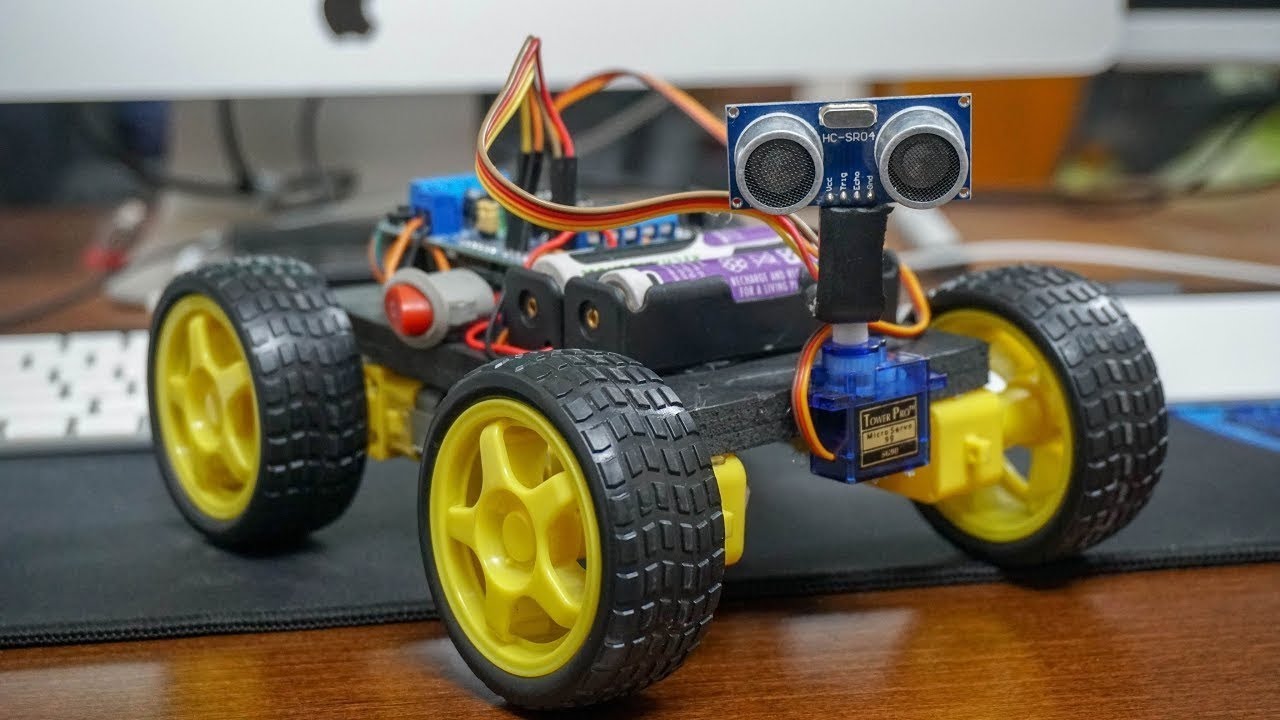
**Obstacle Avoidance Robot V1.0 Design**



*Team 2*

*1- Ahmed Atef*

*2- Amr Gamal El-Abd*

*3- Anas Mahmoud*

*4- Khaled Mustafa*

Table of Contents

[1-Description 3](#_Toc135483440)

[1.1 Hardware components 3](#_Toc135483441)

[1.2 System Requirements 3](#_Toc135483442)

[2-High Level Design 5](#_Toc135483443)

[2.1 Layered Architecture 5](#_Toc135483444)

[2.2 System Flow Chart 6](#_Toc135483445)

[2.3 Drivers Descriptions 2.3.1 DIO Driver 9](#_Toc135483446)

[2.3.2 Timer Driver 9](#_Toc135483447)

[2.3.3 PWM Driver 9](#_Toc135483448)

[2.3.4 EXI Driver 9](#_Toc135483449)

[2.3.5 ICU Driver 9](#_Toc135483450)

[2.3.6 Keypad Driver 10](#_Toc135483451)

[2.3.7 Button Driver 10](#_Toc135483452)

[2.3.8 LCD Driver 10](#_Toc135483453)

[2.3.9 Ultrasonic Driver 10](#_Toc135483454)

[2.3.10 Motor Driver 10](#_Toc135483455)

[2.3.11 Application Driver 11](#_Toc135483456)

[2.4 Modules API’s 12](#_Toc135483457)

[2.4.1 DIO Module 12](#_Toc135483458)

[2.4.2 Timer Module 12](#_Toc135483459)

[2.4.3 PWM Module 12](#_Toc135483460)

[2.4.4 EXI Module 12](#_Toc135483461)

[2.4.5 ICU Module 13](#_Toc135483462)

[2.4.6 Keypad Module 13](#_Toc135483463)

[2.4.7 Button Module 13](#_Toc135483464)

[2.4.8 LCD Module 13](#_Toc135483465)

[2.4.9 Ultrasonic Module 14](#_Toc135483466)

[2.4.10 Motor Module 14](#_Toc135483467)

[2.4.11 App Module 14](#_Toc135483468)

[3-Low Level Design 15](#_Toc135483469)

[3.1 APIs Flow Chart 15](#_Toc135483470)

[3.1.1 DIO API’s 15](#_Toc135483471)

[3.1. Keypad API’s 19](#_Toc135483472)

[3.1. Button 21](#_Toc135483473)

[3.1. Ultrasonic 22](#_Toc135483474)

[3.1. LCD 24](#_Toc135483475)

[3.1. APP 26](#_Toc135483476)

[3.2 Precompiling & Linking Configurations 35](#_Toc135483477)

# 1-Description

## Hardware components

1. Car Components:

1. ATmega32 microcontroller

2. Four motors (M1, M2, M3, M4)

3. One button to change default direction of rotation (PBUTTON0)

4. Keypad button 1 to start

5. Keypad button 2 to stop

6. One Ultrasonic sensor connected as follows 1. Vcc to 5V in the Board 2. GND to the ground In the Board 3. Trig to PB3 (Port B, Pin 3) 4. Echo to PB2 (Port B, Pin 2)

7. LCD

## 1.2 System Requirements

1. The car starts initially from 0 speed

2. The default rotation direction is to the right

3. Press (Keypad Btn 1), (Keypad Btn 2) to start or stop the robot respectively

4. After Pressing Start:

1. The LCD will display a centered message in line 1 “Set Def. Rot.”

2. The LCD will display the selected option in line 2 “Right”

3. The robot will wait for 5 seconds to choose between Right and Left

1. When PBUTTON0 is pressed once, the default rotation will be Left and the LCD line 2 will be updated

2. When PBUTTON0 is pressed again, the default rotation will be Right and the LCD line 2 will be updated

3. For each press the default rotation will changed and the LCD line 2 is updated

4. After the 5 seconds the default value of rotation is set

4. The robot will move after 2 seconds from setting the default direction of rotation.

5. For No obstacles or object is far than 70 centimeters:

1. The robot will move forward with 30% speed for 5 seconds

2. After 5 seconds it will move with 50% speed as long as there was no object or objects are located at more than 70 centimeters distance

3. The LCD will display the speed and moving direction in line 1: “Speed:00% Dir: F/B/R/S”, F: forward, B: Backwards, R: Rotating, and S: Stopped

4. The LCD will display Object distance in line 2 “Dist.: 000 Cm”

6. For Obstacles located between 30 and 70 centimeters

1. The robot will decrease its speed to 30%

2. LCD data is updated

7. For Obstacles located between 20 and 30 centimeters

1. The robot will stop and rotates 90 degrees to right/left according to the chosen configuration

2. The LCD data is updated

8. For Obstacles located less than 20 centimeters

1. The robot will stop, move backwards with 30% speed until distance is greater than 20 and less than 30

2. The LCD data is updated

3. Then preform point 8

9. Obstacles surrounding the robot (Bonus)

1. If the robot rotated for 360 degrees without finding any distance greater than 20 it will stop

2. LCD data will be updated.

3. The robot will frequently (each 3 seconds) check if any of the obstacles was removed or not and move in the direction of the furthest object

# 2-High Level Design

## 2.1 Layered Architecture

A screen shot of a computer

Description automatically generated with medium confidence

## 2.2 System Flow Chart

A picture containing text, screenshot, font, number

Description automatically generated

A diagram of a flowchart

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

## 2.3 Drivers Descriptions2.3.1 DIO Driver

Configuration: Consist of 6 API’s

Location: MCAL

Function: used to set pin direction (input or output), pin value (high or low) or read a value from a pin or toggle a pin

### 2.3.2 Timer Driver

Configuration: Consist of 6 API’s

Location: MCAL

Function: used to set a time delay

### 2.3.3 PWM Driver

Configuration: Consist of 5 API’s

Location: MCAL

Function: used to control motor speed

### 2.3.4 EXI Driver

Configuration: Consist of 5 API’s

Location: MCAL

Function: used to handle external events that happen during the execution

### 2.3.5 ICU Driver

Configuration: Consist of 5 API’s

Location: MCAL

Function: The Input Capture Unit Module is used to measure time between to events

### 2.3.6 Keypad Driver

Configuration: Consist of 2 API’s

Location: HAL

Function: used to initialize the keypad, get pressed key

### 2.3.7 Button Driver

Configuration: Consist of 1 API’s

Location: HAL

Function: used to check the button status pressed or not

### 2.3.8 LCD Driver

Configuration: Consist of 12 API’s

Location: HAL

Function: used to initialize the LCD, send command to LCD & display character or string to LCD & jump to specific position on LCD & to clear the LCD & to wright integer or float number on the LCD

### 2.3.9 Ultrasonic Driver

Configuration: Consist of 4 API’s

Location: HAL

Function: used to detect the distance between car & obstacle

### 2.3.10 Motor Driver

Configuration: Consist of 6 API’s

Location: HAL

Function: used to control car moving

### 2.3.11 Application Driver

Configuration: Consist of 2 API’s

Location: App

Function: combine between the drivers API’s to meet the requirement

## 2.4 Modules API’s

### 2.4.1 DIO Module

1- void DIO\_Init\_All(void);

2- en\_dioError\_t DIO\_initpin (DIO\_Pin\_type pin,DIO\_PinStatus\_type status);

3- en\_dioError\_t DIO\_writepin (DIO\_Pin\_type pin,DIO\_PinVoltage\_type volt);

4- en\_dioError\_t DIO\_readpin (DIO\_Pin\_type pin,DIO\_PinVoltage\_type \*volt);

5- en\_dioError\_t DIO\_togglepin(DIO\_Pin\_type pin);

6- en\_dioError\_t DIO\_WritePort (DIO\_Port\_type port,u8 value);

### 2.4.2 Timer Module

1- EN\_timerError\_t TIMER\_2\_init(Timer2Mode\_type a\_mode);

2- EN\_timerError\_t TIMER\_2\_start(Timer2Scaler\_type a\_prescaler);

3- void TIMER\_2\_stop(void);

4- EN\_timerError\_t TIMER\_2\_setIntialValue(u8 a\_value);

5- EN\_timerError\_t TIMER\_2\_OvfNum(double overflow);

6- void TIMER\_2\_DELAY\_MS(double \_delay);

### 2.4.3 PWM Module

1- void pwm\_init();

2- void waveGen();

3- void PWM\_set\_duty(u8 u8\_DutyCycle,u32 u32\_Freq);

4- void waveGen\_hf();

5- void PWM\_set\_duty\_hf(u8 u8\_DutyCycle,u32 u32\_Freq);

### 2.4.4 EXI Module

1- void EXI\_Init(void);

2- void EXI\_Enable(ExInterruptSource\_type Interrupt);

3- void EXI\_Disable(ExInterruptSource\_type Interrupt);

4- void EXI\_TriggerEdge(ExInterruptSource\_type Interrupt,TriggerEdge\_type Edge);

5- void EXI\_SetCallBack(ExInterruptSource\_type Interrupt,void(\*LocalPtr)(void));

### 2.4.5 ICU Module

1- void PWM\_Measure(u32\* Pfreq,u8\* Pduty);

2- static void Func\_ICU(void);

3- void PWM\_Measure\_exi(u32\* Pfreq,u8\* Pduty);

4- static void Func\_ICU\_exi(void);

5- void ICU\_SW(u32\* Pfreq,u8\* Pduty);

### 2.4.6 Keypad Module

1- KEYPAD\_initError KEYPAD\_init(void) ;

2- KEYPAD\_readError KEYPAD\_getpressedkey(u8 \*value) ;

### 2.4.7 Button Module

1- Button\_State Is\_pressed( u8 BUTTON\_PIN , u8 \*value);

### 2.4.8 LCD Module

1- void LCD\_Init(void);

2- void LCD\_WriteChar(u8 ch);

3- void LCD\_WriteString(u8\*str);

4- void LCD\_WriteNumber(s32 num);

5- void LCD\_WriteBinary(u8 num);

6- void LCD\_WriteHex(u8 num);

7- void LCD\_SetCursor(u8 line,u8 cell);

8- void LCD\_Clear(void);

9- void LCD\_ClearLoc(u8 line ,u8 cell,u8 num);

10- void LCD\_WriteNumber\_3D(u16 num);

11- void LCD\_CustomChar(u8 loc,u8\*pattern);

12- void LCD\_PinsInit ();

### 2.4.9 Ultrasonic Module

1-void USCallBackFun(void);

2-u16 US\_getdistance(void);

### 2.4.10 Motor Module

1- en\_MotorError\_t Car\_Motors\_init(void);

2- en\_MotorError\_t Car\_Moving\_FWD(void);

3- en\_MotorError\_t Car\_Moving\_BWD(void);

4- en\_MotorError\_t Car\_Rotate\_Right(void);

5- en\_MotorError\_t Car\_Rotate\_Left(void);

6- en\_MotorError\_t Car\_Stop(void);

### 2.4.11 App Module

1- void app\_init();

2- void app\_start();

# 3-Low Level Design

## 3.1 APIs Flow Chart

### 3.1.1 DIO API’s

1. en\_dioError\_t DIO\_initpin (DIO\_Pin\_type pin,DIO\_PinStatus\_type status)

A picture containing text, diagram, sketch, drawing

Description automatically generated

1. en\_dioError\_t DIO\_writepin (DIO\_Pin\_type pin,DIO\_PinVoltage\_type volt)

A diagram of a flowchart

Description automatically generated with low confidence

1. en\_dioError\_t DIO\_readpin (DIO\_Pin\_type pin,DIO\_PinVoltage\_type \*volt)

A diagram of a flowchart

Description automatically generated with low confidence

4- en\_dioError\_t DIO\_togglepin(DIO\_Pin\_type pin)

A picture containing text, sketch, diagram, drawing

Description automatically generated

### 3.1. Keypad API’s

A picture containing text, screenshot, design

Description automatically generated

A picture containing text, screenshot, diagram, design

Description automatically generated

### 3.1. Button

A diagram of a flowchart

Description automatically generated with low confidence

### 3.1. Ultrasonic

A picture containing text, screenshot, design

Description automatically generated

A picture containing text, screenshot, diagram, font

Description automatically generated

A diagram of a flowchart

Description automatically generated with low confidence

### 3.1. LCD

A screenshot of a diagram

Description automatically generated with low confidence

A picture containing text, diagram, screenshot, design

Description automatically generated

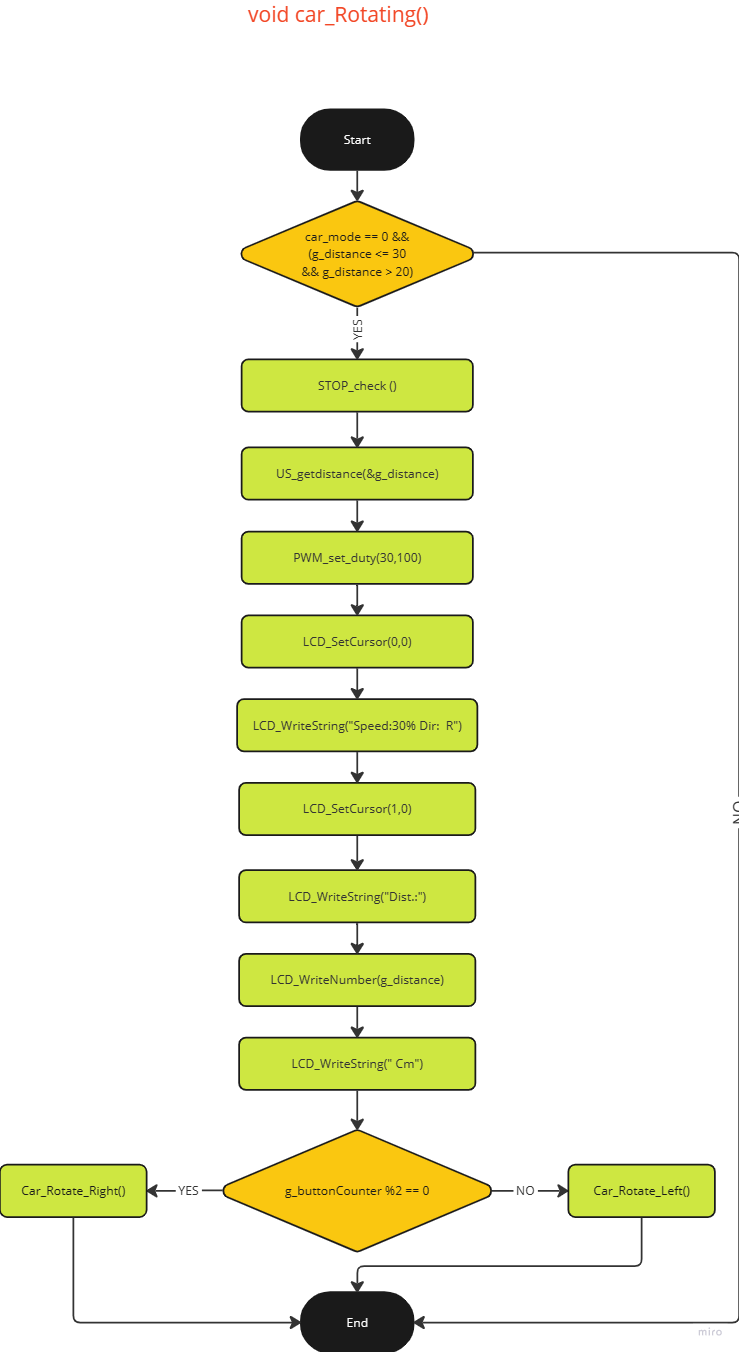
### 3.1. APP

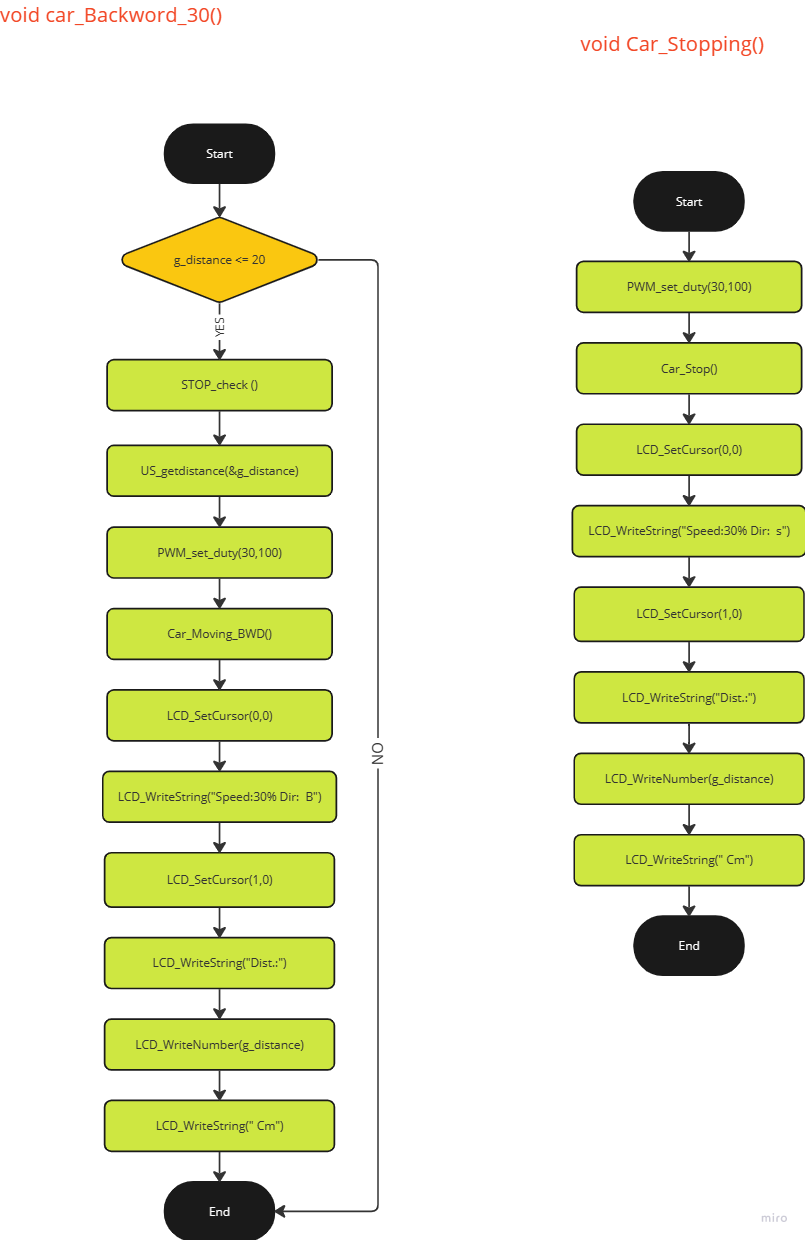
A picture containing text, screenshot, yellow, design

Description automatically generated

A screenshot of a diagram

Description automatically generated with low confidence





A diagram of a flowchart

Description automatically generated with low confidence

A picture containing text, screenshot, design

Description automatically generated

A diagram of a flowchart

Description automatically generated with low confidence

A screenshot of a diagram

Description automatically generated with medium confidence

## 3.2 Precompiling & Linking Configurations

-DIO

A screenshot of a computer code

Description automatically generated with medium confidenceA picture containing text, screenshot

Description automatically generated

A screenshot of a computer program

Description automatically generated with low confidence

-LCD

A screenshot of a computer

Description automatically generated with medium confidence

-Keypad

A screen shot of a computer program

Description automatically generated with low confidence

A screenshot of a computer program

Description automatically generated with low confidence