CmpE 443
Fall 2019
Final Project
Wall Follower Car

Pit Crew
Elif Çalışkan
Baturalp Yörük
Yağmur Kahyaoğlu
Mehmet Berk Kemaloğlu

Jan 3, 2019

Contents

1 Summary

1.1 Test mode

We have worked for 8 days in order to complete this car. The first day of implementation (26.12.19), we finished LEDs and motor for testing part. We used lab templates in order to remember the workflow. We implemented methods for going forward, backward, turning left etc. The second day of implementation, (27.12.19) we wrote the codes for LDR and ultrasonic sensor. At the end of the day, LDR was working but ultrasonic sensor was not giving correct data. The third day of implementation (28.12.19), we skipped ultrasonic sensor and tried to communicate using UART. After successfully communicating, we added Bluetooth part to it. Then, we created methods for handling the input from Bluetooth. The forth day of implementation (30.12.19), we added the missing parts of potentiometer and speedometer. At the end of the day, our board was ready for demo. At the fifth day (31.12.19), we made a demo and moved the pins and components to our car. At the end of the day, our car was ready for test mode.

During the development of the wall following car, naturally, we have encountered some problems. First, we took the board and started to set the components up. We started with LEDs, continued with the motor, ultrasonic sensor, LDRs, potentiometer, UART, Bluetooth, and speedometer. We lost lots of time in the ultrasonic sensor because it was not reading correct values. We were not sure whether the problem was originated from hardware or software. But after three hours, we noticed our ultrasonic sensor was broken. Besides, we have lost time on Bluetooth. We gave 1 bit wrong to one of the HM10's register and it started to wait for THRE interrupt. We couldn't figure out easily. These two were our major problems. Other things have been handled without spending too much time.

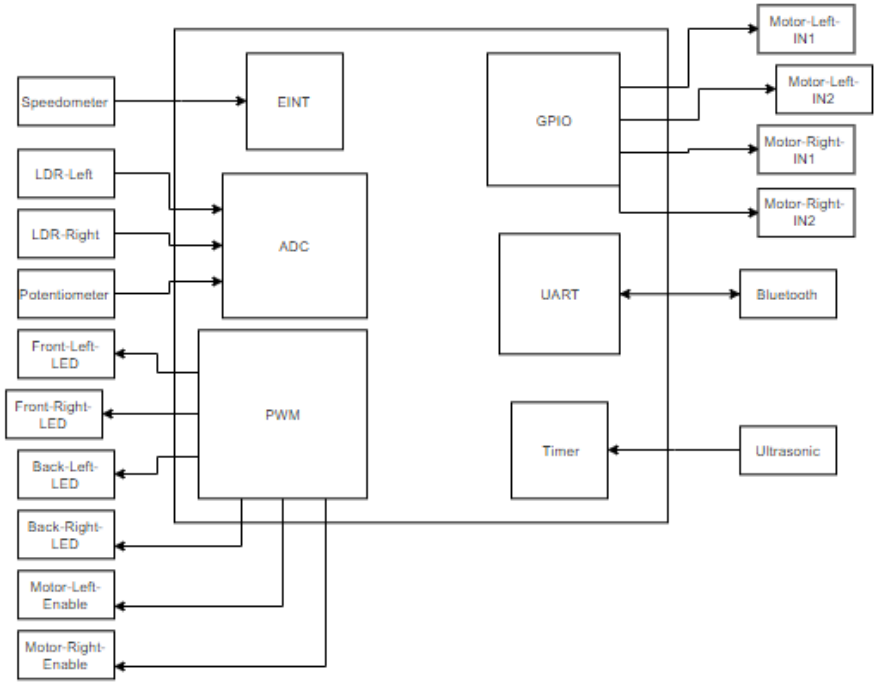
1.2 Autonomous mode

We spend three days on autonomous mode (1.1.20, 2.1.20, 3.1.20). After finishing the test mode, we started to implement autonomous mode. The former part was simple. Implementing the test mode doesn't have many alternatives. However, the autonomous mode has infinite solutions. After the brainstorming stage, we implemented an algorithm. Mainly it works like this: Car goes forward only 1 turn. It measures the difference of distances throughout this step. In the next step, the car turns according to the difference. In this way car always stays between the borders. In the paper the algorithm makes sense. However, in practice, it didn't work well. This algorithm was using lots of flags in order to fulfill the expected functionality. Ultrasonic sensor doesn't always give the correct results. We were not aware of the basics of robotics -usage of median filter. Although we've worked on the implementation of this algorithm hard, we couldn't manage to run it perfectly. Therefore, we had to give up this method. During this time, we have thought of another algorithm that was not

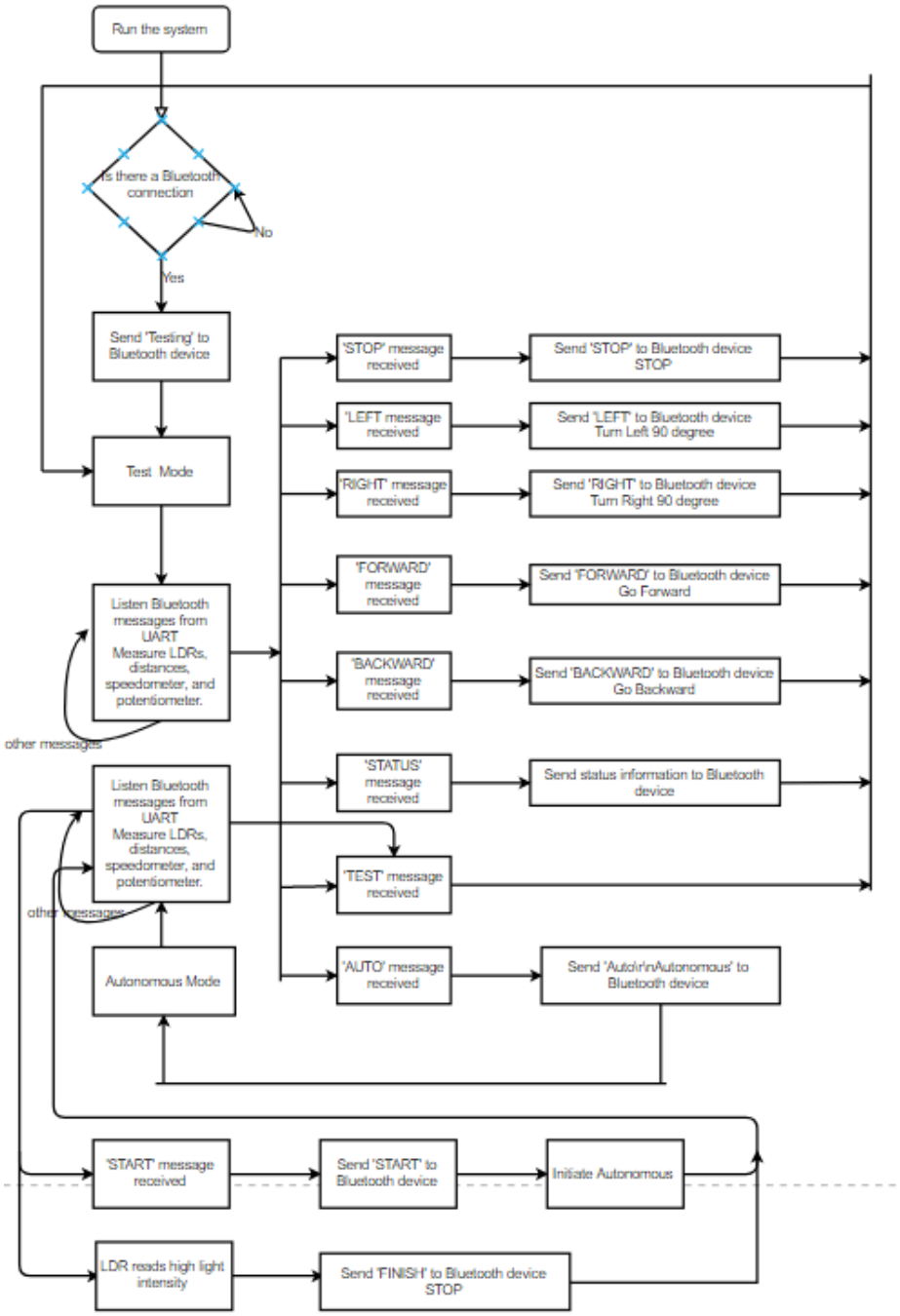
as simple as this. That algorithm works like this: Go forward until the distance changed. Then, turn until the car becomes parallel to the wall. We couldn't implement it too because of the lack of time. Then we developed a simple and working algorithm. At least, now, the car follows a simple path. The algorithm works like the following. It measures the current distance. Then it gives values to duty cycles of the motors' according to the distance. None of us is as good as Edsger W. Dijkstra. We are developers who can develop this code in limited time. We are proud of what we did.

If we need to mention other details, we did them as expected. For extra, we tried to make one of the LED yellow. In order to achieve this perfectly, we had to use one more PWM pin. Because of lack of time, we gave the same PWM output to the red and green input of the RGB LED. So, the colors look red or green from different angles now but from the opposite side it looks yellow.

2 Block Diagram



3 The Flow Chart of the System

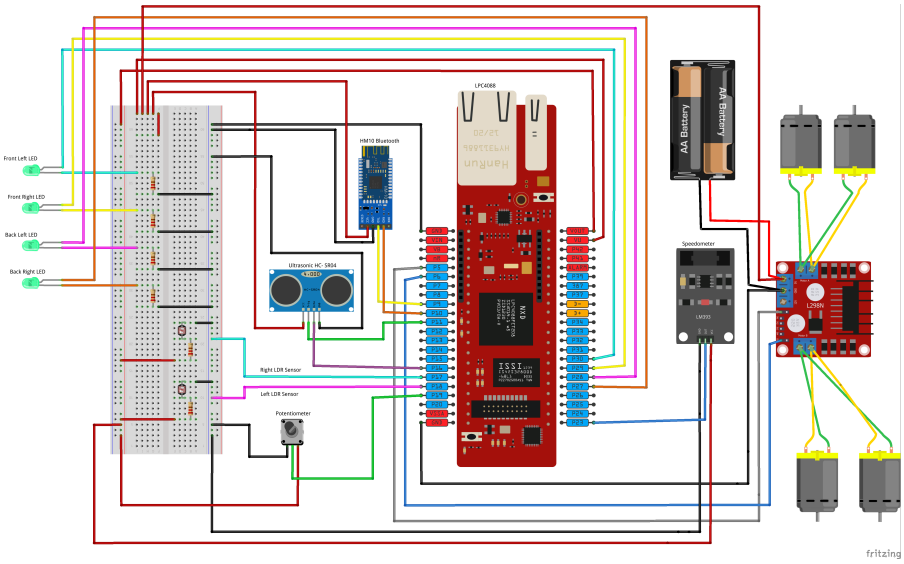


4 Pin connection tables

LPC 4088 Pin	Pin Number	Function	Purpose
P1.2	P30	PWM	Front-Left-LED
P1.3	P29	PWM	Front-Right-LED
P1.5	P28	PWM	Back-Left-LED
P1.6	P27	PWM	Back-Right-LED
P1.24	P5	PWM	Motor-1
P1.23	P6	PWM	Motor-2
P0.0	P9	UART3-TX	Bluetooth
P0.1	P10	UART3-RX	Bluetooth
P5.0	P39	GPIO	Motor1-IN-1
P5.1	P38	GPIO	Motor1-IN-2
P0.4	P34	GPIO	Motor2-IN-1
P0.5	P33	GPIO	Motor2-IN-2
P2.10	P23	EINT0	Speedometer
P0.26	P18	ADC	LDR1
P0.25	P17	ADC	LDR2
P1.30	P19	ADC	Potentiometer
P0.2	P42	UART0-TX	Serial-Communication
P0.3	P41	UART0-RX	Serial-Communication
P1.29	P14	Timer0-Match1	Timer for Ultrasonic
P0.9	P11	Trigger	Ultrasonic
P0.24	P16	Echo	Ultrasonic

Table 1: PIN Connection Table

5 Circuit Schematics



6 How to use guide

Our system is intended to be used to run an embedded system car. The system has a Bluetooth connection. The car can be controlled by an android application. The car can run in 2 modes: 'Test Mode', and 'Autonomous Mode'. After each push button activity, the car echoes the given command and makes an action. In order to learn the light intensity or distance at any time, STATUS button can be pushed. All the sensors, motors and LEDs are defined in the code. They are connected to the corresponding pins. The Autonomous mode can be updated by writing codes into the function 'updateAuto' which belongs to 'main.c' file.

6.1 TEST MODE

The car goes into test mode by default but it can be started with pushing "TEST" button using Bluetooth. In test mode, the car is controlled using FORWARD, LEFT, RIGHT, BACK, STOP buttons. If FORWARD button is pressed, then the front LEDs turn on and the car moves forward until another button is pressed. If LEFT button is pressed, then the left LEDs flicker and the car turns 90 degrees to left. If RIGHT button is pressed, then the right LEDs flicker and the car turns 90 degrees to right. If STOP button is pressed, all LEDs are turned off and the car stops moving. If BACK button is pressed, then the car starts to move backwards while turning on the LEDs at the back of the car. If at anytime, the car sees a light in front of it, it stops moving until

the light is gone. The car's speed can be changed by using the speedometer at the top of the car.

6.2 AUTONOMOUS MODE

The car goes into autonomous mode by pushing TEST button using Bluetooth. After selecting autonomous mode, the car starts working after pushing START button. In autonomous mode, the car follows the wall on the left side. If at anytime, the car sees a light in front of it, it stops moving until the light is gone and sends FINISH signal with Bluetooth.

7 Technical Details

7.1 LEDs

LPC 4088 Pin	**Pin Number**	**Function**	**Purpose**
P1.2	P30	PWM	Front-Left-LED
P1.3	P29	PWM	Front-Right-LED
P1.5	P28	PWM	Back-Left-LED
P1.6	P27	PWM	Back-Right-LED

Table 2: LED Connections

In order to go forward and turn the front LEDs on: PWM0$_{C}ycle_{R}ate$ is set to 20; PWM0 values for Front-Left-LED and Front-Right-LED is set to 100 and others to 0.

In order to go backward and turn the back LEDs on: PWM0$_{C}ycle_{R}ate$ is set to 20; PWM0 values for Back-Left-LED and Back-Right-LED is set to 100 and others to 0.

In order to turn left and flicker the left LEDs: PWM0$_{C}ycle_{R}ate$ is set to 500; PWM0 values for Front-Left-LED and Back-Left-LED is set to 50 and others to 0.

In order to turn right and flicker the right LEDs: PWM0$_{C}ycle_{R}ate$ is set to 500; PWM0 values for Front-Right-LED and Back-Right-LED is set to 50 and others to 0. 1

In order to stop and turn off all the LEDs: PWM0$_{C}ycle_{R}ate$ is set to 500; PWM0 values for all LEDs are set to 0.

7.2 MOTOR

LPC 4088 Pin	Pin Number	Function	Purpose
P1.24	P5	PWM	Motor-1
P1.23	P6	PWM	Motor-2
P5.0	P39	GPIO	Motor1-IN-1
P5.1	P38	GPIO	Motor1-IN-2
P0.4	P34	GPIO	Motor2-IN-1
P0.5	P33	GPIO	Motor2-IN-2

Table 3: Motor Connections

In order to go forward: Motor1 In1 and Motor2 In2 is enabled and the others are disabled.

In order to go backward: Motor1 In2 and Motor2 In1 is enabled and the others are disabled.

In order to turn left: Motor1 In2 and Motor2 In2 is enabled and the others are disabled.

In order to turn right: Motor1 In1 and Motor2 In1 is enabled and the others are disabled.

In order to stop: All pins are disabled. The speed is determined by $\mathrm{PWM}_{Writefunction}$.

7.3 ADC

LPC 4088 Pin	Pin Number	Function	Purpose
P0.26	P18	ADC	LDR1
P0.25	P17	ADC	LDR2
P1.30	P19	ADC	Potentiometer

Table 4: ADC Connections

We are using 2 LDRs which sense light intensity positioned in front of the car, one on the left, the other on right side. They stop the car if the light intensity passes some value.

There is also a potentiometer connected to ADC. It senses the voltage level inside its two legs and changes speed of the car accordingly.

7.4 Ultrasonic

LPC 4088 Pin	Pin Number	Function	Purpose
P1.29	P14	Timer0-Match1	Timer for Ultrasonic
P0.9	P11	Trigger	Ultrasonic
P0.24	P16	Echo	Ultrasonic

Table 5: Ultrasonic Connections

The ultrasonic trigger works at every 60 ms which is the suggested waiting time. Whenever there is a new data available if the car is in auto mode, it will decide on the next move. The distance at every step can be seen with command 'STATUS'. The handler gets called on every rising edge.

7.5 Serial Communication

LPC 4088 Pin	Pin Number	Function	Purpose
P0.2	P42	UART0-TX	Serial-Communication
P0.3	P41	UART0-RX	Serial-Communication

Table 6: Serial Connections

9600 baudrate is used and the values are updated for 8-bit character transfer, 1 stop bits and event parity. Receive data available and THRE interrupts are enabled. After each read data, the char is controlled to see if it is equal to 13(end of the command), then it reads the command by comparing strings. But this part is not called in main method. This was used for trying the communication part.

7.6 EINT

LPC 4088 Pin	Pin Number	Function	Purpose
P2.10	P23	EINT0	Speedometer

Table 7: External Interrupt

The speedometer is connected to External Interrupt. It counts the turn number of the wheels and we are using this functionality for turning according to number of turns. For example, it stops after measuring 6 turns, since it corresponds to a 90 degree turn.

7.7 Bluetooth

LPC 4088 Pin	Pin Number	Function	Purpose
P0.0	P9	UART3-TX	Bluetooth
P0.1	P10	UART3-RX	Bluetooth

Table 8: Bluetooth

9600 baudrate is used and the values are updated for 8-bit character transfer, 1 stop bits and event parity. Receive data available interrupt is enabled. After each new data available, it reads data then finds the selected command by string comparison.

7.8 Sequence Diagrams

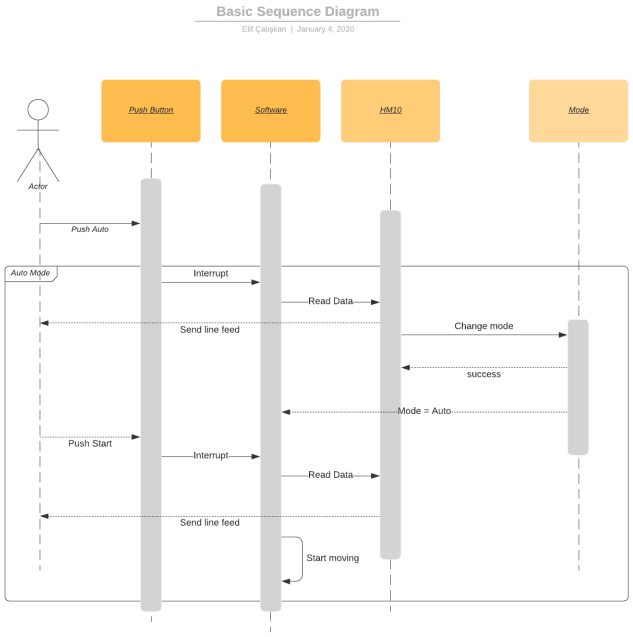


Figure 1: Sequence diagram of stating autonomous mode

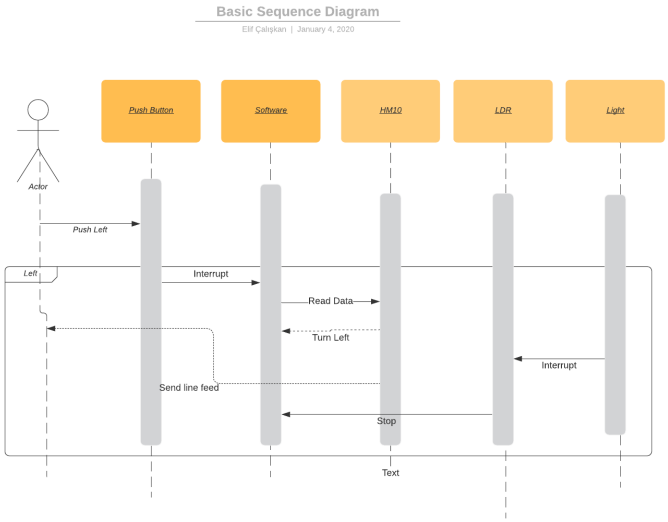


Figure 2: Sequence diagram of pressing left button

8 Expense List

No expenses were made during the process of implementation.

9 References

- UM10562 (LPC408x/407x User manual)
- LPC4088 QuickStartBoard Schematic
- L298N Dual H-Bridge Motor Driver User Guide
- Ultrasonic Ranging Module HC - SR04 Manual
- Cortex^{TM} - M4 Devices Generic User Guide