

# Arduino-Based Automatic Safety Vehicle Control

Othman M.K. Alsmadi, Anas A. Al Jallad, Zaer S. Abo-Hammoud, and Fares J. Al Majali

**Abstract**—To minimize the occurrence of vehicle accidents due to an eye or hand movements while driving, an automatic vehicle human-voice-based safety control is proposed in this paper. The design is implemented using an arduino microcontroller. One of the most important safety control features is the automotive speed control (ASC). In this safety control function, which prevents a vehicle sudden hit, the necessary action is issued based on the vehicle-to-vehicle distance measure. To prevent the driver from attention withdraw and maintain a proper road and situation focus (as especially is the case at night), several other automated safety control operations are provided (i.e., radio, flasher, left flasher, right flasher, wipers, drive, and stop). These features are controlled according to human voice commands using an EasyVR module. The proposed model has been designed, built, and tested providing suitable results as desired.

**Keywords**—Safety-vehicle-control, Arduino microcontroller, voice recognition.

## I. INTRODUCTION

THE problem of vehicle accident is part of an endless list of disasters that could occur anywhere anytime. According to the Association for Safe International Road Travel [1], about 1.24 million die and 50 million are injured on the roads of the world every year. There are many reasons to why vehicle accidents may occur, for instance, carelessness while turning a radio on/off, searching for temperature control, searching for light control, etc. Fig. 1 [2] shows a study for vehicle accident accordance in Jordan. In many parts of the world (i.e., in Jordan), traffic accidents constitute a major social problem. Statistically, they are considered the second leading cause of death.

To overcome such problems, many vehicle manufacturers and automobile device companies have attempted to develop speed control systems to maintain a vehicle safe distance.

Othman M.K. Alsmadi is with the Electrical Engineering Department, The University of Jordan, Amman 11942, Jordan (e-mail:Othman\_mk@yahoo.com).

Anas A. Al Jallad is with the Electrical Engineering Department, The University of Jordan, Amman 11942, Jordan.

Zaer S. Abo-Hammoud is with the Mechatronics Engineering Department, The University of Jordan, Amman 11942, Jordan (email: Zaer\_hr@yahoo.com).

Fares J. Al Majali is with the Electrical Engineering Department, The University of Jordan, Amman 11942, Jordan.

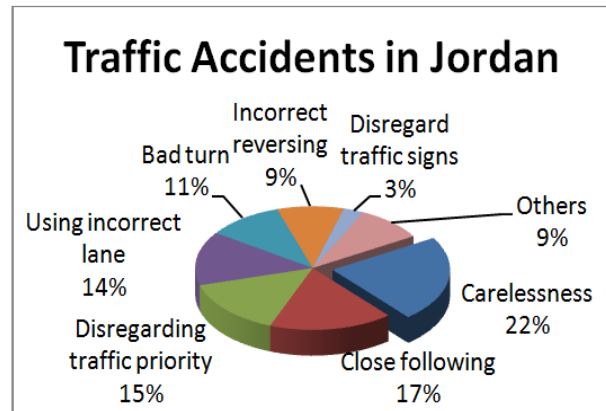


Fig. 1 Causes of car accidents in Jordan

They generally used two methods; one is the Laser-based approach while the other is the Radar-based approach. In the Laser-based type, the systems do not track vehicles in adverse weather conditions nor do they reliably track extremely dirty vehicles. Another implication imposed is that the sensors must be exposed. In the Radar-based type, the system can be hidden behind plastic fascias. The Mitsubishi Inc. [3] was the first automaker to offer a laser-based ASC system in 1995 on the Japanese Mitsubishi Diamante marketed as "Preview Distance Control". More manufactures then used these features such as BMW, Mercedes-Benz, Audi, Honda, and Hyundai.

Researchers have recently started to explore the introduction of additional objectives into the automated vehicle control, e.g., fuel economy and driver desired speed control. Shengbo et al. [4,5] suggested the use of automated speed control to reduce fuel consumption of vehicles and considered issues of driver desired response and hence indicated the necessity to adapt to drivers' individual driving requirements.

Automated vehicle control systems have been recently developed using various detection sensors for speed control, such as long-and-short range radar, wide dynamic range cameras, ultrasonic sensors and laser scanners. Among these detection sensors, radars can quickly provide drivers with reliable information about the velocity, distance, and direction of a target obstacle, as well as information about the vehicle in changing weather conditions. Jeoung et al. [6] stated that three radar sensors are usually needed because two short range radars are used to detect objects in the adjacent lane and one long range radar to detect objects in the desired path. However, it is found that ultra sonic and laser sensors have the advantage of providing fast responses which satisfy most of

the design requirements. The main difference between ultrasonic and laser sensors is the sensing range, i.e. range of ultrasonic is wider, which is a major issue in the ASC systems. Payman and Andrzej [4] have also investigated the issue of automated vehicle control and suggested control algorithms considering the stop & go function with adaptive speed control. Shengbo et al. [5] discussed the economy benefits using automated speed control based on reducing fuel consumption. Others suggest that the best sensors for this objective are the radar sensors [6]. Moreover, Rogowski [7] discussed the main advantages of voice control systems in industrial applications. Englund [8] discussed the effect of G-load on voice control systems and proposed some solutions.

## II. DESIGN PROCEDURE

### A. Automated Speed Control (ASC)

The general design for any ASC system is implemented as seen in Fig. 2. In this paper, the Arduino Mega is used to provide all required functions for the proposed vehicle safety control. The Arduino Mega system is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator along with other necessary connections. It contains all necessary parts needed to support the microcontroller and is powered by a computer with a USB cable [9]. Even though the Arduino system is an educational tool to test new ideas, however, for real implementation, it can be easily converted using an ATmega2560 microcontroller (used in the Arduino mega board) with the appropriate connections.

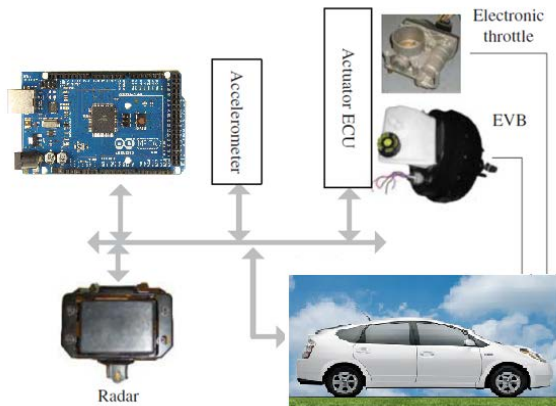


Fig. 2 Configuration of experimental platform

In the proposed work, the ASC is performed using the Arduino map function which, in a way, is similar to a fuzzy controller. The map function structure is: “map (value, fromLow, fromHigh, toLow, toHigh)”. The function remaps a given number from one range to another. For example, a given value of “fromLow” would get mapped into “toLow”, a value of “fromHigh” into “toHigh”, values in-between into different value [9]. However, value from **above** “fromHigh” will be mapped into **above** “toHigh” and hence another function is

used “constrain” to keep the value in a proper range [9]. The value provided by the map function represents the distance measured by the ultrasonic sensor as it falls in a certain range. If the value falls into a lower range, then the vehicle is to be stopped. The proposed ASC can be summarized by the flowchart shown in Fig. 3.

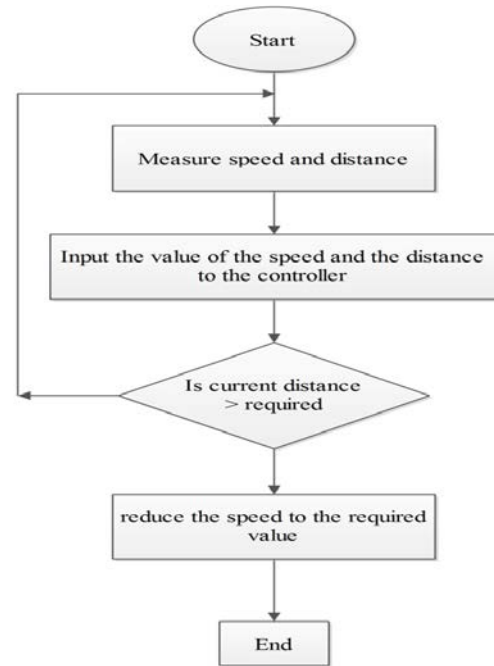


Fig. 3 Speed controlling algorithm

As generally illustrated in Fig. 3, a command is sent to trigger the ultrasonic signal where the time required for the echo signal to return is calculated. The distance is calculated using the following sound equation:

$$Range = Th(V/2) \tag{1}$$

where  $V = 340$  m/s and  $Th$  is the High level time. In the next step, the vehicle speed is calculated using the appropriate sensor, e.g. accelerometer. The values of the distance and speed are entered to the controller algorithm where a test algorithm is made to test if the required distance is greater than the current distance. If the required distance is less than the current distance, then the vehicle should maintain its speed, otherwise, the speed should be reduced to a value that is less than the maximum required speed.

### B. Voice Recognition

There are various applications for voice recognition including voice user interfaces such as voice dialing, call routing, domestic appliance control, search, simple data entry, preparation of structured documents, and voice-to-text processing. In this paper, the purpose of using voice recognition is to maximize the driver safety by allowing the interface with various functions via voice commands to overcome different distraction while driving. Some of the distractions maybe looked at as flasher control, washers, radio,

etc.

Generally, the performance of voice recognition systems is evaluated in terms of accuracy and speed. Accuracy is usually rated with Word Error Rate (WER), whereas speed is measured with real time factor. However, voice recognition (by a machine) is a very complex problem. Vocalizations vary in terms of accent, pronunciation, articulation, roughness, nasality, pitch, volume, and speed. Voice maybe distorted by a background noise, echoes, or electrical characteristics. Hence, the accuracy of voice recognition may generally vary according to the followings:

- Vocabulary size and confusability
- Speaker dependence vs. independence
- Isolated, discontinuous, or continuous voice
- Task and language constraints
- Read vs. spontaneous speech
- Adverse conditions

In this work, a speaker dependent module called EasyVR is used for the voice recognition operation. EasyVR is used as a user interface to allow for voice recognition training. It has 16 groups each group can take up to 32 user-defined Speaker Dependent commands. The groups can be used for different systems or subsystems depending on the number of commands that the user needs to activate.

Using the EasyVR module, the Algorithm of recognizing a given voice is performed for two operations; training and recognition. For training, a spoken word is stored using certain software so as to make the system ready to recognize different words. To recognize a word, the word is spoken into a microphone to provide a certain vector that can be translated into a current (High) to make a certain function. A general part of the vehicle safety control algorithm is given by the following:

```
#include "EasyVR.h"
#define TP 12 //Trig_pin
#define EP 13 //Echo_pin
```

```
boolean flasher= true;
boolean radio= true;
boolean wiper= true;
boolean left = true;
boolean right = true;
boolean move1 = true;
boolean reduce = true;
```

```
EasyVR easyvr(Serial);
Serial.begin(9600);
}
void loop()
{
  check_distance();
  easyvr.recognizeCommand(1);
  while (!easyvr.hasFinished()){
    check_distance();
  }
}
```

```
int index = easyvr.getCommand();
if(index==0||index==1||index==2){
  if(flasher==true){
    digitalWrite(30,HIGH);
    flasher=false;
  }
  else{
    digitalWrite(30,LOW);
    flasher=true;
  }
}
if(index==3||index==4||index==5){
  if(radio==true){
    digitalWrite(32,HIGH);
    radio=false;
  }
  else{
    digitalWrite(32,LOW);
    radio=true;
  }
}
```

The EasyVR shield is a “slave” module communicates via an asynchronous serial interface (commonly known as UART interface) designed with the following specifications:

- Baud Rate: 9600 (default), 19200, 38700, 57600, 115200.
  - Frame: 8 Data bits, No parity, 1 Stop bit.
- The receiver input data line is ERX, while the transmitter output data line is ETX (no handshake lines are used).

The microphone, provided with the EasyVR module, is a unidirectional electric condenser microphone. It is designed with the following specifications:

- Sensitivity = -38dB.
- Load Impedance 2.2K
- Operating Voltage 3V
- Almost flat frequency response in range 100Hz – 20 kHz
- Adapter board

To recognize a voice using the EasyVR module, a “dB” (present as a character) is first sent to the EasyVR to begin the recognition process. Then, a check is performed to assure that there is a serial communication response. Once the response is positive, the module checks for the received character “r” from the EasyVR which indicates that a command has been recognized. Then, a “20” hexadecimal code is sent for the command index and is required to wait until the command is sent from the EasyVR serial communication. As a result, the appropriate command related to the received index is performed. This process may simply be illustrated as seen Fig. 4.

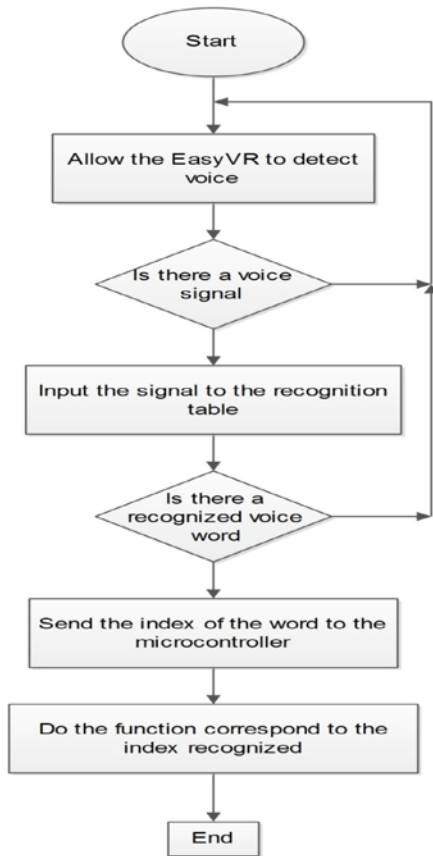


Fig. 4 Voice recognition algorithm

The EasyVR module can save up to 32 different commands. In this work, 7 functions were designed which can be stated as following:

- Flasher
- Left Flasher
- Right Flasher
- Radio
- Wipers
- Go
- Stop

Communication with the EasyVR module uses a standard UART interface compatible with 3.3-5VTTL/CMOS logical levels according to the powering voltage VCC. A typical connection to an MCU-based host is typically specified as seen in Fig. 5 [10].

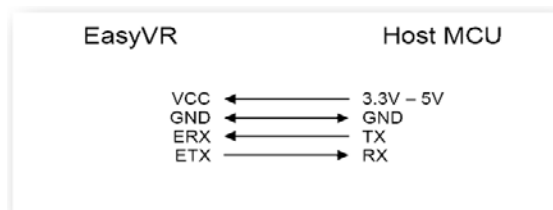


Fig. 5 MCU based host connection

The configuration at power/On is 9600 baud, 8 bit data, No parity, 1 bit stop. The baud rate can be changed as necessary to operate in the range 9600 - 115200 baud.

The communication protocol only uses printable ASCII characters, which can be divided in two main groups:

- Command and status characters, respectively on the TX and RX lines, chosen among lower-case letters.
- Command arguments or status details, again on the TX and RX lines, spanning the range of capital letters.

Each command is sent on the TX line with zero or more additional argument bytes and receives an answer on the RX line in the form of a status byte followed by a zero or more arguments. The proposed design model is presented as shown in Fig. 6.

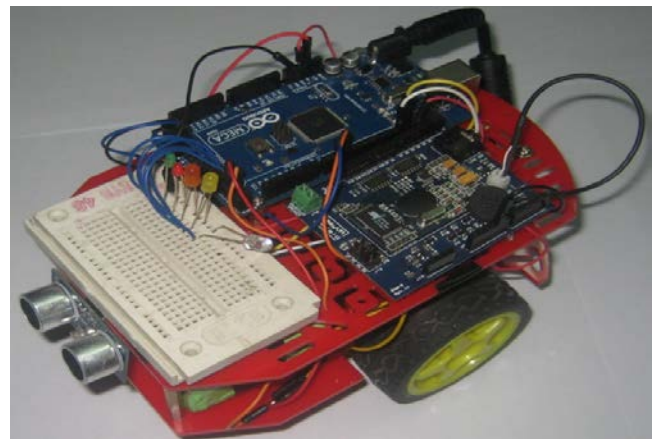


Fig. 6 Design of a vehicle safety control model

### III. CONCLUSION

As it can improve vehicle safety issues and provide the driver with more vehicle operation convenience, which leads to minimizing the number of vehicle accidents, an automated vehicle safety control is presented in this paper. The design is implemented using an Arduino microcontroller. An EasyVR is used for voice recognition operation and training. For the vehicle speed control, the necessary action is issued based on the vehicle-to-vehicle distance measure. To maintain a proper road and situation focus, the design has been provided with different other automated control functions; radio control, flasher control, light control. All of these control options are managed according to human voice commands provided by the EasyVR module. The proposed model has been designed, built, and tested providing suitable results as expected, which show the potential for further future work with enhanced vehicle safety automated operations.

### ACKNOWLEDGMENT

The authors would like to thank the Deanship of the Academic Research at The University of Jordan for supporting this research.

REFERENCES

- [1] ASIRT, “<http://www.asirt.org/home.aspx>”, Association for Safe International Road Travel, 2013.
- [2] H.R. Al-Masaeid, A.A. Al-Mashabkeh, and A.M. Qudah, “Economic Costs of Traffic Accidents in Jordan,” *Journal of Accident Analysis and Prevention*, vol. 31, no. 4, pp.347-357, 1999.  
[http://dx.doi.org/10.1016/S0001-4575\(98\)00068-2](http://dx.doi.org/10.1016/S0001-4575(98)00068-2)
- [3] Mitsubishi, “<http://www.mitsubishi.com/e/mitsubishihp/>”, Mitsubishi.com Committee, 2013.
- [4] P. Shakouri, A. Ordys, and M.R. Askari, “Adaptive cruise control with stop & go function using the state-dependent nonlinear model predictive control approach,” *ISA Transaction*, vol. 51, no. 5, pp. 622-631, 2012.  
<http://dx.doi.org/10.1016/j.isatra.2012.05.001>
- [5] S.E. Li, K. Li, and J. Wang, “Economy-Oriented Vehicle Adaptive Cruise Control With Coordinating Multiple Objectives Function,” *International Journal of Vehicle Mechanics and Mobility*, vol. 51, no.1, pp. 1-17, 2013.
- [6] S. H. Jeong, J. E. Lee, S. U. Choi, J. N. Oh and K. H. Lee, “Technology Analysis And Low-Cost Design Of Automotive Radar For Adaptive Cruise Control System,” *International Journal Of Automotive Technology*, vol. 13, no. 7, pp. 1133–1140, 2012.  
<http://dx.doi.org/10.1007/s12239-012-0116-2>
- [7] A. Rogowski, “Industrially oriented voice control system,” *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 3, pp. 303–315, June 2012  
<http://dx.doi.org/10.1016/j.rcim.2011.09.010>
- [8] C. Englund, “Speech recognition in the JAS 39 Gripen aircraft - adaptation to speech at different G-loads,” Master Thesis, Royal Institute of Technology, Stockholm, 2004.
- [9] Arduino [mega2560](http://arduino.cc/en/Main/arduinoBoardMega2560), “<http://arduino.cc/en/Main/arduinoBoardMega2560>”, 2013.
- [10] EasyVR Arduino Shield, “<http://www.veear.eu/products/easyvr-arduino-shield/>”, Veeear company, 2013.