Project Report

CSEN701: EMBEDDED SYSTEM ARCHITECTURE

WINTER SEMESTER 2022 PROJECT

ARDUINO CONTROLLED CAR

Team Members:

- **Omar Sherif El-Meteny/49-3317/T17**
- **Ziad Tamer/49-1866/T9**
- **Amr Mohamed Samy/49-6656/T17**
- **Rana Wally/49-0534/T17**
- **Abdelrahman Mohamed Mahmoud Aboelmaaty Abdo/49-4016/T18**

Project Description:

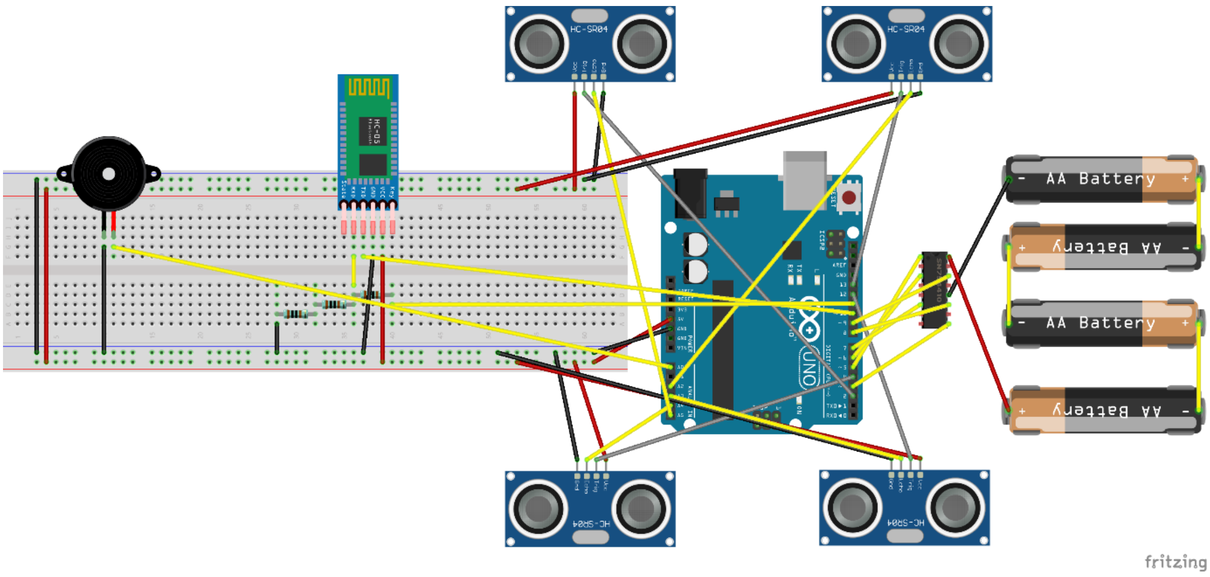
In this project, some of the common car features were implemented using **Arduino Mega** and **Arduino Uno** boards and **FreeRTOS** library. These features are autonomous car-maneuvering system that performs parallel parking using sensors, displaying the car's current gear on a 7-Segment display, using adaptive Headlights to adapt the car's headlights according to the surrounding light intensity, and finally playing MP3 files via a speaker on the car and system should be controlled using LCD-TFT touch screen. The project was divided into 2 sub-projects, one using Arduino Uno board and the other using Arduino Mega. The Arduino Uno board was used for the car and the Arduino Mega was used for the controlling the car. Bluetooth modules were used as a communication port between the two Arduino boards.

Components:

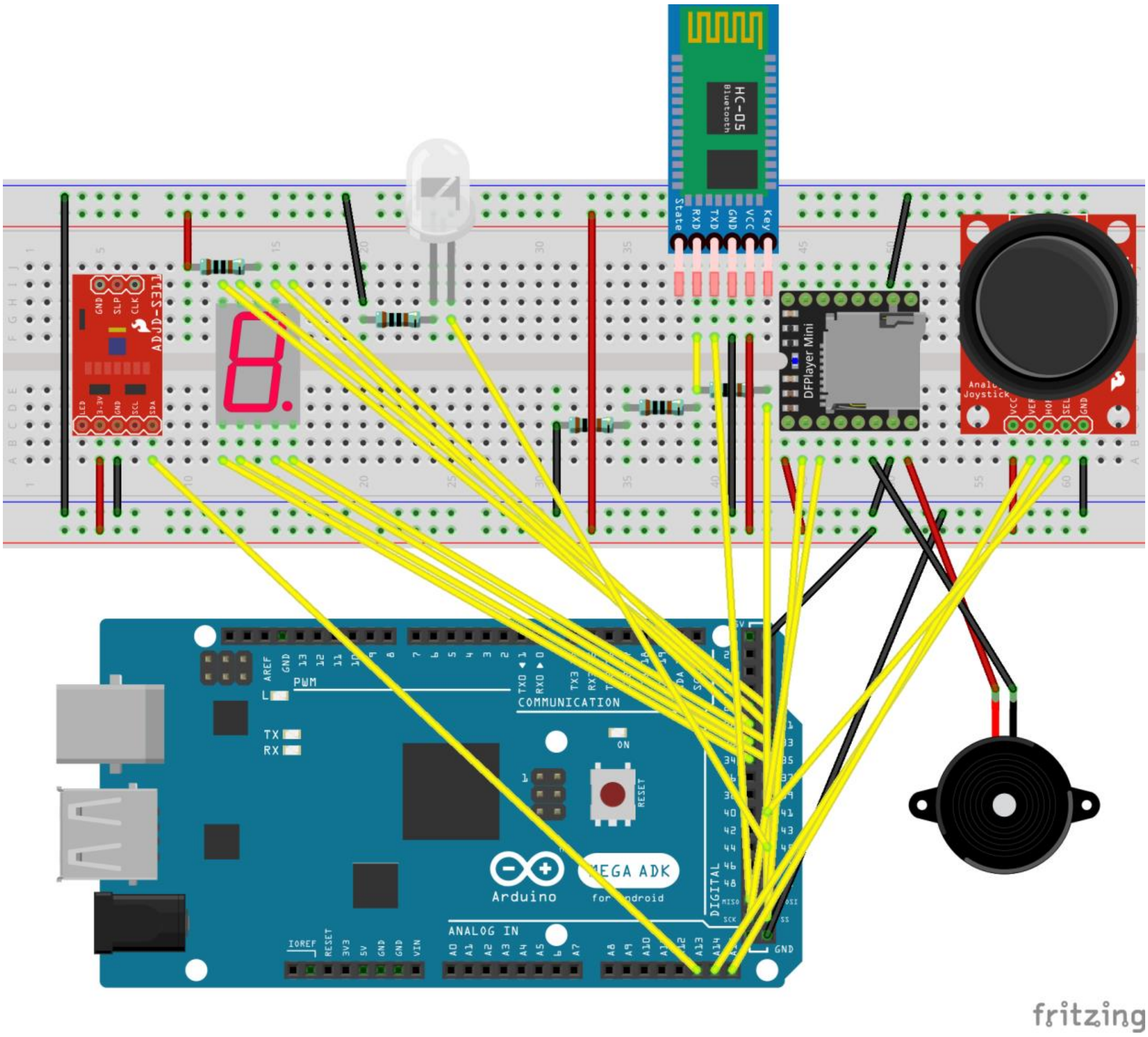
1. **Arduino Uno Board** (used to supply power components on the breadboard and implement functionalities related to the car).
2. **Arduino Mega Board** (used to supply power for the components that are responsible for controlling the car and other components and implement their functionalities).
3. **H-Bridge** (used to control the car's motors by sending the correct signal for each motor according to the functionality needed).
4. **4X Ultrasonic sensors** (used to detect the distance between the car and the surrounding obstacles).
5. **2X HC-05** (used as a communication port between the 2 Arduino Boards).
6. **2X Breadboards** (used to place the components on them and supply power and signals for all components).
7. **Joystick module** (used to control the car's current gear and the car's movement).
8. **MP3 module** (used to play MP3 files downloaded on SD-Card).
9. **LDR sensor** (used to detect light intensity surrounding the car).
10. **0.25W-32Ohm Speaker** (used for listening to the songs played by the MP3 module).
11. **1 Transparent LED** (used as an Adaptive Headlight according to the surrounding light intensity).
12. **7-Segment display** (used to display the car's current gear).
13. **LCD-TFT** (used to play/pause the song, get the next or previous songs, and finally displays the name of the current song).
14. **Several 1KOhm resistors** (used for LEDs and 7-Segment and Bluetooth modules).
15. **9V batteries** (used to supply extra power for the Arduino Boards).
16. **Buzzer** (used as an audible warning and its tone changes according to the distance between the car and the surrounding obstacles).
17. **Jumper wires** (used for pin assignments).
18. **RC-Car** (used to simulate the project).

Project Circuits:

Car Sketch:



Car Remote Sketch:



Project Libraries:

1. **SoftwareSerial** (used to establish serial communication between the Arduinos and the Bluetooth modules and to send commands from the Arduino Mega to the MP3 Player module).
2. **Arduino_FreeRTOS** (used to create tasks for the different features and scheduling those tasks).
3. **DFRobotDFPlayerMini** (used to control the MP3 Player module).
4. **MCUFRIEND_kbv** (used to draw shapes and write text on the LCD-TFT screen).
5. **Adafruit_GFX** (this is a dependency needed by MCUFRIEND_kbv library).
6. **TouchScreen** (used to handle touch screen inputs).

Inputs:

Inputs were taken from the components and assigned to digital or analog input pins of the Arduino Board. This was done by defining new variables and assigning the pins to those variables and those variables were set to inputs in the setup function. Handling those inputs was done by if-else statements and according to the values a certain output was returned.

Outputs:

Outputs were produced by the Arduino and were taken from digital output pins or PWM pins. This was done by defining new variables and assigning the pins to those variables and those variables were set to outputs in the setup function. Handling outputs was done in the FreeRTOS tasks and changed according to the inputs produced by the different components. The outputs were changed by setting the variables as HIGH or LOW if the output was digital or assigning values to the variables from 0 to 255 if PWM was used.

FreeRTOS:

Car Remote Tasks:

1. **Input Task** (used to handle inputs produced from Joystick, Touch Screen and Bluetooth).
2. **7-Segment Task** (used to display the change in the car's current gear).
3. **MP3 Player Task** (used to switch between the songs played by the MP3 Player according to touch screen inputs).
4. **Touch Screen Task** (used to update the screen according to current song).

The Input Task has the lowest priority because the rest of the tasks are sleeping most of the time waiting for their inputs. The other tasks have higher priority than that of the Input Task because they must execute once they are notified that an input is produced before processing more inputs. The Touch Screen Task has a priority lower than the 7-Segment and the MP3 Player Tasks because it takes more time in execution.

Car Tasks:

1. **Buzzer Task** (used to produce tones and changes according to the distance between the car and the surrounding obstacles).
2. **Sensor Task** (used to detect distances between the car and the surrounding obstacles).

3. **Car Control Task** (used to control the car according to the car's current gear with different speeds (**PWM)**).
4. **Bluetooth Task** (used to receive inputs from the Arduino Mega Board and handles those inputs. Moreover, it is used for autonomous parallel parking).

The Bluetooth task has the lowest priority because the rest of the task are sleeping most of the time waiting for their inputs. The other task have higher priority than that of Bluetooth Task because they must execute once they are notified that an input is produced before processing more inputs. The Sensor Task has a priority lower than the Car Control and Buzzer Task because it is considered an input to the Buzzer Task.

Project Problems and Limitations:

The first problem encountered in this project was the code implementation of the LCD-TFT and the Joystick because they were implemented in separate tasks. The problem was that both modules were connected to analog input pins and the Arduino was trying to read the analog values produced by both modules at the same time and this caused problems in the readings. This problem was solved by using a mutex for analog reading so that the Arduino cannot read analog inputs from both modules at the same time. Unfortunately, this approach caused another problem which was in responsiveness. The tasks that take the produced inputs didn't respond with proper timing. At the end, both tasks were implemented in a single task which is the Input Task mentioned above and any other task that was used to read inputs was implemented in that same task (Input Task).

The second problem was in the two libraries used for the implementation of LCD-TFT functionality. MCUFRIEND_kbv library was responsible for reading inputs from specific pins and TouchScreen library was responsible for producing the outputs using the same pins so there was a race condition between the two operations. To solve this problem, a Mutex was used to prevent these 2 operations from executing at the same time.

Another limitation was that, in the library responsible for MP3 functionality, there were no APIs to get the name of the current song, so the song names had to be hardcoded to show them on the LCD-TFT screen.

Team Contribution:

First of all, it was a collaborative team effort since all of us worked together to implement the project code and hardware. To be more specific, each one of us was responsible of a specific part.

1. **Omar Sherif El-Meteny** (implemented car remote code and Bluetooth communication).
2. **Ziad Tamer** (made car remote circuit).
3. **Amr Mohamed Samy** (assembled the car a connected the sensors and buzzer to the car).
4. **Rana Wally** (implemented sensor and buzzer code and car movement code).
5. **Abdelrahman Mohamed Mahmoud Aboelmaaty Abdo** (implemented autonomous parking code).