AGILEHAND

Installation Manual: WP6.2 – BUILD: Agile, Flexible and Rapid Reconfigurable SUITE

AGILEHAND has received the funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101092043.

Document Information

GRANT AGREEMENT NUMBER 101092043	ACRONYM	AGILEHAND	
FULL TITLE	Smart grading, handling, and packaging solutions for soft and deformable products in agile and reconfigurable lines		
START DATE	01-01-2023	DURATION	36 months
PROJECT URL	https://agilehand.eu/		
WORK PACKAGE	WP6 – BUILD: Agile, Flexible and Rapid Reconfigurable SUITE		
LEAD BENEFICIARY	UPM		
RESPONSIBLE AUTHOR	Mazzuto Giovanni, Ortenzi Marco, Croci Stefano, Iacovanelli Matteo		
CONTRIBUTIONS FROM	UPM		
TARGET AUDIENCE	Solution user(s)		
CONTENT	Installation manual		
ABSTRACT	The scope of this document is to guide the user(s) through the installation instructions of the solutions developed within WP6.2 – BUILD: Agile, Flexible and Rapid Reconfigurable SUITE of the AGILEHAND project.		

Disclaimer

TABLE OF CONTENTS

1. Introduction

1.1. Purpose of the Document

This Installation Manual provides detailed instructions for setting up, installing, and maintaining the **AGILEHAND** Digital twin system hardware and software components. It is intended to guide technical personnel through the complete installation process to ensure the system operates reliably and effectively. This document complements the User Manual by focusing on the technical setup, configuration, and troubleshooting of the system environment.

1.2. Who Should Use This Manual

This manual is intended for technical staff responsible for installing, configuring, and maintaining the **AGILEHAND** Digital twin system. It is designed for system integrators, IT personnel, and engineers who have experience with hardware setup, software installation, and troubleshooting in industrial or laboratory environments. Operators and end-users should refer to the User Manual for daily operation instructions.

1.3. System Overview

The **AGILEHAND** Data-Driven Digital Twin system is an AI-powered solution for creating real-time digital replicas of production and logistics processes in the food manufacturing sector. It connects to company databases to leverage historical and real-time data for demand forecasting, resource allocation, and process optimization.

The system integrates sensors, GPU-equipped processing, and AI-based simulation tools to analyze scenarios, detect anomalies, and support reconfiguration to maintain optimal production performance.

This manual covers hardware installation, software setup, and system initialization, ensuring reliable data acquisition, accurate simulations, and seamless Digital Twin operation.

2. Prerequisites

2.1. Minimum Hardware Requirements

This section lists all hardware and accessories required to set up the **AGILEHAND** WP6 solution:

- **Processor** Intel(R) Core(TM) i7 10gen
- **Installed RAM:** 16 GB
- **System Type:** 64-bit Operating System, x64-based processor, IN10

2.2. Software Requirements

- **Docker Desktop (Windows, Mac, or Linux).**
- **Python.**
- **Node.js (for frontend deployment).**
- **Npm (for frontend deployment, included in Node.js).**

2.3. External Dependencies

No dependence on external applications.

3. Installation

3.1. Environment preparation

Ensure that all dependencies, including Docker, Python, and npm, are installed. Clone the repository from the official GitLab project and configure the backend and frontend environments as needed.

3.2. Step-by-step Installation process

- **Manual Installation:** Requires configuring backend and frontend, installing dependencies, and launching services manually.
- **Docker Installation:** Uses a *docker-compose.yml* file to deploy the application.

Manual Installation:

Make sure you have Python 3.12, Node.js, and npm installed. You can verify the installation by running the following commands:

- **Check version:**

```
python -V
node -v
npm -v
```

- **Navigate to the "./subsystems/backend" subdirectory of the project:**

```
cd <main directory>/subsystems/backend
```

Install the required Python libraries:

```
pip install -r requirements.txt
```

Start Python

```
python main.py
```

- **Navigate to the "./subsystems/frontend" subdirectory of the project:**

```
cd <main directory>/subsystems/frontend
```

Install the React libraries:

```
npm install
```

Start the web server:

```
npm run dev
```

The user interface will be accessible here: http//localhost:3000

Docker installation:

Make sure you have Docker and Docker Compose installed. You can verify the installation by running the following commands:

```
docker --version
```

Check Docker Compose Version

```
docker-compose --version
```

Steps to Install and Start the Containers

- **Navigate to the "orchestration" subdirectory of the project (where *`docker-compose.yaml`* and *`.env`* are located):**

```
cd <main directory>/orchestration/docker/
```

- **Modify URLs and Ports:**

If you need to change the backend URL and the ports for backend and frontend, you can modify the content of the .env files:

  - **./orchestration/docker/.env (PYTHON_APP_PORT, REACT_APP_PORT)**
  - **./subsystems/frontend/.env (REACT_APP_BACKEND_URL if you change the backend server)**

- **Build and Start the Containers:**

  To build the images and start the containers (frontend and backend), run:

```
docker-compose -f docker-compose.yaml -p lc_dis up
```

  This command performs the following actions:

  - **--build: Builds the images for the services defined in their respective `Dockerfile`s (for both frontend and backend).**
  - **docker-compose up: Starts the containers based on the built images.**

- **Interact with the Containers:**

  To monitor the logs and check for any errors or issues, run:

```
docker-compose logs
```

  If you want to access the shell of one of the containers, run:

```
docker exec -it <container_name> /bin/bash
```

  For example, to access the backend container:

```
docker exec -it <backend_container_name> /bin/bash
```

  To stop the containers, run:

```
docker-compose down
```

4. Troubleshooting Setup Issues

This section addresses common problems users may encounter during the installation or initial system setup.

Issue	PossibleCause	Suggested Action
Dependencies not installing	Outdated pip, incompatible OS or Python version	Upgrade pip (pip install --upgrade pip); verify compatibility in requirements
Script crashes at launch	Incorrect script or missing files	Ensure you're running the correct script and that all models and configs are present

5. Frequently Asked Questions (FAQs)

Q1: Do I need to train or retrain the AI model before using the system?
A: No. All models are pre-trained for the supported use cases. No manual training or annotation is required by the user.

Q2: Can I use the system in other manufacturing process?
A: No. The current version supports only these specified use cases. Extending the system to other products would require additional data collection, annotation, and AI model development.

Q3: Is an internet connection required to operate the system?
A: No. Once installed the local, the system runs entirely offline on the local machine.

Q4: How do I properly stop the Digital Twin system during operation?
A: To stop the Digital Twin system, press **Ctrl + C** in the command line window where the digital Twin script is running. This safely terminates the process. Avoid closing the terminal abruptly to prevent possible data loss or corrupted logs.