AI & SUSTAINABILITY IN VET EDUCATION

Name of the Author/s: GR Team

Title of WP: WP3 – AI-SAVES@ProjectHub

Activity Reference: 3.2 – Project template for AI-SAVES@ProjectHub

Logo:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Table of Contents

Table of Contents **2**

1. Project Title 3

2. Project Category 3

3. Project Description 3

4. Keywords 4

5. Target Audience 4

6. Project Objectives 4

7. Project Timeline 4

8. Resources Provided 5

9. Tools and Technologies Used 5

10. Step-by-Step Implementation Plan 5

11. Evaluation Criteria 9

12. Languages Available 9

13. Contributors 9

14. Feedback and Continuous Improvement 10

Submission Notes **10**

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

AI-SAVES@PROJECTHUB

---

1. Project Title:

Remote Controlled Car Using Raspberry Pi

---

2. Project Category:

AI / Robotics / IoT – STEM Maker Project

---

3. Project Description:

This project demonstrates how to build a remote-controlled robotic car using a Raspberry Pi. The vehicle can be controlled wirelessly through a web interface, smartphone, or computer. The system uses a Raspberry Pi microcomputer connected to motor drivers and DC motors, allowing users to control the direction and movement of the car in real time.
This project introduces learners to robotics, embedded systems, wireless control, and basic programming. It can also be extended with computer vision and AI algorithms, transforming the vehicle into a self-driving robot prototype.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

4. Keywords:

robotics
Raspberry Pi
remote control car
IoT robotics
embedded systems
STEM education
AI robotics
smart vehicles

---

5. Target Audience:

Vocational education and training (VET) students
Robotics clubs and STEM laboratories
Beginner programmers interested in robotics

---

6. Project Objectives:

Learn the basics of robotic vehicle design

Understand how Raspberry Pi controls motors

Develop basic Python programming skills

Learn about wireless communication and remote control systems

Explore how robotics platforms can be expanded with AI and computer vision

---

7. Project Timeline:

Estimated duration: 8–10 hours

Phase	Time
Introduction to robotics and Raspberry Pi	1 hour
Hardware assembly	2 hours
Electronics wiring	2 hours
Programming	2 hours
Testing and calibration	1–2 hours

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

8. Resources Provided:

Hardware

Raspberry Pi (3 / 4 / Zero)

Motor driver module (L298N or similar)

2–4 DC motors

Robot car chassis

Wheels

Battery pack

Jumper wires

---

9. Tools and Technologies Used:

Software

Web interface or remote control application

Python

Raspberry Pi OS

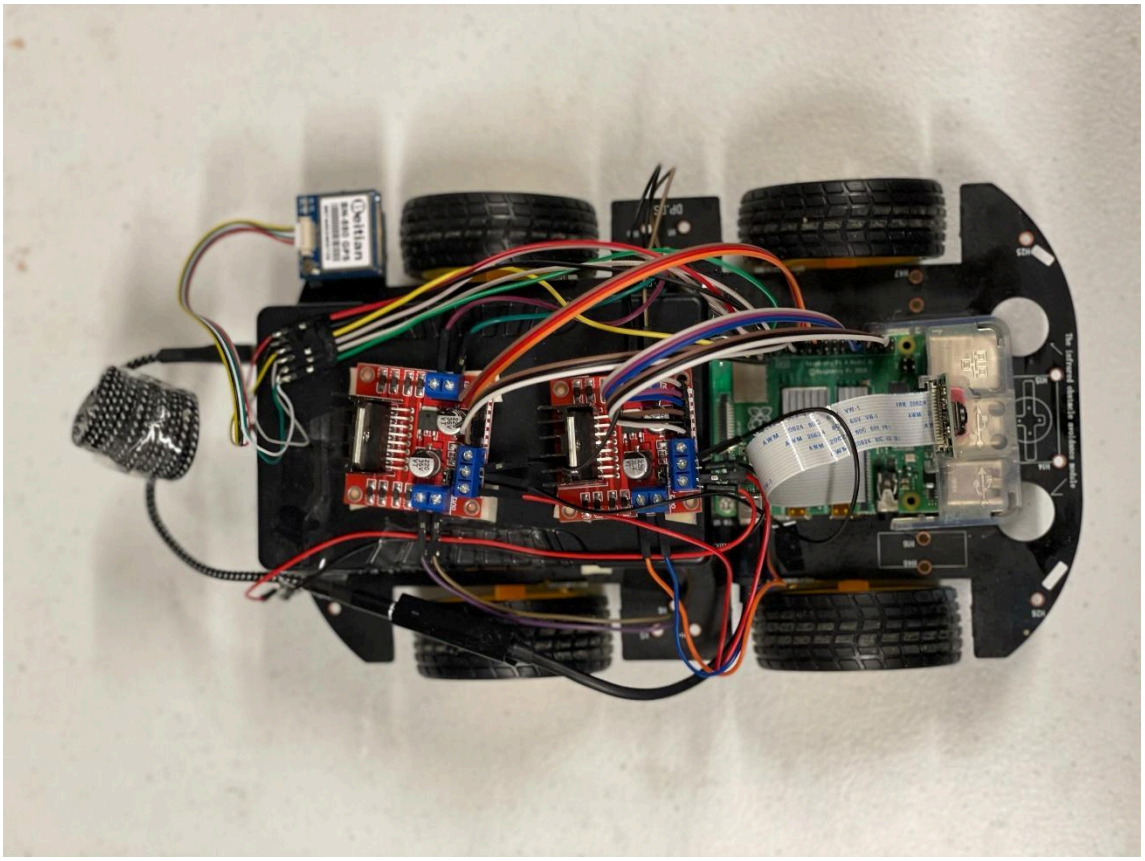
GPIO libraries

---

10. Step-by-Step Implementation Plan:

Co-funded by the European Union

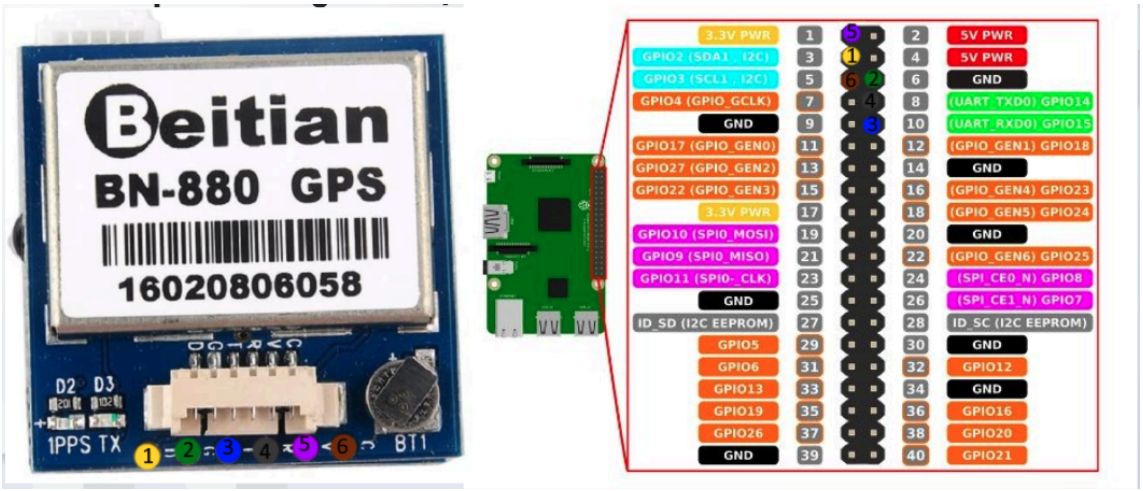
This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Step 1: Install OS Images
Install OS Images
Materials: computer (Mac, Ubuntu 18.04, or Windows), microSD card, SD adapter

Basically, the Raspberry Pi runs the Raspbian OS, so we have to install the OS images on a microSD card. To do so, you need an SD adapter so a microSD card can be inserted into a regular laptop/computer's SD port (through which the images will be installed). Then, the microSD card is inserted into the Pi.

Follow the instructions here: https://www.raspberrypi.org/documentation/install...



Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Step 2: Setting Up the Pi
Setting Up the Pi
Setting Up the Pi
We need to be able to work on the Raspberry Pi. Raspberry Pi is a single-board computer, so it can directly connect to a monitor. Be sure to connect the HDMI (using an adapter), USB (you'll likely need a USB hub), and power ports. You'll also need a keyboard and mouse (which you'll connect using USB).

You can refer to the image to become familiar with the Raspberry Pi desktop. You'll see a manila folder icon, which is the equivalent of a Finder (Mac) or Windows Explorer. Essentially, it a visual of your files, downloads, directories, etc. For our purposes, it's how you'll manage your code. You'll also see a black screen icon, which is the terminal window. This is where you will write your code.

Open up a terminal window (there should be an icon in the upper left section of the screen).

Run sudo raspi-config to get a menu like the one above
Enable the P6 serial port
Enable the P5 I2C
If you need more guidance: https://www.raspberrypi.org/documentation/configu...
From here, you have two options:

1. Continue using a monitor

Every time you want to work on the Pi, just do the same thing as above
Connect the peripherals and work directly on the pi
2. Run the pi headless (using ssh)

https://www.raspberrypi.org/documentation/remote-a...
Essentially, you can log in to the pi remotely from another machine to avoid having to connect peripherals
To do this, you MUST enable ssh on the pi by following the instructions at the link above
Create a repository for the project. Then, in the terminal window, make a new directory for the project on the pi. You can edit your files within the terminal window (vi) if more comfortable, or using a GUI if less comfortable.

If you are unfamiliar with the command line, refer to these links:

https://product.hubspot.com/blog/git-and-github-tu...

https://www.codecademy.com/learn/learn-the-command...

or do a Google search.

Step 3: Prereading
Prereading
If you want to learn more about how all this stuff works (which I recommend), I would do some reading (there are plenty of resources and videos online!). But, here's a brief overview of what to know if you are just looking to get started.

The Raspberry Pi board - a single-board computer; know where these things are (see image

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

above):
MicroSD card slot
Mini HDMI port
Micro USB port
Micropower port & LED that indicates that the Pi is being powered
Camera connector
Processor (and heat sink if you have one)
40-pin header & GPIO pins
Know the SDA/SCL pins, MOSI/MISO pins, and TX/RX pins
Communication protocols
UART: universal asynchronous reception and transmission
I2C: inter-integrated-circuit
SPI: serial peripheral interface
Step 4: Assembling the Chassis
Assembling the Chassis
You should have a chassis. This is the base frame of the car. Assemble the chassis the way the kit specifies.

If you bought the one I linked:

To start, break off the eight rectangles along the dotted indents (be careful!).

Then, using the nuts and bolts included in the kit, and a screwdriver, assemble the pieces, motors, and wheels until they resemble the image above (which is of the bottom of the car, the top should be clear):

Note: be sure that the orientation of the motors is correct!

Now, take jumper wires and solder the wire

Once you have completed the assembly, the base of the car is assembled. Now, do a quick check to ensure that all of the motors are working properly by directly powering the motors. Check to see that the white rods turn. If you switch the black and red wires, the rod should turn in the opposite direction. Check all four wheels.

Step 5: Motor Controllers
Motor Controllers
Each one motor controller can control two wheels. Depending on whether you bought one or two drivers, there a difference in the way the vehicle turns: whereas controlling all four wheels enables the car to turn about its center, only controlling the rear wheels will cause the car to turn in an arch.

For now, we will focus on using one controller. If you bought two, repeat these steps for the second driver. Note: you can run them in parallel by making one driver power the other.

Here is the reading to understand how these modules work: https://components101.com/modules/l293n-motor-dri... I'd recommend reading the specifications.

Use the jumper wires provided to get the motor controllers working. Again, using a breadboard first might be more efficient. Here is where you have options: I recommend using

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

a 12V power source for the driver(s) and a 5V power source for the pi. But, you can use the pi to power the motor driver(s), too. There is a jumper that differentiates between 12V and 5V -- you should read the specs of the module when doing the wiring.

Step 6: Camera Module
Camera Module
If you use a pi board with limited RAM, then it will only support the basic Pi Camera module. If you purchased a Pi 4 with more RAM (around 8 GB), you can try and implement the OpenCV camera. There are resources online to do so.

For now, though, you can work with the camera module, which will suffice.

Readings:

https://gist.github.com/willprice/abe456f5f74aa95...
https://www.hackster.io/ruchir1674/video-streamin...
https://gist.github.com/willprice/abe456f5f74aa95...
For the code, you can run the following command:

git clone https://github.com/miguelgrinberg/flask-video-str...

NOTE: Credit goes to Miguel Grinberg for the camera code.

Step 7: GPS and Compass
GPS and Compass
Use the following resources to get the GPS and compass working. The wiring is also in the diagram above.

Readings for GPS:

https://maker.pro/raspberry-pi/tutorial/how-to-use...

https://maker.pro/raspberry-pi/tutorial/how-to-use...

The specific GPS module that I linked also has an embedded compass.

Sample code to read compass data: https://github.com/ozzmaker/BerryIMU.git

Another example NMEA format (parse string for coordinate data):
https://github.com/ozzmaker/BerryIMU.git

Again, credit for the GPS code goes to ozzmaker.

Step 8: Displaying Data on the Web Interface Using Flask
Displaying Data on the Web Interface Using Flask
This allows you to stream the GPS and compass data, as well as video, to a web server.

We are going to use Flask, a web framework that helps you build a web application.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

11. Evaluation Criteria:

Criterion	Description
Hardware assembly	Robot car assembled correctly
Electronics	Correct wiring of motors and driver
Programming	Car responds correctly to commands
Functionality	Vehicle moves reliably
Creativity	Additional improvements or features

---

12. Languages Available:

English

---

13. Contributors:

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

17. Feedback and Continuous Improvement:

AI obstacle avoidance

autonomous driving algorithms

IoT remote monitoring

mobile app control

Submission Notes:

Please ensure that all required documents and resources are attached to your submission. The AI-SAVES@ProjectHub review team will evaluate the submission based on the provided information. Once approved, the project will be published on the hub, and partners will be notified.

Co-funded by the European Union

This project has been funded with support from the European Commission. This communication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.