Radboud University 

Design of Embedded System

Mars Rover Project Report

Luca Poli[1124244], Razvan Potcoveanu[1123076]

Version: 1.0.0

Table of Contents

1. Planning

1.1 Requirements

1.1.1 Definition

Group	Code	Priority	Description
Functionality	F_SFT	M	The rover should perform any mission safely (without falling of the edge or getting stuck in the lakes)
	F_MSQ	M	The rover should perform the missions in sequential order (from the first)
	F_COL	M	The rover should be able to detect colors, to find or avoid them
	F_DIS	M	The rover should be able to detect distant objects
	F_IMP	M	The rover should be able to detect impact with other objects
	F_ARM	M	The rover should be able to use the measurement arm
	F_POB	S	The rover should be able to push another object
	F_FED	S	The rover should give feedback (led or sound) on mission complete or errors
	F_PRK	C	The rover should be able to park
	F_DNS	W	The rover should be able to dance and sing
Usability	U_NHM	M	The rover should perform any mission without human interaction
	U_ELG	M	The user should be able to see the error log
	U_ALG	S	The rover should keep an activity log
	U_TLG	C	The rover should keep track of time required for a mission
Reliability	R_ESR	M	The rover should stop and restart after a critical error
	R_BRC	M	After a Bluetooth connection lost, the rover should try to reconnect
	R_SER	M	The rover should be able to work normally even with some sensors read errors
Performance	P_CTM	M	The rover should be able to perform the missions in a reasonable time
	P_BCM	M	The two bricks should respond to each other's instructions immediately
	P_TPR	M	The rover should response to trigger in order of their priority
	P_IEL	S	The rover should log the errors immediately
	P_EFM	W	The rover should perform the missions in an efficient way
Supportability	S_COL	M	The user should be able to configure the colors to detect (and how to react to them)
	S_MSP	S	The user should be able to configure the speed of the motor
	S_ARM	S	The user should be able to configure on which objects the rover has to use the measurement arms
	S_MOV	C	The user should be able to configure the movement actions parameters (angle to turn, distance, …)
	S_FED	C	The user should be able to configure the missions feedback
	S_POB	C	The user should be able to configure which objects the rover has to move
	S_FSN	W	The user should be able to add and use custom sounds for feedback

1.1.2 Timetable

Week number	Activities
0: 22/11 – 27/11	Requirements definition and planning; Sensor/actuators mapping; DSL typepal refactor; Refine of DSL grammar
1: 29/11 – 05/12	Basic must have functionalities [F_SFT, F_COL, F_DIS, F_IMP, F_ARM, U_NHM, U_ELG, R_SER, S_COL]
2: 06/12 – 12/12	Test and refine of implementation of F_SFT; Finish implementation of previous functionalities; Implementation of [F_MSQ, F_POB, F_FED, R_ESR, R_BRC, P_CTM, P_BCM, U_ALG]
3: 13/12 – 19/12	Finish implementation of previous functionalities; Implementation of [P_TPR, P_IEL, S_MSP, S_ARM, S_MOV, F_PARK]
4: 20/12 – 26/12	Finish implementation of previous functionalities; Implementation of [P_IEL, S_MOV, S_FED, S_POB, U_TLG]
5: 27/12 – 02/01	Finish implementation of previous functionalities; Implementation of [S_FSN, P_EFM, F_DNS]
6: 03/01 – 09/01	Testing and refining

1.2 Sensors/Actuators Mapping

1.2.1 Proposal

	EV3 Brick1	EV3 Brick2
Actuators	Left motor Right motor	Measurement arm motor
Sensors	Color left Color right Color mid Ultrasonic back	Touch left Touch right Touch back Ultrasonic front

Explanation:

Because of the delay in communication between the two bricks we decided to go with this configuration for brick1 (motor and safety related sensors) because it’s going to assure that we don’t drive off the table. For brick2 we have put the ultrasonic front sensor, because we can increase the detection range to account for the delay.