Cover page

Team Name: Active and High

Team Number
52

Implemented features:

1- **Lane Keeping Assist (LKA)**
2- **Control Indicators (CI)**
3- **Sound System (SS)**

Team members:

Name	ID	Tutorial #	Email
Ahmed Hatem Mahmoud	49-17964	20	Ahmed.semeda@student.guc.edu.eg
Kareem Mohamed Eid	49-19194	14	Kareem.eid@student.guc.edu.eg
Abdelrahman Hussein	49-17771	19	Abdel-rahman.mahmoud@student.guc.edu.eg
Mahmoud Sayed	49-17764	19	mahmoud.sayed@student.guc.edu.eg
Abdelrahman Said	49-18227	19	abdulrahman.said@student.guc.edu.eg

a) Brief description about our project idea and approach:

Modern, autonomous vehicles come with many integrated systems that perform different tasks to implement a set of features. However, not all of these tasks are of the same priority, so it's the System's job to decide which of these features is of more importance and which task to execute first.
In this project, we will be implementing these three features:

● **Lane Keeping Assist (LKA)**:

In this feature, the car keeps moving along one specific lane and when it drifts away from it, an alarm will be fired to alert the driver and the car returns itself back to the lane.

● **Control Indicators (CI)**:

includes 2 features:
1. Current Gear: The user can change the gear using a physical joystick and it reflects on a 7-segment display showing values (0,1,2,3) that represents the values of (P,D,R,N) of an automatic car.
2. Adaptive Headlights: The car adapts its light depending on the reading from a Light Sensor and 3 LEDS.

● **Sound System (SS):**

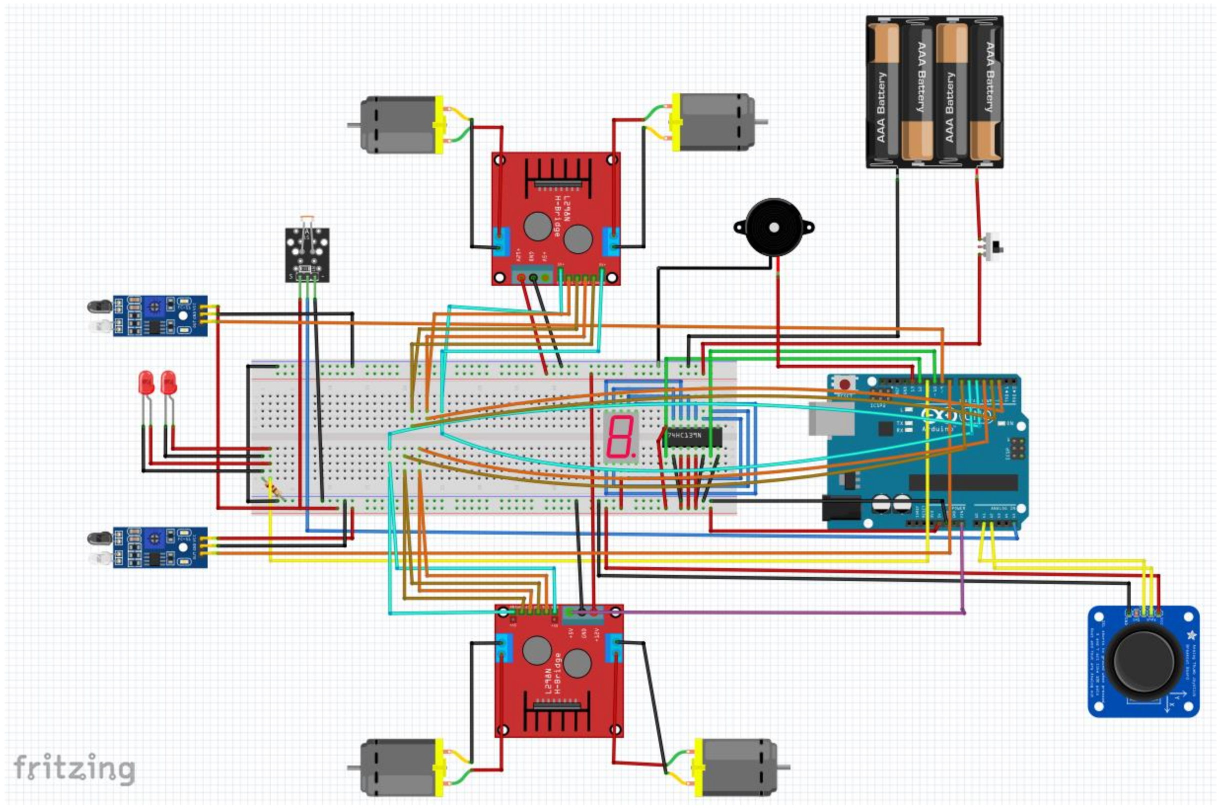
An MP3 module that communicates with the touchscreen so the user can play/stop a song, change and listen to different songs using the previous and back buttons on the touch screen.

b) Components

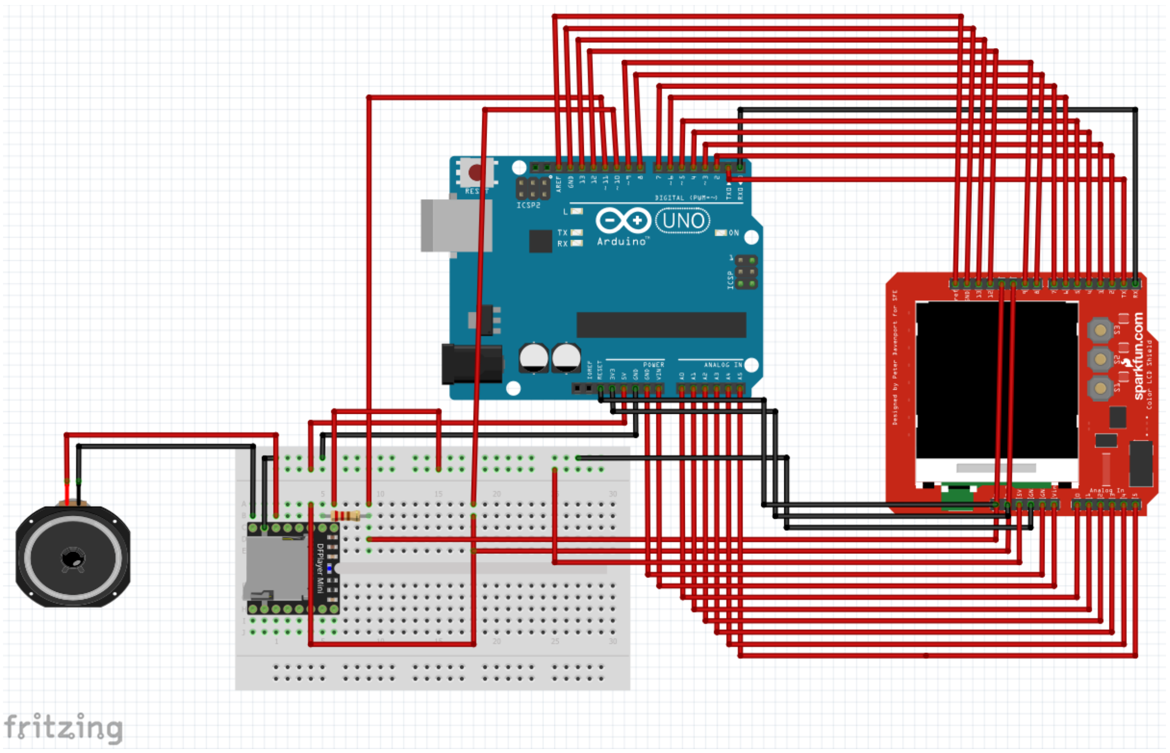
1. 2x Arduino UNO boards: one board is used for control all of the car, gears, and lighting functionalities. The other one is used for controlling the touch screen along with the mp3 module.
2. MP3 module: used for playing songs from the SD card.
3. Speaker: outputs the selected song to be played.
4. LCD TFT 2.4 Touch Screen shield: used for controlling the played songs.
5. 4-wheel car kit: this is the main body of the car along with 4 separate DC motors one for each wheel.
6. L298N H-Bridge Module Motor Driver: used for controlling the speed and direction of the motors.
7. 2x IR sensors: used for capturing the borders of the lane in which the car is moving.
8. 3x 3.7V Li-ion batteries: main source of power.
9. Analog stick: the gearstick that is used for shifting gears.
10. 7-segment display: shows the currently selected gear.
11. SN54LS47 decoder: used for decoding the signal carrying the current gear from the Arduino to be displayed on the 7-segment.
12. LDR sensor: used for capturing the current intensity of surrounding light.
13. LEDs: the headlights of the car.
14. Breadboards: to hold all the components in place.
15. Jumper wires: used for connections.
16. Switch: to turn the car on and off.

c) projects full circuit:

1. features (LKA) and (CI)



2. feature (SS)



d.1) The names of the libraries used:

● For (LKA and CI) We used Arduino_FreeRTOS Library to manage Tasks and Scheduling them.

● For (SS) we used:

1. Adafruit_TFTLCD: To operate the touch screen.
2. Adafruit_GFX: To draw on the touchscreen.
3. SoftwareSerial: To define serial channels for communication between the Arduino and the MP3.
4. DFRobotDFPlayerMini: import the functions of mp3.
5. MCUFRIEND_kbv tft: import functions to create buttons and print on the screen

d.2) The functions used:

1.	touchscreen:	initiate the touch screen
2.	myDFPlayer.volume():	change the volume of the mp3
3.	myDFPlayer.play():	play the mp3
4.	dow.initButton():	initiate button on the screen
5.	dow.drawButton():	doesn’t show the button
6.	tft.fillRect(40, 80, 160, 80, WHITE):	shows a white block given its dimension
7.	tft.println("001.mp3"):	print on the touchscreen
8.	tft.setCursor(85,120):	set the co-ordinates of the text displayed
9.	myDFPlayer.start():	run the mp3
10.	myDFPlayer.pause():	pause the mp3
11.	myDFPlayer.next():	start the next song
12.	myDFPlayer.previous():	start the previous song
13.	dow.justPressed():	check if the button is pressed
14.	displaym(song):	display the songs name

f) Handling inputs:

- IR Sensors: both of the IR sensors used in this project are digital sensors, they return 1 on facing a black surface, and 1 otherwise. The left and right sensors are connected to the 9 and 8 digital pins on the Arduino respectively. Decision on the motors' speed and direction are taken based on the readings of the sensors.
- LDR Sensor: the light sensor is an analog one connected to the A5 pin of the Arduino. It returns values ranging from 0 to 1023, the lower the value the higher the intensity of the surrounding light. These values are used in calculating the brightness of the LEDs.
- Analog Stick: the analog stick moves in 2 directions, X and Y. The center of the stick corresponds to the values (0,0). Moving the stick to the positive or negative directions of the axes will result in increasing or decreasing the returned value of that axis. The values of the X axis are sent through the analog pin A2 on the Arduino, and the Y values are on pin A1. These values are used in choosing the gear to be displayed on the 7-segment.

g) Handling outputs:

Motors: the motors movement is controlled by the values read by the IR sensors, and it's managed by the signals sent to the H-bridge. One H-bridge controls the left wheels, and the other controls the right ones. Each motor has 3 pins on the H-bridge that controls its motion, one PWM that controls the motor's speed, and the other 2 pins control its direction (move forward, move backward, stop). To make the car keep in lane, we check if the left sensor is reading black and the right one is not -meaning the lane is turning right- then we make the left wheels move forward and the right wheels turn backwards in order to make the car turn right, and if the right sensor is reading black and the left one is not -meaning the lane is turning left- then we make the right wheels move forward and the left wheels turn backwards in order to make the car turn left. If both are reading black then we reached the end of the track, and if both are not reading black then it means the car is moving in a straight line and the lane is not turning in any direction so all the wheels are moving forward.

7-Segment: gear selections from the analog stick are reflected on the 7-segment display. The readings of the analog stick are mapped to 4 values (0,1,2,3) that represent the 4 states of gears.
the mapping is as follows:
X<-400 : 0
X>400 : 1
Y<-400: 2
Y>400: 3
these values are binary encoded into 2 bits and sent to the decoder through the 2 digital pins on the Arduino 10 and 12. The decoder handles the job of reading these 2 bits and display their corresponding value on the 7-segment.

LEDs: the LEDs are connected to pin 11 on the Arduino as their brightness will vary according to the readings of the LDR sensor, so they had to be connected to a PWM pin. To make the LEDs glow brighter in darker environments we mapped the range of values read by the LDR sensor (0 - 1023) to the range of values that can be written to a PWM pin (0 - 255). Reading a 0 from the sensor means it's in a fully bright environment so 0 is written to the LEDs and it's turned off, while reading 1023 means it's in a fully dark environment and it maps to 255 to be written to the LEDs and it's glowing with full brightness. All the intermediate values are mapped to their corresponding values along the range.

h) Explain how the features were prioritized and divided into tasks using freeRTOS

- For (LKA and CI) Since we implement the 2 features on the same Arduino, so we divided them into 3 tasks (task1 for LKA, task2 for current gear, task3 for light intensity). LKA has the highest priority, followed by Task2 of the gears, and finally Task3 with the lowest priority.

- All the requirements of the SS features are handled on a standalone Arduino, and this Arduino performs the single task of playing songs on the MP3 module using the Touch Screen, so we created a single task for it.

i) The problems or limitations faced during the implementation of your project.

Two main limitations were faced during this project:

1- The motors are not all of the same health and strength, some motors are stronger and faster than others with the same PWM value given to them both.
solution: the speed of the motors were analyzed and for the slower ones we gave higher PWM values than the other faster ones.

2- The monitor shield would cover all the pins of the Arduino board if mounted on it - even though it does not use all the pins on the Arduino board-, leaving no room for the MP3 module to be connected to the same Arduino.
solution: the screen was alternatively mounted on 2 breadboards with jumper wires leaving more room for the MP3 module.

3- The screen remained white when we tried to put its code in a freeRTOS task, not rendering any of the buttons and the MP3 was completely incontrollable.
solution: the main issue was with the size of the task heap, as it was initialized with 1000, but after some research we found that it should be a smaller value that is a multiple of 2, so a heap size of 256 was chosen and the screen works well now.

j) How did you divide the work among the team members?

Ahmed Hatem:

- Handling the (LKA) features.
- Controlling the LEDs using the LDR sensor.

Abdelrahman Hussein:

- Handling the features of the gears (reading the analog stick and reflecting on the 7-segment).

Kareem Eid:

- Creating tasks using freeRTOS for different features on both Arduinos.

Mahmoud Sayed &
Abdelrahman Said:

- Both shared the requirements of the (SS) features.
- Connected the Touch Screen along with the MP3 module to the Arduino.