ITR : Project

April 4, 2020

1 Introduction

This project consists in programming a small robot car to make it go through a maze without destroying it and to stop when arrived. The robot has a basic OS (OSEK/VDX), two motors, a sonar and two bumpers. We started by implementing a small library to help us to develop the tasks. Then we implemented some tasks and alarm to drive the robot and make it avoid the walls.

2 The librairy we implemented

Because we are on an embedded system we have only few functions to work with we implemented some useful functions to make our work easier.

2.1 Make it go straight and backward

We implemented two functions *void go_front(int degree)* and *void go_back(int degree)*. Those two functions turn the wheels of *degree*° at a speed v which have been chosen by us. We use only 20% of the speed of the motors to not loose too much precision on those functions.

2.2 Make it turn

We had different solutions to make our robot turn and we choose to set a wheel speed at the value v and the other wheels whith the speed $-v$. Like that the center of the robot does not move while it turns.

2.3 Printer

We also implemented to make a proper print on the screen with the function *printf(char*),int**. This function reads the *char** until it finds a or a then it print the line up to there. If it finds a % while reading it print a member of the *int** (the *i*eme if it's the *i*eme %).

Each time we print a line, we move the cursor to the following line and when the screen is full we clear it and print on the first line again.

3 Tasks and alarm

The robot main task consists in going forward, and to do it again. But we do not want to destroy the walls so we used alarms to check if we are going into a wall, if we are arrived, if a wall is approaching us too close.

3.1 Bumpers check

We have a task that checks either or not we are running into a wall. This task is triggered by an alarm and have a higher priority than the main task. The alarm has a cycle of 100ms to not destroy the walls too much while still executing the main task. When we touch a wall if only one bumper is touching then we are turning in the other direction. If both bumpers are touching it means we went too far so we are going back a little and turning left. We are checking if a wall is near and if it is the case we turn right twice (one to cancel the previous turn and another to go right). Just after that we are checking if there is another wall because if it is the case it means we are in a dead-end so we are arrived.

3.2 Sonar check

The bumpers are checking if we are touching a wall, but we want to avoid the walls as much as possible so if we can turn before touching them it will be better! If there is a wall just behind us then we are turning left. If there is still a wall it means that there is a corner so we turn right twice. And again, if there is another wall it means we are arrived.