Board 83/Work in Progress: The Magic Orb: A Mechatronics Demonstration and Course Project to Attract Next-generation Engineering Students

Dr. Ruoshi Zhang, University of Louisville

Ruoshi Zhang has accepted the position of Senior Research Scientist at the Louisville Automation and Robotics Research Institute. His research interests include sensor technology, microtechnology, robotics, and embedded systems. He is the author of four journal papers and more than ten conference papers. He currently teaches two courses for the ECE department at the University of Louisville: ECE412 Introduction to Embedded Systems and ECE500 Introduction to Advanced Microcontrollers.

Nathan George, University of Louisville

Nathan George is currently a junior level undergraduate student at the University of Louisville studying computer science. His academic interests include artificial life, automation, real-time simulation, and user experience.

Prof. Dan O Popa, University of Louisville

Dan Popa has over 30 years of research experience in robotics and automation. He is currently the Director of the Louisville Automation and Robotics Research Institute (LARRI) at UofL and the Head of the Next Generation Research Group (NGS). His early research work included adaptive force control and motion planning for nonholonomic robots. In 1998, he joined the Center for Automation Technologies at Rensselaer Polytechnic Institute, as a Research Scientist, where he focused on precision robotics and micromanufacturing. In 2004, he became an Assistant and then an Associate Professor of Electrical Engineering at the University of Texas at Arlington. Since 2016, he has been the Vogt Endowed Chair in Advanced Manufacturing and a Professor of Electrical and Computer Engineering at University of Louisville. Dr. Popa's research in focused on two areas: 1) social and physical human–robot interaction through adaptive interfaces, robot tactile skins, and facial expressions; and 2) the design, characterization, modeling, and control of microscale and precision robotic systems. Dr. Popa is the recipient of several prestigious awards and the author of over 300 peer reviewed conference and journal articles, mainly in IEEE and ASME publications. He has been very active in the IEEE Robotics and Automation Society (RAS), including extensive competition, workshop, conference, and journal service.

[WIP] The Magic Orb: A mechatronics demonstration and course project to attract next-generation engineering students

Abstract

Open-house events hosted at university labs for K-12 students, typically feature academic research which often requires prior knowledge of the field to fully appreciate its significance. This disconnection often fails to capture the interest of young audiences. Furthermore, these demonstrations may lack interactive elements that prompt curiosity, preventing students from being able to experiment with the demonstrations in a hands-on way. In response to this challenge, we created this device as a "primer" to stimulate students' interest, preparing their appetite for more in-depth academic content. This paper outlines a student-developed project aimed at engaging young visitors at a robotic research institute during open-house events. This interactive device, named the “Magic Orb” demonstrates a figurative orb that hovers on top of the user's palm as it is being moved. To implement and understand this experiment, students utilize applied knowledge in mechanical, electrical, and computer engineering. The orb will also be used in the future as a course project in three related courses. It aligns with the paradigm of mechatronics education as it encompasses electronics sensor integration, motor control, microcontroller programming, mechanical design, and 3D printing skills.

Keywords

Mechatronics, electrical engineering, mechanical engineering, 3D printing, EPS32, ToF sensor

1. Introduction

Traditional engineering education focuses on a focused aspect of disciplines such as mechanical, electrical, or computer engineering. Typical electrical engineering graduates have gained the necessary skills to implement motor control with a power circuit and a microcontroller when appropriate mechanical hardware is presented. On the other hand, a typical mechanical engineering graduate will be able to 3D print a mechanical structure driven by a motor and may need help to understand how to control and interface with it. Once in the industry, graduates will face challenges in the modern job market, because jobs increasingly require multidisciplinary knowledge across different engineering domains.

The Magic Orb uses a distance sensor to detects operators hand and creates a floating effect to an "orb" on the operator’s palm. The “Magic Orb” demonstrator described in this paper is inspired by one of the outreach events at our newly established Robotics Institute at the University of Louisville, with the goal that it be used in a freshman course, as well as follow-on courses in the department of Electrical and Computer Engineering. With a multidisciplinary teaching goal, this project covers mechanical, electrical, and computer science and engineering in the integrative area of mechatronics [1] [3]. Considering the simplicity of its design, it can be introduced to an early academic career course project [2] [4], such as "ENGR 110 Engineering Methods, Tools, and Practice I", which is a first-year introductory course for engineering students. This course introduces essential engineering tools and skills for success in engineering, for example, computer-aided design (CAD) , 3D printpracticing, and basic programming. All the design components including the CAD model and the firmware can be given to the student to practice their hands-on skills. Introducing projects with hands-on experience in the first year has a positive impact on students' motivation [5]. In later courses, the Magic Orb can be used in the embedded system junior-level course, "ECE412 Introduction to Embedded Systems", to teach programming skills and practice design tasks such as sensor interfacing, interrupt, pulse-width modulation (PWM), etc. Controller design for the Magic Orb, can be algorithms can be discussed in the senior level control course "ECE 560 Control Systems Principles".

The Magic Orb also addresses challenges faced by our emerging robotics and mechatronics degree programs to both attract and explain engineering concepts to young audiences at outreach events such as local science fairs. In events like this, humanoid, quadruped robots, or drones have regularly been used to demonstrate human-robot interaction technologies. For example, we have pioneered the use of speech and facial expressions, with instruction robots such as Zeno robot [6] in order to invite participating children to follow a programmed body movement. After attending a couple of such events, the author finds that although these projects bear scientific significance, they are less efficient in inspiring questions and initiating conversations due to lack of a more direct interaction with the children. The demonstration aspect of the design is a work in progress and more data needs to be collected in outreach events.

2. Experiment Description

A pulley driven by a servo motor is the basic mechanism, and it was printed with a 3D printer in our prototype. It offers an opportunity for the students to design, fabricate, and optimize simple mechanical structures. A hobby-level servo motor is installed in the base fixture and drives the pulley to control the position of the orb. Motor control can be practiced, such as pulse width modulation (PWM) signal generation. The "eye" of the system is a time-of-flight (ToF) sensor that accurately tracks the palm location of the operator. The "brain" of the project is a microcontroller module that uses an ESP32. It is a popular module among both hobbyists and serious scientific researchers. It gained its popularity because it supports the Arduino software framework which is extremely easy to use and teach to students who are not very familiar with programming on embedded systems. For serious learners, the ESP32 module also provides ample learning opportunities. They can use the native toolchain from Espressif to explore C or C++ programming on the microcontroller platform. Finally, the entire design can be powered by a 5V USB power bank, making it portable.

3. Mechanical Design

To lift the orb into the air, its mechanical design needed to fulfill the following requirements including:1) Each component needed a dedicated resting spot; 2) Visually minimal so the orb looks like it is floating. 3) 3D printable for ease of implementation; 4) Modularity, which allows all the parts to be optimized or replaced and 5) the design needs to be mechanically sturdy.

A solution that fits all these requirements was implemented with a string and pulley system. The string can be made almost invisible, and the pulley system allows for a very simple mechanical design. Figure 1 shows 3D renderings of our model with different colors for each separate part.

The design can be separated into a supporting leg, the main pole and base, the motor pulleys, the top pulley, and the bearing peg as indicated in Figure 1.

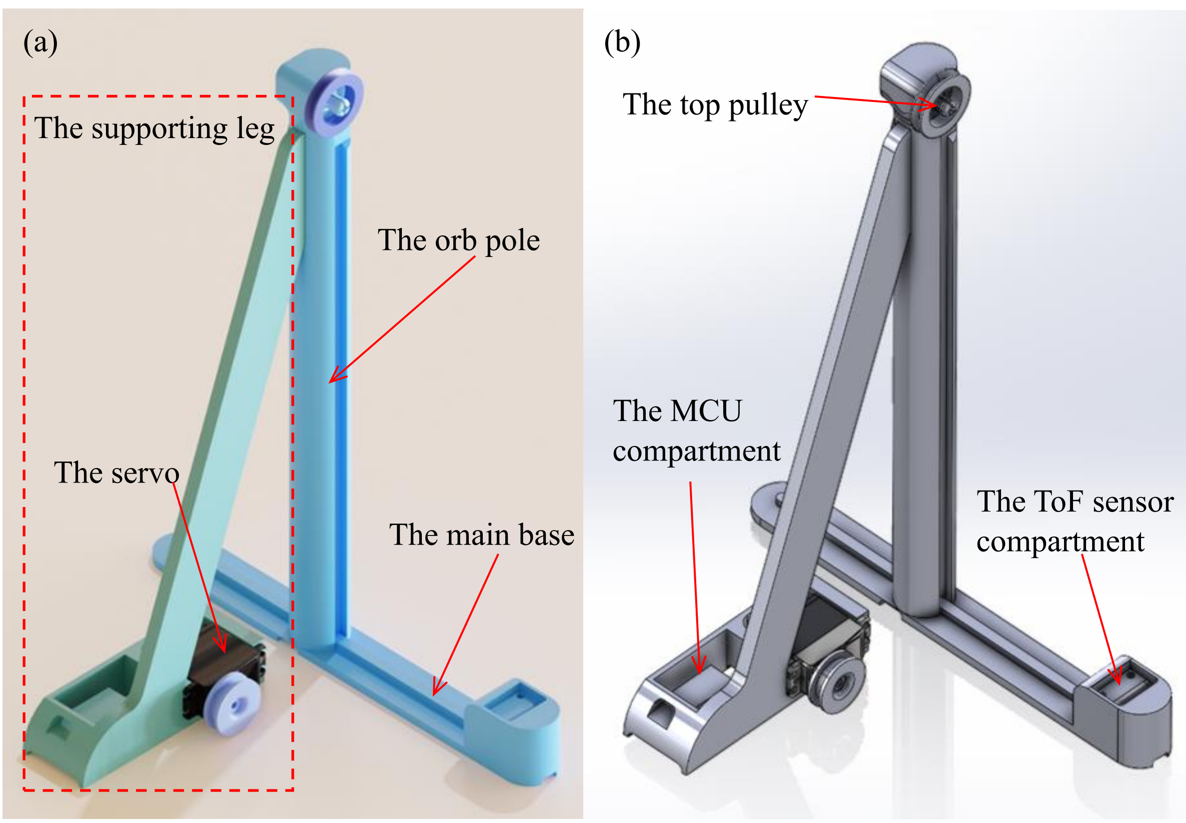

Figure 1. 3D rendering of the design in (a) different colors, and (b) the same color.

3.1 The supporting leg

The supporting leg's main purpose is to stabilize the design side-to-side, as shown in Figure 2.

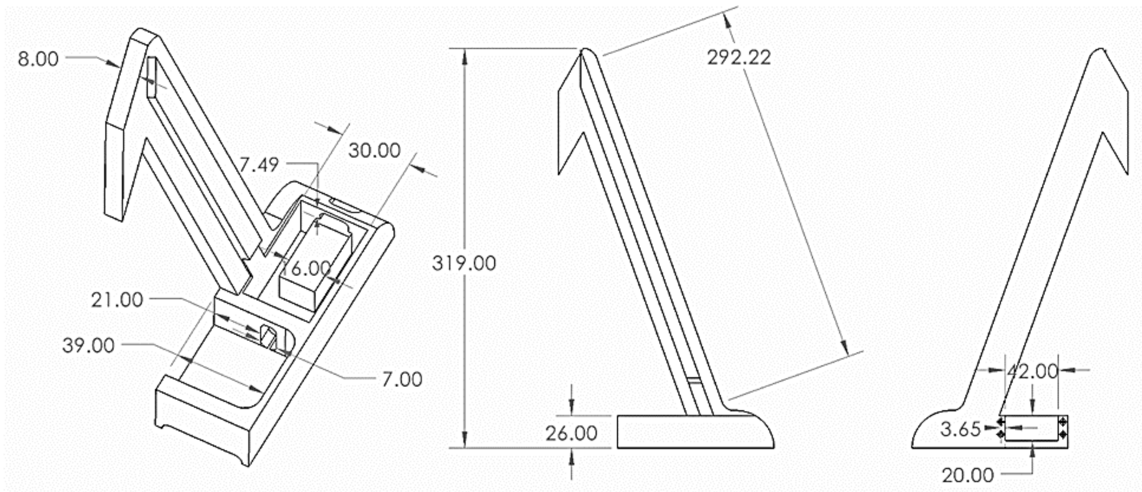

Figure 2. CAD drawing of the supporting leg (uses mm).

The supporting leg plays an important role in the design because it holds the ESP32 and the motor. The main stem of this part also has a rib to improve mechanical strength. A gap underneath the footprint allows for the routing of wires underneath the device. In practice, a large compartment in the base is preferred since it simplifies the routing of the wires. A hole connects between the ESP32 and the motor compartment allowing for passing the wire through.

This part has an angled connector on top that allows it to slide easily together with the base. The angled design prevents it from sliding out easily, as it doesn't rely only on friction to hold it in.

3.2 The main base and the orb pole

This part's main purpose is to elevate the orb's height while maintaining structural integrity. Thus, the orb pole is designed thick.

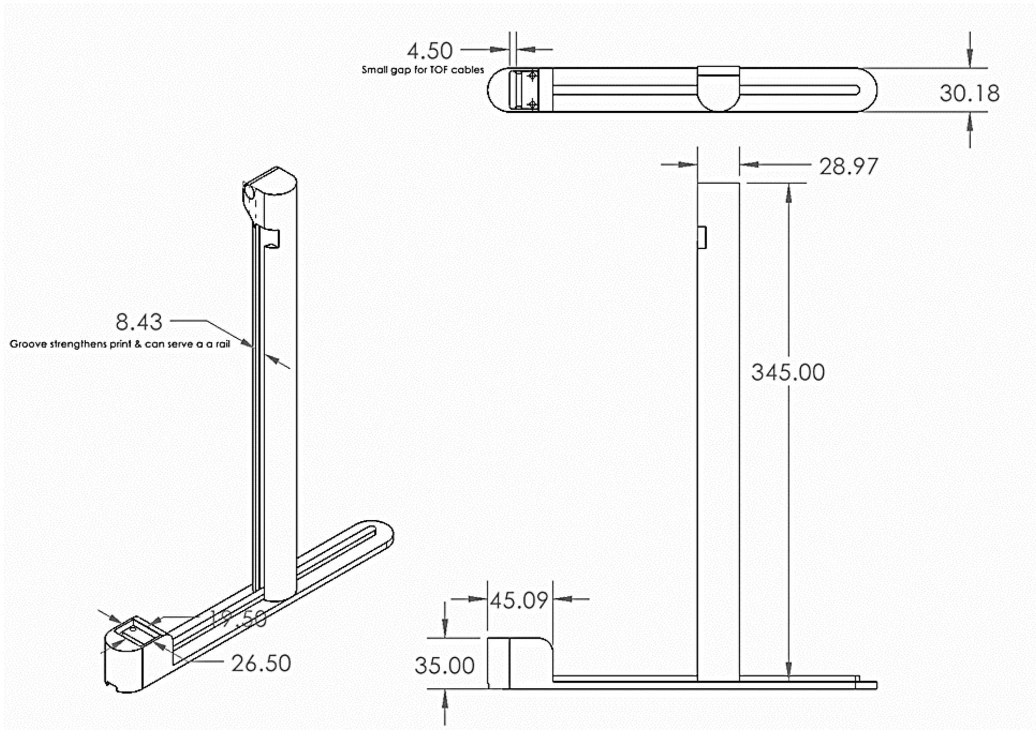

Figure 3. CAD drawing of the main base (uses mm).

A groove and rail running down the length of the pole can be used to align the orb horizontally, at the same time, it gives extra resistance to bending in a similar way that a rib would. A rib is added to the base to strengthen the structure.

At the front side of the base, there is a raised section that holds the ToF sensor. It has a small gap that allows the wires that come off the bottom of the sensor to run down under the base and make an electrical connection to the microcontroller.

Finally, at the top of the pole, there is a hole that allows inserting the bearing peg. This allows easy iteration on the peg or switching out different bearings without having to reprint.

3.3 The servo spool

The servo spool holds a length of wrapped-up string, and the string attaches to the servo motor.

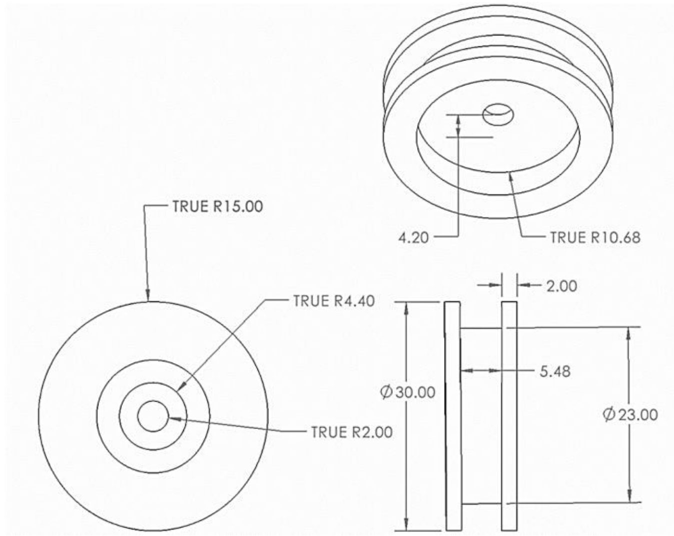

Figure 4. CAD drawing of the servo spool (uses mm).

An inexpensive hobbies level servo motor from Parallax (model 900-00360) is selected as the actuator.

3.4 The bearing peg

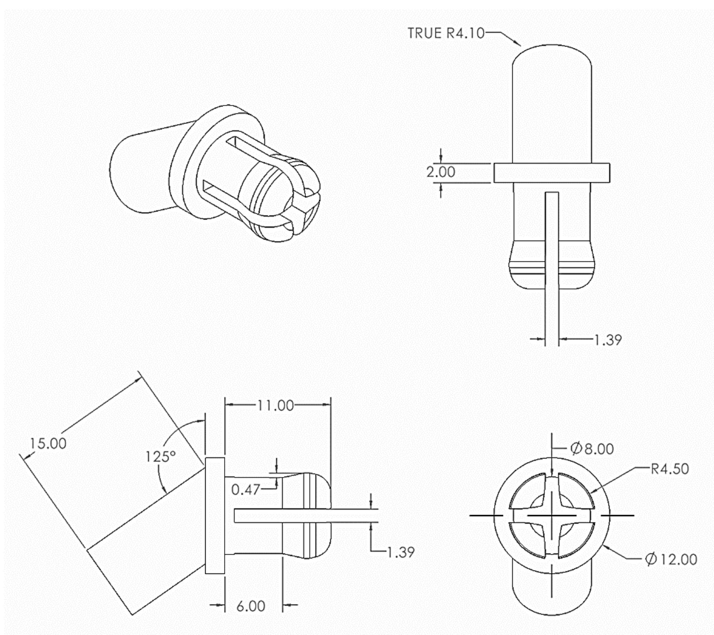

Figure 5. CAD drawing of the bearing peg (uses mm).

This peg is an easy snap-on, snap-off connection for a 608-2RS ball bearing, while it is compatible with any bearing with an inner diameter of 8mm. This bearing is small which allows the pulley to be low profile and not overwhelm the rest of the design. The four beveled prongs can be pushed together towards the center, decreasing the diameter of the shaft. This allows a bearing to slip over the beveled portion and be held on by the plastic's resistance to deformation. The angled connector slides into the top of the main base.

3.5 The top pulley

This is a simple pulley that slips onto the outside of a 608-2RS ball bearing.

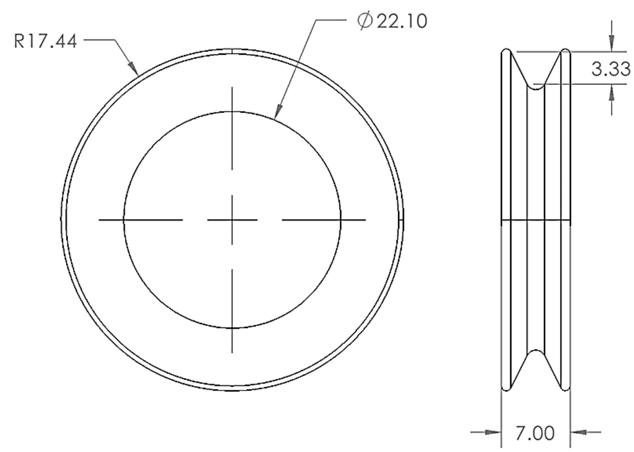

Figure 6. CAD drawing of the top pulley (uses mm).

Generally, 3D printers will tend to print holes slightly smaller than their exact dimension. So, to get a snug friction fit the inner diameter of the pulley, was made slightly larger than the outer diameter of the ball bearing. Since a tight fit is better and can be sanded if it is too small, it was not made too much larger than the bearing diameter.

3.6 Electrical design

Three main electrical parts are used in this project:

1. The Parallax Inc. 900-00360 continuous rotation servo motor.
2. An Espressif ESP32 WROOM-32 Microcontroller module.
3. The VL53L4CD Time-of-Flight distance sensor module from Adafruit.

3.7 Servo selection

We chose a continuous rotation servo in our design. This is because positional rotation servos are only capable of up to 360° of maximum rotation, leaving them only capable of winding up a certain amount of string depending on the spool's radius. Increasing the spool's radius is undesired as we want the motor to stay close to the ground where it is easy to mount, and we want it to be low profile to increase stability. Whereas a continuous rotation servo can wrap up an arbitrary length of string regardless of the spool's diameter. Lastly, continuous rotation servo servos are controlled by speed rather than angular position like a normal servo, thus choosing a servo with a built-in encoder was important.

3.8 Microcontroller selection

The ESP32 microcontroller was chosen for its small, affordable, and incredibly flexible form factor. It supports the Arduino software framework, known for its wide community support and ease of use. For experienced users and educators who wish to offer a deeper understanding of firmware development, the Espressif IoT Development Framework (IDF) provides low-level access with a C language application programming interface (API). The ESP32 also has wireless capabilities which in the future could be used to make the Magic Orb fully wireless or enable other interesting wireless demo features.

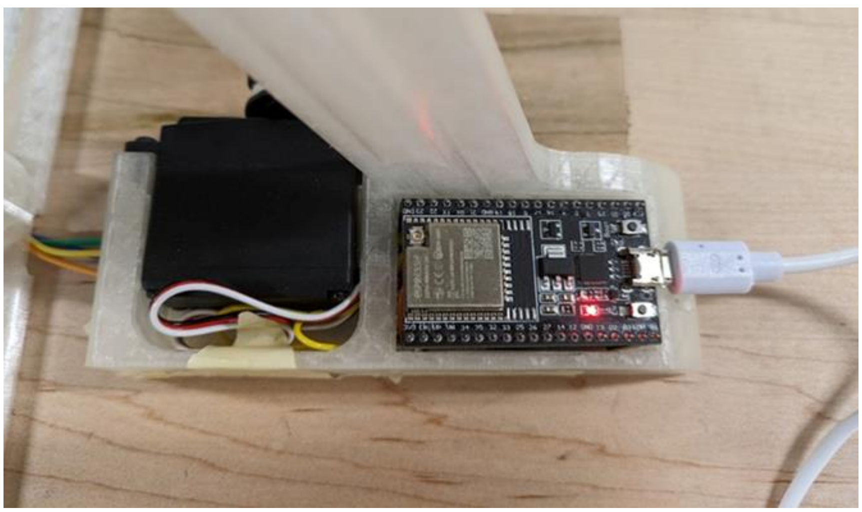

Figure 7. Foot of the design, where an ESP32 module and a servo motor have been installed.

3.9 Distance sensor selection

Finally, the VL53L4CD ToF sensor module from Adafruit was chosen as it offers precise and fast distance sensing (up to 100Hz sampling rate). It additionally gives good educational value as it can teach how distances can be measured using time and our knowledge of physical constants. The same lesson can be taught using ultrasonic sensors, however, using a laser may be more interesting to younger students. Besides, it offers teaching opportunities to discuss the infrared spectrum of light. The ToF sensor uses an IR laser which is invisible to humans. However, a camera can see the infrared light. The infrared laser can be shown to the students live using a camera, while being invisible during normal use. It was chosen over an ultrasonic sensor for its size, reliability, and accuracy across its full range, without suffering from blind spots. It additionally, has a very narrow FOV which allows it to measure the distance only in a straight line rather than a wide cone which could cause problems for this project.

Figure 8. The installed ToF sensor.

3.10 Electronic assembly

Each component was assembled with prototyping wires, and the ESP32 is powered by a small battery bank connected by a micro-USB cable. The wiring is routed underneath the fixture, ensuring it remains discreet and does not detract from the presentation. Figure 9 shows the connection diagram of the design. The servo and the ToF module draw 5V and 3.3V power from the ESP32 module accordingly. The ToF sensor module communicates with the MCU through the I^2C bus. The servo is controlled by a single I/O pin from the ESP32 MCU via a PWM signal, and it sends angular position feedback to the MCU from its onboard hall effect-based encoder. The ESP32 module draws nearly 78mW of power by itself, the servo draws around 1W, and the ToF module draws roughly 72mW. A power bank can support hours of operation.

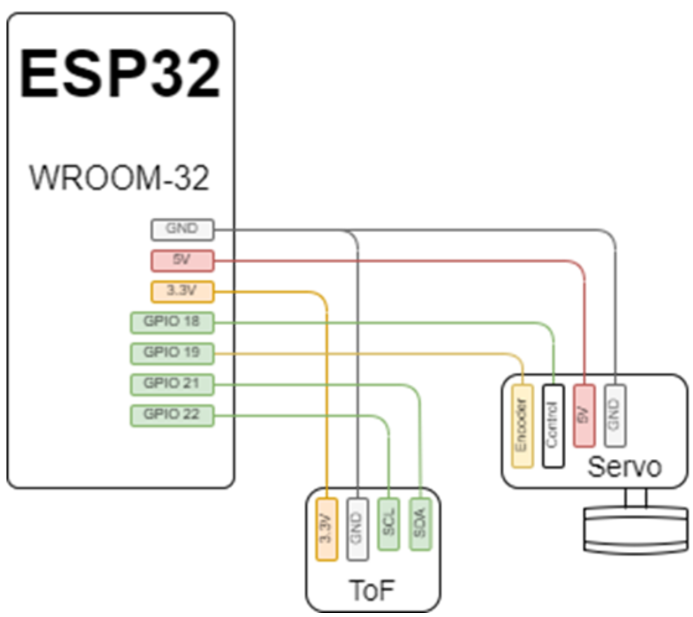

Figure 9. Electrical connection diagram.

4 Software development

The control software was written in C language using the Espressif IDF in combination with the Arduino framework. There are two main parts to the software.

1. A control algorithm for the 360-degree servo.

2. The main behavior logic for the motion of the Orb.

Both parts used a mix of Arduino and Espressif libraries, however the bulk of the lower-level code is in the control algorithm. For example, the number of revolutions done by the servo is stored in flash memory on the ESP32 using the Espressif preferences library. The Arduino IDE works with the ESP32 almost out-of-box. However, the Espressif IDF, while also compatible out of the box, requires more setup to use. Using the Arduino framework with the Espressif IDF, while not difficult, requires manually downloading the Arduino C libraries and integrating them with your IDF workspace. Overall, the Espressif IDF provides many potentially useful functions, but for most simple projects the Arduino framework is plenty sufficient.

4.1 Servo control algorithm

The servo control algorithm allows getting and setting the target angle of the servo to an arbitrary absolute number such as 640° or –90°. This capability makes it much easier to write the main logic of the program since we can just map the target height linearly to an angle on the servo. Creating this algorithm is not entirely trivial since the servo is controlled by speed and not by position like a conventional servo. However, we can make good use of the built-in encoder to detect the rotation of the servo from 0-360°. It's important to note that when the encoder passes 360° it wraps back around to 0°. This is called the relative rotation in the following discussion.

Appropriate logic is necessary to handle this wrapping behavior. If the encoder jumps from 360° to 0°, a full revolution has been completed. This event is tracked by a variable so we can calculate the absolute revolution degrees we have accumulated from 0°. Thus, the following formula is used:

$$\theta_{absolute} = 360° \cdot N + \theta_{relative}$$

Where *N* is the number of full revolutions made and $\theta_{relative}$ is the encoder reading converted to degrees.

We must account for negative rotations as well. If the encoder jumps from 0° to 360°, we need to recognize that the servo has just "undone" a full turn. And if we are rotating in the negative direction, our $\theta_{relative}$ should be treated as its inverse: $(360° - \theta_{relative})$. One last thing: since *N* is now representing our direction *and* the number of rotations, it should never equal 0. With this in mind, we get our new formula, represented piecewise as it is easier to understand and implement:

$$\theta_{absolute} = \begin{cases} N > 0 & 360° \cdot (N-1) + \theta_{relative} \\ N < 0 & -360° \cdot (N+1) - (360° - \theta_{relative}) \end{cases}$$

4.2 Setting the absolute rotation

Once we can read the absolute rotation, we can simply move the servo towards that rotation until we reach the target. A naïve algorithm set to do this is likely to overshoot or not stop smoothly. To fix this, we can slow the rotation linearly as we approach the target.

4.3 System flow chart

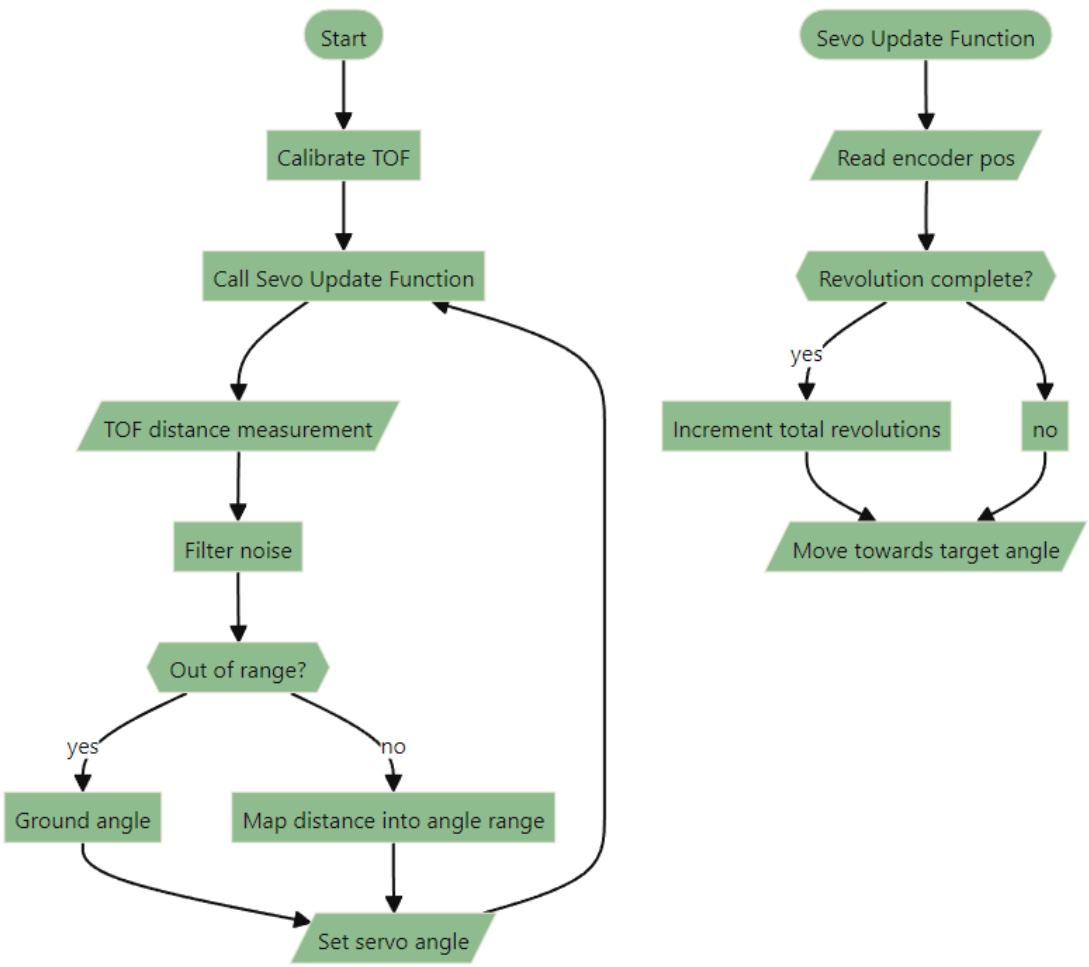

Figure 10. The Magic Orb application program flow chart.

The main code is made up of a servo control class implementing the previous algorithm (Right of Figure 10), as well as the main file containing the basic logic flow (Left of Figure 10.). The main logic flow is mainly comprised of a while loop that runs forever. We start by updating the servo and then move on to reading the ToF distance. Noise is filtered from the distance by taking a moving average. A moving variance is also taken (the variance of the moving window of data). When nothing is in range of the ToF sensor the variance of the data increases dramatically as it is mostly random noise. However, when something is within range the variance is much lower.

If the measurement is out of range we allow the servo to return to the ground, otherwise we map the distance to an angle. This is done using the circumference of the spool (linear distance traveled per revolution). Finally, we set the target angle on the servo which will then be updated in the next loop.

5. Results and Discussion

The 3D printing process was relatively simple. Each part was designed to have at least one large flat face on which to print. The only special consideration was that to make a large display we had to use a larger printer. Simple downward-angled slide-together connectors were created for each component to allow the whole assembly to be modular. For some of these connections, it was useful to wrap the connector in tape until the friction was at a desired level. The parts were then sliced using Cura and printed in clear PLA on a Creality CR-10 (used for its large print height).

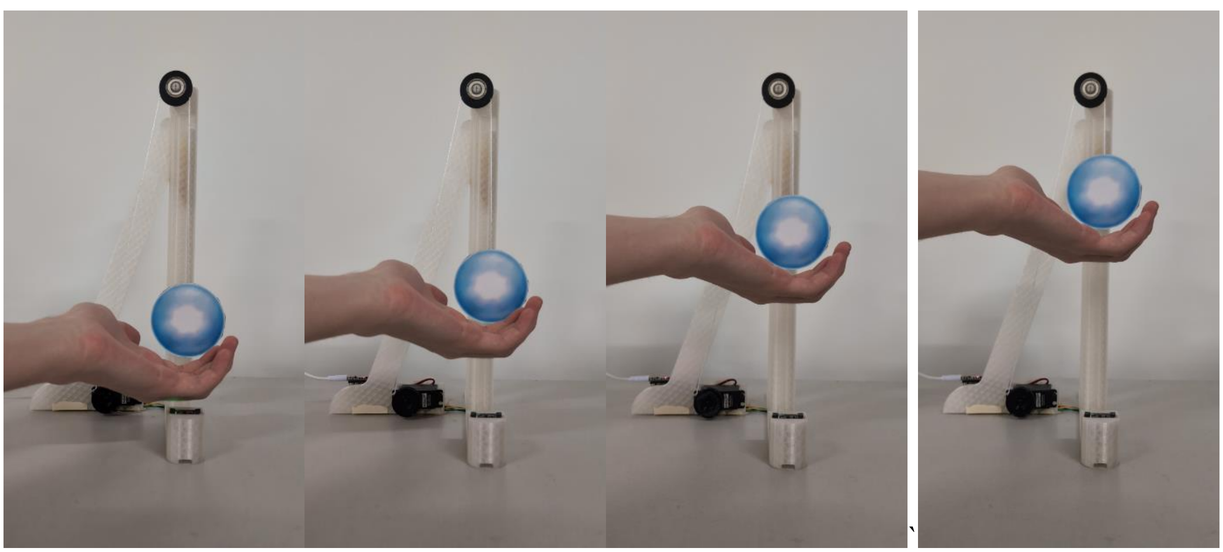

Figure 11. Demonstration of operation.

When the device is turned on the first thing to happen is it lowers the ball to the “ground” if it isn’t already there. It then sits static until a hand (or any other object) is placed above the ToF sensor. The motor then turns and raises the orb so that the bottom of the orb lines up with the position of the hand (so it looks like it is on top). If the hand is removed (or if it is raised out of range) the orb is lowered back to the ground. The orb is also not allowed to be raised higher than the pulley, so it does not fall off. It takes around one second for the orb to go from the bottom to the top and the same time top to bottom. However, since the hand is usually being moved smoothly the orb often stays very close to level with the hand. The reaction time is almost unperceivable by plain observation, and it starts moving almost as soon as the hand starts to move.

The Magic Orb was demonstrated to K-12 cohorts of students for the calendar year 2023 and included over 2000 visitors to our robotics facility. It was often a specific point where students stopped and asked questions and competed with much more expensive robotic platforms in terms of fun and interest [6][7][8]. In our most recent open house event in Spring 2024, about 150 K-12 students between the ages of 8 and 16 have visited the Magic Orb project. About 20 students came back a second time to the display on their own accord in order ask further questions and play with the orb. Of these visitors, 10 students came back yet again to play with the orb and ask more questions. This indicates that the Magic Orb is an attractive demo capable of sparking further interest in engineering.

6. Conclusions

The Magic Orb project can be used in either introductory or advanced courses in the engineering curricula. This demonstration integrates concepts covered in all three related engineering topics: mechanical, electrical, and computer programming. The Magic Orb is also highly customizable for the instructor to cover freshman and upper classmen concepts. When used as a demonstration for outreach events, the demo offers a great opportunity to attract young audiences in order to spark interest and introduce basic engineering concepts.

References

[1] J. G. Cherng, B. Q. Li, and N. Natarajan, "Development of a Mechatronics Course for Senior Mechanical Engineering Students," in *120th ASEE Annual Conference & Exposition, June 23-26, 2013*

[2] H. Vasquez and A. A. Fuentes, "Integration of Sensors and Low-Cost Microcontrollers into the Undergraduate Mechanical Engineering Design Sequence," in *120th ASEE Annual Conference & Exposition, June 23-26, 2013*

[3] M. Barger and R. Gilbert, "New Mechatronics Education Initiatives in 2-year Programs," in *2018 ASEE Annual Conference & Exposition, Salt Palace Convention Center, Salt Lake City, UT, June 24-27, 2018*

[4] S. A. Yost and D. D. Maggio, "The Mechatronics Road Show: Building on Success in Mechatronics Curriculum Development," in *2004 American Society for Engineering Education Annual Conference and Exposition*

[5] J. R. Haughery, D. R. Raman, A. L. Kaleita, and S. A. Freeman, "Towards Understanding the Impacts, Why's, and What's Behind Mechatronic-based Projects and Student Motivation," in *2017 ASEE Annual Conference & Exposition, Columbus, Ohio, June 24, 2017*

[6] I. Ranatunga, M. Beltran, N. A. Torres, N. Bugnariu, R. M. Patterson, C. Garver, and D. Popa, "Human-Robot Upper Body Gesture Imitation Analysis for Autism Spectrum Disorders, " in *International Conference on Social Robotics, Bristol, UK, October 27-29, 2013*

[7] M. Moosaei, S. K. Das, D. Popa, and L. D. Riek "Using facially expressive robots to calibrate clinical pain perception, " in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Vienna, Austria, March 6-9, 2017*

[8] S. Abubakar, S. K. Das, C. Robinson, M. N. Saadatzi, M. C. Logsdon, H. Mitchell, D. Chlebowy, and D. Popa, "Arna, a service robot for nursing assistance: System overview and user acceptability," in 2020 *IEEE 16th International Conference on Automation Science and Engineering (CASE), Hong Kong, China, August 20-21, 2020*