# Autonomous Car Implementation Based on CAN Bus Protocol fot IoT Applications

A.M. Elshaer
Department of Computer Engineering,
Higher Institute of Engineering and Technology
El-Boheira, Egypt
ahshaer1@yahoo.com

M. M. Elrakaiby
Department of Computer Engineering,
Higher Institute of Engineering and Technology
El-Boheira, Egypt
mm_elrakaiby@yahoo.com

Mohamed E. Harb
Department of Electronics and Communication Engineering
Higher Institute of Engineering and Technology
Alexandria, Egypt
moh.harb@yahoo.com

***Abstract*—This work aims to design an autonomous vehicle that able to sensing the surrounding environment. The self-driving vehicle can moves in certain path between two points A and B. Autonomous vehicle have five electronic control units (ECU0 through ECU4) will communicate over a control area network (CAN) bus. Every control unit ECU0 and ECU1, controls three of the ultrasonic sensors installed in the front and back of the vehicle. The middle ECU2 consists of ultrasonic sensor and global positioning system (GPS) to provide the main control unit ECU4 with current location to take the decision to move and send it to the moving control unit ECU3. ECU3 controls the motor drivers, that able to take a fast decision depending on the data collected from other ECUs. ODROID-XU4 is the main control unit that connected to a Microsoft camera 5000 and internet flash. ECU4 captures all data published on the CAN bus and classified based on the identifier of each ECU according to emergency or location data. GPS sensor updates the location on Google maps every 5 second and stops the vehicle if the vehicle reaches to point B producing a good internet of things (IoT) application.**

**Keywords—smart vehicle; CAN bus; electronic control units; IoT**

## I. Introduction

Over the past 50 years, several writers have expected the emergence of self-driving vehicles in the future, and their ideas have been reflected in many films and literary novels. Eureka Prometheus Project was the largest R&D project that tries to create first autonomous vehicle in the last century in Europe [1]. Today vehicle manufacturers compete to produce autonomous vehicle with “Level 5 Full automation” System performs the lateral and longitudinal dynamic driving task in all situations encountered during the entire journey, no driver required. Tesla vehicle which is a semi-autonomous vehicle is classify under “Level 3 conditional automation” to reach to level 5 the vehicle computer need to sense local environment, classify different kinds of objects that it detect in real time and need to collect more data, from here another problem appeared in front of the vehicle manufacturers [2-3].

Every vehicle has electrical units that called ECU. Every ECU does a specific function in the vehicle like control the brakes and airbags or monitors the fuel level and engine temperature…etc. In order to make the vehicle smart and more safety new ECUs or nodes were added to the vehicles, it is not a problem. The problem was how to connect between these different ECUs to deliver the data from ECU to another. In order to solve the problem of wiring in the vehicles, Bosch started to develop a new serial communication protocol in the early1980s. In 1986, Bosch introduced The Controller Area Network (CAN) bus to the world [4]. The first use of CAN bus was in 1988 BMW-8 Series vehicles. The last version of CAN bus is CAN with Flexible Data-Rate (CAN FD) and it released from Bosch in 2012, this version speed reach to 8 Mbit/sec and allows to sending 64 byte of data in every message. The CAN bus is also using to run diagnostics on the vehicles to report the status of each ECU in the vehicle this feature make the fix of vehicle's problem easier. Today all vehicles in the world use CAN bus as a main serial communication protocols between most of ECUs for several reasons [5,6]:

- Easy to add / remove nodes.
- Node breakdown does not bring down the bus.
- Low wire count.
- Nodes independences

This paper aims to build an autonomous vehicle (self-driving vehicle) that can able to pick up any one that requests a tour through the graphical user interface GUI at the monument area. Driverless vehicles pick-up the person to start the tour from point A. Vehicles navigate over a planned path, based on Google maps and GPS sensors that provides the proposed hardware modules by end-point B to stop the vehicle. The proposed model is dividing into hardware and software structures.

Thanks to Bright skies® Technologies for providing by a Microsoft LifeCam HD-5000 720p HD and control Vector VN8972 CAN case to test and simulate the linking between five ECUs based on CAN bus protocol.

## II. SYSTEM COMPONENTS

### *A. Electronic components*

ODROID-XU4, the Heterogeneous Multi-Processing (HMP) Samsung Exynos5422 Cortex™, OS Ubuntu MATE 16.04, were supplied from Hard kernel co., South Korea. MCP2551 CAN bus transceiver microcontroller, MCP2515 SPI to CAN, external CAN bus hardware, AT90CAN128 microcontroller, ATMEGA32 microcontroller, GPS, and ultrasonic sensors, were brought from Future electronics, Egypt. Microsoft LifeCam HD-5000 720p HD was supplied Bright Skies® Company, Egypt.

### *B. Implementation of Autonumous Vechile*

The proposed hardware structure consists of five ECUs (ECU0 through ECU4) and CAN bus. Every ECU have microcontroller, sensors, and power supply circuits and communicate with other ECUs using CAN bus as shown in Figure 1. ECU0 and ECU1 will control the vehicle movement front and back. ECU2 will supply ECU4 with a current location and the destination via GPS sensor. ECU3 responsible the decision of movement or the brakes if the ECU1 or 2 detect any obstacles around 3 m. ECU4 is the main control unit have a high priority in the proposed system. The main board ODRIOD-XU4 with Ubuntu MATE 16.04 operating system will transmit and receive data through CAN bus protocol. CAN bus allow two different ECUs to communicate with each other based on two wires only CANL and CANH to reduce the number of wires in the vehicle.

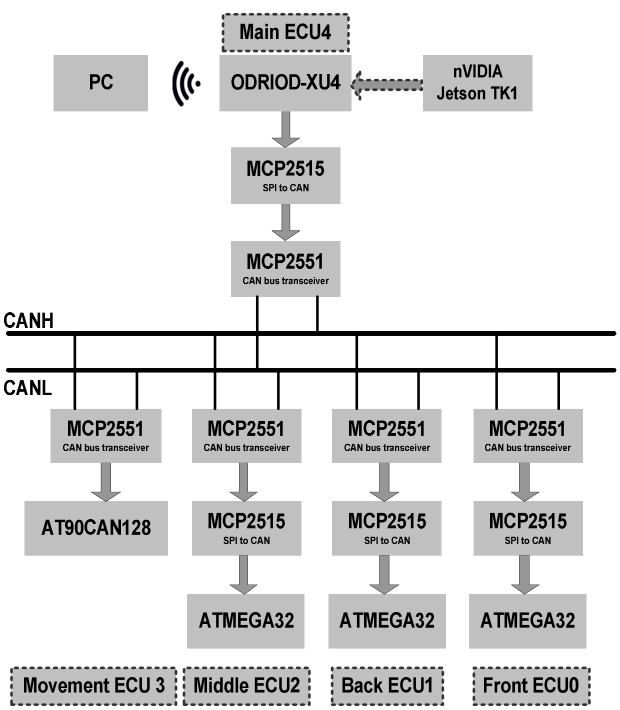


Fig. 1. Hardware structure of the proposed model with 5 ECUs and CAN bus.

*1) Front and Back ECUs:* ECU0 and ECU1 represent the front and back ECU respectively. Figure 2, shows ECU0 and ECU1 will connected to the CAN bus via MCP2551 CAN bus transceiver and MCP2515 SPI to CAN, directly to ATMEGA32 to controls the three ultrasonic sensors in the front and rear of the vehicle. The mission of these ECUs is to capture the data provided from the ultrasonic sensors and send it to another ECUs based on MCP2515 to CAN bus. CAN protocol is an ISO standard (ISO 11898) for serial data communication. CAN bus allow to different ECUs to communicate with each other based on 2 wires only CANL and CANH to reduce the number of wires in the car. Figure 2 shows CAN bus standard data frame. Every CAN bus data frame is called message and every node on CAN bus has unique identifier that should not be repeated and determined by the programmer, or manufacturer. Each message published on the bus has an ID, data and overhead. These ECUs is very important because it will alarm if any barrier close to the vehicle in front or behind in range of 3 meters.

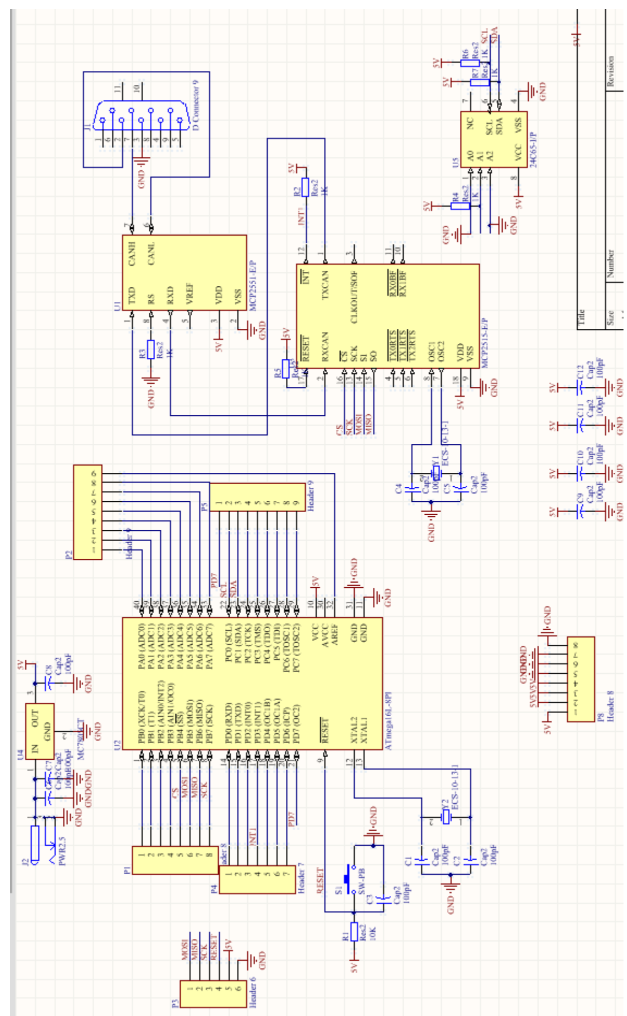

Fig. 2. Circuit schematic of the ECU0 and ECU1 .

*2)Middle ECU2:*This unit controls two ultrasonic sensors install in both sides of the vehicle, GPS and compass install in the middle of the vehicle. The mission of this ECU is to send the data collected from the ultrasonic, GPS and compass via CAN bus to ECU 4. ECU2 will provide the user with current location of the vehicle point A from the GPS sensor. ECU4 is the main board will receive the information from ECU2 and will apply GPS information to Google Maps to show the current location on a real map and this will appear to the user on GUI screen installed on the vehicle in real time.

*3) Movement ECU3:* AT90CAN128 is the main microcontroller of the ECU3, which connected to the CAN bus via MCP2551. ECU3 controls the movement of the vehicle based on controlling the drivers of DC motors. ECU3 can take a fast decision depend on the data of ultrasonic sensors around the vehicle provided from another ECUs. Fast decision like stop, brakes the vehicle reduce the possibility of a vehicle colliding with the obstacles on the road. Figure 3 represents photo of the movement ECU PCB Board.

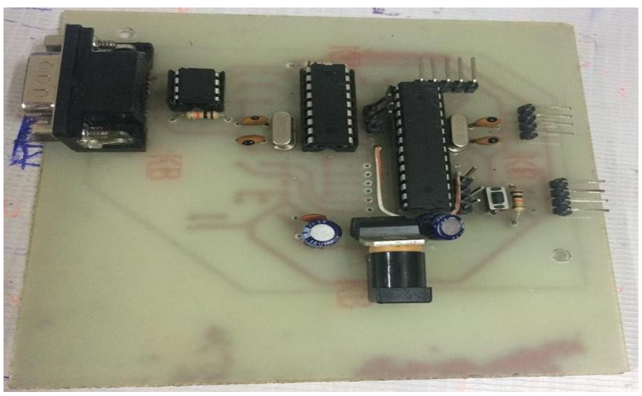

Fig. 3. PCB design of the movement ECU.

*4) Main ECU4:* The main control unit of the proposed hardware architecture is ODROID-XU4 Heterogeneous Multi-Processing technology. ODROID-XU4 has CPU from Samsung which is Exynos5422 Cortex™ - A15, 2 Ghz and Cortex™ - A7 Octa core , 2 GB DDR3 RAM. ECU4 connect to high quality camera (Microsoft LifeCam HD-5000) and Internet flash via USB3 port. The GUI interface shown in Figure 4 applied on the ECU4. GUI used to on/off the vehicle and receive all data from all ultrasonic devices installed on the vehicle and run CAN bus trace program. Figure 5 depicts CAN bus trace program is capture all messages which send on CAN bus and start to classify it depend on it identifier then classify it into location data and emergency data and pass it to GUI program and store this data on external memory SD vehicled. Classified data will applied to TensorFlow (deploy phase) with training phase based on external board NVIDIA Jetson TK1 developer kit. The application will provide the main board by:

a) Locate the vehicle location on Google maps using GPS data, update this location every 5 sec online, and stop if the vehicle reach to point B.

b) Analyses the data collected from all ultrasonic sensors ECU0, 1, 2 and take the decision left, right, or stop and send the decision to ECU3.

c) Analysis the data came from the HD camera, recognize the road and traffic signs then take the decision, and send the order to ECU3 via CAN bus

## C. Software structure

The software structure it consists of low-level and high-level software structure.

*1) Low level Software:* Low-level software will used with ECU0 through ECU3 based on C for embedded on Motor Industry Software Reliability Association MISRA-C rules. AUTOSAR helps to write a code that another companies and supplies can deal with and can understand. AUTOSAR based on software layer concept; the proposed low-level software architecture will imitate the AUTOSAR layer concept. For example the flowchart shows in Figure 6 describe the software structure of the main ECU.

*2) High Level Software:* The high-level software architecture for ECU4 ODRIOD-XU4 that manages the orders collected from all ECUs. An algorithm to recognize 4 Traffic Signs based on python is described in Figure 7.

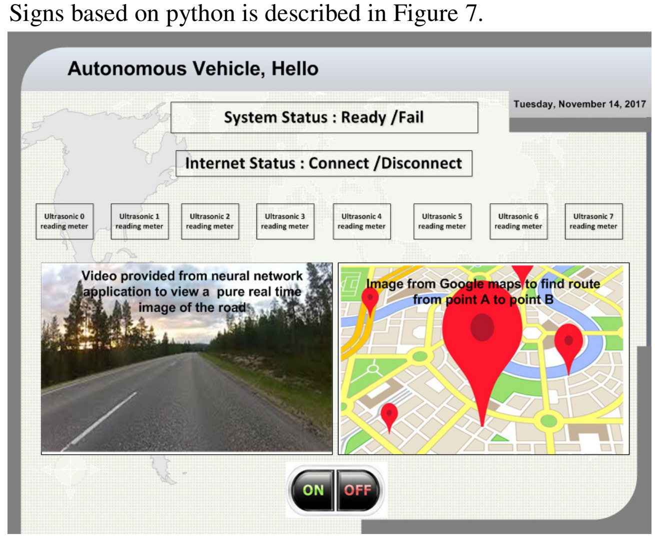


Fig.4. The GUI interface applied on the main ECU4

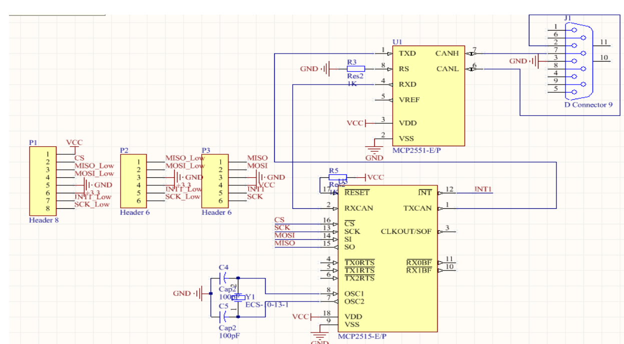

Fig.5. CAN Bus Module schematic

CONCOLUSION

A smart vehicle based on IOT was successfully fabricated. An autonomous vehicle that can be able to pick up any one that requests a tour through the graphical user interface GUI. Autonomous vehicle that moves from point A to point B by avoiding all obstacles, people and identify the traffic lights. Vehicles navigate over a planned path, based on Google maps and GPS sensor that provides the proposed hardware modules by end-point B to stop the vehicle. The proposed hardware consists of 5 ECUs and communicate with each other via CAN bus. All 5 ECUs and accessories will simulate and interface with the CAN bus hub.

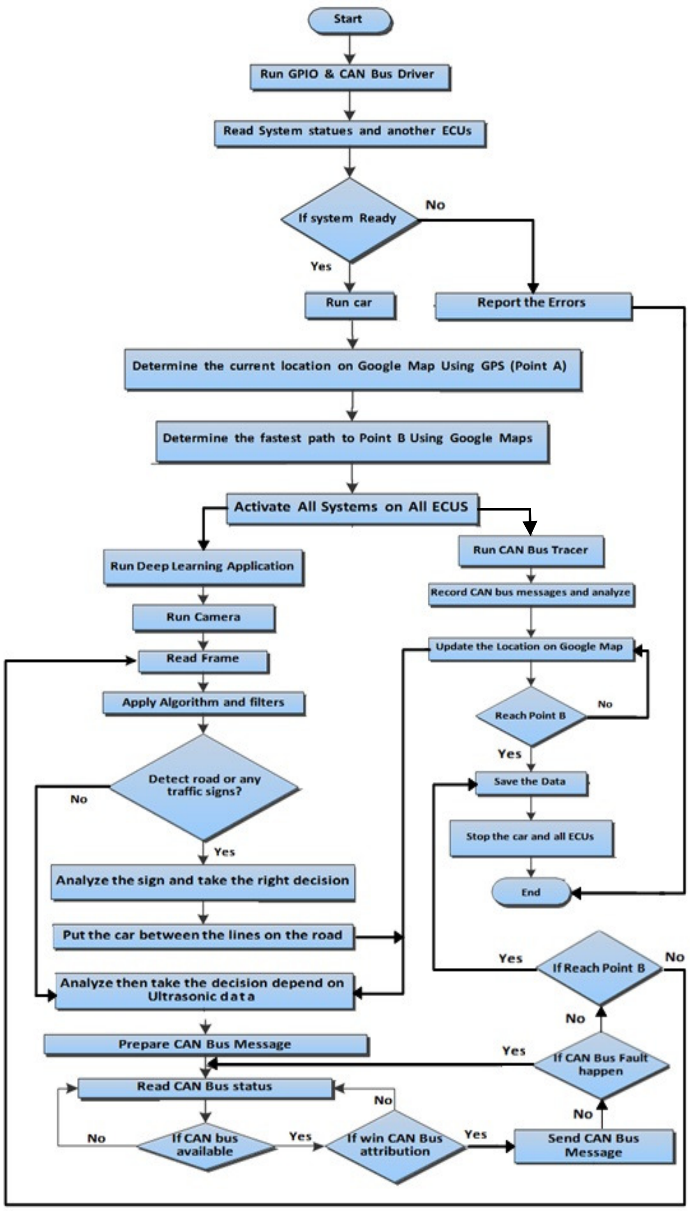


Fig.6. Main ECU software structure.

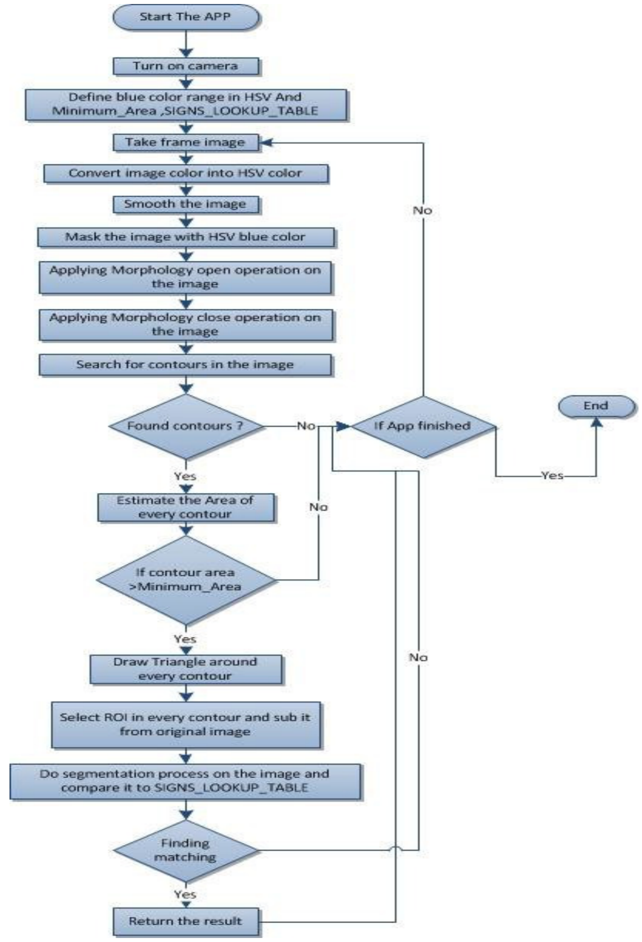


Fig. 7. Flowchart of the traffic recognition algorithm

## References


[1] C. Urmson, J. Anhalt, M. N. Clark, D. Duggins, T. Galatali, C. Geyer, et al., “Autonomous driving in urban environments: Boss and the urban challenge,” J. F. Robot., vol. 25, no. 8, pp. 425–466, 2008.

[2] Tesla (2016), "Model 3", www.teslamotors.com/model3. last access 18.05.2017.

[3] S. L. Poczter and L. M. Jankovic, “The google vehicle: driving toward a better future,” Journal of Business Case Studies, vol. 10, no. 1, pp. 1–7, 2014.

[4] L. Laursen, “Volvo to test self-driving vehicles in traffic,” IEEE Spectrum - Tech Talk, 2013.

[5] J. E. Naranjo, M. Clavijo, F. Jimenez, O. Gomez, J. L. Rivera, and M. Anguita, “Autonomous vehicle for surveillance missions in off-road environment,” in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 98–103.

[6] A. Hussein, P. Martin-Plaza, D. Martin, A. de la Escalera, and J. M. Armingol, “Autonomous off-road navigation using stereo-vision and laser-rangefinder fusion for outdoor obstacles detection,” in 2016 IEEE Intelligent Vehicles Symposium (IV), 2016, pp. 104–109.