CSUN | COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

ECE 595RL

Autonomous Vehicle using MSP432

Final Project

Group 8
Nethania Morales
Shantanu Telrandhe
Instructor: Professor Shahnam Mirzaei
Aaron Nanas
05/13/2024
Spring 2024

Introduction:

Autonomous vehicles, also known as self-driving or driverless vehicles, are revolutionizing the transportation industry. These advanced cars use a combination of sensors, software, and hardware to navigate roads, detect obstacles, and make decisions without human intervention. This project achieved the ability to perform such tasks without human interference and respond to the current situation in real time. We used concepts like PWM (Pulse Width Modulation), timers, time delays, interrupts and the UART to accomplish our goal. This project consists of four major functions to be performed in various real-time conditions:

1. Localization: The robot keeps moving straight until it detects an obstacle in front of it. It will change its speed depending on the distance between it and the obstacle.
2. Traffic light/signal detection: The PMOD COLOR Sensor detects the traffic lights colors, and the robot acts accordingly.
3. Overtake: The robot overtakes once the front obstacle doesn't move for a set amount of time.
4. Collision detection and handling: Handle_collision() is a function that gets activated once the bumpers switches are pressed and STOPS the robot wherever it is. An SOS alarm is activated through the Piezo Buzzers.

Components Used:

The table below shows a list of all the components that were used for the production of this final project. The table includes a basic description of each component, as well as the quantity of each component and their manufacturer.

Description	Quantity	Manufacturer
MSP432 LaunchPad	1	Texas Instruments
USB-A to Micro USB Cable	1	N/A
TI-RSLK MAX Chassis	1	Pololu
PMOD COLOR	1	Digilent
Sharp GP2Y0A21YK0F Analog Distance Sensor 10-80cm	3	Pololu
Bumper Switches	6	Pololu
Piezo Buzzer	1	TDK

Pins Used:

The tables below show how each component was wired to the main microcontroller, the MSP432 LaunchPad. There is a detailed description showing what pins the components are connected to.

Right GP2Y0A21YK0F Analog Sensor	MSP432 LaunchPad Pin
Vout (White Wire)	P9.0 (A17)
VCC (Red Wire)	VCC (5V)
GND (Black Wire)	GND

Left GP2Y0A21YK0F Analog Sensor	MSP432 LaunchPad Pin
Vout (White Wire)	P9.1 (A16)
VCC (Red Wire)	VCC (5V)
GND (Black Wire)	GND

Center GP2Y0A21YK0F Analog Sensor	MSP432 LaunchPad Pin
Vout (White Wire)	P6.1 (A14)
VCC (Red Wire)	VCC (5V)
GND (Black Wire)	GND

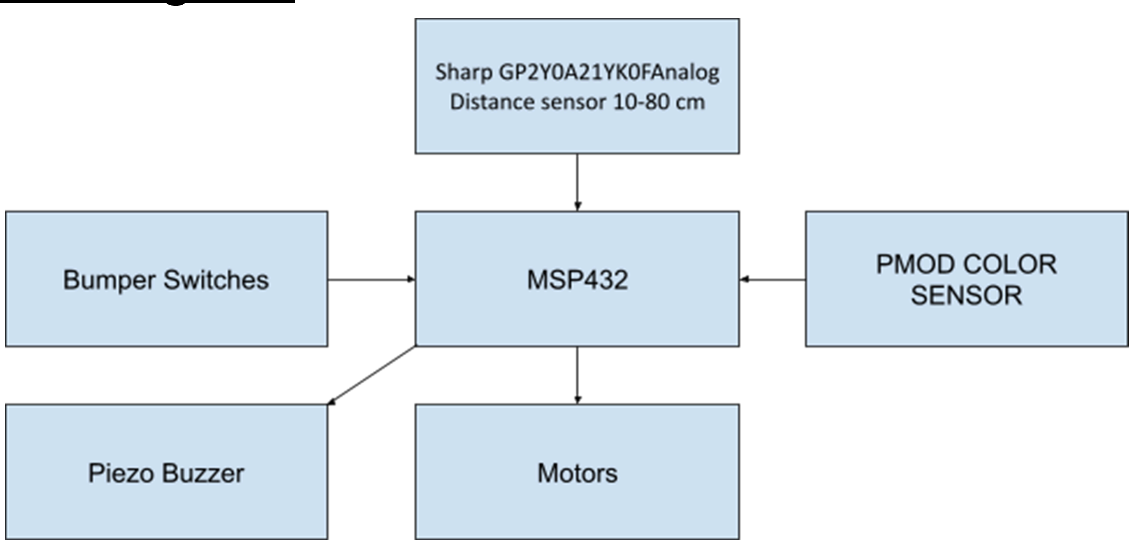
PMOD COLOR Pin Name	MSP432 LaunchPad Pin
IO1 / ~INT Pin 1	Not Connected
IO2 / LED_EN Pin 2	P8.3
SCL Pin 3	P6.5 (SCL)
SDA Pin 4	P6.4 (SDA)
GND	GND
VCC	VCC (3.3V)

Left Bumper Switch	MSP432 LaunchPad Pin
GND	GND
BMP0	P4.0
BMP1	P4.2
BMP2	P4.3

Right Bumper Switch	MSP432 LaunchPad Pin
BMP3	P4.5
BMP4	P4.6
BMP5	P4.7
GND	GND

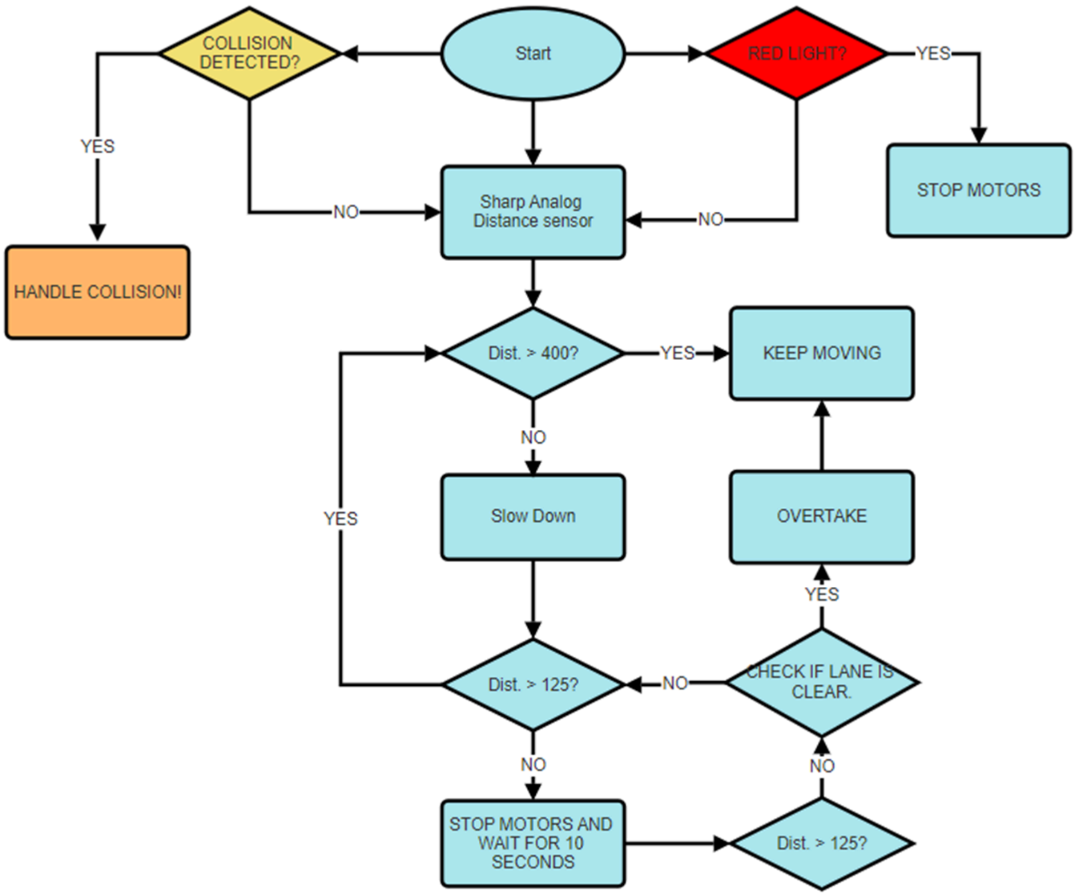
Piezo Buzzer	MSP432 LaunchPad Pin
VCC	P1.6
GND	GND

Block Diagram:



Flowchart:

Below is a basic flowchart of how our algorithm works. This flowchart allowed the team to conceptualize the basis of the project.



Analysis and Results:

To start, the general idea of this project was to mimic an "autonomous/unmanned car" using all of the components listed above. In order to do this, some rules had to be established. The first function to be discussed in the Lane_Controller() function. With the use of the analog distance sensors, if the robot detected that there were no objects (i.e. people, cars, animals, etc.) in front of it, then the robot would move forward at full speed. The Motor_Forward() function was used to set the PWM signals for both motors at a value of 3500. Next was tackling the issue of the robot getting in range of cars. If the robot sensed an object or "car" at a distance of 500mm, the robot would begin to slow down. The Move_Forward function was used to update the motor PWM signals to a value of 1500. If the robot detected an object in a range of 200mm, then the robot's motors came to a stop using the Motor_Stop() function. If for whatever reason a car starts to back up towards the robot and comes into a range of 100mm, the robot slowly moves backward.

Next is discussing the Overtake() function. The idea was that if the robot is stuck behind a "car" for a certain amount of time, the robot will go around the "car" or object. Inside the Lane_Controller() function, the robot will wait for a certain amount of time after the motors have stopped. For the purpose of displaying this action, the amount of time used in the code was 5000ms. The robot will then check if there is still a car in front of it at a distance of 200mm or less. If not, the robot will move forward. If there is a car still in front of the robot, the Overtake() function is called. This function handles the action of going around said object. The first check that this function does is it again checks if there is a car in front of it. This is just an extra measure for better accuracy. If so, it calls the Lane_Controller() function again. If there is still a car in front of the robot, then it starts the process of overtaking. The robot turns about 90 degrees to the left and then stops. It then uses the analog sensors to detect if there is an object in front of it. If there is an object next to it (or a car in the adjacent lane), the robot will turn 90 degrees to the right. This process will continue until there is no car in front of the robot or no car on the left side of the robot. Once there was no car in the adjacent lane, various motor functions were used to go around the car. Before merging into its original lane, the robot will use the analog sensors to check if there is a car or object to its right. If there is no object or car detected, then the robot will merge to its original lane and move forward. However, if there is a car or object detected where the robot wants to merge, the robot will turn back to its original position and just stay in the same lane. This entire process is a continuous loop.

Next is to talk about how the PMOD COLOR sensor and module was implemented into this project. This sensor can detect red, green, blue, and white. The PMOD_Controller() function is used to read the data from the sensor. For this project, a green light represents go and the red light represents stop. For the purpose of simplicity, the LED on the robot (LED2) was used to simulate a traffic light. Inside the function, if a red light is detected, the robot stops. If a green light is detected, the robot calls the Lane_Controller() function.

Lastly was the implementation of collision detection. Bumper switches were used to prevent collision. At any point in time where the bumper switches were hit, the robot would stop and play a sound through the piezo buzzer. This was accomplished using the SysTick timer and handler.

Conclusion:

In conclusion, we were able to achieve all the desired outcomes for this project. Every function in the robot was fully functional and worked perfectly. We were able to learn many new concepts through the course of the project. In summary, this project successfully showcased the potential of autonomous vehicles powered by the MSP432 microcontroller. By leveraging concepts like PWM, timers, interrupts, and UART communication, we have demonstrated the capability to navigate roads, detect obstacles, interpret traffic signals, overtake slow-moving obstacles, and handle collisions autonomously.The future applications of this technology are vast and promising. Through enhancement within delivery services to serving all types of industries (agriculture, etc.), autonomous vehicles/robots can offer endless opportunities. This project showcases the potential this robot can have in offering lots of solutions for different applications. With further development of autonomous robots, they can become an important part of our society. This project has shown how configurable and flexible it can be. tThe MSP432 LaunchPad and all components used are just a starting point, imagine the greater possibilities with unlimited access to resources. This project has built a great foundation that can be built upon in the future to further improve how autonomous vehicles/robots work.