# 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\_0600

Name: User interface

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

# Sofware Architectural Design Requirements

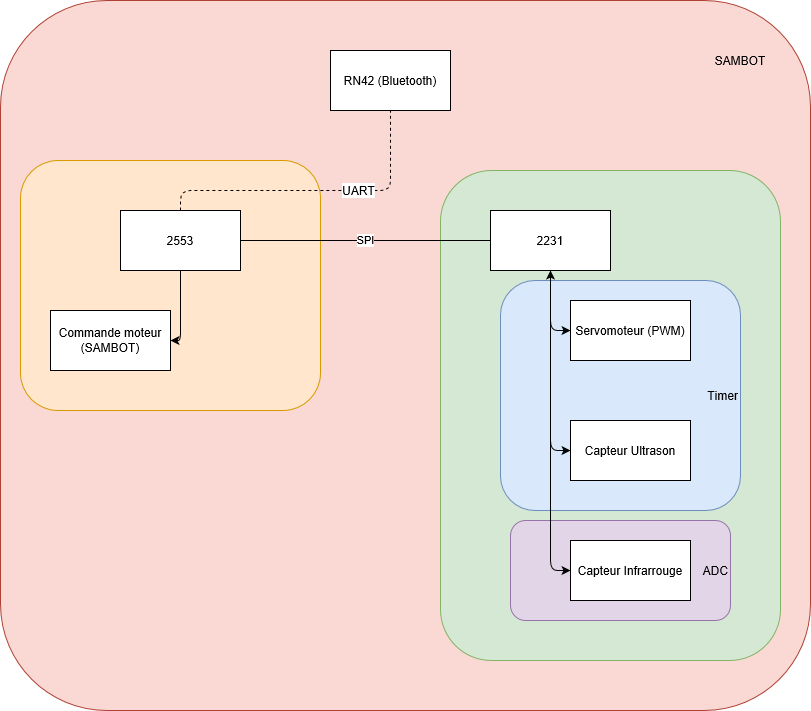
A two-wheels robot needs to be designed that roll and detects obstacles and void in front of it.

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 :

Module :

**ADR\_0200**

Name : Obstacle detection

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

Module :

**ADR\_0300**

Name : ultrasound sensor

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

**ADR\_0400**

Name : Void detection

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

Covers :

Module :

**ADR\_0500**

Name : Manual mode

Text : The user shall control the robot by sending commands to it with the MSP2553 via UART connection

Covers :

Module :

**ADR\_0600**

Name : Autonomous mode

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

Covers :

Module

**ADR\_0700**

Name : Communication

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

Covers :

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)

**ADR\_0900**

Name : Turn around

Text : the robot shall be able to turn around left or right and run backward.  
Covers :

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

Module :

# Software Detailed Design Requirements

## DDR\_00100

Name: Turning right

Text: The SamBot shall turn 90° when the

Cover:

Function: turn\_right

## DDR\_00200

Name: Change to manual mode

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

Covers:

Function: Main

## DDR\_00210

Name: Change to Autonomous mode

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

Covers:

Function: Main

## DDR\_00300

Name: cycle\_Autonomouse

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

Covers:

Function: Autonomous\_drive

## DDR\_00400

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:

Function:

## DDR\_00500

Name: void alert

Text: The MSP22311 shall send an alert message to the MSP2753 in case of void detection

Covers:

Function:

## DDR\_00600

Name:

Text:

Covers:

Function:

## DDR\_

Name:

Text:

Covers:

Function: