# SamBot Specification

# Software System Requirements

Introduction :

This document aims to list the global requirement for the SamBot software.

## 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.

## 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 obstacle by its own in Autonomous Mode

## SYS\_0700

Name: User interface

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

# Sofware Architectural Design Requirements

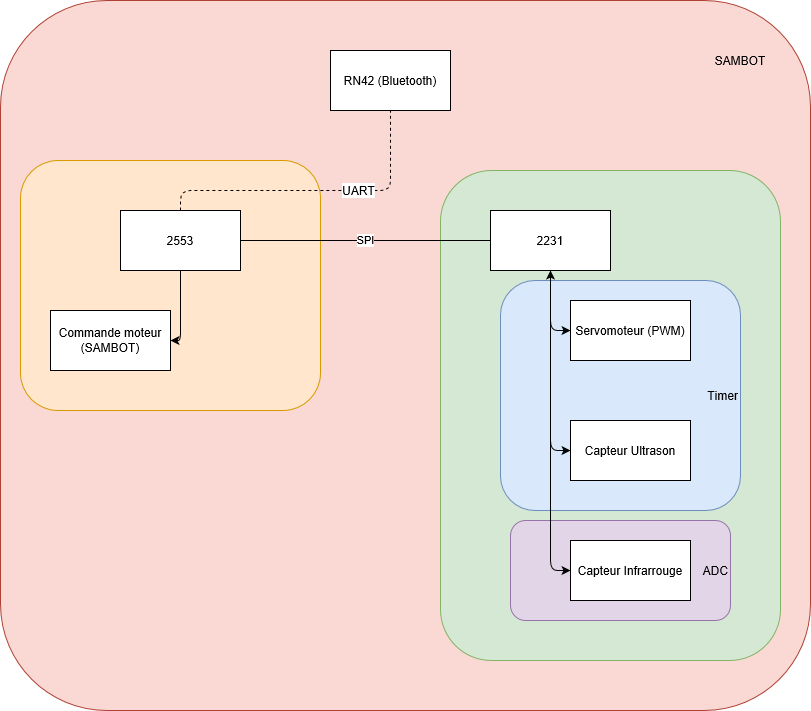
A two-wheels robot needs to be designed that roll and detects obstacles and void in front of it.

This document lists the requirements of the sofware 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.

This 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:

## ADR\_0250

Name: Obstacle reaction

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

Covers: Sys\_0200

Module:

## ADR\_0300

Name: ultrasound sensor

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

Covers: Sys\_0200

Module:

## ADR\_0400

Name: Void detection

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

Covers: Sys\_0300

Module:

## 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 going on.

Covers: Sys\_0300

Module:

## ADR\_0500

Name: Manual mode

Text: The user shall control the robot by sending commands to it with the MSP2553 via UART connection, from a terminal.

Covers: Sys\_0500

Module:

## ADR\_0600

Name: Autonomous mode

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

Covers: Sys\_0600

Module:

## ADR\_0700

Name: Communication

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

Covers: Sys\_0100

Module:

## ADR\_0800

Name: Structure

Text: the robot shall have two processors:

MSP2553, which controls robot’s motors (direction) and sends the commands to the second processor

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

Covers: all

Module:

## ADR\_0900

Name: Turn around

Text: In manual mode, the robot shall be able to turn left, turn right, run straight on and run backward regarding user instructions.  
Covers: Sys\_0500

Module:

## ADR\_1000

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:

## ADR\_1100

Name: Help

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

Covers: Sys\_0700

Module:

## 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:

# Software Detailed Design Requirements

## DDR\_00100

Name: Turning right

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

Cover: ADR\_0900

Function: turn\_left

## DDR\_00300

Name: Running

Text: If the user write “z” in the terminal, The SamBot shall go on.

Cover: ADR\_0900

Function: Run

## DDR\_00400

Name: Run backward

Text: If the user write “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 write “M” in the terminal, the state machine shall turn in Manuel Mode.

Covers: ADR\_0500

Function: Main

## DDR\_00600

Name: Change to Autonomous mode

Text: If the user write “A” in the terminal, the state machine shall turn in Autonomous Mode.

Covers: ADR\_0600

Function: Main

## DDR\_00700

Name: cycle Autonomous

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

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

Function: Autonomous\_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: void alert

Text: The MSP2231 shall send an alert message to the MSP2553 in case of void detection.

Covers: ADR\_0400

Function:

## DDR\_01000

Name: obstacle alert

Text: The MSP2231 shall send an alert message to the MSP2553 in case of obstacle detection.

Covers: ADR\_0250

Function:

## DDR\_01100

Name: Debug mode

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

Covers: ASR\_1200

Function:

## DDR\_

Name:

Text:

Covers:

Function:

## DDR\_

Name:

Text:

Covers:

Function:

## DDR\_

Name:

Text:

Covers:

Function: