International Journal of Research in Engineering, Science and Management
Volume-3, Issue-6, June-2020
www.ijresm.com | ISSN (Online): 2581-5792

Integrated Digital Driving System Using CAN Protocol

K. B. Kaveri[1], K. R. Rakesh Aradhya[2*], S. Shilpa[3], M. Y. Punith[4], Shivaraj Mallappa Gadyal[5]
[1]*Associate Professor, Department of Electrical and Electronics Engineering, Vidya Vikas Institute of Engineering and Technology, Mysore, India*
[2,3,4,5]*Student, Department of Electrical and Electronics Engineering, Vidya Vikas Institute of Engineering and Technology, Mysore, India*
*Corresponding author: rakesharadhya4646@gmail.com

***Abstract*: Over the past few years, many recognition algorithms proposed to assist in the area of driver safety, but very few of them are both accurate and fast enough for real-time processing. This project aims to develop algorithms for high accuracy and real-time performance in the area of pedestrian detection, beam control, drowsy driver, fuel monitoring, temperature monitoring, obstacle detection and gas detection from an automobile. Pedestrian recognition is one of the most challenging problems. There have been many recognition algorithms proposed for purposes such as prevention of traffic accidents by using vehicle cameras. Consequently, engineers and planners have started to implement new road treatment and intersection designs with the aim of reducing the risk of vehicle-pedestrian crashes. Roundabouts are one such treatment that has gained considerable attention among planners and engineers. This project introduces an effort to design and implement such real-time oriented algorithms that are highly adaptive to the road and traffic scenes based on domain-specific knowledge on road, vehicle and control.**

***Keywords*: Pedestrian, Drowsy detection, Beam control, Fuel level, Gas level, Temperature, CAN (Controlled Area Network) Protocol, Nextion.**

1. Introduction

Automotive Electronics has been witnessing a major change from the analog world to the digital world to accommodate the rapidly growing technology so that the driving experience is made better, safer and at the same time, the end-user is provided with a variety of features that he can utilize in the vehicle. This change in the automotive industry is driven by a driving mechanism called the digital driving behaviour of the vehicle.

Due to the rise in automobile use over the last century, and its continued rise today, road accidents have become a prominent cause of injury and death. The knowledge in the field of computer vision continues to grow, so too does the realization of its potential benefits in the area of driver safety. Over the past few years, many recognition algorithms have been proposed to assist in the area of driver safety, but very few of them are both accurate and fast enough for real-time processing. This major qualifying project aims to further develop one of these algorithms for high accuracy and real-time performance in the area of pedestrian detection, beam control, drowsy driver, fuel monitoring, temperature monitoring, and obstacle and gas detection from an automobile. Detecting pedestrians in an image has proven to be a challenging task for many researchers due to the wide variability in possibilities. Posture, clothing, size, background, and weather all can be impactful on the appearance of an image. Real time pedestrian detection and tracking is considered as a critical application. Night time driving is riskier as compared to day time driving because of poor visibility especially in the case of senior citizens. While traditional methods of segmentation using thresholding, background subtraction and background estimation provide satisfactory results to detect single objects, noise is produced in case of multiple objects and in poor lighting conditions. To overcome these difficulties, a new method is proposed for detecting and tracking multiple moving objects on night-time lighting conditions. The method is performed by integrating both the wavelet-based contrast change detector and locally adaptive thresholding scheme. In the initial stage, to detect the potential moving objects contrast in local change over time is used. To suppress false alarms motion prediction and spatial nearest neighbor data association are used.

Sensing vehicles ahead and traffic situations during driving are important aspects in safe driving, accident avoidance, and automatic driving and pursuit. We designed a system that is capable of identifying vehicles ahead, moving in the same direction as our car, by tracking them continuously with an in-car ultra-sonic. The fundamental problem here is to identify vehicles in changing environment and illumination. Although there have been numerous publications on general object recognition and tracking, or a combination of them, not many of these techniques could successfully be applied in real time for in-car video, which has to process the input on-the fly during vehicle movement. This project introduces an effort to design and implement such real-time oriented algorithms and systems that are highly adaptive to the road and traffic scenes based on domain-specific knowledge on road, vehicle, and control. Pedestrian recognition is one of the most challenging problems in the field of computer vision. There have been many recognition algorithms proposed for purposes such as prevention of traffic accidents by using vehicle cameras. For embedded systems, however, a recognition algorithm that achieves not only high accuracy but also real-time processing in an environment with limited resources is required. Pedestrian detection is also an essential and significant task in any intelligent video surveillance system, as it provides the fundamental information for semantic understanding of the video footages.

2. Problem Formulation

- An incessant vehicle monitoring and communication between multiple nodes with respect to alerts for driver required more complex process. Existing structure uses multigateway network which has specific limitation for continuous studies upkeep.
- Study up keeping, monitoring, decision making, alert generation & control.

The overall hardware assembly and software coding provides a good challenge to achieve the prescribed digital driving behaviour in real-time and make the vehicle operate efficiently. Issues in the design can be categorized into hardware and software challenges. The hardware challenges include designing the circuits for each of the nodes, interfacing the sensors, interfacing the display unit and realizing the CAN bus between the nodes and interfacing master node and to MATLAB to get alerts like pedestrian detection, high beam, low beam & drowsy detection. The software challenges include programming each of the master & slave node to carry out monitoring tasks using each of the sensors, convert the analog signals to digital data (wherever applicable) and pass that digital data (control information) to the master node, process information from the slave nodes and raise alarms if necessary and additional image processing to identify pedestrian, light beams, drowsy.

3. Objectives

- To design a prototype to ADAS using CAN.
- To develop an HMI interface for generating alerts using Nextion display. Enabling communication between master and salve using CAN protocol.
- To integrate sonar to detect obstacle. To integrate MQ7 in for Gas detection. To integrate temperature & fuel sensor to slave for continuous monitoring.
- Pedestrian detection & alert generation using image processing. Headlight detection & beam change using image processing. Front camera with drowsy detection using image processing.

4. Proposed Methodology

The whole design can be divided into two stages. First stage is the development of the three individual nodes with necessary hardware interface and software to implement the various tasks performed by them. Second stage is to implement CAN network between the master node and the slave nodes and develop additional software to achieve CAN based communication among the three individual nodes.

Master node design

The master node of the system is implemented using a STM32F103C8T6 32bit ARM Controller. The hardware modules in MASTER Node circuit includes the Nextion TFT display unit, alarms for each of the system parameters in case of violation of user specified constraints. Serial port communication is also implemented between the master node and the host PC for MATLAB to update the pedestrian & headlight. The software part includes the MATLAB on the host PC for the serial port communication based HD camera acquisition, pre-processing & decision making feature is developed using image processing and microcontroller programs for displaying the current status of the vehicle, and for monitoring the current status of the vehicle by processing various control information received from the slave nodes in the system.

Slave node design

Slave node is the system is implemented using the same STM32F103C8T6 ARM controller used for master node implementation. The hardware modules in slave node circuit include the fuel level, temperature unit and gas detection unit. The software part includes microcontroller sub-routines to set up the current status of the vehicle, sensing any obstacle by using an ultrasonic combination and sending the current status signal to the display.

A. Hardware description

1) Master node

Figure 1 shows the Peripheral Interface of Master Node. Master node which does multiple operations and makes the decision as according to the input provided by slave node and MATLAB. Master node is designed using STM32 series Arm controller operating at 72 MHz with the execution speed 1.2dmips.

- To provide visual information to this module 3.2-inch TFT Nextion display is with various field is used which gives various alerts in graphical part. 40KHZ Ultrasonic transducer is used to detect obstacle and generate the alerts. This sensor can scan the range of 400cm with scan angle of 32 degree. Electrical burst signal of 10microsecond is sent through transmitter in the form of sound. This procedure is called trigger. Once the trigger is issued echo will be received from another transducer which converts sound in to electrical signal. Depending on the echo high time distance will be calculated with respect to sound equation.
- Four white led is used to simulate the headlight. In this led’s two led’s will be lit all the times and another two is controlled by STM. Beam changes information to the STM

is received by MATLAB which checks the high intensity light through camera and process the information to the hardware. L293D h-bridge motor driver with a 5V dc motor is used to simulate the speed variation with respect to pedestrian & obstacle detection.

- Master node receives other information like temperature, fuel & sensor status from slave node using CAN. Over all process is handled by ARM parallelly with the help of RTOS with various task and priority. Communication between master & slave node is done with the help of CAN at 100Kbs.

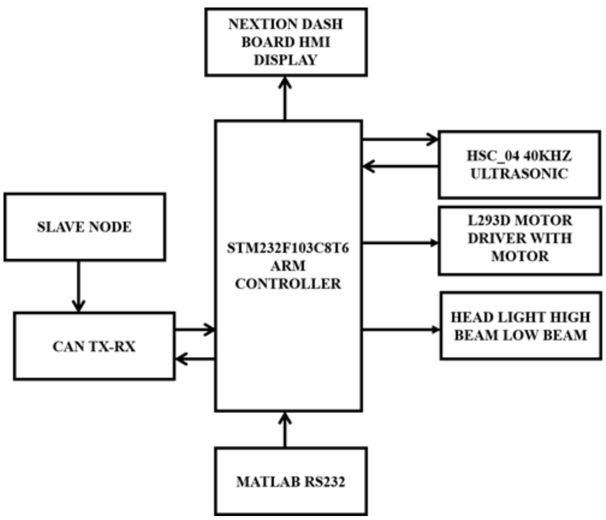

Fig. 1. Block diagram of peripheral interface of master node

2) Slave node

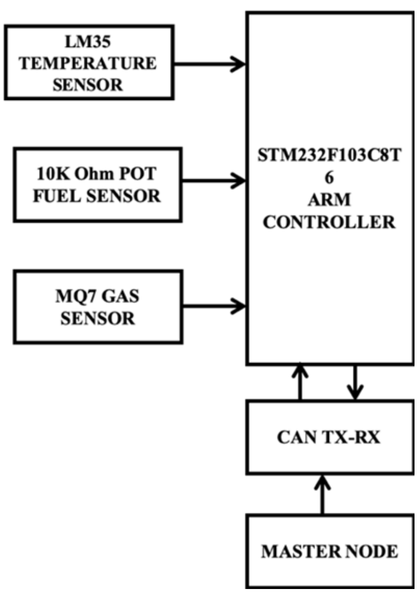

Fig. 2. Block diagram of peripheral interface to slave node

Figure 2 shows the Peripheral Interface to Slave Node. Slave node which does multiple operations and makes the decision as according to the input provided by master node and sensors. Slave node is designed using STM32 series Arm controller operating at 72 MHz with the execution speed 1.2dmips. Temperature measurement is done using LM35 analog sensor which provides voltage has according to variation to temperature.

- 12-bit inbuilt ADC is used to convert voltage into digital values and convert in to degree centigrade.
- Fuel measurement is simulated using 10kohm pot configured as voltage divider. As according to voltage variation from 0 to 3.33volts is mapped as 0-100value by using 12-bit ADC channel 1.
- MQ7 carbon monoxide sensor with a platinum wire coated with SN02 on ceramic tube with equipped with heater electrodes is helpful in detecting co2.
- All the processed information from sensors is sent to master node using CAN.

B. Software description

RTOS

Real Time Engineers Ltd. Developed RTOS as free and open source specifically for small footprint embedded system to fit for corresponding implements. Only function list that are minimal like management of memory, task handling, synchronization etc. without providing for communication in network for file system to excess hardware driver externally. It has the following features like task on Pre-emptive and can support multiple architecture of controller by its specific manufacturer that is written in C and different compiler of C. There is no limitation for number of tasks that are allowed to execute in a desired time with their priorities affords hardware for long operational time by implementing semaphores, binary and queues.

Binary semaphores:

To synchronize tasks, the simplest method used is binary semaphores which are not suitable as resource or input polling.

Handling Interrupts:

RTOS tasks or kernel provides interrupt handling method otherwise it will be handled by calling an hardware instruction. The priority over other interrupts will depend on an interrupt number. Handler is an interrupt, for semaphore operation must be realized using xSemaphoreGiveFromISR() instead of xSemaphoreGive().

Critical Sections:

For any context change that are protected as a part of code must prevent from being corrupted or cut, mixed of an I/O operation with another due to calculation must be prevented. RTOS gives two mechanisms to protect- by context change and by scheduler operation or an interrupt event.

5. Circuit Diagram and Working

Master node:

- PA10 pin is configured with UART transmission port connected to RxD of Nextion display.
- PA11 pin is configured as an input to USBCONN data

minus (D-).

- PA12 pin is configured as an input to USBCONN data plus (D+).
- PB7 pin is configured with L293D H-bridge motor of IN1 (U3).out1 & out2 pint is connected with motor.
- PB8 pin is a slave receiver from CAN reception pin.
- PB9 is configured as an input to output of SIG pin (send input to output pin).
- PB12 is configured as an input to output of LED.

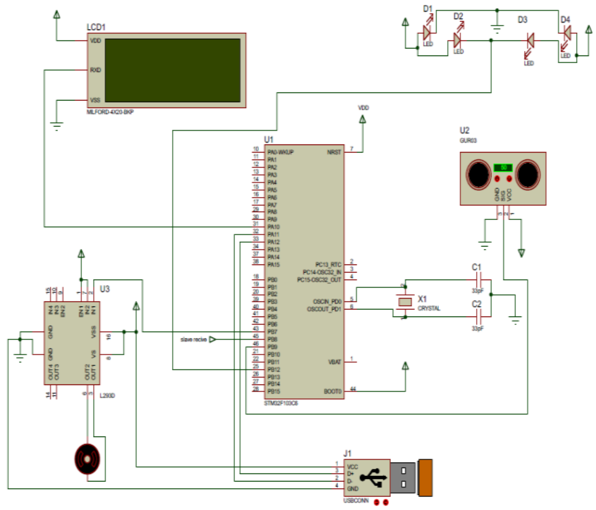

Fig: 3 Design diagram of master node using STM32F103C6

Slave node

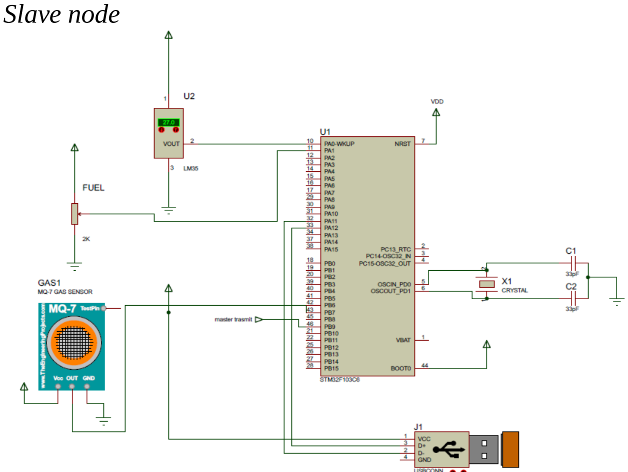

Fig: 4. Design diagram of slave node using STM32F103C6

- PA0 pin is an Analog channel as an input connected from LM35 (temperature sensor).
- PA1 pin is an input connected from Fuel Sensor (10K Ohm POT).
- PA11 pin is configured as an input to USBCONN data minus (D-).
- PA12 pin is configured as an input to USBCONN data plus (D+).
- PB7 pin is configured as an input to MQ-7 Gas sensor of output pin.

Image processing

Statistics show that a disproportionate amount of road fatalities occur at night time despite the greatly reduced volume of traffic on roads. An image processing system to detect and track vehicles using their lamps is presented. As the appearance of vehicle lamps in video can vary depending on camera hardware, a camera configuration process is implemented. A night-time pedestrian detection system, based on the processing of Far-Infrared video, is presented. A pre-processing step is introduced, which compensates for distortion caused by clothing, using vertically-biased morphological closing. Automotive safety is an important factor in vehicle design and technology. Vehicle safety assessment programmes, such as the Euro NCAP, have made both manufacturers and consumers more safety-conscious with regards to road vehicles. Driver in-alertness is an important cause for most accident related to the vehicles crashes. Driver fatigue resulting from sleep deprivation or sleep disorders is an important factor in the increasing number of the accidents on today's roads. Drowsy driver warning system can form the basis of the system to possibly reduce the accidents related to driver's drowsiness. The purpose of such a system is to perform detection of driver fatigue. By placing the camera inside the car, we can monitor the face of the driver and look for the eye-movements which indicate that the driver is no longer in condition to driver is no longer in condition to drive. In such a case, a warning signal should be issued. This paper describes how to find and track the eyes. One of the most significant categories of on-road entities that drivers need to be aware of are other road vehicles. Driver assistance systems that automatically track the positions of other road vehicles can be used for numerous functions, such as: collision warning, blind spot monitoring and automatic cruise control. A camera is an inexpensive and versatile sensor to utilize for this task, but at night the amount of visual information available in video is extremely limited compared with day time.

While driving at night, vehicles on the road are primarily visible by red coloured rear facing lamps and clear coloured headlamps. While all vehicle lamps will differ in appearance, they must adhere to automotive regulations which provide a set of characteristic features that can be utilized by image processing systems for identification. Lighting conditions in road environments can vary dramatically at night as ambient illumination can commonly consist of a combination of: vehicle lamps, street lights, infrastructure lighting and moonlight. Any system for identifying automotive lamps at night must be able to adapt to these conditions or mitigate their impact. It is beneficial to use standard colour cameras for the task of detecting other vehicles as they are low-cost, readily available and may already be present on the vehicle performing other functions, such as display of blind zones to the driver. CMOS image sensors are low cost and are widely available, but they have an extremely limited dynamic range in comparison to the human eye. While a human eye is sensitive from 0.1 lux to

10,000 lux, a CMOS camera sensor is normally only sensitive in the range of 6 to 15 lux. Headlamp and tail-lamp reflections can be prevalent on the road surface, particularly in wet weather conditions. This could lead to false positive identification of lamps in some instances, especially if the reflections are high in intensity.

Pedestrian and drowsiness detection using viola jones algorithm

Driver drowsiness is one of the major reasons which lead to these mishaps. we are discussing a Real Time Drowsiness Detection System which could determine the level of drowsiness of the driver. This system considers both the closing of eyes as the constraints for determining the drowsiness. Viola Jones Algorithm is used for facial features detection. Primary attention is given to faster detection and processing of data. Driver drowsiness will be detected by checking whether the eyes are closed over some particular consecutive frames.

Measures for Detection of Drowsiness

Several measures are available for the measurement of drowsiness which includes the following:

- Vehicle based measures.
- Physiological measures.
- Behavioural measures

Behavioral Measures

Certain behavioral changes take place during drowsing like

- Yawning
- Amount of eye closure
- Eye blinking
- Head position

Eye Blinking Detection

In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it’s really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver. In this stage, we find the actual state of the eye that if it is closed or open or semi closed or open. The identification of eyes status is most important requirement.

Poor contrast of eyes generally creates lots of problems in its detection. After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver. Though detection of eye does not look complex but the actual process is quite hectic. In this case it performs the detection of eye in the specified region with the use of feature detection. We send a warning if we obtain that the eyes are in open state or semi open state up to a particular threshold value. If the system detects that the eyes are open then the steps are repeated again and again until it finds a closed eye.

Vehicle headlight detection using blob analysis

There are 7 steps in the Headlight Detection Algorithm.

- Image Acquisition
- Image Greyscaling
- Noise Filtering
- Image Binarization
- Morphological operations to the image
- Light Blob area calculation
- Conversion of High Beam to Low Beam

The above steps have been discussed in detail

Image Acquisition

The video is taken at a fixed point without changing the position of the camera. The video can be taken at a lower resolution for reducing the processing time. The video sequence is acquired from a camera. A video sequence needs to be sampled in to multiple frames before applying the image processing techniques and those set of frames are the inputs to the subsequent stages of the system. The function called win video (), which is a built-in function in MATLAB is one such method.

Image Grey Scaling

Converting colour images into grayscale images is the first step of the pre-processing. Grey scaling removes the colour values of an image and simplifies computational time significantly compared to a colour RGB image. There are several algorithms to convert a colour RGB image to a grayscale image. The function called rgb2gray ( ), which is a built-in function in MATLAB is one such method.

Noise Filtering

After grey scaling the image, it had to go through the filtering process to filter out any noise in the image. A wide variety of filtering algorithms are available to detect and remove noise, leaving us the required information. The filter used in this work is a median filter, which has a nonlinear operation and is often used to reduce the "salt and pepper" noise. Median filter is more effective than convolution when the goal is to simultaneously reduce noise as well as preserve the edges. The function medfilt2 ( ) is a built-in function in MATLAB, which performs the median filtering operation. The output image from the previous operation is passed through the filter to obtain a noise free image.

Image Binarization

After the filtering process, we have to determine the threshold to convert the grayscale image to binary image. The function im2bw () is a built-in function in MATLAB, which performs the thresholding operation and converts the image from grayscale to binary.

Morphological Operations to the Image

To enhance the binary image, we apply different morphological operations, in the binary image, some areas of the headlight have black pixels which need to be filled to get more accurate results. The function imfill ( ) is a built-in function in MATLAB, which performs the filling operation on the binary image.

Light Blob Area Calculation

we then calculate the number of white pixels present in the binary image.

Conversion of High Beam to Low Beam

International Journal of Research in Engineering, Science and Management
Volume-3, Issue-6, June-2020
www.ijresm.com | ISSN (Online): 2581-5792

After calculating the area of the number white pixels we then compare it to our threshold value. This threshold value is calculated from a binary image in which the vehicle is at a distance of certain distance for headlight detection.

6. Results and Discussions

The figure 5 shows the Hardware design of the model. It comprises of Nextion HMI display, Ultrasonic sensor, MQ7 gas sensor, LM35, H bridge motor driver, STM32 Processor, DC to DC converter, 12v battery, Bluetooth receiver, 10k Pot, DC motor and LED light.

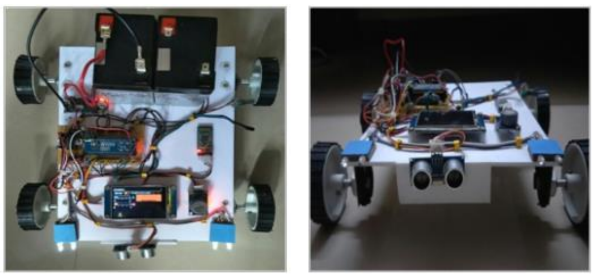

Fig. 5. Hardware design

When there is any erroneous circumstances MATLAB detects it and sends the signal via Bluetooth. This signal will be received by the Bluetooth receiver in the model and sends the signal to the STM32 Processor which will be processed and the warning alert will be displayed in the Nextion Display.

Pedestrian detection

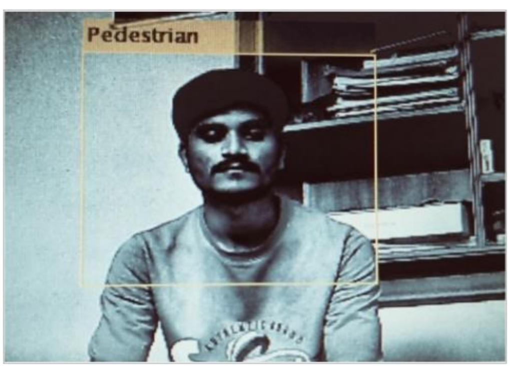

Fig. 6. Pedestrian detection

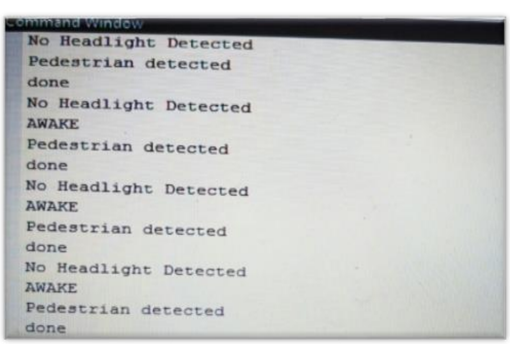

Fig. 7. MATLAB command window

When the pedestrian comes in front of the vehicle, camera records the image and this is processed by image processing using MATLAB as shown in figure 6 and in the MATLAB command window pedestrian detected command is shown as shown in figure 7. Warning alert will be displayed in Nextion display as "PEDESTRAIN" as shown in figure 8 and the vehicle stops.

Fig. 8. Nextion display

Headlight detection

When a vehicle is approaching from the opposite direction, camera records the light from the approaching vehicle and this is processed by image processing using MATLAB. The headlight remains in high beam if there is no vehicle in the opposite direction as shown in the figure 9. When there is a vehicle approaching from the opposite direction the headlight changes to low beam as shown in figure 10 and in the MATLAB command window Light Detected command is shown as shown in figure 11. High beam and Low beam alert is displayed in Nextion display as shown in figure 12.

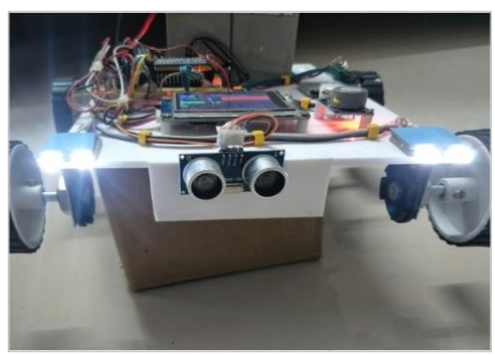

Fig. 9. High beam

Fig. 10. Low beam

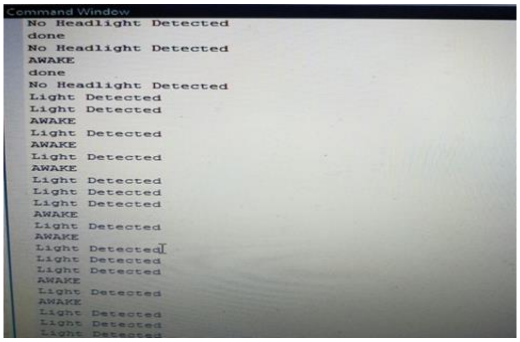

Fig. 11. MATLAB command window

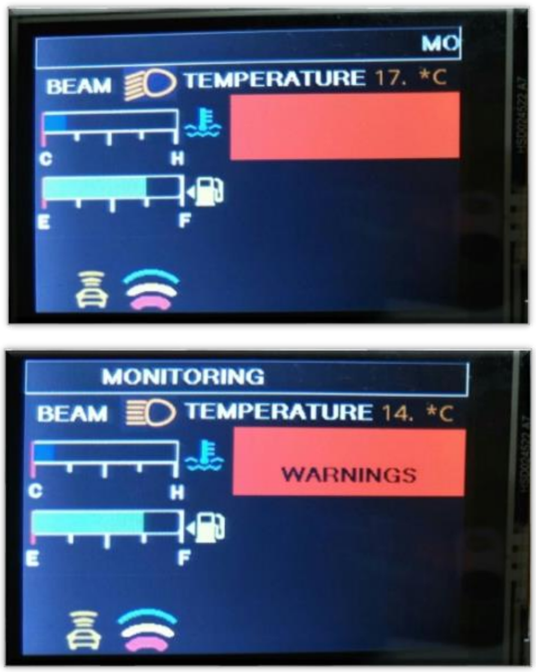

Fig. 12. Nextion display

Drowsy detection

The camera on the dash board will be continuously monitoring the driver and this is processed by image processing using MATLAB as shown in figure 13 and in MATLAB command window awake and drowsy command is shown as shown in figure 14. If the driver is awake the command window shows awake and if the driver is drowsy the command window shows drowsy.

Fig. 13. Drowsy detection

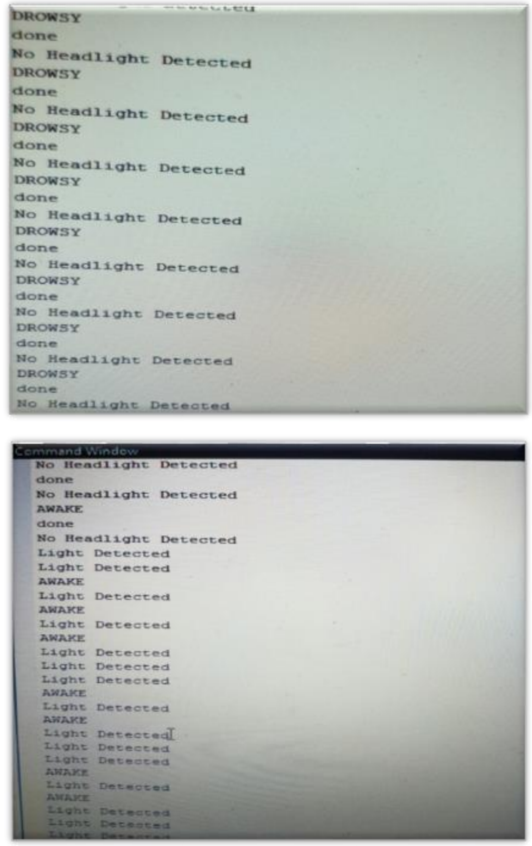

Fig. 14. MATLAB command window

Obstacle detection

When there is an obstacle in front of the vehicle, it will be detected by the ultrasonic sensor and sends the signal to the STM32 processor which will be processed. Nextion display shows the warning as "COLLISION PROXIMITY" as shown in figure 15 and the vehicle stops.

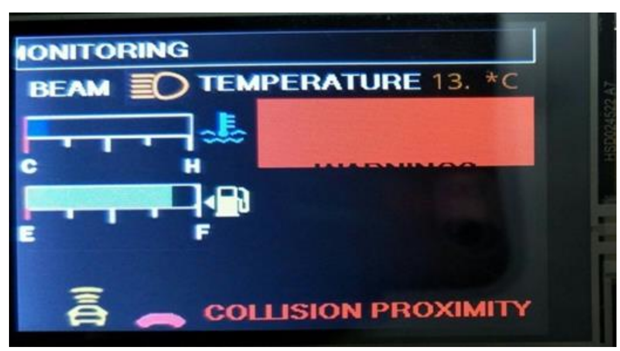

Fig. 15. Nextion display

Gas detection

The MQ7 sensor detects the CO2 level in the vehicle. If the level is beyond the limit the sensor sends a signal to the STM32 Processor which will be processed. Nextion display shows the warning as "CO ALERT" as shown in figure 16.

Fig. 16. Nextion display

Temperature detection

The LM35 sensor detects the temperature level in the vehicle and the temperature level will be continuously monitored by STM32 Processor. This will be displayed continuously on Nextion display if the temperature level is beyond the limit Nextion display shows the warning as "TEMPERATURE ALERT" as shown in figure 17.

Fig. 17. Nextion display

Fuel level

The fuel level is continuously monitored by STM32 Processor and the fuel level will be continuously displayed on Nextion display. If the fuel level is less than 25% Nextion display shows the warning as "LOW FUEL ALERT" as shown in figure 18.

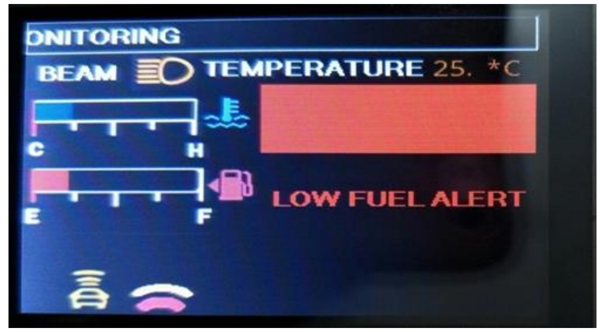

Fig. 18. Nextion display

7. Conclusion

- CAN is ideally suited in applications requiring a high number of continuous messages with high reliability in automobiles. Since CAN is message based and not address based, it is especially suited when data is needed by more than one location and system-wide data consistency is mandatory.
- Fault detection is also a major benefit of CAN.
- The safety measures includes temperature & fuel monitoring, gas detection, pedestrian detection, headlight detection & drowsy driver detection and accident alert system are provided in this safety system.

8. Future Scope

As research and development is an endless process, there is always a chance to improve any system.

The following improvement can be done:

- Lane Monitoring
- Emergency Vehicle Alert
- Navigation using Nextion HMI Display
- Dash Cam Implementation.

References

[1] Mohammed Ismail, "Design and Development of a Vehicle Monitoring System Using CAN Protocol," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 11, November 2015.
[2] Hazim Hamza, "Night Time Car Recognition Using MATLAB," M-Eng in Electronic Systems, 2013.
[3] Sina, "Vehicle Counting and Speed Measurement Using Headlight Detection."