Embedded Systems Project Report

Team 32 (SHYNK):

1. Name: Nada Abdel Rahman Hegazy
   ID: 49-1384
   Email: nada.mohamedhegazy@student.guc.edu.eg
2. Name: Farida Tarek Kassem
   ID: 49-1122
   Email: farida.kassem@student.guc.edu.eg
3. Name: Yasmine Khaled El Serafy
   ID: 49-1224
   Email: yasmine.elserafy@student.guc.edu.eg
4. Name: Salma Ihab
   ID:49-0480
   Email: salma.ihab@student.guc.edu.eg
5. Name: Hana Mohamed Saeed
   ID: 49-2127
   Email: hana.abdelhakim@student.guc.edu.eg

Project idea:

The project idea is to assemble a car with 3 features:

- lane keeping assist
- Control Indicators
- Radio System

Those features are to be created using FreeRtos by creating tasks with priorities. Our approach was to create 3 tasks one for the lane, one for the control and one for the radio. The Arduino Uno board had the l293d motor shield on it as well as 2 Line Sensors and the led used for alerting the driver. This task is continuous and separate than the other 2 tasks meaning that it is always running, we thought that the car should always be running no matter what.

On the Arduino Mega Board we had the TFT LCD touch screen to control the radio, the radio module, 2 LEDs, Light dependent resistor sensor, Joystick and 7 segment display to display the gear.

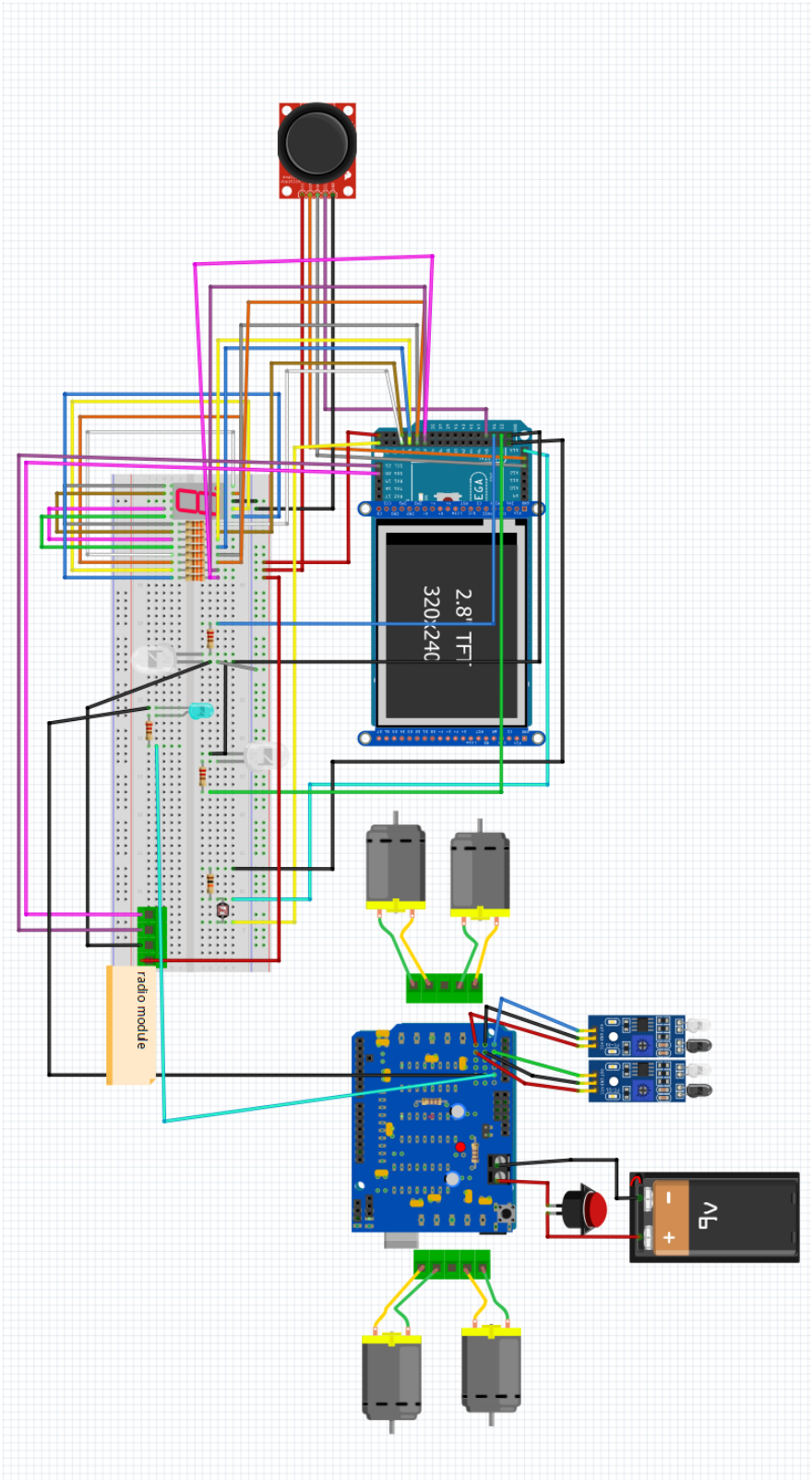
We divided those to 2 periodic tasks one for control and one for the radio and touch screen with the higher priority going to the control however the tasks sleep for a very short period of time so it actually looks as if both tasks are running at the same time

Components and their functionalities:

- 2 Lane Sensors: the sensors detect when there is a black line (lane) so that if both sensors don't detect a line all 4 motors will run forward, and if the left sensor detects a black line then the left motors continue to run forward while the right motors run backward with a much greater speed in order to steer the car back into the lane, and if the right sensor detects a black line then the right motors will run forward while the left motors run backward with a much greater speed, and if both sensors detect a line then all motors are supposed to stop.

- Blue LED: The LED is driven high when either one of the sensors or both detect a black line so that the driver is alerted that the car is being steered back to the lane
- TEA5767 FM Radio Module: radio broadcast receiver module that is based on the TEA5767 IC product. The module is equipped with adjustable radio antenna and can be connected to speaker (we connected ours to headphones). It uses frequency modulation or FM signal with frequency range from 70 MHZ to 108 MHZ where you can tune the radio using software to select the receiving frequency.
- TFT LCD touch screen: A touch screen to turn the radio on/off and change the radio station.
- Joystick: A joystick to control the gear displayed on the 7 segment display where up represents P, down represents N, right represents D, left represents R
- 7 Segment display: Displays the gear decided by the joystick
- LDR sensor: inputs analog value based on the light intensity the sensor is exposed to
- 2 RED LEDS: based on the LDR sensor value, if it is under 100 turn on both LEDs, if it is under 300, turn on 1 LED otherwise, turn off both LEDs
- L293d Motor shield: A shield for Arduino Uno board to control the 4 DC Motors with external power supply sourced from 3 lithium batteries. It holds the 2 Line sensors and the blue led as well
- 4 DC Motors to run the car wheels forwards and backwards
- 2 Arduino Boards: connected to all sensors ,actuators and shields, programmed with the logic to handle them to run the car correctly

Circuit Diagram:



We couldn't find in fritzing any radio modules so we replaced it with an IC with 4 connections since the radio module had 4 connections (VCC, GND, SDA, SCL) and those connections are represented in the diagram.

Libraries:

- **Arduino_FreeRTOS.h:** Operating system to create and manage tasks on the microcontroller
- **Adafruit_GFX.h:** To create graphics on the LCD screen
- **Adafruit_TFTLCD.h:** Library for the LCD screen
- **TouchScreen.h:** Library for the touchscreen
- **Wire.h:** I2C Library as the radio uses the I2C Bus
- **radio.h:** Library for the radio
- **MCUFRIEND_kbv.h:** Library to control the TFT LCD screen
- **TEA5767.h:** Library for the Tea5767 radio to control it
- **semphr.h:** Binary semaphore library for the 2 tasks
- **AFMotor.h:** Library to control the 4 DC Motors

Inputs:

For inputs we had the 2 Lane sensors to detect the black lines of the lanes when it detects a lane it inputs 1 otherwise it inputs 0.

We have 2 input analog pins as well for the X and Y values of the joystick and based on their values we detect whether it's right, up, down or left and therefore P,R,N,D

And an analog input pin for the LDR sensor and based on its value we light up the LEDs accordingly

Outputs:

For outputs we had the Blue LED as output that lights up (driven high) when any of the Lane sensors inputs 1 as well as the 4 DC Motors that run according to the Lane sensors. If both sensors read 0 then all 4 motors run forward with speed 80, if one sensor reads 1 and one 0 then the motors on the 1 side run forward with speed 80 and the motors on the 0 side run backwards with speed 200

We had 2 Red LEDs as outputs depending on the analog value from the LDR sensor. If the LDR reads a value less than 100 then both LEDs are driven high, if the value is less than 300 then only 1 is driven high and the other low otherwise, both LEDs are driven low. The 7 segment display is based on the X and Y values from the joystick input to display P R N D.

In addition, there is the output sound from the radio module

Tasks:

As stated in our approach we decided that the all the lane related sensors and actuators (Arduino UNO board) are to be given the highest priority by placing them in a continuous task that is unrelated to the other 2 features.

On the MEGA board there were the radio and the control features specified in 2 periodic tasks we gave the higher priority to the control task and specified the tasks sleeping time to be 80 ms for the radio and 100 ms for the control, therefore it gives the impression of parallelism

Limitations:

We had some limitations throughout our project assembling time, firstly we had a problem with the soldering of the wires for some reason, some of our wires kept falling out which caused a lot of time being wasted going back and forth to solder. Then we had a LDR module that didn't work at all and we couldn't figure out why so we ended up getting a LDR sensor with a 10k resistor instead which worked perfectly fine. Also the 7 segment display took a lot of time and trials and errors in order to finally find a connection that worked where we used 8 330 ohm resistors for each segment. Then we had a big power problem at first our only power source for the car and Arduino UNO was our 3 lithium batteries and the car and lane sensors were working just fine until they were not, suddenly everything stopped working and we thought the sensors burnt out and we got new ones however after the new ones also didn't work and after consultation with the TAs it turned out to be a power

problem, apparently the batteries should only power the 4 car motors and the boards need another power source (power bank) thankfully that solved the problem and everything else worked smoothly from that point forward with the exception of the lane keeping assist sometimes failing to steer back the car on time.

Division of Work:

1. Nada: LKA, radio and scheduling
2. Hana: LKA and scheduling
3. Yasmine: LDR sensor and corresponding LEDs
4. Salma & Farida: Joystick ,7-segment display and Touch Screen