CVPRO COMPETITION KIT

TECHNICAL DOCUMENT

🔧 What is the CVPro Competition Kit?

The **CVPro Competition Kit** is a **4-wheel robot DIY kit** tailored for high-performance in AI and robotics competitions, especially aligned with the **WRO (World Robot Olympiad) Future Engineers category**. With modular hardware and AI-driven capabilities, this kit allows participants to build, program, and train their robot to tackle real-world tasks using vision and sensor data.

🛠️Hardware Provided:

This robust kit includes all essential hardware components to build and operate a fully functional AI-powered robot:

1. **CVPro Controller (ESP32-based)**
   A high-performance microcontroller that supports Wi-Fi and Bluetooth—ideal for real-time AI applications and sensor integration.
2. **Servo Motor with Ackerman Steering System**
   Ensures precise directional control, emulating the steering mechanism of real-world vehicles for efficient turns.
3. **DC Motor with Optical Encoder**
   Powers the robot's forward and backward motion while providing feedback on speed and distance for accurate navigation.
4. **Color Sensor**
   Detects coloured paths or zones on the ground—useful for line-following or zone-based tasks.
5. **Six Ultrasonic Sensors**
   - **Front (3)**: Detect obstacles ahead with increased accuracy.
   - **Rear (1)**: Enables reverse movement safety.
   - **Left (1) and Right (1)**: For lateral obstacle detection and wall-following tasks.
6. **Push Button**
   A simple interface to start the robot's programmed tasks—ideal for quick competition launches.
7. **Programmable RGB LED**
   Visual feedback through color coding—useful for status indication, error alerts, and task confirmation.

8. **Mobile Phone Holder**
   Designed to securely hold a mobile device for image processing, remote control, or augmented vision applications.

💻 Software Support:

The CVPro kit comes with full software support for both programming and competition-specific applications:

1. **Arduino IDE with Required Libraries:**
   - Easy-to-use programming environment.
   - Compatible libraries for sensor interfacing, motor control, and AI features like object detection.
2. **Android Mobile App for WRO Round 2:**
   - An Android mobile application designed specifically for WRO Round 2.
   - It features an object detection model capable of identifying red and green boxes.
   - Additionally, it supports custom object detection and image classification capabilities.

🔧 Technical Notes:

🔌 About the CVPro Controller:

1. **Charging and Discharging:**
   Charging is enabled **only when the robot is powered OFF** to ensure safety and optimal charging efficiency.
2. **Charging Indicators:**
   - A **green LED** indicates that the robot is **fully charged** (only visible when the bot is OFF).
   - A **red LED** indicates that the battery is **currently charging**.
   - Ensure the robot can run for approximately **50 minutes** after a full charge.
3. **Battery Specification:**
   Equipped with a **3.7V, 3200mAh Li-ion single-cell battery** for extended operation time.
4. **Power Module:**
   Supplies stable power to the **controller board**, **sensors**, and **motors**.
5. **Motor Driver:**
   Dedicated module for controlling the **DC and servo motors** with precision.

6. **ESP32 Microcontroller:**
   A powerful dual-core microcontroller that supports both **wired (via Type-C port)** and **wireless communication (Wi-Fi and Bluetooth)**. It handles sensor inputs, motor control, and AI-based algorithms.

About the CVPro Controller:

1. The servo in the kit is **calibrated with a centre angle of 100 degrees**.
2. When adjusting the servo, ensure the angle remains within a **±20° range** from the centre—**between 80° and 120°**.
   Operating beyond these limits may cause **mechanical strain or damage to the servo**.

Pinouts for CVPro Controller:

Kindly refer to the pins provided in the above table for programming firmware.

Function	Port Type	Port No	GPIO Pins
Motor (Battery Operated Motor)	USB 3.0	1	32,33
F1US & F2US (Front 1 & Front 2 Ultrasonic Sensors)	USB 3.0	2	F1-16,14 and F2-25,26
BUS (Back Ultrasonic Sensor)	USB 3.0	3	17,19
SM (Servo Motor)	USB 3.0	4	27
RUS & LUS (Right & Left Ultrasonic Sensors)	USB 3.0	5	Right-2,23 and Left-5,18
FUS (Front Ultrasonic Sensor)	USB 2.0	6	12,4
CS (Color Sensor)	USB 2.0	7	22,21
RGB LED	-	-	15
NSLEEP For Motor	-	-	13
Battery Voltage Reading	-	-	39
DPDT Push Button	-	-	34
Optical Encoder	-	-	36

🔌 Connecting Mobile Phone to CVPro Controller:

To establish a proper connection between your mobile phone and the CVPro Controller, follow these steps:

1. **Type-C OTG Cable:**
    - Plug the **female end** of the **Type-C OTG cable** into the **male end** of the **Type-C USB cable**.
2. **Connect to Mobile Phone:**
    - Insert the **Type-C end** of the OTG cable into your **mobile phone's Type-C port**.
3. **Connect to CVPro Controller:**
    - Plug the **Type-C end** of the USB cable into the **Type-C port on the CVPro controller PCB**.
4. **Grant Permissions:**
    - If the connection is successful, your **mobile phone will prompt for permission** to access the connected USB device.
    - Tap **“Allow”** to enable communication between the phone and the CVPro Controller.

🔧 Working Procedure:

Step 1: Download the Bot Test Codes

Download the Bot Test Codes folder from the official GitHub repository:
🔗 https://github.com/robotixdevteam/CVPro_Competition_Kit/tree/main

---

Step 2: Download and Install Arduino IDE

Follow the instructions in this link: https://support.arduino.cc/hc/en-us/articles/360019833020-Download-and-install-Arduino-IDE.

For Windows:

- Visit https://www.arduino.cc/en/software
- Click on **"Windows Win 10 and newer, 64 bits"** to download the installer.
- Install the Arduino IDE and launch the application.
- In Arduino IDE, go to **Tools > Upload Speed**, and set the speed to **115200**.

For macOS:

- Visit https://www.arduino.cc/en/software
- Click on **"macOS Intel, 10.15 'Catalina' or newer, 64 bits"**.
- Download and install the application.
- Open Arduino IDE, go to **Tools > Upload Speed**, and set the speed to **115200**.

For Ubuntu/Linux:

- Visit https://www.arduino.cc/en/software
- Click on **"Linux AppImage 64 bits (X86-64)"** to download the file.
- Locate the downloaded. AppImage file in your file manager.
- Make it executable:
    - Right-click the file and choose **Properties**.
    - Under **Permissions**, check **"Allow executing file as program"**.
- Double-click the file to launch the Arduino IDE.

Step 3: Install Required Libraries in Arduino IDE

1. Open **Arduino IDE**.
2. Go to **Boards Manager**:
    - Search for ESP32, select **version 2.0.17**, and click **Install**.
3. Go to **Library Manager** and install the following libraries:
    - **NewPing** – Version 1.9.7
    - **FastLED** – Version 3.7.0
    - **ESP32Servo** – Version 1.2.1
    - **Adafruit TCS34725** – Version 1.4.4

Step 4: Upload Bot Test Codes

- Open the downloaded **Bot Test Codes** folder.
- Inside, you will find individual test programs for various motors and sensors.
- Open each test code in the Arduino IDE and **upload them one by one** to the CVPro bot for verification and testing.

🔧 Troubleshooting Process:

🪟 CP210x Driver Issues:

If your device is not recognized in the Arduino IDE or you're unable to upload code, the issue may be due to a missing or improperly installed CP210x USB to UART driver. Follow the steps below based on your operating system:

🔧 For Windows:

- Visit the Silicon Labs official driver download page:
  🔗 https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads
- Click on "CP210x Universal Windows Driver" to download the installer.
- Run the installer and follow the on-screen instructions.
- After installation, restart your computer.
- Reconnect your device and check if it appears under Tools > Port in Arduino IDE.

🍏 For macOS:

- Go to the same official driver page:
  🔗 https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers?tab=downloads
- Download the "CP210x VCP Mac OSX Driver".
- Open the installer and complete the installation.
- After installation, go to System Preferences > Security & Privacy, and allow the system extension if prompted.
- Restart your Mac and then reconnect the device.
- Check if the port /dev/cu.SLAB_USBtoUART appears in the Arduino IDE under Tools > Port.