MU5EBB12

- E-WheelChAIr

E-WheelChAIr

Smart Embedded System for Electric Wheelchairs

For all files see the github link : https://github.com/RenaudFookIng/E-WheelChAIr

License ROS2
Image not found or type unknown Image not found or type unknown

---

Origin of the project :

- Occupational therapist and Medical Doctor couldn’t prescribe a Power Wheelchair if the user driving it could injure themselves or others.
- In consequence, the user with massive motor impairment and cognitive or visual impairment don’t have power wheelchair and they depend of the other to move in their home and the city.

Description

E-WheelChAIr is an open-source project designed to enhance autonomy and accessibility for electric wheelchairs. It integrates:

- A **ROS2-based architecture** for real-time control and sensor fusion.
- **3D-printed mechanical parts** (sensor mounts, joystick adapters).
- **Modular design** for easy customization and collaboration with partners like Sorbonne University.

□□ Visual Overview

Wheelchair Views

Front View Right View Rear View Left View
Image not found or type unknown Image not found or type unknown Image not found or type unknown Image not found or type unknown

Project Structure

Note: All ROS launch files are centralized in the `e_wheelchair_launch` package for easier management.

```
E-WheelChAIr/
├── hardware
│   ├── 3d_models
│   │   ├── BackRest_Fixation/
│   │   │   ├── Fixation_BackRest.iges
│   │   │   ├── Fixation_backrest_part1.stl
│   │   │   └── Fixation_backrest_part2.stl
│   │   ├── Camera_Module/
│   │   │   ├── Camera_module.iges
│   │   │   ├── Fixation_camera.stl
│   │   │   ├── Fixation_camera_up.stl
│   │   │   └── Support_camera.stl
│   │   ├── ServoControl_Joystick/
│   │   │   ├── BasePart.stl
│   │   │   ├── FourchetteAxeX.stl
│   │   │   ├── FourchetteAxeY.stl
│   │   │   ├── Servo_controller_Joystick.iges
│   │   │   ├── Support_servo_joystick.stl
│   │   │   └── Visuel_Joystick_servomoteur_support.png
│   │   └── Ultrasonic_Module/
│   │
│   ├── arduino/
│   │   ├── ewheelchair_controller/
│   │   │   ├── ewheelchair_controller.ino
│   │   │   └── lecture_arduino.py
│   │   ├── ARDUINO_GUIDE.md
│   │   ├── FILES_SUMMARY.md
│   │   ├── README.md
│   │   └── test_arduino.py
│   │
│   └── Image_processing/
│       ├── cmd.txt
```

```
|       ├── ROS.txt
|       ├── vision_config_gpu.py
|       ├── vision_config_mac.py
|       ├── vision_system_gpu.py
|       ├── vision_system_mac.py
|       ├── yolo11n.pt
|       └── yolov8n.pt
|
├── resources
|   ├── docs
|   └── utils
|
├── src
|   ├── arduino_bridge/
|   |   ├── arduino_bridge
|   |   |   ├── __init__.py
|   |   |   └── arduino_bridge_node.py
|   |   ├── resource
|   |   |   └── arduino_bridge
|   |   ├── test
|   |   |   ├── test_copyright.py
|   |   |   ├── test_flake8.py
|   |   |   └── test_pep257.py
|   |   ├── package.xml
|   |   ├── setup.cfg
|   |   └── setup.py
|   ├── custom_msgs/
|   |   ├── msg/
|   |   |   ├── EmergencyData.msg
|   |   |   ├── Joystick.msg
|   |   |   ├── ObstacleDetection.msg
|   |   |   ├── ServoCommand.msg
|   |   |   ├── UltrasonicArray.msg
|   |   |   ├── VisionObstacle.msg
|   |   |   └── WyesIntent.msg
|   |   ├── CMakeLists.txt
|   |   ├── package.xml
|   |   └── setup.py
|   ├── e_wheelchair_launch/
|   |   ├── e_wheelchair_launch
```

```
|   |   |   ├── __init__.py
|   |   |   └── launch/
|   |   ├── resource/
|   |   |   └── e_wheelchair_launch
|   |   ├── package.xml
|   |   ├── setup.cfg
|   |   └── setup.py
|   ├── master_node/
|   |   ├── config/
|   |   |   ├── master_config.yaml
|   |   |   └── servo_config.yaml
|   |   ├── master_node/
|   |   |   ├── __init__.py
|   |   |   ├── master_node.py
|   |   |   └── master_node_bis.py
|   |   ├── package.xml
|   |   ├── setup.cfg
|   |   └── setup.py
|   ├── pc_vision_bridge/
|   |   ├── pc_vision_bridge
|   |   |   ├── __init__.py
|   |   |   └── pc_vision_bridge_node.py
|   |   ├── resource
|   |   |   └── pc_vision_bridge
|   |   ├── test
|   |   |   ├── test_copyright.py
|   |   |   ├── test_flake8.py
|   |   |   └── test_pep257.py
|   |   ├── package.xml
|   |   ├── setup.cfg
|   |   └── setup.py
|   └── wyes_teleop/
|       ├── resource
|       |   └── wyes_teleop
|       ├── test
|       |   ├── test_copyright.py
|       |   ├── test_flake8.py
|       |   └── test_pep257.py
|       ├── wyes_teleop
|       |   ├── __init__.py
```

```
|   |   └── wyes_teleop_node.py
|   ├── package.xml
|   ├── setup.cfg
|   └── setup.py
|
├── .gitignore
├── CABLAGE_SCHEMA.md
├── E-WheelChAIr.mp4
├── LICENSE
├── parametrage_raspberry_hotspot.txt
├── README.md
├── TESTING_GUIDE.md
└── yolov8n.pt
```

Prerequisites

Hardware

- Electric wheelchair with servo-controlled joystick (using Miuzei MZ996 servos).
- Sensors: Intel Realsense (depth camera), HC-SR04 (ultrasonic), Arduino (I/O interface).
- Joystick or alternative control input device.

Software

- ROS2 Jazzy.
- colcon (build tool for ROS2).
- OnShape (for 3D modeling).
- Git LFS (for versioning large 3D files).
- Python 3.9+ with pyserial and pyyaml packages.

Installation and Setup On your computer

1. Clone the Repository

```
git clone https://github.com/RenaudFookIng/E-WheelChAIr.git
cd E-WheelChAIr
```

2. Build the ROS Workspace

```
colcon build --symlink-install
source install/setup.bash
```

3. Launch the System

```
ros2 launch e_wheelchair_launch ewheelchair_all.launch.py
```

Installation and Setup On your Raspberry PI

1. Cloner le dépôt en mode "sparse-checkout"

```
git clone --filter=blob:none --no-checkout https://github.com/RenaudFookIng/E-WheelChAIr.git
cd E-WheelChAIr
```

2. Active le mode sparse-checkout et exclut le dossier 'hardware'

```
git sparse-checkout init --cone
git sparse-checkout set --no-cone /* !/hardware/
```

3. Extrait les fichiers (sauf 'hardware')

```
git checkout main
```

Build the ROS Workspace

```
colcon build --symlink-install
source install/setup.bash
```

ROS Launch Files Centralization

All ROS launch files have been centralized in the `e_wheelchair_launch` package to provide:

- **Simplified Management**: Single location for all launch configurations
- **Easier Maintenance**: No duplication across packages
- **Consistent Structure**: Uniform approach to system launching
- **Better Organization**: Clear separation between nodes and launch configurations

Available Launch Files

Launch File	Description	Components Launched
`ewheelchair_all.launch.py`	Complete system with all sensors and processing	6 nodes
`ewheelchair_joystick_servo.launch.py`	Control servo with joystick and Ultrasonic sensor	2 node
`ewheelchair_arduino_vision.launch.py`	Control servo with joystick and Ultrasonic sensor and Wide-angle camera processing	3 node
`ewheelchair_wyes_teleop.launch.py`	Control servo with joystick, Ultrasonic sensor and Wyes Glasses	3 node

ROS Packages

Package	Description
`arduino_bridge`	Receives data from Arduino sensors
`custom_msgs`	Custom ROS message definitions

Package	Description
`e_wheelchair_launch`	Launch files for the complete system
`master_node`	Main control node for servo-controlled joystick
`pc_vision_bridge`	Bridge between PC vision system and ROS
`wyes_teleop`	Keyboard-based teleoperation interface

Pour se connecter au Hotspot WiFi :

- Nom du Hotspot : E-Wheelchair
- Mot de passe : EWheelChair2025

Pour se connecter à la plateforme : Nom de la plateforme : ewheelchair2

```
ssh ewheelchair2@192.168.4.1
```

mdp : ewheelchair

3D Models

- **Location**: `3d_models/`
- **Formats**: STL and STEP files for 3D printing and simulation.
- **Tools**: Designed in OnShape.
- **Note**: Use `git lfs pull` after cloning to retrieve 3D files.

Recent Changes

Version 0.2.0 (Current)

- **Major Architecture Change**: Replaced Sabertooth motor controller with Miuzei MZ996 servo-based joystick control
- **New Package**: `arduino_bridge` for Arduino-based servo management
- **Updated**: `master_node` now publishes joystick commands instead of motor commands
- **Updated**: `visualization` package removed motor speed plotting
- **Improved**: Safety features with neutral position on emergency stop

Version 0.1.0

- Initial release with Sabertooth motor controller
- Basic sensor integration (ultrasonic, camera)
- ROS2 Jazzy compatibility

Collaboration

- **University of Milan**: Partner for accessibility validation.
- **Contributions**: Pull requests and issues are welcome!
- **License**: Apache 2.0 (see LICENSE).

Contact

- **Maintainer**: Renaud JANET (renaud.janet@etu.sorbonne-universite.fr or renaud.janet@proton.me)
- **Github link** : https://github.com/RenaudFookIng/E-WheelChAIr