**SAM Bot – Software System Requirements**

Table of contents

[Introduction 2](#_Toc66957413)

[General diagram 2](#_Toc66957414)

[Software System Requirements 3](#_Toc66957415)

[Software Architectural Design Requirements 5](#_Toc66957416)

[Software Detailed Design Requirements 7](#_Toc66957417)

# Introduction

We are going to do a robot which can move alone (automatic mode) or manual mode with a wireless Bluetooth command.

(The communication will be done either wirelessly with a Bluetooth module and a digital remote control on smartphone, or with the PC’s control guest.)

Our robot will use an IR sensor positioned on a servomotor to analyse its environment and avoid obstacles.

# General diagram

Input (Control guest or BT)

UART

MSP430 2553

SPI

MSP430 2231

Analog

PWM

IR Sensor

Servomotor

# Software System Requirements

This document lists all the **system requirements**, for the **software only.**

Every requirement is composed of:

* One unique ID following this pattern: SSR\_XX,
* A name, which is always a small introduction of the requirement,
* A text, describing what is this requirement for.

**SSR\_01**

Name: MSP430 2553 & MSP430 2231 Communication

Text: These two microcontrollers shall communicate with SPI communication.

**SSR\_02**

Name: IR sensor

Text: The IR sensor shall detect obstacles.

**SSR\_03**

Name: Servomotor

Text: The servomotor shall scan the environment of the robot.

**SSR\_04**

Name: Move

Text: The robot shall move in all directions.

**SSR\_05**

Name: Avoid obstacles

Text: The robot shall avoid moving or static obstacles.

**SSR\_06**

Name: Two operating modes

Text: The robot shall have two modes of operation: automatic or manual. The automatic mode shall allow for autonomous travel while avoiding obstacles. The manual mode shall allow for user to control the robot with PC’s or smartphone’s commands.

**SSR\_07**

Name: Bluetooth module

Text: The robot shall have a Bluetooth module in order to receive user’s information remotely.

**SSR\_08**

Name: Control guest

Text: The robot shall have a control guest in order to receive user’s information. The control guest will mainly be used to test the code on the robot.

# Software Architectural Design Requirements

This document lists the **requirements** of the **software architectural design.**

Every requirement is composed of:

* One unique ID following this pattern: SADR\_XX,
* A name, which is always a small introduction of the requirement,
* A text, describing what is this requirement for.

**SADR\_01**

Name: Control guest

Text: The system shall allow the user to enter commands via the control guest.

Covers: SSR\_01 & SSR\_06 & SSR\_08

Module: MSP430 2553

**SADR\_02**

Name: Menu

Text: The system shall allow the user to look the menu to know the different possible commands.

Covers: SSR\_01 & SSR\_08

Module: MSP430 2553

**SADR\_03**

Name: When obstacle

Text: When an obstacle is detected, the system shall avoid it.

Covers: SSR\_02 & SSR\_05

Module: MSP430 2231

**SADR\_04**

Name: Move with commands sent by control guest

Text: When the user uses the control guest to send commands, the system shall execute these commands.

Covers: SSR\_04 & SSR\_06 & SSR\_08

Module: MSP430 2553

**SADR\_05**

Name: Move with commands sent by Bluetooth module

Text: When the user uses the application Bluetooth to send commands, the system shall execute these commands.

Covers: SSR\_04 & SSR\_06 & SSR\_07

Module: MSP430 2553

**SADR\_06**

Name: Choose the operating mode

Text: The user shall choose the operating mode of the system.

Covers: SSR\_06

Module: MSP430 2553

# Software Detailed Design Requirements

MSP430 2553 Functions:

**SDDR\_01**

Name: main

Text: Initialize the functions and calls the command interpreter.

Covers: SADR\_01

**SDDR\_02**

Name: init\_BOARD

Text: TI LaunchPAD card initialization function.

Covers: SADR\_01

**SDDR\_03**

Name: init\_UART

Text: UART initialization function.

Covers: SADR\_02 & SADR\_04 & SADR\_05

**SDDR\_04**

Name: envoi\_msg\_UART

Text: Emission function of a character string.

Covers: SADR\_02 & SADR\_04 & SADR\_05

**SDDR\_05**

Name: interpreteur

Text: User command interpretation function.

Covers: SADR\_02 & SADR\_04 & SADR\_05

**SDDR\_06**

Name: init\_USCI

Text: USCI initialization function FOR SPI ON UCB0.

Covers: SADR\_03

**SDDR\_07**

Name: Send\_char\_SPI

Text: Function to send a character on USCI in 3-wire SPI MASTER Mode.

Covers: SADR\_03

**SDDR\_08**

Name: init\_move

Text: Initialize the PWM signal to control the speed of the robot.

Covers: SADR\_04 & SADR\_05

**SDDR\_09**

Name: avancer

Text: set the duty cycle to move forward.

Covers: SADR\_04 & SADR\_05

**SDDR\_10**

Name: arretRobot

Text: set the duty cycle to stop the SamBot.

Covers: SADR\_04 & SADR\_05

**SDDR\_11**

Name: tournerDroite

Text: set the duty cycle to turn right.

Covers: SADR\_04 & SADR\_05

**SDDR\_12**

Name: tournerGauche

Text: set the duty cycle to turn left.

Covers: SADR\_04 & SADR\_05

**SDDR\_13**

Name: moveAuto

Text: Initialize IR sensor, use ADC library and turn right when there is an obstacle.

Covers: SADR\_06

**SDDR\_14**

Name: ADC\_init

Text: Initialize the ADC library.

Covers: SADR\_03

**SDDR\_15**

Name: ADC\_Demarrer\_conversion

Text: Convert the output to an integer.

Covers: SADR\_03

**SDDR\_16**

Name: ADC\_Lire\_resultat

Text: Return Conversion value.

Covers: SADR\_03

MSP430 2231 Functions:

**SDDR\_17**

Name: main

Text: Initialize the SPI in slave mode and follow the instructions of the master

Covers: SADR\_03

**SDDR\_18**

Name: servo\_PWM

Text: Initialize a PWM signal and move the servo motor from right to left

Covers: SADR\_03

**SDDR\_19**

Name: servo\_stop

Text: Stop the servomotor

Covers: SADR\_03