MINI-PROJET 2 | *Information systems*
CAN Lift

Hes·so VALAIS WALLIS
Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

Part 1: Project description and evaluation

1.1 Introduction

The main objective of this laboratory is to realize the driving of an elevator according to realistic rules. This elevator will be controlled through a CAN bus. You will have to realize all the logic to control the elevator (control the motors, opening and closing the doors…) on the micro-controller and send the commands (via CAN) to the elevator. You can use the previous project (Modbus) as a starting template for your project. You also need to download the required files for the CAN library (`.c` and `.h`) that are available on http://sin.begincoding.net/.

1.2 List of commands and rules

Appendix 1 gives the list of the CAN messages supported by the elevator. Take some time to study those messages, as they provide your only interface to the elevator itself.

1.2.1 Basic rules

These rules at managing the displacement of the lift with simplicity and security. With those rules, the displacement speed is limited and no sequence optimizations are required. Those rules are as follows:

1. the lift shall not move when a door is open;
2. if a door is open, it remains open for a moment (3 seconds) and after that, they automatically close;
3. if a user presses a button, the light next to this button has to be turned on and the lift shall move to the corresponding floor, except if the lift is already stopped at this floor. In that case, the door shall open;
4. when the lift reaches the selected floor, the floor buttons lights and the lift button light turn off;
5. the lift shall slow down before stopping at a defined floor;
6. if the door is closing and simultaneously the button outside the lift is pressed on the same floor, the door will re-open immediately.

1.2.2 Extended rules

The goals of these rules are to manage the displacement of the lift in order to increase the transfer capacity.

1. if a user presses a button in the lift and the door is open, the door could close immediately and the lift could move
2. the lift could move faster when the floor to reach is more than one floor away from its current position
3. when a user at a floor needs to take the elevator, the elevator could stop if it is not too close to this floor
4. when a user at a floor needs to go up and the lift goes down, the lift shall not stop. The same applies to the other direction.
5. when the lift moves in a direction and after a stop needs to go up and down, it has to go to the floor further in the same direction before to change its direction

1.3 Project objectives and evaluation

The objective of the project is to implement all the rules. The evaluation for the microcontroller part of the semester is twofold: code presentation and oral examination.

1.3.1 Code presentation

An oral presentation **during the last lecture** together will be done by the group to present the work done. The **evaluation** of this presentation is based on a 5 minutes' presentation (without slides), which is followed by 5 minutes of questions. This evaluation takes place in front of a PC with the presence of the group and an evaluation board composed of professor(s) and of a collaborator. For this evaluation, the group shall present on **one page** the structure of the software and the relations between the developed functions.

As a **minimum**, the students group shall implement a C program integrating the basic rules defined above. As an **option,** extended rules can be implemented. They are optional and are considered a bonus which can improve your mark but not lower it.

Items evaluated during this presentation are: code quality, functional performance of the implemented solution, your one-page summary as as answers to questions.

1.3.2 Oral exam

In addition to the evaluation of the project which takes place during the last lecture, an oral exam will take place during the exam week. This exam will start with theoretical questions about programming micro-controllers and will be followed by specific questions about your implementation. For the oral exam, you must TAKE YOUR CODE WITH YOU (on paper).

CAN Lift

	Name	CAN ID	Remote Frame [R]	Message Data Byte 0	Byte 1	Byte 2	Description	ET/TT/RR*	Event / Command / Response
Msg from Controller	*ID_REQUEST*	0x7FF		-	-	-	Request for all lifts to provide their ID	RR (Request)	At startup, is a broadcast
	MOTOR_SPEED	Lift ID + 0x020		-128 to 127	-	-	Set speed of lift motor (INT8, 0 is stop)	-	
	LED_LIFT	Lift ID + 0x040		LEDs	-	-	Set and clear LEDs in lift	ET	On button pressed and stage reached
	LED_FLOOR	Lift ID + 0x060		LED_UP	LED_DOWN		Set and clear LEDs on floors	ET	On button pressed and stage reached
	DOOR_STATE	Lift ID + 0x080		0	-	-	Close the lift door	ET/TT	On button pressed, time elapsed to close the door automatically
				1	-	-	Open the lift door	ET	On stage reached, button pressed
	STATUS REQUEST	Lift ID + 0x0A0	R	-	-	-	Request lift status (sensors, motors, doors)	CR (Request)	
	RESET	Lift ID + 0x0C0		-	-	-	Reset the lift	ET	After a crash
	LIGHT_CAB	Lift ID + 0x0E0		0	-	-	Turn OFF light in cab	ET/TT	After a time elapsed without events, turn off
				1	-	-	Turn ON light in cab	ET	When the lift is in action

	Name	***CAN ID***		***Message Data Byte 0***	***Byte 1***	***Byte 2***	***Description***	***ET/TT/RR****	***Event***
Msg from Lift	*ID_RESPONSE*	LiftID + 0x7E0		-	-	-	Lift ID returned	RR Response	Answer to ID_REQ command
	BUTTON_LIFT	Lift ID + 0x100		BTN	0	-	Button pressed	ET	User has pressed / released a button in a lift
					1	-	Button released		
	BUTTON_FLOOR	Lift ID +0x120		BTN_UP	BTN_DOWN	0	Button pressed	ET	User has pressed / released a button on a floor
						1	Button released		
	STATUS RESPONSE	Lift ID + 0x0A0		SENSORS	...	...	Stage sensors status (0 between floors)	RR (Response)	Response for a STATUS command
				...	-128 to 127	...	Current speed of motor	ET	On each change of a status parameter
				...	...	0	Door is closed		(For motor speed : every 250 ms when the speed
				...	...	1	Door is opening		is varying)
				...	...	2	Door is closing		(Doors: only open / close status)
				...	...	3	Door is open		
	ERROR	Lift ID +0x140		0	-	-	Crash occured lift higher as stage 7	ET	Lift goes higher as stage 7 (motor not stopped)
				1	-	-	Crash occured lift lower as stage 0		Lift goes lower as stage 0 (motor not stopped)
				2	-	-	Stopped between stage		Motor has stopped and lift is between two stages
				3	-	-	Start moving with door not closed		Motor start with door not closed
				4	-	-	Open a door with motor not stopped		Door open command with motor not stopped
	RAIL LIFT	Lift ID + 0x300		-	-	-	Between each floor are 11 pulses	ET	On each pulse of rail lift sensor

	CAN speed	125 kB/s						ET/TT*	Event Triggered, Time Triggered
	CAN ID format	11 bits						RR	Request Response
	Lift IDs	0x01 to 0x1E		hardware coded in EEPROM			ID 0x1F is a broadcast address		
	LEDs & BTN in lift	one bit per floor (example floor 1 : 0b00000010)							
	LEDs & BTN in floors UP	one bit per floor (example LED_UP floor 6 : 0b**0**1000000), bold is always 0							
	LEDs & BTN in floors DOWN	one bit per floor (example LED_DOWN floor 6 : 0b0100000**0**), bold is always 0							*vers: 3.2*