MOQL

# System Software Requirements

Objective :

The objective of this project is to allow a communication between 2 microprocessors which are place on a « SAMBOT ». The user should be able to control the « SAMBOT » with a Bluetooth connection through a control screen. The « SAMBOT » can stay autonomous by interacting independently with the environment with his sensors.

SSR\_0100

Name : Servo Motor

Details : The software shall control the elements of the Servo Motor.

SSR\_0200

Name : Ultra Sonic Sensor

Details : The sensor shall detect any obstacle in front of the « SAMBOT ».

SSR\_0300

Name : Infrared Sensor

Details : The sensor shall prevent the « SAMBOT » to fall of any edge.

SSR\_0400

Name : Communication SPI between Microprocessors (MSP430G2553 – MSP430G2231)

Details : The two microprocessors shall be able to communicate with each other.

SSR\_0500

Name : Commands Control

Details : The MSP430G2553 shall be able to control the multiples motors of the « SAMBOT » to move safely in his environment.

SSR\_0600

Name : Control Screen

Details : The user shall be able to select multiples commands to control the movements of the « SAMBOT » through an interface.

SSR\_0700

Name : Communication UART

Details : The user instructions shall be send to the MSP430G2553 thanks to the RN42 (Bluetooth technology).

# Documentation Architecture générale / Architectural Design Requirements

ADR\_0100

Name : ServoMotor Rotation

Covers : SSR\_0100

Details : Control the rotation and the speed rotation of the Servo Motor

Module : 2231-SM

ADR\_0200

Name : Calcul of distance

Covers : SSR\_0200

Details : Perform the multiples calculations needed to measure the distance with the obstacle

Module : 2231-US

ADR\_0300

Name : Ground detection

Covers : SSR\_0300

Details : Detect the presence of ground to prevent the « SAMBOT » to fall

Module : 2231-IF

ADR\_0400

Name : SPI Communication

Covers : SSR\_0400

Details : Allow the communication between the two Microprocessors with specific protocols

Module : 2231-SPI, 2553-SPI

ADR\_0500

Name : Data Interpretations

Covers : SSR\_0400, SSR-0500

Details : Execute a specific command following the data receive

Module : 2553-Commands

ADR\_0550

Name : Motors Controls

Covers : SSR\_0500

Details : Control the sense of rotation and the speed of the motor for each wheel

Module : 2553-Commands

ADR\_0600

Name : User Interface

Covers : SSR\_0600, SSR\_0700

Details : Allow the user to interact with the « SAMBOT » with a window displaying the commands

Module : 2553-ScreenControl

ADR\_0700

Name : UART Communication

Covers : SSR\_0700

Details : Allow the communication between the User Interface and the MSP430G2553 with specific protocols

Module : 2553-Commands, 2553-UART

# Detailled Design Requirements

DDR\_0100

Name : ServoMotor Rotation Limits

Covers : ADR\_0100

Details : Control the limits of rotation of the Servo Motor

Function :

DDR\_0150

Name : ServoMotor Speed

Covers : ADR\_100

Details : Control the speed rotation of the Servo Motor

Function :

DDR\_0160

Name : ServoMotor Calculs

Covers : ADR\_0100, ADR\_0200

Details : Make all calculations necessary for the position and the speed of the rotation of the Servo Motor, in accordance with the Ultrasonic sensor.

Function :

DDR\_0200

Name : Input Ultrasonic Sensor

Covers : ADR\_0100, ADR\_0200

Details : Treat the input data from the Ultrasonic sensor to make calculations

Function :

DDR\_0250

Name : Calculations Ultrasonic Sensor

Covers : ADR\_0200

Details : Make all necessary calculations for measuring the distance of the obstacle

Function :

DDR\_0300

Name : Input Infrared Sensor

Covers : ADR\_0300

Details : Treat the input data from the Infrared Sensor and send it to Infrared Reactions

Function :

DDR\_0350

Name : Infrared Reactions

Covers : ADR\_0300, ADR\_0500

Details : Treat the data from Input Infrared Sensor before selecting and sending the correction needed for the motors

Function :

DDR\_0400

Name : Format SPI

Covers : ADR\_0400

Details : Convert the data in the correct format receives from multiples functions

Function :

DDR\_0450

Name : Communication for SPI

Covers : ADR\_0400

Details : Establish the communication before sending the data from Format SPI

Function :

DDR\_0500

Name : Treatment of data

Covers : ADR\_0500

Details : Treat the data receives from Communication for SPI and select the correct changes to implement to the system

Function :

DDR\_0550

Name : Send Orders

Covers : ADR\_0500, ADR\_0550

Details : Send the changes to implement to the system to the correct functions

Function :

DDR\_0600

Name : Creation of the Interface

Covers : ADR\_0600

Details : Create a window where the user can interact with the « SAMBOT »

Function :

DDR\_0650

Name : Display of Commands

Covers : ADR\_0600

Details : Display the multiples commands the user can use to control the « SAMBOT »

Function :

DDR\_0660

Name : Selection of Command

Covers : ADR\_0600, ADR\_0700

Details : Save the selected command of the user and send the order to the correspondent function of the system

Function :

DDR\_0700

Name : Format UART

Covers : ADR\_0700

Details : Convert the data in the correct format receives from Selection of Command

Function :

DDR\_0750

Name : Communication for UART

Covers : ADR\_0700

Details : Establish the communication before sending the data from Format UART

Function :

# Unitary Tests

Execute a test relative to every command of the User interface and check the reaction

* UT\_0110 : Advance
* UT\_0120: Back Off
* UT\_0130 : Turn Right
* UT\_0140 : Turn Left

Execute a test for every motor engine (ServoMotor + Wheels engines) on their speed, limits and their sense of rotation

* UT\_0210 : ServoMotor
  + UT\_0211 : Change of Speed
  + UT\_0212 : Change of the Sense of rotation
  + UT\_0213 : Speed limit UP
  + UT\_0214 : Speed limit DOWN
* UT\_0220 : Wheel engine LEFT
  + UT\_0221 : Change of speed
  + UT\_0222 : Change of the sense of rotation
  + UT\_0223 : Speed limit UP
* UT\_0230 : Wheel engine RIGHT
  + UT\_0231 : Change of speed
  + UT\_0232 : Change of the sense of rotation
  + UT\_0233 : Speed limit UP

Execute a test to verify the good transmission of data between the elements with SPI & UART

* UT\_0310 : Send an echo from the Launchpad 2553 to 2231
* UT\_0320 : Send an echo from the Launchpad 2231 to 2553
* UT\_0330 : Use a terminal with Bluetooth to send a command to the “SAMBOT”

Execute a test to verify if the Infrared sensor detect correctly and the distance of detection

* UT\_0410 : Distance of detection (>5 millimeters)
* UT\_0420 : Detection of ground on multiples surfaces

Execute a test to verify if the Ultrasonic sensor detect an obstacle, the distance of detection (maximum & minimum) and if he works correctly with the Servomotor

* UT\_0510 : Detection of an obstacle by the Ultrasonic Sensor on the Servomotor turning
* UT\_0520 : Distance of detection (>10 centimeters) with the Servomotor turning
* UT\_0530 : Distance of detection (<20 centimeters) with the Servomotor turning

# Integrity Tests

IT\_0100 : The multiples commands send with the User Interface must be carry out in the respect of security and the integrity of the “SAMBOT” (relative to UT\_01xx)

IT\_0200 : The wheels engines must adapt to the commands send to the “SAMBOT” (relative to UT\_01xx, UT\_0220, UT\_0230, UT\_03xx)

IT\_0300 : SPI & UART signals must be as fast as possible with an obligation of being under 0,5 second (relative to UT\_03xx)

IT\_0400 : The multiples sensors must be useable as the “SAMBOT” is moving (relative to UT\_01xx, UT\_03xx, UT\_04xx, UT\_05xx)

IT\_0500 : The “SAMBOT” must be able to correctly react to multiples obstacles in his environment (no danger of integrity or security) (relative to all UT)

# Acceptance Test

AT\_1000 :

The « SAMBOT » must be able to adapt to multiples environments and the different datas send by the sensors