Team Name: 2ERD-uino
Team number: #5

Team Members:

Ahmed Hesham Wahba	*T-20*	*49-5423*
Omar Moataz	*T-20*	*49-0359*
Ahmed Reda El-Demery	*T-20*	*49-13168*
Michael Raouf	*T-20*	*49-0831*
Omar Sherif Ali Hassan	*T-20*	*49-3324*

Brief Description

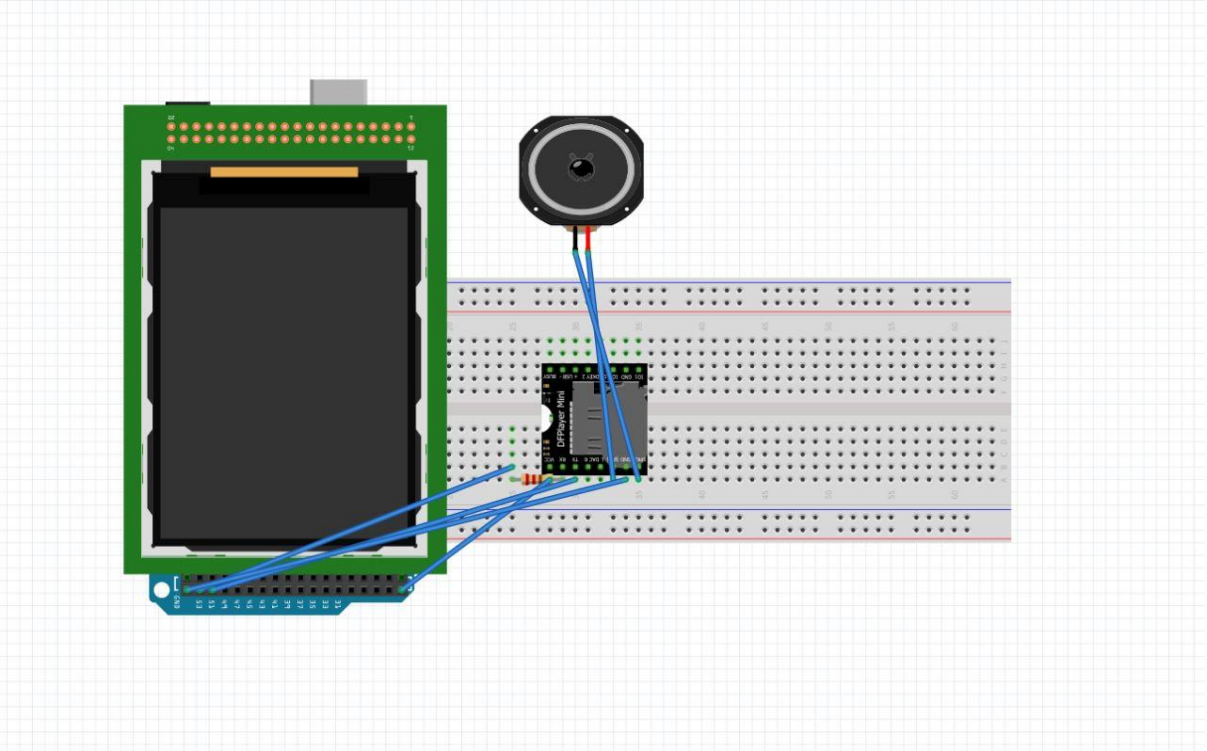
Our project idea is to build a smart car with some of the safety and entertainment features present in the modern day vehicles. We chose to implement a ***Lane Keeping Assist (LKA)*** as the safety feature in our car. The car should move between 2 lanes, similar to real life and the car detects as soon as it departs it's lane and steers gently the car towards the lane while also alerting the driver as soon as the car exits its lane. As our second feature, we implemented control indicators such as changing the gear and reflecting the current gear on a 7 segment display while also monitoring the surrounding light intensity and changing the light intensity of the car headlights accordingly. Last but not least, as an entertainment feature in our car, we chose to implement a sound system, where users can download music on an SD card and play them through an external speaker and control the playback using a touch screen.

Components Used

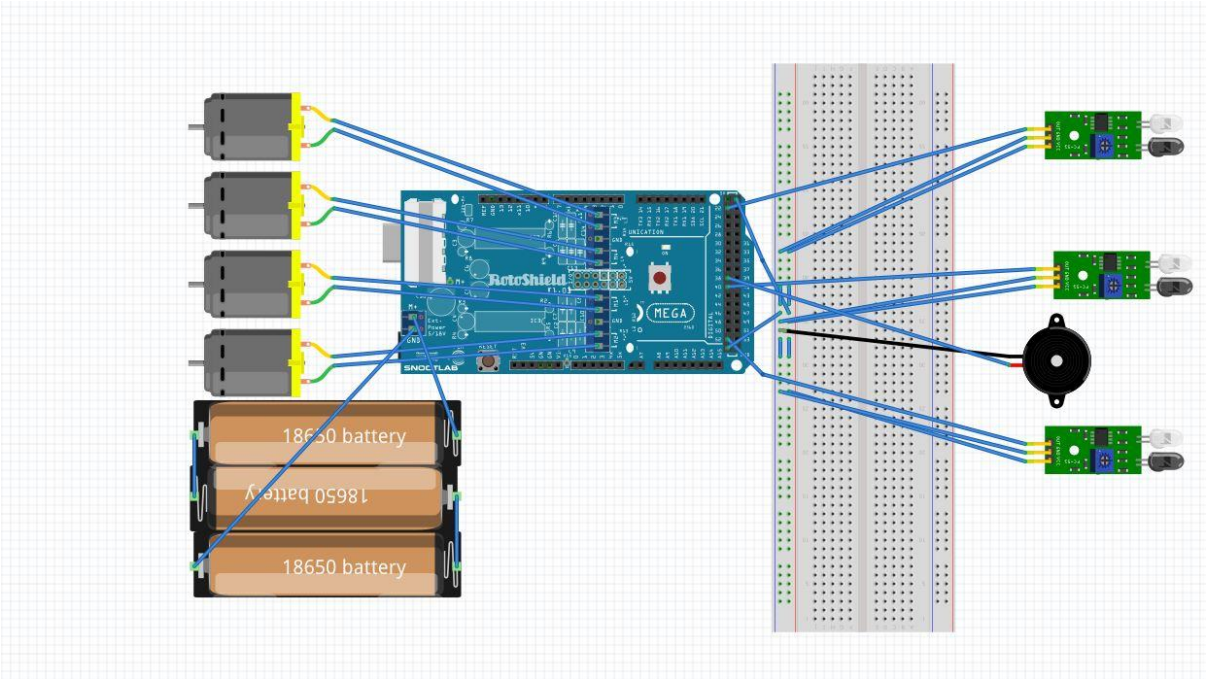
To implement the previously mentioned features, we needed to use several components connected together to achieve the expected functionality. To implement the Lane Keeping Assist, we needed a car chassis to build up our car and 4 dc motors for the wheels to spin in order for the car to move. We also needed a motor driver to be able to control the speed of each of the 4 motors as well as their direction of rotation. Infrared sensors were used to detect if the cart departs out of its lane as we made the lanes using 2 duck tapes on the sides and the Infrared changes reading when it recognizes a black surface, signifying that the car is exiting its designated lane. A buzzer was also used to alert the driver when the car leaves its lane. Secondly, we used a joystick to change the gear from P -> R -> N -> D and the other way around. The current gear is always displayed on a 7 segment display. We used light LEDs to represent the car headlights and based on the reading of the light sensor which reads how much light is in its surrounding and sets intensity accordingly. If the light reads too much light around, the car headlights intensity is decreased as higher intensities would be of no use in such a scenario and vice versa where if there is little to no light around, the LEDs light at full intensity to light up the road ahead. Finally, implementing the car's sound system required the use of a LCD TFT screen to control the music playback and its GUI was designed and implemented using the code, where there exists 3 buttons namely: Previous, Pause/Play and Next. An mp3 module connected to an SD containing the songs were used to play the music as well as an external speaker to hear the actual song playing.

Schematics & Circuits

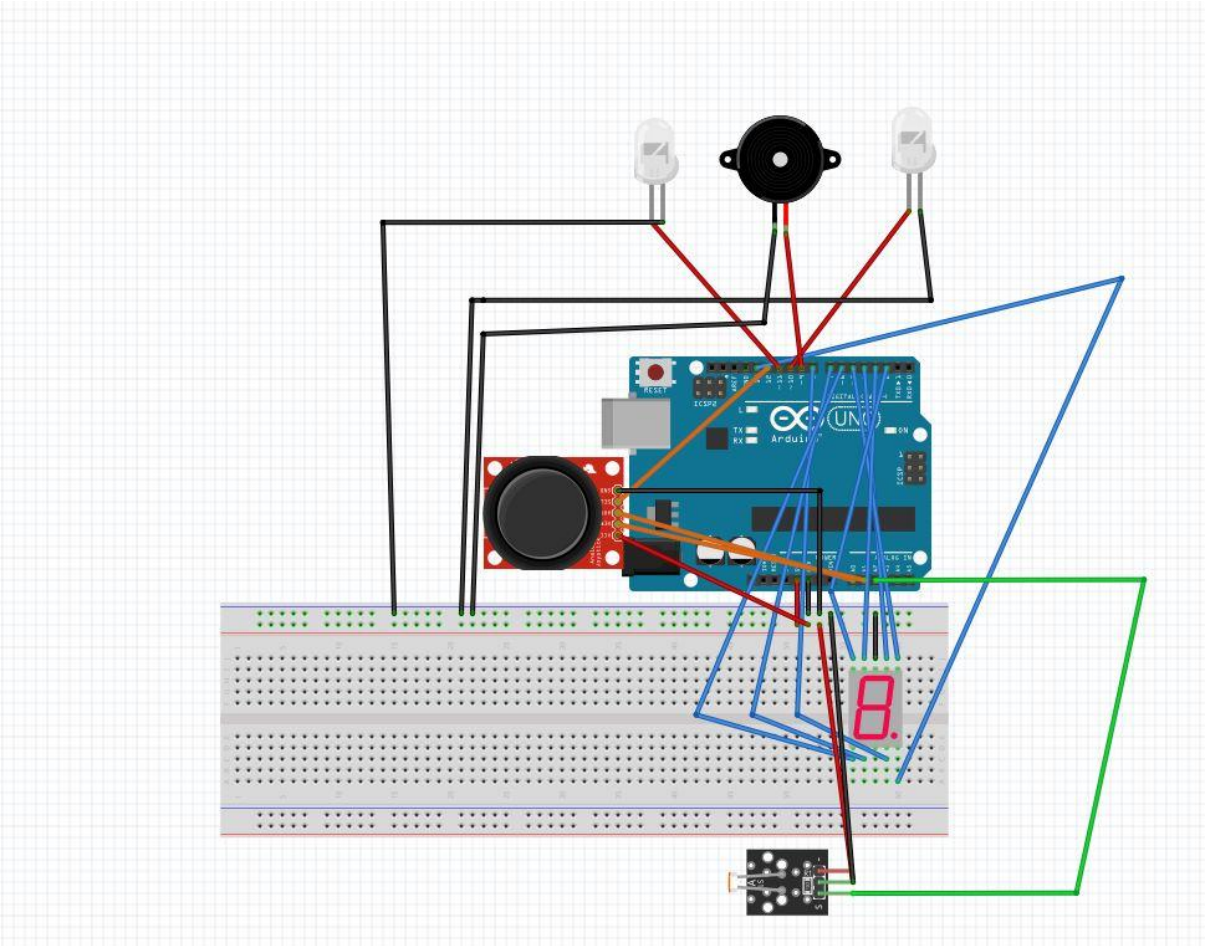
- Sound System Circuit containing MP3 module, external speaker along with the LCD and the arduino UNO connecting all together.



- Lane Keeping Assist Schematic where there exists the 12V batteries, the 4 DC Motors, the 3 Infrareds used, the buzzer used to alert the driver as well as the arduino MEGA.



- Control Indicators Circuit Diagram where shown below is the Joystick, the 2 LEDS representing headlights, the buzzer, the LDR sensor, the 7 segment display and the arduino board used to connect all these components together.



Libraries

- AF_Motor library: was used to communicate with the motor shield/driver to be able to control the speed of each of the 4 motors as well as their direction of rotation.

- Adafruit_GFX, MCUFRIEND_kbv.h library: were used to draw the screen GUI and place the 3 music control buttons as well as the song number currently playing.

- TouchScreen.h library: for buttons to be interactive and clickable and determine where the user touched the screen to implement expected logic accordingly.

- SoftwareSerial.h library: to be able to communicate with the mp3 module and play the songs present on the SD card, besides change the music playback according to the buttons clicked.

- Arduino_FreeRTOS.h library: to be able to schedule standalone tasks in the arduino according to their priority where a round robin algorithm is used when tasks of the same priority exist.

Inputs & Outputs

- Input sensors were connected to the arduino 5V pin and ground pins along with the pin where the signal is read. This input is different from one sensor to the other as some needed to be connected to an analog pin like the Joystick used to change the car gear and the light sensor sensing the light intensity in its surrounding environment. Other sensors like Infrared were connected to a digital pin while some other input components were complex and required several pins on the arduino board to be functioning like the MP3 module and LCD.

- Speaking of the Sound System connections, the MP3 module needed its Tx and Rx connected to 2 digital pins on the arduino besides the ground. 2 of its pins (SPKR 1 & SPKR 2) were connected to the external speaker. The LCD was connected to a full arduino UNO consuming almost all its pins.

- To implement the control indicators features, Joystick had 5 pins, 2 of which are 5V and Ground and 2 were connected to analog pins to know the direction of movement occurred on the joystick as they record the x and y values in this direction and an SW pin where it's responsible for clicking the button in the Joystick. The 7 segment had 10 pins where one pin was connected to the GND and 8 other pins were connected to 8 digital pins on the arduino. The LDR sensor was connected to 5V, GND and to an analog pin. Each of the car headlights were represented using 2 light LEDs that had their positive ends connected to 5V and their negative ends to the arduino's GND.

- The motor shield/driver had 2 pins connected to both terminals of the 12V batteries holder. Also, the shield was connected to an arduino MEGA consuming almost all its pins similar to the LCD connection to the arduino UNO. 3 Infrared sensors were used to detect if the car moves out of its lane. Each IR sensor had 3 connections: 5V, GND and a digital pin on the arduino. The buzzer used to alert the driver was connected also to an output digital pin and a GND pin on the arduino.

RTOS

- *In Arduino Uno :*
    - Task 1 : 7 segment display and the gear had a priority of **3**.
    - Task 2 : Light Sensor Detection and the 2 bulbs had a priority of **2**.
    - Task 1 had a higher priority (the higher the number assigned, the higher the priority) because the gear in real life controls the car movement, which is much more important than the car headlights. It was given a small periodic time to be able to reflect much faster and Task 2 was scheduled periodically as the light sensor detects when it is dim around and lightens the LEDs representing the car headlights and if the environment is bright, it shuts down the LEDs.

- *In Arduino Mega 1 :*
    - Task 1 : LCD with MP3 Module (it was the only task running in the arduino) so no scheduling was needed in this case because there were no competitors as a result.

- *In Arduino Mega 2 :*
    - Task 1: The Car Movement (it was the only task running in the arduino) so no scheduling was needed in this case because there weren't any competitors but it needed delays when IR Sensors detects the lanes and then the positioning adjusting needed some delay so that the car takes its time for adjusting its position properly and while fixing its position the buzzer buzzes to alert the driver that adjustments are occuring at the moment.

Problems & Limitations

1. One of the top problems faced was the high cost of all the electrical components needed to be bought, especially after the last economical crisis, the prices skyrocketed and a load of money was spent to compile all the needed components.

2. The second limitation was the unavailability of a plethora of the components needed and we had to gather the items needed from over 5 places, both online and offline.

3. Too many sources of error are a huge obstacle when unexpected behaviour happens during debugging the cause of the error leading to this result. Sources of errors for a single problem may include jumper wires connecting components together, breadboard, logical error in the code, switching 5V and GND, USB cable connecting arduino to PC to upload code, arduinos having pins not working in the 1st place, only to name a few!

Work Distribution

- We all worked equally in the control indicators features.

- 2 of our team members namely Ahmed Reda and Michael Raouf worked on the Lane Keeping Assist where they were responsible for connecting batteries, motor shield and dc motors together and writing the movement logic of the motors.

- 3 other members namely Ahmed Hesham, Omar Moataz and Omar Sherif were responsible for the sound system feature where they had to connect the MP3 module, external speaker and LCD screen together.

- Omar Sherif was assigned the RTOS of the car and how to schedule and assign priorities to the tasks wanting to execute.