ACC - ARC

Created by:

RISHIYENDRA PRASAAD A/L VIAYAKUMARAN
(prasaadrishiyendra@gmail.com)
GitHub : Rishiyendra

MUHAMMAD NUR AIMAN BIN RAMZAN
(DEM JKM PSAS)
(burungunta0@gmail.com)
GitHub : Mnz05

Verified by:

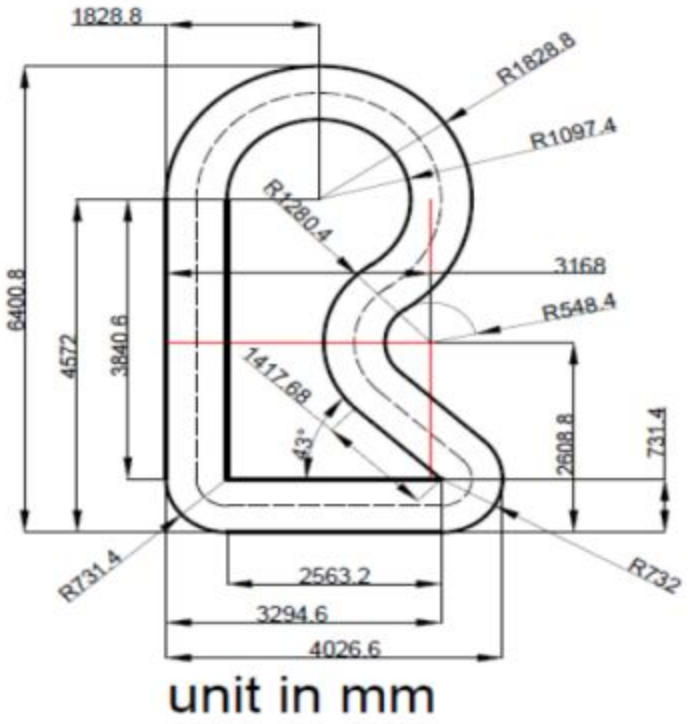
NUR AKHYAR BIN NORDIN
(JKE PSA)
(ayozzet@gmail.com)
GitHub : ayozzet

Robot And Arena Specification

Sub Kategori	ACC ARC
Maksimum Berat Robot	Tiada had
Dimensi Robot:	Kereta ARC adalah menggunakan kuasa bateri sahaja: Tidak melebihi dari 7.5” (190mm) wheelbase, gandar ke gandar. Panjang kereta : 300mm - 550mm Lebar kereta : 150mm - 350mm Tinggi kereta : tidak melebihi 450mm Bateri hendaklah diikat rapi dengan menggunakan velkro atau pengikat jenis lain agar tidak mudah tertanggal semasa pertandingan.
Kawalan Robot:	Automatik sepenuhnya (Fully Autonomous)

Jenis Motor Yang dibenarkan	Bebas
Bekalan Kuasa	Bateri sahaja dan jenis bateri adalah bebas.
Sistem Kemudi	Sistem kemudi Ackerman dengan satu atau dua sistem paksi dibenarkan.
Jenis Pacuan	4WD dan 2WD dibenarkan dengan 4 roda sahaja
Ringkasan Pertandingan:	Setiap kereta ARC yang bertanding perlu menamatkan perlumbaan secara ‘Time-Attack” (pusingan awal) dan side by side (pusingan kalah mati) dari garisan permulaan ke garisan penamat dan mengikut peraturan pertandingan yang ditetapkan. Kereta yang mencatatkan mata tertinggi dan masa terpantas akan dinobatkan sebagai pemenang pertandingan.

Saiz gelanggang	8m*10m (Anggaran)
Warna dan Dimensi:	Gelanggang : Material banting Garisan : Putih (50mm lebar) ; Kuning (50mm lebar) putus-putus Lebar Trek : Minimum 700mm Kon : Tinggi minimum : 18 cm ; Panjang dan Lebar maksimum : 14 cm ; Warna : Oren
Material Gelanggang	Banting yang di lapik papan



unit in mm

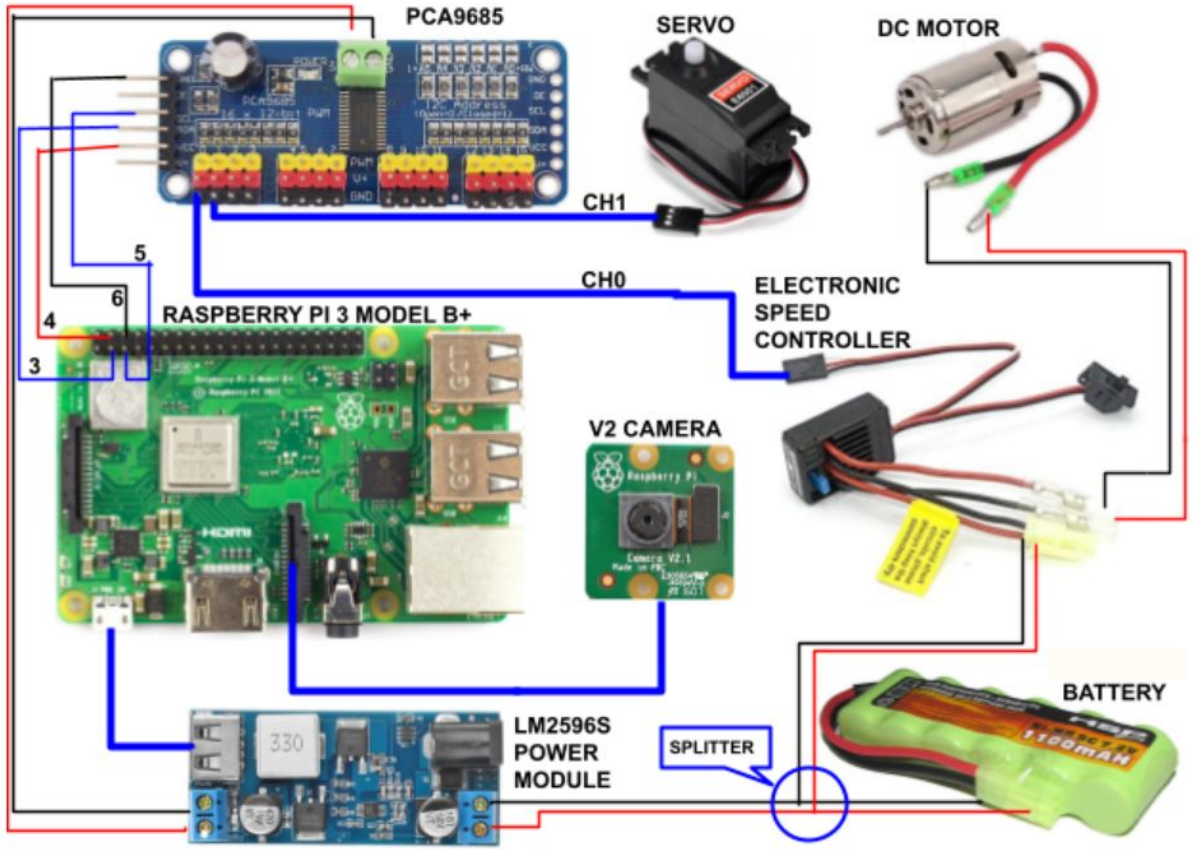
7 buah kon oren 12" akan diletakkan secara rawak di tepi trek.

Hardware

Below is the list of hardware components required for this project:

- Raspberry Pi 4 (Pi4)
- Camera Module OV5647 or Camera Module IMX219
- HSP 94186 RC Car Chassis
- 11.1V 2500mAh 20C Li-Po Battery with T-plug Connector (3S LiPo)
- Adafruit PCA9685 16-Channel Servo Driver
- Brushed Waterproof ESC (Model: 03058)
- LM2596S DC-DC Buck Converter Module

Full Connection (Stand-alone)



Software

The following software applications should be installed:

Link to Download file:

1. Visual Code
2. Raspberry Pi Imager
3. DonkeyCar

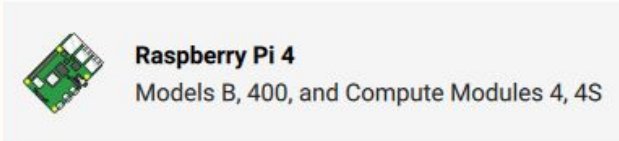
Visual Code

- After installing Visual Studio Code, open it and navigate to the Extensions tab. Then, search for and install the 'Remote - SSH' extension

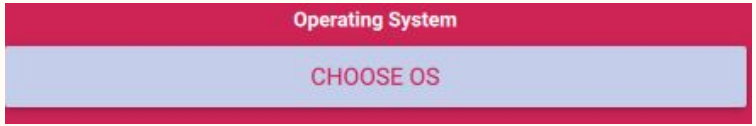
Raspberry Pi Imager

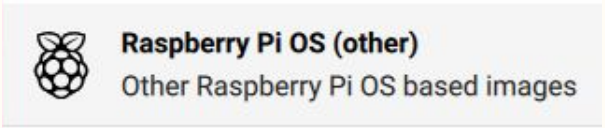
- After installing **Raspberry Pi Imager**, follow these steps:
- Open **Raspberry Pi Imager**.

- Click **'Choose Device'** and select **'Raspberry Pi 4'**.



- Click **'Choose OS'** and go to **'Raspberry Pi OS (Other)'**.

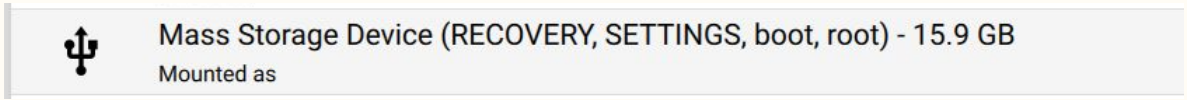




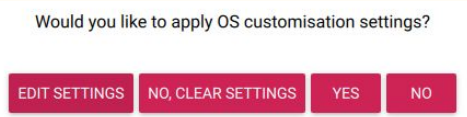
- Select **'Raspberry Pi OS (Legacy, 64-bit) Full (Debian Bullseye Version)'**.

Raspberry Pi Imager

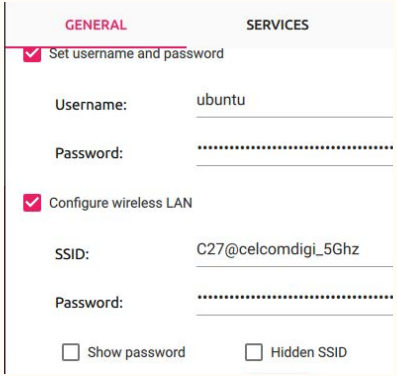
- Next, click 'Choose Storage' and select the memory card you want to write the OS image to.

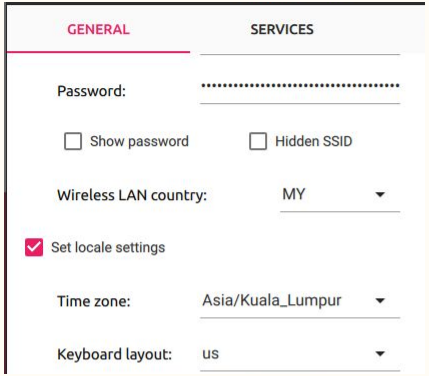


Raspberry Pi Imager

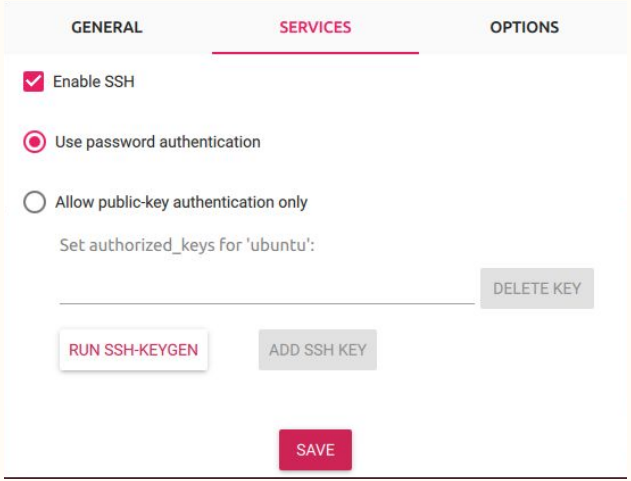


- Click 'Next' at the bottom right corner, then select 'Edit Settings'.
  - Under the **General** tab:
    - Enable **'Set username and password'**,
    - Enable **'Configure wireless LAN'**,
    - Enable **'Set locale settings'**.
    - Then, enter your **username**, **password**, **Wi-Fi credentials**, **timezone**, and **keyboard layout**.

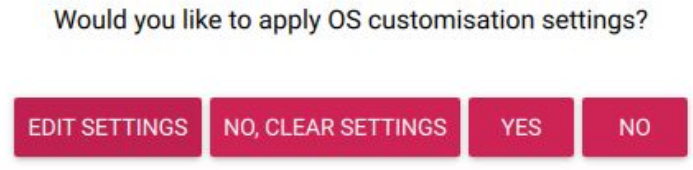




Raspberry Pi Imager



- Switch to the **Services** tab:
  - Enable **SSH**,
  - Select **'Use password authentication'**.
- Finally, click **'Save'** to apply the settings.
- After saving the settings, click **'Yes'** to begin burning the OS to the memory card.
- If prompted to format the memory card, confirm by selecting **'Yes'**.



Donkeycar

- After opening the link, follow these steps to install software on your host PC:
- Select **LINUX**.
- Install **virtualenv** on your pc:
  - `sudo apt install python3.8-venv`
- In your PC terminal, create a Python environment folder:
  - `mkdir <your folder name>`
  - `cd <your_folder_name>`
  - `python3 -m venv <your_environment_folder_name> --system-site-packages`
- After creating the Python environment folder, activate it:
  - `source <your_folder_name>/<your_environment_folder_name>/bin/activate`
- To check if the environment is active, you'll see the environment folder name next to your username in the terminal, indicating that it's successfully activated.


```
rishiyendra@rishi:~$ cd test/
rishiyendra@rishi:~/test$
```

```
rishiyendra@rishi:~$ mkdir test
```

```
rishiyendra@rishi:~/test$ python3 -m venv test_env
rishiyendra@rishi:~/test$
```

```
rishiyendra@rishi:~/test$ source test_env/bin/activate
(test_env) rishiyendra@rishi:~/test$
```

Donkeycar

Install DonkeyCar on PC

- Upgrade pip and install required system dependencies:
  - `pip install --upgrade pip setuptools wheel`
- Install DonkeyCar Python Code For PC:
  - `mkdir ~/projects`
  - `cd ~/projects`
  - `git clone https://github.com/autorope/donkeycar`
  - `cd donkeycar`
  - `git checkout release_5_0`
  - `pip install -e .[pc]`
- Create your DonkeyCar project folder:
  - `donkey createcar --path ~/<your_folder_name>/mycar`
  - Replace <your_folder_name> with your DonkeyCar folder name.

Donkeycar

- After completing the software installation on the host PC, proceed to install the software on DonkeyCar (Raspberry Pi 4):
  - **Insert the memory card** with the burned OS into your **Raspberry Pi 4** and boot it up.
  - Once booted, open a terminal on the Raspberry Pi and run the following commands to update and upgrade the system:
    - sudo apt-get update --allow-releaseinfo-change
    - sudo apt-get upgrade
  - Launch the Raspberry Pi configuration utility:
    - sudo raspi-config
    - In the raspi-config menu:
      - Go to **Interfacing Options** and **enable I2C**.
      - Go to **Advanced Options** and select **Expand Filesystem** to make full use of your SD card storage.
  - After completing these steps, **reboot your Raspberry Pi** to apply the changes:
    - sudo reboot

Donkeycar

After rebooting your Raspberry Pi 4, follow these steps to set up the DonkeyCar environment:

- Open a terminal on your Raspberry Pi and create a Python virtual environment with system site packages:
  - `python3 -m venv env --system-site-packages`
- Add the environment activation command to your .bashrc file so it auto-activates on startup:
  - `echo "source ~/env/bin/activate" >> ~/.bashrc`
  - `source ~/.bashrc`
- Install the required system dependencies:
  - `sudo apt install libcap-dev libhdf5-dev libhdf5-serial-dev`
  - `pip install --upgrade setuptools wheel pip`

Donkeycar

Install DonkeyCar Python Code for Raspberry Pi

- Clone and install DonkeyCar:
  - `mkdir ~/projects`
  - `cd ~/projects`
  - `git clone https://github.com/autorope/donkeycar`
  - `cd donkeycar`
  - `git checkout release_5_0`
  - `pip install -e .[pi]`
- Verify TensorFlow installation:
  - `python -c "import tensorflow; print(tensorflow.__version__)"`
- Create your DonkeyCar project folder:
  - `donkey createcar --path ~/mycar`

Donkeycar APP and Calibration

Donkeycar APP

- **After rebooting your Raspberry Pi 4, follow these steps to connect via SSH and set up your DonkeyCar project:"**
  - On your Raspberry Pi 4, open a terminal and type:
    - `ifconfig`
  - Look for the **IP address under wlan0 (inet)** and write it down.
- On your **PC**, open a terminal and connect to the Raspberry Pi using SSH:
  - `ssh <your_pi_username>@<raspberry_pi_ip_address>`
- Press Enter. If successful, your terminal will now be connected to the Raspberry Pi.
- Install I2C tools and verify your **PCA9685 servo driver** connection:
  - `sudo apt-get install -y i2c-tools`
  - `sudo i2cdetect -y 1`
  - You should see a grid with a detected I2C address (usually 0x40) if the PCA9685 is connected properly.

Donkeycar Calibration

Steering Calibration Instructions:

- In your Raspberry Pi terminal (via SSH), run the steering calibration command:
  - `donkey calibrate --channel 0 --bus=`
  - Enter values in the range of **0 to 1500** to find the appropriate PWM values for **far left** and **far right** steering.
- Open Visual Studio Code and connect to your Raspberry Pi via **Remote - SSH**.
  - Open the mycar folder located in your home directory.
  - Inside the mycar folder, open the config.py file.
  - In config.py:
    - Go to **line 83** in config.py to find PWM_STEERING_THROTTLE, and **line 98** to find I2C_SERVO. Use the calibrated values to update the following fields accordingly:
      - STEERING_RIGHT_PWM
      - STEERING_LEFT_PWM

Donkeycar Calibration

Steering Calibration Instructions:

- **Testing & Fine-Tuning Steering:**
  - If the car **steers left without turning**, adjust STEERING_LEFT_PWM **closer to neutral**.
    - Example: If STEERING_LEFT_PWM = 460 and STEERING_RIGHT_PWM = 290, reduce the left value slightly, e.g., 458.
  - If the car **steers right without turning**, adjust STEERING_RIGHT_PWM **closer to neutral**.
    - Example: If STEERING_LEFT_PWM = 460 and STEERING_RIGHT_PWM = 290, increase the right value slightly, e.g., 292.

Donkeycar Calibration

Throttle Calibration Instructions

- In your Raspberry Pi terminal (via SSH), run the throttle calibration command:
  - `donkey calibrate --channel 1 --bus=`
- Enter values in the range of **0 to 1500** to identify the correct PWM values for **forward**, **center (stopped)**, and **reverse** throttle positions.
- Open **Visual Studio Code** and connect to your Raspberry Pi using **Remote - SSH**.
  - Navigate to the mycar folder in your home directory and open it.
  - Inside the mycar folder, open the config.py file.
  - In config.py:
    - Go to **line 83** to locate PWM_STEERING_THROTTLE. And Go to **line 98** to find the I2C_SERVO section.
    - Use your calibrated values to update the following fields:
      - THROTTLE_FORWARD_PWM
      - THROTTLE_STOPPED_PWM
      - THROTTLE_REVERSE_PWM

Train

Donkeycar Setting On Pc Host And Raspberry Pi4

This setting is for to change `config.py` file in both Pc Host and Raspberry Pi4

Configuration File Edits:

- **Line 371**:
  - Adjust the batch_size according to your GPU capability or training preference. A value of batch_size = 64 is generally sufficient to lower GPU load during training.
- **Line 566**:
  - Set AUTO_RECORD_ON_THROTTLE = False (change from True).
- **Line 645**:
  - Set AUTO_CREATE_NEW_TUB = True (change from False).
- **Line 386**:
  - Set AUTO_CREATE_NEW_TUB = False (change from True).

Donkeycar Setting On Pc Host And Raspberry Pi4

Configuration File Edits:

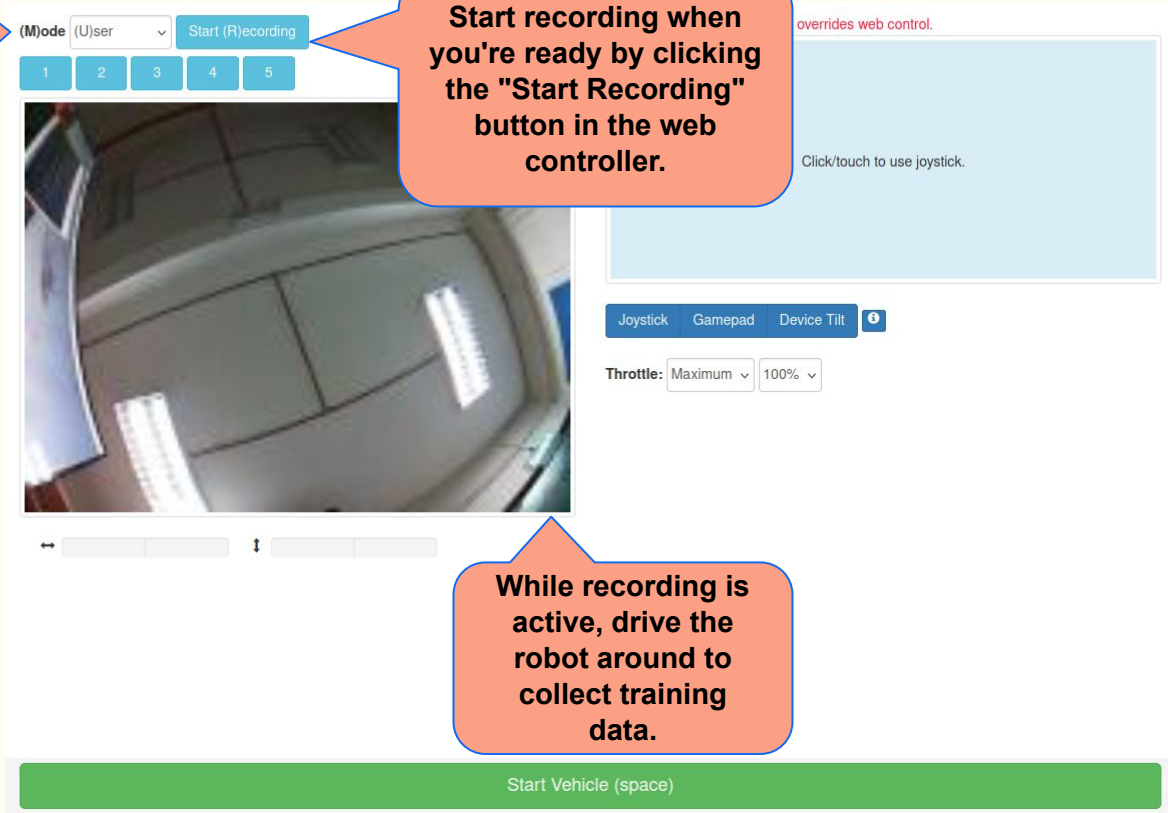
- **Line 567**:
  - Update CONTROLLER_TYPE = 'xbox' depending on the type of remote control you are using:
    - **For an RC remote**, change 'xbox' to 'pigpio_rc' in the CONTROLLER_TYPE setting.
      - Then, in **lines 593 and 594**, update the GPIO pin numbers to match your **throttle** and **steering** signal pins.
      - Then, in **lines 596 to 599**, update the **throttle** and **steering** values based on the results from the **Throttle and Steering Calibration** section.
      - **For RC remote users:** Refer to the Installation Guide for detailed instructions on installing and configuring the **GPIO daemon (pigpio)**, which is required for pigpio_rc to work properly.
    - For a **PS3 or PS4 controller**, change 'xbox' to 'ps3' or 'ps4' accordingly.

Donkeycar WebView

Starting the DonkeyCar Web Controller

- In your Raspberry Pi terminal, navigate to your DonkeyCar project folder:
  - `cd ~/mycar/`
- To start the web server with a **joystick or RC controller**, run:
  - `python manage.py drive --js`
- If you're using **keyboard control only**, run:
  - `python manage.py drive`
- On your PC or phone, open a web browser and go to:
  - `<your_pi_ip_address>:8888`
  - Replace <your_pi_ip_address> with the actual IP of your Raspberry Pi.

Make sure to set the mode to `user` in the web controller before driving the car.



Start recording when you're ready by clicking the "Start Recording" button in the web controller.

While recording is active, drive the robot around to collect training data.

- **Each time you start recording, a new tub (data folder) will be created automatically.**
- Aim to collect a **minimum of 10,000 images** to ensure sufficient training data.

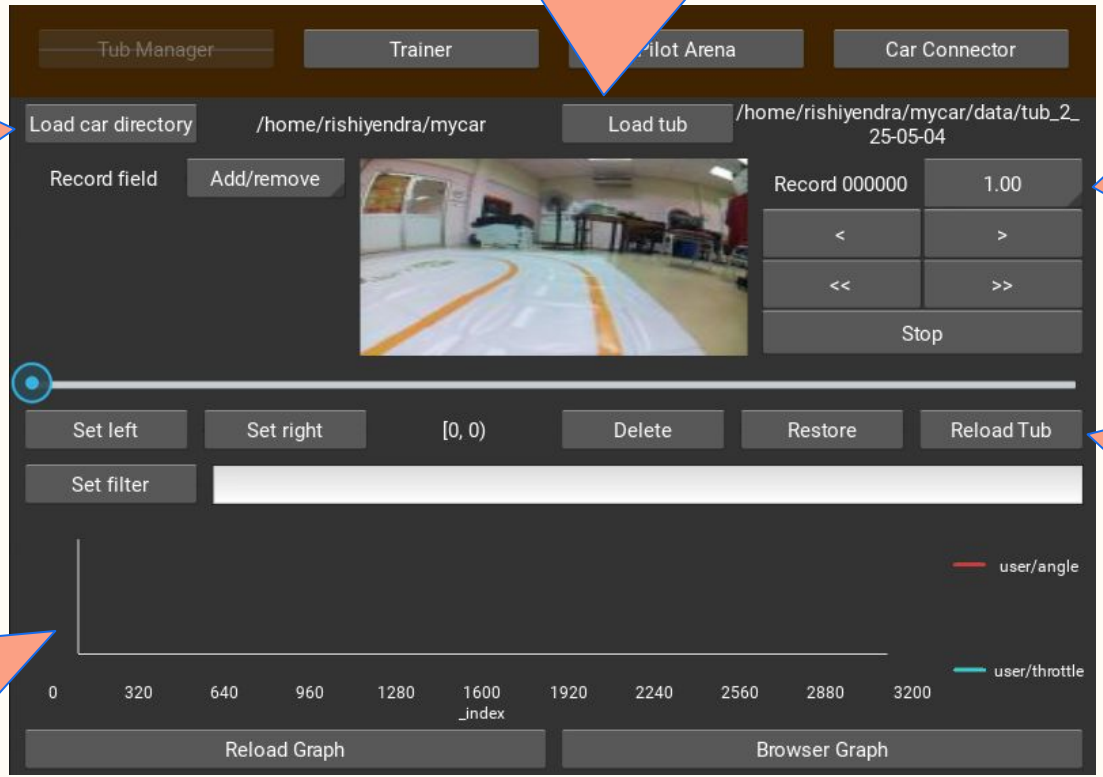
Train

After you finish recording data:

- Use **FileZilla** (refer to the Installation Guide for detailed setup instructions) to transfer the **tub folder** from your **Raspberry Pi** to your **PC** at the following location:
  - mycar/data/
- Open a terminal on your PC, activate your Python environment, and navigate to the project directory:
  - `cd mycar/`
  - Run the following command for a **one-time setup** of xclip (used for clipboard access):
    - `sudo apt-get install xclip`
  - `donkey ui`
- In the **Donkey UI**,
  - click on **"Tub Manager"** to view and manage your recorded data.

Load the tub folder from your Raspberry Pi 4 here

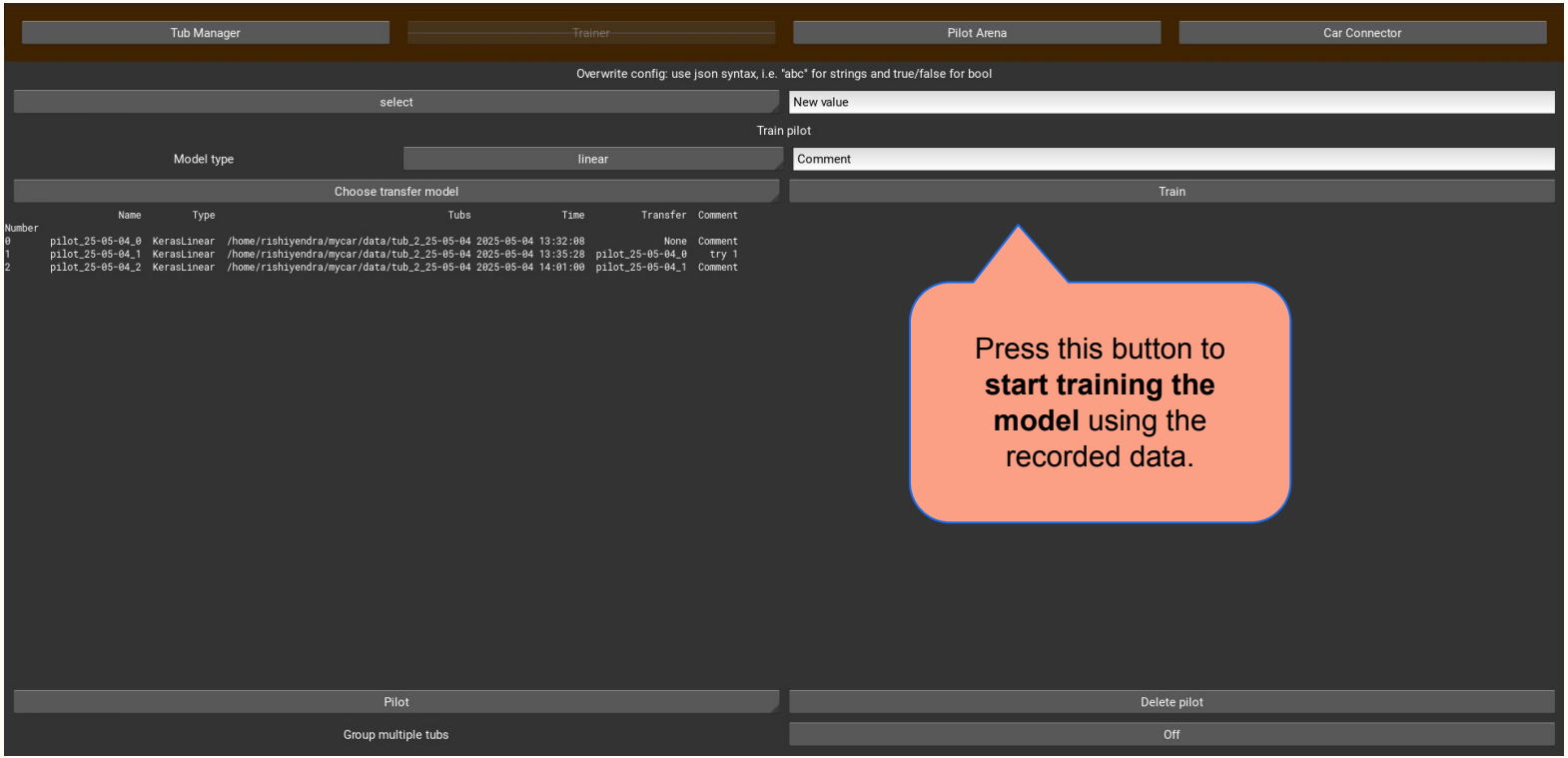
Load the mycar/ folder



Player Area: This section allows you to **play**, **pause**, **rewind**, or **fast-forward** the recorded driving footage for review and training adjustments.

This section displays a **graph of throttle and steering values**, allowing you to visually inspect how they change over time during the recording.

Edit Area: This section allows you to **delete individual frames** you don't want, or **restore previously deleted frames** as needed.



Press this button to **start training the model** using the recorded data.

This section is used to **add notes or comments** for the user, helping to document observations or mark specific data model.

Drive

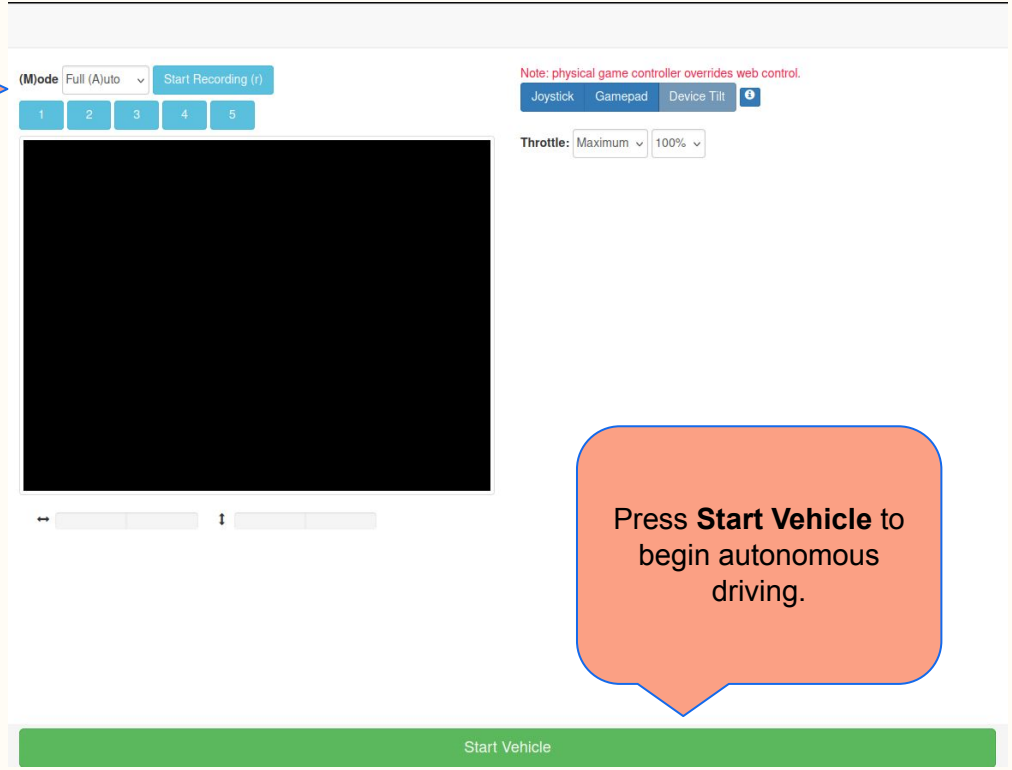
Drive

After Training Your Model:

- **Use FileZilla** to transfer the trained model folder from PC to your Raspberry Pi 4 located at:
  - ~/mycar/models/
- On your **Raspberry Pi 4 terminal**, run:
  - `python manage.py drive --model ~/mycar/models/mypilot.tflite`
- Open your web browser and go to your DonkeyCar web server (`<pi_ip>:8888`). In the web interface:
  - Set the **mode to "full auto"**.
  - Click **"Start Vehicle"** at the bottom of the page to begin autonomous driving.

Drive

Set this to **Full Auto** to enable autonomous driving.



Press **Start Vehicle** to begin autonomous driving.

Questions?

Reference

- Donkey Car Website

Installation Guide

Note for RC Remote Users:

- To use pigpio_rc, you need to start the **pigpio daemon**:
  - Install the system daemon
    - `sudo apt-get update`
    - `sudo apt-get install pigpio`
  - Install python support (with donkey environment activated)
    - `pip install pigpio`
  - Start the daemon
    - `sudo systemctl start pigpiod`
  - Enable the daemon on startup
    - `sudo systemctl enable pigpiod`

Installation Guide

Note for FileZilla Installation (Linux)

- **Download** the FileZilla tar.xz package from the official website.
- Open your terminal and run the following commands:
  - `cd ~/Downloads`
  - `tar -xf FileZilla_3.69.1_x86_64-linux-gnu.tar.xz`
  - `cd FileZilla3/`
  - `cd bin/`
  - `./filezilla`