# Software Engineering for Cyberphysical Devices: Design Sheet

Pex Jan-Mathijs, Thys Ferre and Vandenberk Seppe

## System Requirements

The system needs to be capable of moving around an industrial facility to survey the area and measure various data elements. This means that the device should be able to move under user control as well as being directed to specific points within a room. The device doesn't require the ability to carry more than its own weight. Power for the device should come from the robot itself, necessitating a battery-operated solution without the need for cables. The company intends to control the robot through a web browser, meaning its operational range is constrained by network coverage.

The industrial facility may feature multiple floor types, including the possibility of wet and dusty floors, as the robot's primary purpose is to detect errors. Although the rooms won't have stairs, the potential for ramps exists.

Ultimately, the robot should be capable of performing inspection, repair, and maintenance tasks.

## Mechanical Design

- Tracks are more versatile and stable then wheels. They also provide more traction across multiple floor types. Possible addition of weight won't be much of a problem, because of the weight distribution and the higher payload capacity compared to regular wheels. Finally, wheeled robots may have more maneuverbility in tight spaces, tracks can still navigate efficiently enough through industrial environments.
- The idea is to build a low profile bot, with a telescopic arm, so it fits better in to tight spaces.

## Electrical Design

- **Raspberry Pi**: The Raspberry Pi serves as the brain of the robot, controlling its behavior and components. It processes sensor data, executes algorithms, and sends data over the cloud to our interfacing software.
- **Sensors**: Our robot is equipped with an array of sensors for perception and environmental awareness. These sensors include:
  - **Distance Awareness**: Our system will use a Lidar sensor for distance perception. This in combination with a GPS sensor, can help determine the robots location.
  - **Inertial Measurement Unit (IMU)**: Provides data on orientation, acceleration, and angular velocity for navigation and motion control.
  - **Camera(s)**: Enables visual perception and object recognition for tasks such as navigation and object manipulation.
  - **Environmental Sensors**: Measure parameters such as temperature, humidity, and atmospheric pressure for environmental monitoring and adaptation.
  - **Safety Sensors**: A safety sensure, such as a touch sensor can prevent the robot from doing unwanted behavior.
- **Actuators**:

  - **DC Motors**: Used for driving mechanisms.
  - **Servo Motors**: Provide precise control over angular position for tasks such as camera orientation.

- **Power Supply**: The power supply system based on batteries, provides electrical energy to the robot's components. It includes voltage regulators and power distribution circuits to ensure stable and reliable operation.

## Software Design

4. Software: o Python woooooo o Specify the control algorithms for motion planning, collision avoidance, and task execution. 1. D* lite 2. Lidar with threshold for collision avoidance, bumper with pressure sensor as an extra safety measure in case the lidar mechanism fails. o Include safety features and emergency stop procedures.

## Safety Systems

Our robot will have safety systems, so when it bumps into objects or enters restricted areas the robot won't be allowed to move any further, and will require to drive back in the distance it came from to avoid more damage. This will also require a re-calculation of the path it needs to take. The addition of a emergency stop in the web-interface and a physical button on the robot, will allow human interference to stop the robot at any time.

## Human-Machine Interaction (HMI)

6. Human-Machine Interface (HMI): o Describe the user interface for programming, monitoring, and troubleshooting. 1. Web interface, robot is keyboard controlled (with ps4 option ofc) , visible map on which to set goal points (points to navigate to) Also some other controls and sensor interface. Live video feed of camera via WebSockets

o Specify the level of automation and manual control options. 2. Given a point on the map the bot should be able to in a safe and pid controlled way find its way to the point without extensive stopping, preferrably in one smooth motion o Include error messages and diagnostics.

## Environmental Considerations

7. Environmental Considerations: o Address the operating temperature, humidity, and other environmental conditions. 1. No such sensors but i *walnoot* suggest we fake them in the web interface because it looks coool as fuck o Consider ingress protection (IP) ratings for protection against dust and water.

## Maintenance

o Outline maintenance procedures, including lubrication and inspection schedules. o Specify spare parts availability and ease of replacement. o Include documentation for troubleshooting and repairs.