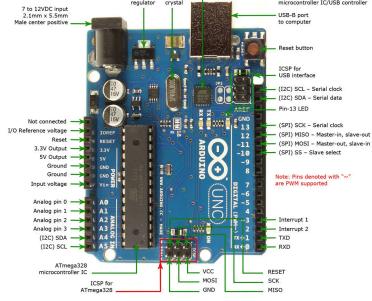
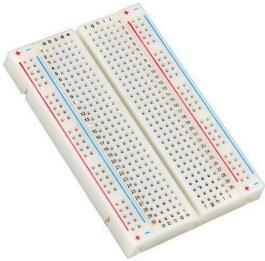
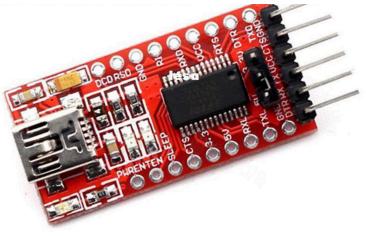
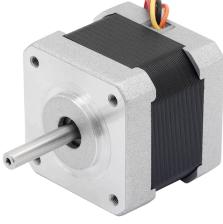
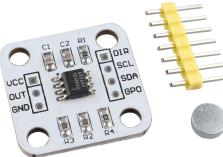


Hardware Component	Image	Specification & purpose
Arduino UNO		<p>Specification:</p> <ul style="list-style-type: none"> Microcontroller: ATmega328P Operating Voltage: 5V Digital I/O Pins: 14 (6 PWM) Analog Input Pins: 6 Clock Speed: 16 MHz <p>Purpose:</p> <ul style="list-style-type: none"> -central controller for the game -manages inputs and outputs of the game <p>Potential scenario:</p> <ul style="list-style-type: none"> The computer receives a camera view of the bottom of the table from the ESP32 CAM. The computer processes the camera view to decide on next movement instructions based on the current game camera view. The Arduino receives instructions from the computer. The Arduino sends control signals to the motor drivers according to the instructions received. The motor drivers interpret the control signals from the Arduino and adjust power output to the motors. The motor drivers provide the necessary power to the motors, enabling precise movement of the foosball poles.

A4988 Motor Drivers (for Side Motion of Foosball Poles)		<p>Specification:</p> <ul style="list-style-type: none"> Voltage Range: 8-35V Current per Phase: Up to 2A (with adequate cooling) Microstepping: Up to 1/16 step <p>Purpose:</p> <ul style="list-style-type: none"> -Controles stepper motor* <p>Potential Scenario:</p> <ul style="list-style-type: none"> -Receives direction signals from arduino - Translate these direction signals into controlled and precise electrical pulses for the stepper motor -Results the stepper motors to deliver controlled and precise side-to-side motion of the poles. <p>*manages side to side motion of the poles</p>
17HS4401 Stepper Motors (for Side Motion)		<p>Specification:</p> <ul style="list-style-type: none"> Step Angle: 1.8° (200 steps per revolution) Holding Torque: 4.2 kg·cm Voltage: 12V Current per Phase: 1.7A <p>Purpose:</p> <ul style="list-style-type: none"> -delivers side-to-side motion of the foosball poles. -stable and controlled movement

Capacitors		<p>(Connected to A4988 Motor Drivers)</p> <p>Specification:</p> <p>Capacity: 100 µF (typical for motor driver circuits)</p> <p>Voltage Rating: At least 25V (adjust according to motor power supply voltage)</p> <p>Purpose: The capacitors are used to stabilize the power supply for side-to-side motion's motor driver.</p> <ul style="list-style-type: none"> -smoothes out voltage fluctuations -protects against spikes, -ensures consistent power delivery to the motors and -prevents damages to the driver
Breadboards		<p>Specification: Prototyping boards</p> <p>Purpose: Platform for wiring connections between components</p>
Red Button (Game Start/Reboot)		<p>Specification: Push-button switch, ??? V</p> <p>Purpose: Begin, end reset functionality to the game</p> <p>When the button is pressed, it sends a signal to the Arduino, which then initiates the game start sequence, ends the game or reboots the system in case of a problem .</p>

LED Strip (Illumination for Camera View)		<p>Specification:</p> <p>LED strip</p> <p>Operating Voltage: ??</p> <p>Control Protocol: Digital</p> <p>Purpose:</p> <ul style="list-style-type: none"> - Placed at the bottom of the foosball table to better the view for the ESP32 CAM to facilitate more accurate and precise image processing for game state analysis from the computer.
ESP32 CAM (Camera for Image Processing)		<p>Specification:</p> <p>Processor: ESP32 with integrated Wi-Fi and Bluetooth</p> <p>Camera: OV2640 sensor (2MP)</p> <p>Operating Voltage: 3.3V (ESP32) with 5V input supported</p> <p>Purpose:</p> <ul style="list-style-type: none"> - Provides data of the game to the computer <p>Potential scenario:</p> <ul style="list-style-type: none"> -captures real time images of the game -processes image frames -sends data to the computer for in depth analysis - Computer sends action commands to Arduino for the next state actions of the poles and computer aided players.

FTDI Adapter (For Programming and Configuring the ESP32 CAM)		<p>Specification: USB to Serial TTL converter 3.3V</p> <p>Purpose: - programming and configuring the ESP32 CAM</p>
Stepper Motor (for the Rotary Motion of Poles)		<p>Specification: Model: NEMA 17 (example)</p> <p>Step Angle: 1.8 degrees (200 steps per revolution)</p> <p>Holding Torque: Typically around 0.5-1.5 Nm, depending on the exact model</p> <p>Rated Voltage: 12V</p> <p>Current per Phase: ~1-2A</p> <p>Purpose: - provide precise rotary motion to the poles. - Each motor is attached to a pole, allowing for controlled rotation</p>
As5600 (Magnetic Encoder, for the Stepper Motors for the Rotary Motion of the Poles)		<p>Specification: Type: Magnetic encoder Interface: I2C or Analog output Resolution: 12-bit (providing 4096 distinct positions per revolution) Operating Voltage: 3.3V or 5V</p> <p>Purpose: - Provides feedback on the position of each stepper motor's shaft, - ensures the correct orientation and rotation of the foosball poles. - feedback loop that allows the system to know the exact angle of the poles, enabling precise control, avoiding undesired pole motions</p>

<p>Digital Screen (For the Score Display)</p>		<p>Specification:</p> <p>Type: 16x2 LCD Display with I2C Interface (e.g., LCD1602)</p> <p>Operating Voltage: 5V</p> <p>Communication Protocol: I2C (typically uses two pins, SDA and SCL, for communication with Arduino)</p> <p>Backlight: Adjustable (usually blue or green)</p> <p>Purpose:</p> <ul style="list-style-type: none"> -displays the current score chart -Counter of the game till it ends -Game start and Game over indication at the beginning and end of the game <p>Potential Scenario:</p> <p>Arduino sends an update to the board based on the game state, which is triggered by signals from sensors or by processed image data from the ESP32 CAM, sent through the computer.</p>
<p>L298N Motor Driver (for Stepper Motor Control for rotary motion)</p>		<p>Specification:</p> <p>Voltage Range: 5-35V</p> <p>Maximum Current: 2A per channel</p> <p>Dual H-Bridge: Controls two motors independently</p> <p>Purpose:</p> <ul style="list-style-type: none"> -manages the power delivery to the rotary stepper motors based on control signals from the Arduino -adjusts the speed and direction of each motor

WEMOS D1 R32		<p>Specification:</p> <p>Processor: ESP32, dual-core</p> <p>Operating Voltage: 3.3V (with onboard regulators for 5V)</p> <p>Communication: Wi-Fi, Bluetooth, I2C, SPI, PWM, ADC, and more</p> <p>Purpose:</p> <ul style="list-style-type: none"> -main controller of the foosball table's rotary motion motor system. - processes input signals, calculates desired pole positions, and sends control signals to the motor drivers. <p>Reference:</p> <p>cs358-resources/code/actuators/SimpleFOC/stepper_L298N/WEMOS_D1_R32/stepper_L298N_D1.ino at main · epfl-cs358/cs358-resources</p>
Optical Switch End Stop sensor		<p>Specification</p> <p>Operating Voltage: 5V.</p> <p>Mounting: Fixed near the extremities of the rotary range of the foosball poles.</p> <p>Wiring: Connected to digital input pins on the Arduino to detect when each end-stop is triggered.</p> <p>Purpose</p> <p>defines the physical limits of movement for each motor-controlled foosball pole.</p> <p>detect when the pole has reached the farthest allowed rotation in either direction.</p> <p>-prevents the motor from over-driving the poles beyond their intended range</p>