Introduction

This document outlines the key components and technical considerations involved in the wearable robotic assistive device. Each section provides a detailed explanation of one hardware component, including its function, selection criteria, and reasoning behind the chosen specifications.

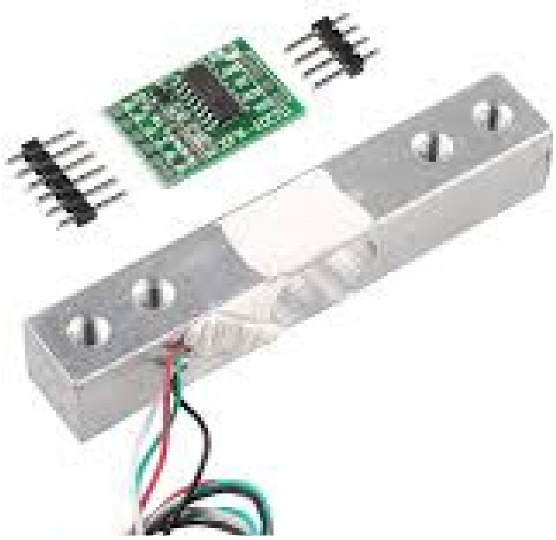
Component 1: Load Sensor + HX711

Purpose and Application in the System:

To monitor how much weight the child is bearing during movement, we integrated a 10kg Strain Gauge Load Cell paired with an HX711 Weight Sensor Amplifier Module. This combination serves as a feedback system to measure the support ratio—whether the child is fully, partially, or not at all bearing their body weight.

In our assistive robot, it is critical to determine how much support the child needs at any moment. The load cell is mounted under the child's foot, allowing us to detect vertical ground reaction forces. If minimal weight is detected, it implies that the robot is providing most of the support. Conversely, a higher force reading indicates that the child is actively participating in the walking process.

The HX711 amplifier conditions and converts the analog signal from the load cell into digital data, which can then be read accurately by the microcontroller. The 10kg capacity of the sensor makes it suitable for children weighing up to 15 kg, as each leg is estimated to support approximately 7–8 kg during motion.



Component 2: Servo Motors

Overview and Role in the System:

Servo motors are used to control the movement of the child's hips, knees, and ankles by providing precise angular rotation. Their accurate response to control signals allows the robot to guide or assist the child's walking motion, making them essential for converting electronic commands into natural leg movements.

Motor Selection:

Servo motors without built-in feedback systems were selected due to their cost-effectiveness, which makes them suitable for graduation projects. Although the absence of a built-in feedback mechanism presents a minor limitation, a solution has been developed and will be addressed in a later section. This choice ensures a practical balance between performance requirements and budget constraints.

To determine the appropriate torque for each joint, we conducted basic calculations using estimated parameters:

- a child aged 3 years
- weighing approximately 15 kg
- around 100 cm in height.
- **Hip Joint Servo**

Since only one leg is supported by each hip, we considered 1/4 of the total body weight (about 3.75 kg) acting on one side. The estimated distance from the hip to the foot is 50 cm, so we placed the center of mass at 25 cm (0.25 m).

Using the torque formula:

Torque = Force × Distance

Force = 3.75 kg × 9.81 m/s^2 ≈ 36.8 N

Torque = 36.8 N × 0.25 m = 9.2 N·m ≈ 92 kg·cm

To ensure safe and smooth operation under dynamic conditions, we selected a servo with 150 kg·cm torque (model DS51150), which provides extra strength and reliability.

2. **Knee Joint Servo**

To estimate the required torque for the knee joint, we considered half of the leg's mass (about 1.875 kg) acting at half the distance from the knee to the foot (0.125 m).

<u>Using the torque formula:</u>

Force = 1.875 kg × 9.81 m/s² ≈ 18.4 N

Torque = 18.4 N × 0.125 m = 2.3 N·m ≈ 23 kg·cm

To ensure safe movement and provide a buffer for dynamic motion, we selected a servo motor rated at 60 kg·cm (model RDS5160), which is more than double the required torque for reliable assistance.

3. **Ankle Joint Servo**

The ankle joint supports a smaller portion of the leg's weight. We estimated about 0.94 kg acting at a distance of 0.1 m from the ankle.

<u>Using the torque formula:</u>

Force = 0.94 kg × 9.81 m/s² ≈ 9.2 N

Torque = 9.2 N × 0.1 m = 0.92 N·m ≈ 9.2 kg·cm

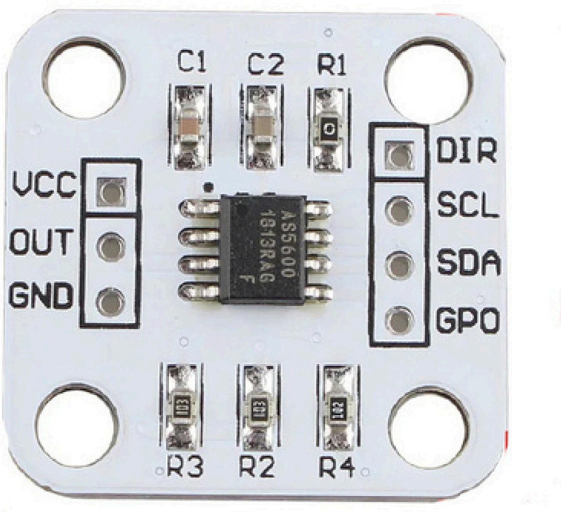
To ensure smooth and reliable movement, we selected a servo motor rated at 25 kg·cm (model DS3225), which provides nearly 3× the required torque—ensuring safety and mechanical headroom.

Component 3: Rotary Encoder AS5600

In order to provide accurate feedback on joint positions, a feedback mechanism was essential. Initially, we considered using servo motors with built-in feedback, which would have simplified implementation. However, their high cost made them impractical for a graduation project with a limited budget.

We then explored the use of an Inertial Measurement Unit (IMU), which offers orientation and acceleration data. While the IMU is a powerful sensor, it is highly sensitive and captures all movement—including unintended motions from the child. To use it effectively, we would have needed complex filtering algorithms to isolate servo-induced movement from the child's natural shifts, adding unnecessary complexity.

Ultimately, **we selected rotary encoders as the ideal solution.** They are low-cost, easy to integrate, and provide precise angular position feedback. When connected to each servo shaft, the rotary encoder accurately reports the servo's rotation angle in real-time, allowing us to monitor and adjust movement with confidence. This ensures 100% reliable positional feedback without the need for external filtering or advanced processing.



Component 4: PCA9685 PWM SERVO DRIVER

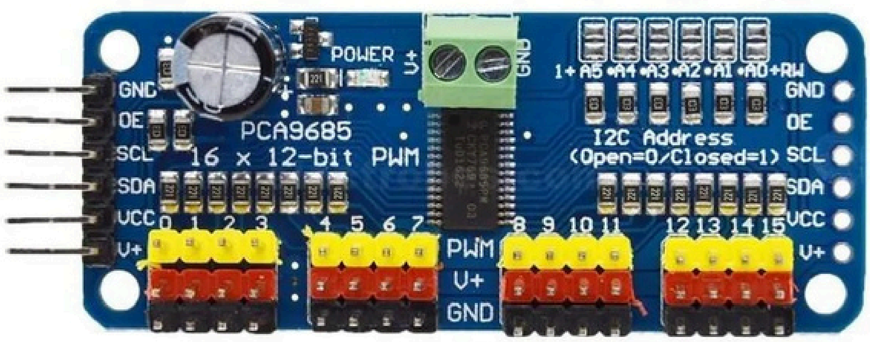
The PCA9685 is a 16-channel PWM (Pulse Width Modulation) driver that allows precise and independent control of multiple servo motors. It communicates with the Raspberry Pi via the I²C protocol, significantly reducing the number of GPIO pins required for controlling several servos simultaneously.

Reason for Selection:

In our project, multiple servo motors are used to actuate the hip, knee, and ankle joints on both legs. Controlling all of them directly from the Raspberry Pi would consume a large number of GPIO pins and place additional processing load on the main controller. By using the PCA9685 module, we can:

- Control up to 16 servos with just two I²C pins (SDA and SCL).
- Achieve stable and smooth servo motion, as the module generates PWM signals independently without being affected by Raspberry Pi processing delays.
- Easily scale the system to accommodate more actuators if required in future upgrades.

This choice ensures reliable servo control, minimizes wiring complexity, and provides flexibility for future system expansion.



Component 5: EMG

Purpose and Function in the System:

The Electromyography (EMG) sensor measures the electrical activity produced by skeletal muscles during contraction. In our project, EMG sensors are placed on the muscle groups responsible for moving the hips, knees, and ankles. By detecting muscle activation, the system can determine when the child is attempting to initiate movement, making it an ideal replacement for the flex sensors in Phase 2 of our rehabilitation program.

Reason for Selection:

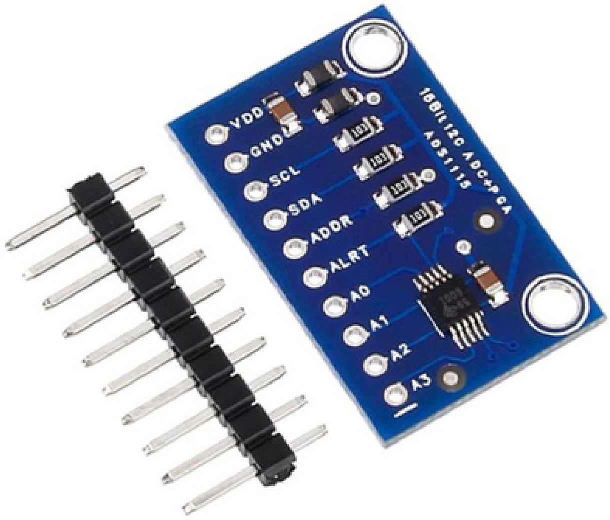
- Direct muscle intent detection: Unlike flex sensors, which detect joint bending, EMG sensors capture the muscle activation signal before visible movement occurs, allowing for faster and more accurate responses.
- Ease of use: Modern EMG modules are simple to set up, easy to program, and provide clear analog output that represents muscle activity.
- Improved training quality: Detecting real muscle engagement ensures the child is actively participating, which is essential for effective rehabilitation.

Connection Method:

The EMG sensor outputs an analog voltage proportional to the muscle's activity level. Since the Raspberry Pi cannot read analog signals directly, an Analog-to-Digital Converter (ADC) module is used between the EMG sensor and the Raspberry Pi. The ADC converts the analog EMG signal into digital data, which the Raspberry Pi processes to determine when to provide assistive movement.

Component 6: ADC ADS1115

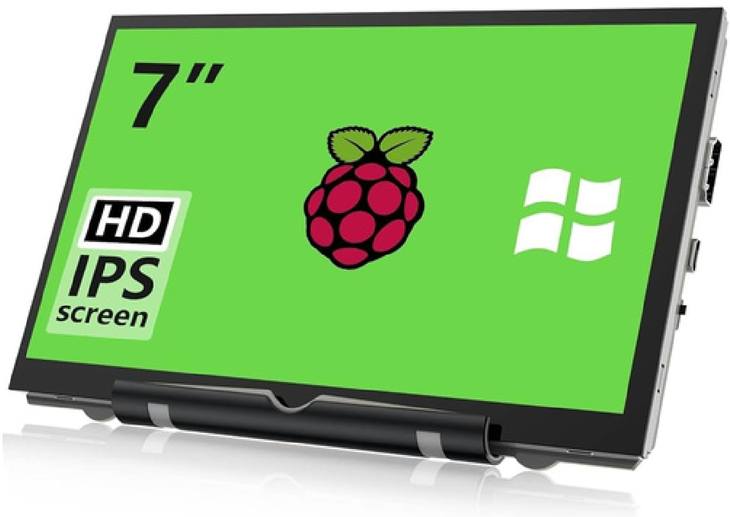
The ADS1115 is a 16-bit Analog-to-Digital Converter that enables the Raspberry Pi to read analog signals from sensors such as EMG. Since the Raspberry Pi lacks analog input pins, the ADS1115 is essential for converting these signals into digital data. It offers 4 input channels, high accuracy, and communicates via I²C, requiring only two pins (SDA and SCL), making it efficient and easy to integrate.



Component 7: Hamtysan Screen and USB Mini Speaker

The Hamtysan screen is a compact, high-resolution display used to present interactive visual feedback, such as animated faces and motivational graphics, to engage the child during training sessions. It connects directly to the Raspberry Pi via HDMI or compatible display interface.

The USB mini speaker provides clear audio output for encouraging voice prompts. It is powered and connected via USB, allowing the Raspberry Pi to play pre-recorded or AI-generated motivational phrases using text-to-speech software. Together, the screen and speaker create an interactive AI-based motivation system that enhances the child's rehabilitation experience.



Component 8: Microcontroller RASPBERRY PI 4

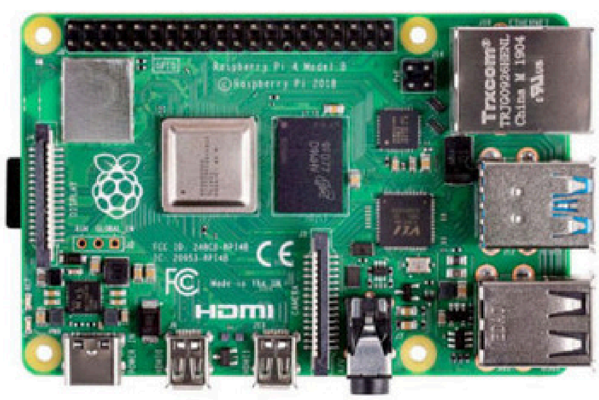
The Raspberry Pi 4 serves as the main controller of the system, managing sensor data processing, servo motor control, and AI-based interaction features. It was chosen for its high processing power, versatility, and compatibility with a wide range of peripherals, such as displays, speakers, and wireless communication modules.

Analog Input Limitation:

The Raspberry Pi 4 does not have built-in analog input pins, meaning it cannot directly read analog signals from sensors like EMG. To address this, we integrated the ADS1115 Analog-to-Digital Converter, which reads analog sensor outputs and sends the converted digital data to the Raspberry Pi via I^{2}C.

Servo Motor Control:

While the Raspberry Pi supports PWM output, it has only two hardware PWM channels, which is insufficient for controlling all the servo motors in our system. To overcome this, we use the PCA9685 PWM Servo Driver, which provides 16 independent PWM channels. This ensures smooth and precise control of multiple servos while using only two I^{2}C pins on the Raspberry Pi.



Component 9: Battery

To select a suitable battery for the robot, both voltage and current requirements of all components were considered.

Voltage Requirements

The battery voltage must meet the needs of all components:

- Microcontroller: Operates at either 3.3V or 5V
- Servo Motors:
  - DS51150 (Hip): 6V–8.4V
  - RDS5160 (Knee): 6V–8.4V
  - DS3225 (Ankle): 6V–7.4V
- Sensors:
  - Flex sensors, rotary encoders, load cells typically require 5V

Based on these requirements, a 7.4V LiPo (2S) battery was selected, which is compatible with all servo motors. A DC-DC buck converter is used to regulate the voltage and provide a stable 5V output for the microcontroller and sensors.

Current Requirements

To estimate the total current needed, the peak current consumption of each component was considered:

component	DS51150 (Hip)	RDS5160 (Knee)	DS3225 (Ankle)	Microcontroller	Sensors (combined)	Total Estimated
quantity	2	2	2	1	17	-
peak current	~5A	~3A	~2A	~3 A	~0.05A	-
total peak current	10A	6A	4A	3A	~0.85A	~23.9A

⚠️ A margin was included for unexpected load variations. Therefore, a battery capable of delivering at least 25A continuous discharge is recommended.

Component 10: DC-DC Buck Converter

Purpose and Function in the System:

The DC-DC buck (step-down) converter is used to reduce the 7.4 V output from the battery to a stable 5 V supply for the Raspberry Pi, display, speaker, and sensors. This ensures all low-voltage components receive clean and consistent power, regardless of battery voltage fluctuations during operation.

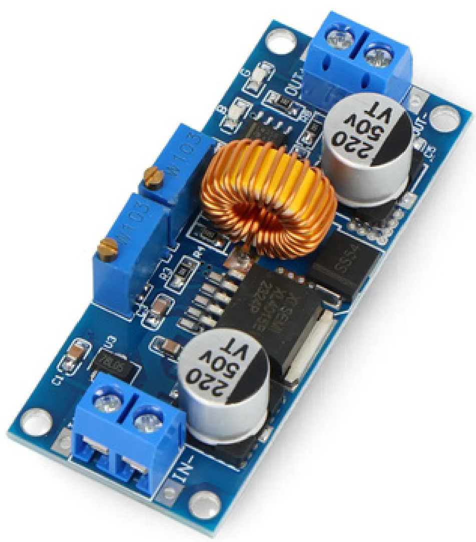
Reason for Selection:

While the servo motors are powered directly from the 7.4 V battery, the Raspberry Pi and other electronics require a regulated 5 V input. A high-efficiency buck converter was selected because:

- It provides stable voltage under varying loads.
- It minimizes energy loss and heat generation compared to linear regulators.
- It supports higher current output, ensuring all components operate reliably without voltage drops.

Choosing the Optimal Converter:

Based on our system's estimated peak current (~3.85 A @ 5 V), we selected a buck converter rated for at least 10 A continuous output. This provides a safety margin to handle startup surges and future system upgrades. A model with high efficiency (≥ 90%) and built-in protections (overcurrent, overtemperature, and short-circuit) was preferred to ensure reliability and safety during operation.



User Interface (UI) – Design and Functionality

To enable a physical therapist (or rehabilitation trainer) to control and monitor the robot easily, a graphical user interface (GUI) will be developed using a laptop.

Purpose of the Interface

The interface allows the therapist to:

- Select the training phase (e.g. Phase 1 or Phase 2).
- Choose the exercise type (walking or static support).
- Set values such as:
  - Training duration
  - Support level
- View real-time feedback, including:
  - The load applied on the child's leg (from load sensors).
  - The support percentage provided by the robot.
  - Whether the child completed the session.
- Save session data for future evaluation, such as:
  - Last performed activity
  - Session duration
  - Progress tracking

How the Interface Communicates with the System

1. The therapist inputs data through the laptop GUI.

• Example: selects Phase 2 + "Static Exercise" + 10-minute duration.

2. The GUI program formats this information, usually into a simple string or data packet.

3. Since the Raspberry Pi 4 runs directly on the same laptop (connected via USB/Ethernet/Wi-Fi), the data is sent directly to the Raspberry Pi without the need for a Bluetooth module.

• Python scripts running on the Raspberry Pi handle incoming data.

4. The Raspberry Pi 4 receives the data and :
• Interprets the command.
• Activates specific servo motors through the PCA9685 PWM driver depending on the selected mode.
• Adjusts movement speed, angles, and duration.
5. In real-time, the Raspberry Pi also reads sensor data from:
• Load cells (to measure weight or force on legs).
• EMG sensors (to detect muscle activation indicating movement intent).
• Rotary encoders (for precise motor position feedback).
6. These sensor readings are sent back to the GUI on the laptop over the same direct connection.
7. The GUI updates instantly to show:
• Current force readings.
• Support ratio.
• Session progress.
8. The Raspberry Pi also controls an interactive Hamtysan display and USB mini speaker to provide AI-generated motivational feedback via ChatGPT and text-to-speech. This feature is only used to encourage the child and does not control the robotic hardware.
9. At the end of the session, the GUI saves the data locally in a database file for progress tracking and future reference.