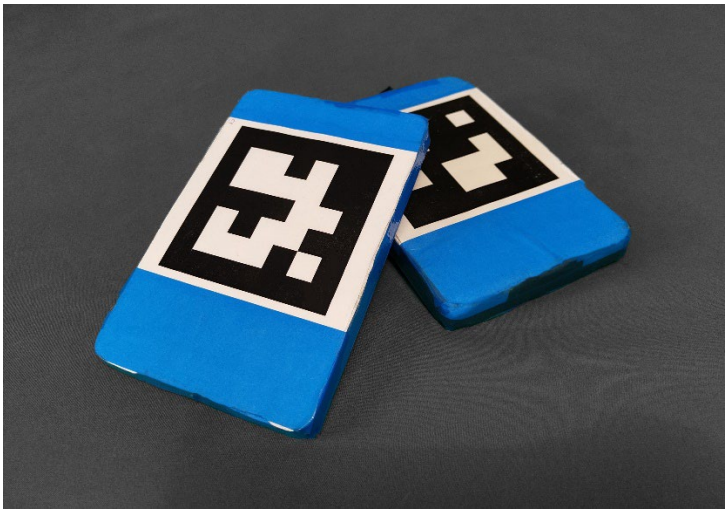
Speech-to-text

For the purpose of speech-to-text, we have utilized the speech to textV website. The website is accessed through the use of the Selenium library. The library allows for the automation of the browser, which enables the implementation of the speech-to-text functionality.

One possible solution to address this issue is to utilize an application programming interface (API) that provides speech-to-text functionality. However, after careful consideration, it was determined that this approach was not optimal due to cost implications and implementation complexity. As an alternative, a solution was chosen that is both cost-effective and easy to implement, and does not impose excessive demands on computer resources.

Pose estimation

For the purpose of pose estimation, the project team has chosen to use ArUco trackers, which are part of the OpenCV library. This is because ArUco markers are an efficient and accurate type of fiducial marker that can be easily detected and identified in image or video streams. By attaching multiple ArUco markers to the robot's limbs and joints, the position and orientation of these markers can be tracked to estimate the robot's pose with a high degree of accuracy.



In order to collect information about the location and orientation of the human body's joints and calculate angles for the robot, ArUco trackers will be attached to the human body. At this time, no other solutions have been selected for pose estimation purposes.

More information

To obtain further details, including access to our source code and future plans, we kindly invite you to visit our Github repository:

https://github.com/Med0kin/BETA-Humanoid-robot