# SamBot Specification

# Software System Requirements

# Introduction

A two-wheels robot needs to be designed in order to roll and detect obstacles and void in front of it.

This document lists all the system requirements.

Every requirement is composed of:

* One unique ID following this pattern: SYS\_XXXX (Four digits),
* A name, which is always a small introduction of the requirement,
* A text, describing what is this requirement for.

## SYS\_0100

Name: Autonomous bot

Text: The SamBot shall communicate without physical connection with a computer terminal.

## SYS\_0200

Name: Obstacle detection

Text: SamBot shall detect obstacles in front of it.

## SYS\_0300

Name: Void detection

Text: The SamBot shall detects void in front of it.

## SYS\_0400

Name: Debug mode

Text: The SamBot shall provide debug value to the computer on demand.

## SYS\_0500

Name: Manual Mode

Text: The SamBot shall follow instructions from the computer in Manual Mode

## SYS\_0600

Name: Autonomous Mode

Text: The SamBot shall roll and avoid obstacles by its own in Autonomous Mode

## SYS\_0700

Name: User interface

Text: The SamBot shall provide an easy-use interface through the terminal

## SYS\_0800

Name: Movement

Text: The SamBot shall be able to move around its environment

# Software Architectural Design Requirements

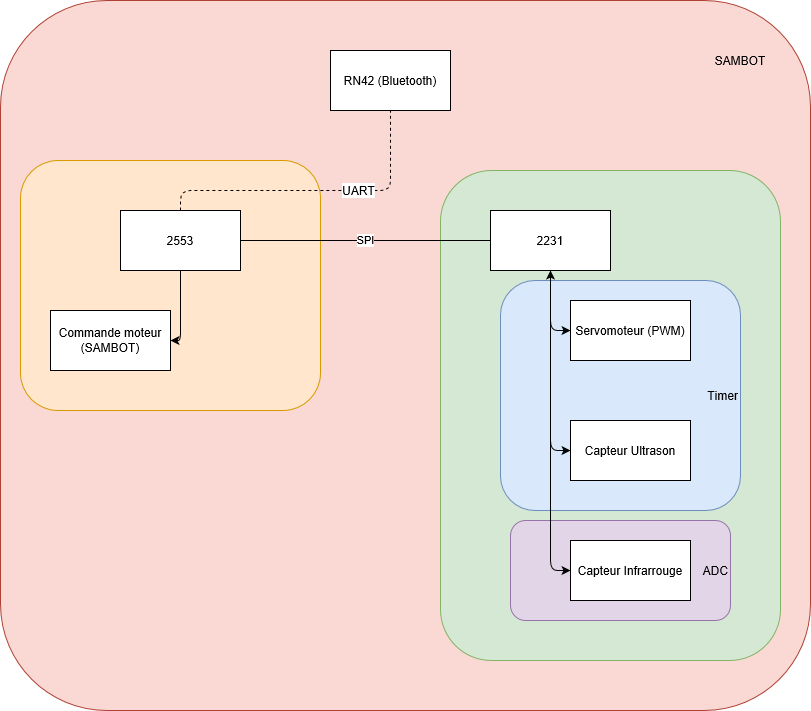
A two-wheels robot needs to be designed in order to roll and detect obstacles and void in front of it.

This document lists the requirements of the software and the hardware architectural design.

Every requirement is composed of:

* One unique ID following this pattern: ADR\_XXXX (Four digits)
* A name, which is always a small introduction of the requirement
* A text, describing what is the requirement for.

Here is the schematic of the system:



## ADR\_0100

Name: Roll

Text: The robot shall roll in a reasonable speed.

Covers: Sys\_0500, Sys\_0600

Module: Movement

## ADR\_0200

Name: Obstacle detection

Text: The robot shall detect obstacles in front of it with an ultrasound sensor.

Covers: Sys\_0200

Module: Obstacle\_Detection

## ADR\_0250

Name: Obstacle reaction

Text: When an obstacle has been detected, the robot shall stop and turn right before running forward.

Covers: Sys\_0200

Module: Obstacle\_Detection

## ADR\_0300

Name: ultrasound sensor

Text: The sensor will be placed on a servomoteur that will move within a 180° area

Covers: Sys\_0200

Module: Obstacle\_Detection

## ADR\_0400

Name: Void detection

Text: The robot shall detect a void in front of it with an infrared sensor.

Covers: Sys\_0300

Module: Void\_Detection

## ADR\_0450

Name: Void reaction

Text: When a void has been detected, the robot shall stop, run backward for a few inches and turn around before running forward.

Covers: Sys\_0300

Module: Void\_Detection

## ADR\_500

Name: infrared sensor

Text: the infrared sensor will be placed at the edge of the robot in order to detect the void early

Covers: Sys\_0300

Module: Void\_Detection

## ADR\_0600

Name: Manual mode

Text: The user shall be able to control the robot by sending commands to it via the MSP2553 with a UART connection (Bluetooth), from a terminal of a computer.

Covers: Sys\_0500

Module: ManualMode

## ADR\_0700

Name: Autonomous mode

Text: the robot shall be able to manage itself without the intervention of the user

Covers: Sys\_0600

Module:AutoMode

## ADR\_0800

Name: Communication

Text: The robot shall communicate with the user with a UART connection via Bluetooth

Covers: Sys\_0100

Module: Com

## ADR\_0900

Name: Structure

Text: the robot shall have two processors:

MSP2553 (master), which controls robot’s motors (direction). It receives the user’s commands via and sends the commands to the second processor

MSP2231 (slave), which receives the commands via SPI connection and handles the servomoteur and the two sensors (ultrasound and infrared)

Covers: all

Module:

## ADR\_1000

Name: Turn around

Text: The robot shall be able to turn left, turn right, run forward and backward  
Covers: Sys\_0800

Module: Movement

## ADR\_1100

Name: Help

Text: When the user enters a wrong command, the robot shall display the help assistance.

Covers: Sys\_0700

Module: Help

## ADR\_1200

Name: Debug mode

Text: the debug mode shall display the measures of all the sensors on real time when the user demands it.

Covers: Sys\_0400

Module: Debug

# Software Detailed Design Requirements

# Introduction

A two-wheels robot needs to be designed in order to roll and detect obstacles and void in front of it.

This document lists the requirementsof the software and hardware detailed design.

Every requirement is composed of:

* One unique ID following this pattern: DDR\_XXXXX (Five digits),
* A name, which is always a small introduction of the requirement,

A text, describing what is this requirement for.

## DDR\_00100

Name: Turning right

Text: If the user writes “d” in the terminal, The SamBot shall turn 90° right.

Cover: ADR\_0900

Function: Turn\_Right

## DDR\_00200

Name: Turning left

Text: If the user writes “q” in the terminal, The SamBot shall turn 90° left.

Cover: ADR\_0900

Function: Turn\_Left

## DDR\_00300

Name: Run forward

Text: If the user writes “z” in the terminal, The SamBot shall run forward.

Cover: ADR\_0900

Function: Run\_Forward

## DDR\_00400

Name: Run backward

Text: If the user writes “s” in the terminal, The SamBot shall run backward.

Cover: ADR\_0900

Function: Run\_Backward

## DDR\_00500

Name: Change to manual mode

Text: If the user writes “M” in the terminal, the state of the machine shall change to Manual Mode.

Covers: ADR\_0500

Function: Main

## DDR\_00600

Name: Change to Autonomous mode

Text: If the user writes “A” in the terminal, the state of the machine shall change in Autonomous Mode.

Covers: ADR\_0600

Function: Main

## DDR\_00700

Name: cycle Autonomous

Text: When the SamBot is in autonomous mode, it shall alternatively move forward a little and get sensors information.

Covers: ADR\_0600, ADR\_0400, ADR\_0200

Function: Auto\_drive

## DDR\_00800

Name: User input help

Text: If the Sambot receives any input not defined in the menu then it shall invoke UI\_help to guide the user

Covers: ADR\_1100

Function: interpreter

## DDR\_00900

Name: Debug mode

Text: If the user writes “B” in the terminal, it shall display the measures of all the sensors in real time.

Covers: ADR\_1200

Function: Debug

## DDR\_01000

Name: Manual Mode

Text: In manual mode, the MSP2231 shall continue to send the sensors’ information to the MSP2553, but the microcontroller shall ignore them.

Covers: ADR\_0600

Function: Manual\_mode

## DDR\_01100

Name: Sending information

Text: the MSP2231 shall always send the sensors’ information to the MSP2553

Covers: ADR\_0300, ADR\_0500

Function: Sensors

## DDR\_01200

Name: Void detection

Text: the MSP2553 shall command the robot to stop, run backward and turn around if the value received from the infrared sensor (sent by the MSP2231) indicates a void.

Covers: ADR\_0400, ADR\_0450

Function: Void\_detection

## DDR\_01300

Name: Infrared’s value conversion

Text: the MSP2231 shall convert the analogic value of the infrared sensor into a numeric value.

Covers: ADR\_0400, ADR\_0450

Function: Infrared\_Sensor

## DDR\_01400

Name: Ultrasound’s value conversion

Text: the MSP2231 shall convert the analogic value of the ultrasound sensor into a numeric value.

Covers: ADR\_0400, ADR\_0450

Function: Infrared\_Sensor

## DDR\_01500

Name: Obstacle detection

Text: the MSP2553 shall command the robot to stop and turn right if the value received from the ultrasound sensor (sent by the MSP2231) indicates an obstacle.

Covers: ADR\_0200, ADR\_0250

Function: Obstacle\_Detection