

Black Box with Intelligent Vehicle System

Prafulla U. P, Yuvasree G. S, Ramya Gowda, Rashmitha H. R.
Students, Department of EEE,
Ghousia College of Engineering, Affiliated to VTU,
Belagavi

Rajalakshmi
Assistant Professor, Dept. Of EEE,
Ghousia College of Engineering, Affiliated to VTU,
Belagavi

Abstract – The main purpose of this paper is to develop a prototype model of the Vehicle Black Box System VBBS that can be installed into any vehicle all over the world. This prototype can be designed with minimum number of circuits. The VBBS can contribute to constructing safer vehicles, improving the treatment of crash victims, helping insurance companies with their vehicle crash investigations, and enhancing road status in order to decrease the death rate.

Keywords- Vehicle, Black Box, Microcontroller, Computer interface.

I. INTRODUCTION

According to the World Health Organization, more than a million people in the world die each year because of transportation-related accidents [1]. In order to react to this situation, the black box system draws the first step to solve this problem that crosses national boundaries and threatens the safety and health of people worldwide. Introduced to a part of the United States market in 1999, the black box system proved to be efficient [2]. However in the latter case, the system was embedded in the vehicle [3-5]. Therefore, in addition to improving the treatment of crash victims and the road status in order to decrease the death rate, constructing safer vehicles, and helping insurance companies with their vehicle accidents investigations, the main purpose of this paper is to develop a black box system that can be installed to any vehicle all over the world.

Like flight data recorders in aircraft, “black box” technology can now play a key role in motor vehicle crash investigations [6]. A significant number of vehicles currently on the roads contain electronic systems that record information in the event of a crash. That is why it is so important to have recorders that objectively track what goes on in vehicles before, during and after a crash as a complement to the subjective input that is taken usually from victims, eye witnesses and police reports.

This system is committed mainly to two approaches. The first one is how to detect and record data from the vehicle. The second is how to present the data recorded to the user in a simplified way. To implement the first approach, some major components and different type of sensors were used. While the second approach was implemented using a Visual Basic .NET computer program. This program receives the data serially from the black box memory, presents it in real-time graphics and finally saves it to a formal excel report for future use.

In order to know what type of sensors should be installed into the vehicle, research was carried out to identify the main information needed for better accident analysis. After filtering the information and taking into consideration what could be done and what could help the most, the following data were found to be the most important ones needed after an accident: Belt status, Road condition, Brake status, Speed Measurement, Position of the accident, Main Lights status.

In this paper, we describe in section II the hardware resources dedicated to VBBS system. The software part is subject of the section III. Finally, a conclusion is given in section IV.

II. HARDWARE RESOURCES

The hardware part consists on the sensors and the black box installed into the vehicle. This part mainly collects the status of the sensors and saves it to the microcontroller’s EEPROM.

A. Sensors

1. Speed Sensor

An inductive proximity sensor HYP-18RL8P is used to detect the number of wheel turns per unit of time [7]. This sensor can be used for the detection of all good electrical conductors (materials). It has a sensing distance of maximum 8 mm, and a Current Voltage 12 – 24VDC. Note that in case it detects some metal; it will have an output voltage of exactly what was given to its input voltage. However, when there is no detection, the output will be zero volts.

Because of the 14VDC of its output, a voltage divider was placed between the sensor’s output and the microcontroller’s input pin to decrease the voltage to 5VDC. Finally in order to detect the speed in the PICBasic Pro, the ‘COUNT’ command was used.

2. Water Sensor

The water sensor module used is very sensitive to water level, and its sensitivity can be increased by turning the potentiometer. This module consists of two uncovered and unconnected wires equidistant of each other, as seen in the figure 1. When this sensor detects water, the two wires will be connected through water conductivity, and the status of the sensor will be on.

The module is implemented inside the black box, whereas the sensors are placed in the front of the vehicle behind the front wheels to detect the first change in the road status.

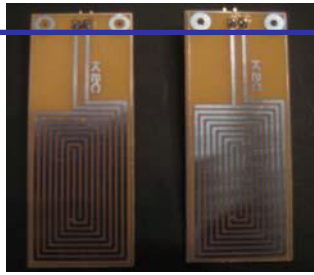


Fig. 1. Equidistant Wires

3. Switches

To stimulate accident sensors and belt sensors push button are used [8]. They have three pins, one pin for the common (C) and the other two for normally closed (NC) and normally open (NO). In the black box system, nine push buttons were used. One of them is used as the belt sensor and the eight others as the accident sensors. In all cases the NO pin was used because a zero volt was needed as input to the microcontroller, when the push button is pressed. This justifies the use of pull up resistors for all these nine inputs of the PIC.

a- Accident Sensors

Because the purpose is to detect only accidents that hit the chassis of the vehicle, only eight pushbuttons are used around the vehicle. The distribution of sensors is shown in figure 2.

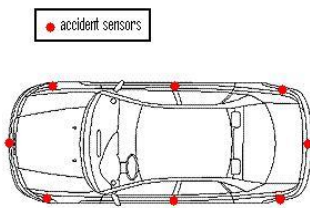


Fig. 2. Accident Sensor Distribution

When an accident is detected logic '0' will be applied to the corresponding input pin of the microcontroller. In other cases the input of the 8 pins connected to the accident sensors is logic '1' due to the pull up resistors connected to these pins.

b- Belt Sensor

One push button is used to detect the place of the seat belt during the drive. The seat belt of the driver is only taken into consideration in this paper, but can be extended to include all the belts of the vehicle, depending on the traffic regulations of each country.

The push button is placed on the seatbelt and gives a logic 'zero' when the belt is used and a logic '1' when the belt is not placed by the driver.

4. Lights Sensor

For a particular vehicle, the important lights in the analysis of an accident are the flashers, the brake lights and the rear lights. The rear lights are needed in the analysis to know the direction of the vehicle. The brake lights are needed to show the brake status seen by the driver behind, before the accident. Finally, the flashers will also be useful in the analysis

of the accident in order to determine whether the driver has used them properly.

This sensor consists of a small magnetic switch, connected to the input of the microcontroller, and a bobbin wire all around it. This wire is in series with the light connection of the vehicle. Whenever the light is "ON", a current will pass in the wire, creating a magnetic field. This magnetic field will cause the switch to close. Thus the microcontroller will detect 5V in case of the flashers, and 0V in case of the other lights. However, an additional circuit was connected between the sensor's output and the microcontroller's input pin for the flashers sensors. This circuit converts an ON/OFF period to an ON period. This circuit is added so the sensor's status will be ON for the total duration when the flasher is turned ON. Figure 3 shows the different parts of the sensor.

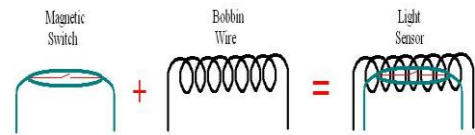


Fig. 3. Light Sensor

5. Brake Sensor

The brake sensor is a type of switch implemented in the vehicle underneath the brake footstep. This switch controls the brake lights. In order to know if the driver pushed the brake during the accident, this switch is connected to the input of the microcontroller. However as mentioned above, the brake sensor has an output of 14V. So it needs a voltage divider in order to have 5V at the input of the brake switch pin of the microcontroller.

B. Digital processing

In order to control all these sensors and their inputs, a digital process can be used [9]. As prototype a microcontroller

is selected to control the VBBS. This will allow the control circuit to be realized by a minimum of circuits. Thus resulting in less maintenance, minimization of the occupied space on a PCB, and reduced costs.

For this prototype, the main need was a large EEPROM, to enable recording as much data as possible about the accident, and a large amount of inputs. Thus, PIC16F877A was used because it has 8 Kbytes of Flash Program Memory, 368 bytes of Data Memory, 256 bytes EEPROM data memory, 15 interrupts, 8 input channels, 5 I/O Ports, and many other characteristics.

1. Microcontroller's Connections

The inputs to the microcontroller, which contain information about the accident, are distributed as follows: 8 pins for the 8 accident sensors, 8 for the lights (flashers included), 1 for the speed sensor, 1 for the belt sensor, 1 for the brake switch, 1 for the water sensor, and 1 pin for the serial switch. The outputs of the microcontroller are three pins for the LEDs, and two pins for serial transmission and reception. The three LEDs will show the user whether the black box is recording, waiting, or transmitting.

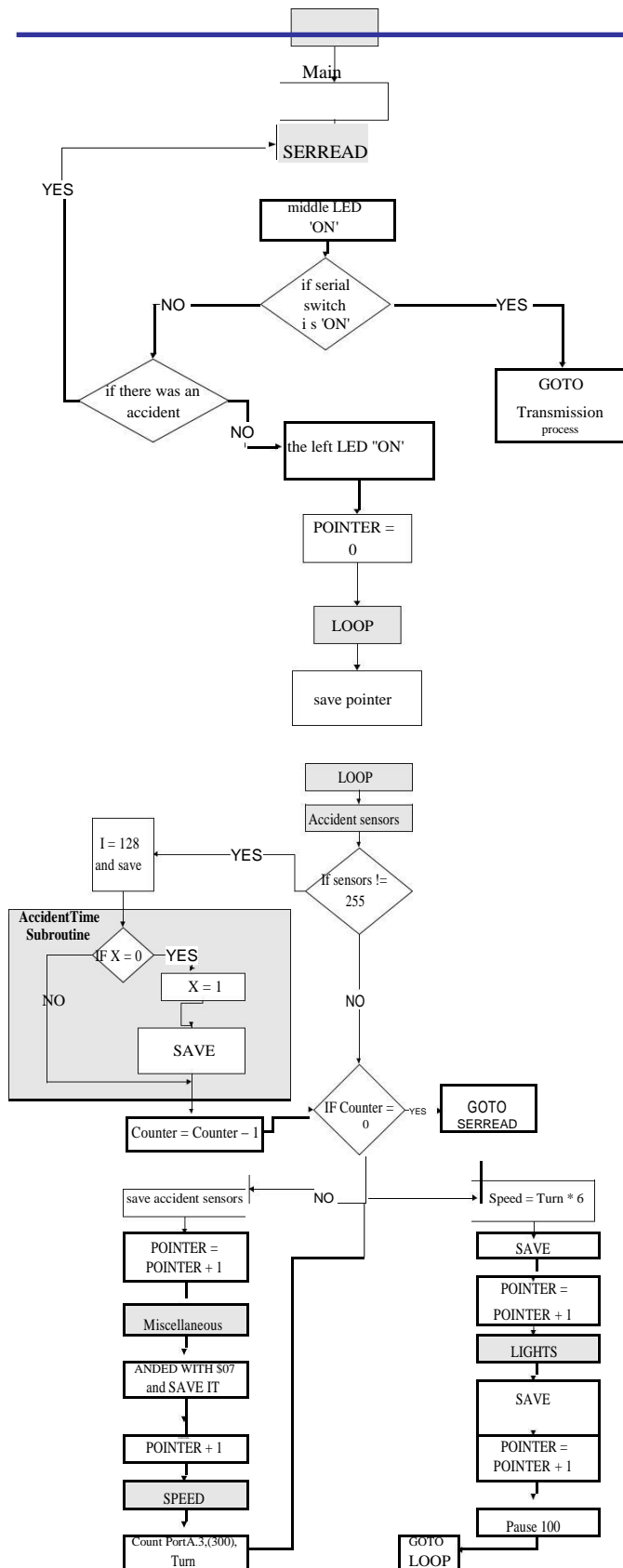


Fig.5. Flow Chart of the Main Program

After an accident, the microcontroller will take 20 samples of each sensor or a total of 10 seconds of data after the accident

before going to sleep. This sleep mode ends if the switch on PortA.4 is switched to '0'. When this happens, the microcontroller program will go to the "SerialTest" where the data saved on the EEPROM will be transmitted serially to the Visual Basic.NET program in order to be visualized using an interface that could be helpful in analyzing the accident. The value of the Pointer is sent first, followed by the Accident Pointer and finally the remaining data will be transmitted in a loop from the oldest sample to the newest one. Add to that, a method of handshake is used between the microcontroller and the VB.NET program in order to be sure that there is no data lost during the transmission. The overall explanation of the microcontroller's program is found in the flowchart of the figure 5.

C. Data Transmission

One of the main characteristics of this system is that only the authorized people could access the true interpretation of the black box data. This is due to the interface between the microcontroller and the computer, in addition to the specific programs written for both of them.

As described previously, the data related to the accident is saved in the EEPROM of the 16F877A PIC. An interface is thus needed to transmit this data from the EEPROM to a specific computer, where a VB.NET program can use them to begin the analysis.

The choice of the microcontroller's transmission protocol was the standard asynchronous format using 8 data bits, no parity bit and one stop bit with a 9600 baud rate. Since the complexity is in the interpretation of the data and not in the transmission, the need was for a format that guarantees minimum simplicity with maximum reliability. In addition, a MAX232 is used as an intermediary station, to connect the PIC to the serial port of the computer.

III. SOFTWARE RESOURCES

After covering the hardware part of the Black Box system, it's now time to take a look at the main functions of the Visual Basic .NET program and how the user will be shown the data recorded before, during, and after the accident. The main duties of this program are to receive data serially, interpret, and finally display it to the user in a clear and simple way.



Fig. 6. Main Page of the VB.NET Program

The VB .NET program consists of two functional and four graphical forms. One of the functional forms is the Main Page as shown in the figure 6. The EXIT button is used to exit the program while the SERIAL button is used for the serial reception of data. Also found on this page are the steps that the user should follow in order to have a successful transmission.

Once the SERIAL button is pressed; the serial port settings will be set as the ones used for data transmission in the microcontroller, 9600 baud rate, 8 data bit, no parity, and 1 stop bit. In order to have a reliable transmission, a handshake program was written to interface the microcontroller and the VB.NET program. The VB.Net program sends an 'A' character to the PIC before receiving every byte. Once it is received, the program will save it into its corresponding place. After the transmission is completed successfully the corresponding bytes are converted to bits and the program will move on to the second functional form, the Display Page. Figure 7 shows this page which contains 7 buttons; 4 for display purposes, 1 to open and save the report excel file, the EXIT and the BACK buttons. Each one of these buttons has its own mouse click event. So each one of them has its own program that will run once clicked on.

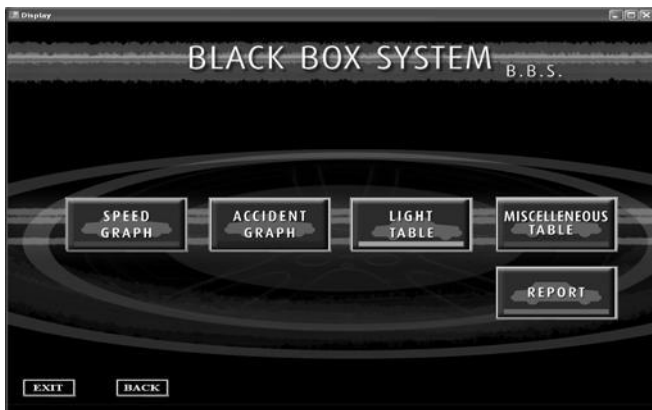


Fig. 7. Display Page of the VB.NET Program

In order to show how the program displays the recorded data in a clear and simple way using its 4 display forms, an accident example will be given and the result will be shown in the display forms. In all the forms, an interval of 31 seconds will be used from the oldest value recorded to the newest one. In this example the following will be considered:

- The speed will vary before the accident from 20 to 70 km/h. Then it will decrease to 0km/hr after the accident.
- The accident position will be next to the driver and after 3 seconds the vehicle is hitting again in its left back.
- The right flashers will be on from the fifth to the tenth second.
- There will be water on the road before the accident, so the water sensor will be ON during the time interval of the accident.

The brake and the brake lights will be ON between 7.5 seconds and 8.5 seconds then between the 20 and 27.5 seconds.

- The belt sensor will be off.
- The other lights unspecified during this example were off.

When the user presses on the “Speed Graph” button, the following form (figure 8) will open.

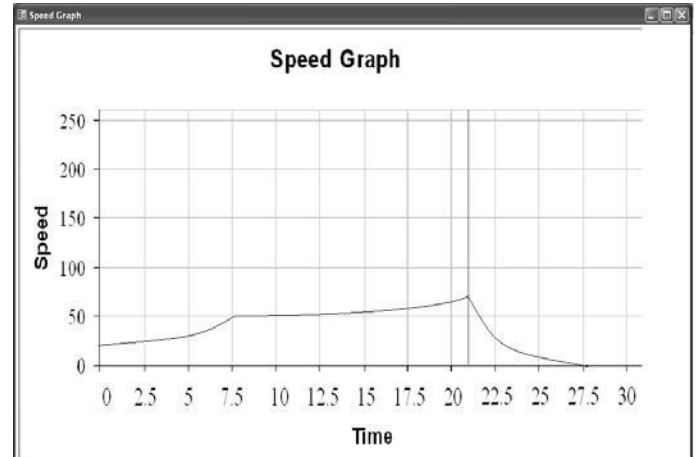


Fig. 8. Speed graph sample

As shown in figure 8, the y-axis shows the speed from 0 to 255km/h and the x-axis shows the time in second from 0 to 31 seconds. Also it is important to know that the red line represents the accident time.

Time (Sec.)	Bell	Water Sensor	Drinkes
0	OFF	OFF	OFF
0.5	OFF	OFF	OFF
1	OFF	OFF	OFF
1.5	OFF	OFF	OFF
2	OFF	OFF	OFF
2.5	OFF	OFF	OFF
3	OFF	OFF	OFF
3.5	OFF	OFF	OFF
4	OFF	OFF	OFF
4.5	OFF	OFF	OFF
5	OFF	ON	OFF
5.5	OFF	ON	OFF
6	OFF	ON	OFF
6.5	OFF	ON	OFF
7	OFF	ON	OFF
7.5	OFF	ON	OFF
8	OFF	ON	OFF
8.5	OFF	ON	OFF
9	OFF	ON	OFF
9.5	OFF	ON	OFF
10	OFF	ON	OFF
10.5	OFF	ON	OFF
11	OFF	ON	OFF
11.5	OFF	ON	OFF
12	OFF	ON	OFF
12.5	OFF	ON	OFF
13	OFF	ON	OFF
13.5	OFF	ON	OFF
14	OFF	ON	OFF
14.5	OFF	ON	OFF
15	OFF	ON	OFF
15.5	OFF	ON	OFF
16	OFF	ON	OFF
16.5	OFF	ON	OFF
17	OFF	ON	OFF
17.5	OFF	ON	OFF
18	OFF	ON	OFF
18.5	OFF	ON	OFF
19	OFF	ON	OFF
19.5	OFF	ON	OFF
20	OFF	ON	OFF
20.5	OFF	ON	OFF
21	OFF	ON	OFF
21.5	OFF	ON	OFF
22	OFF	ON	OFF
22.5	OFF	ON	OFF
23	OFF	ON	OFF
23.5	OFF	ON	OFF
24	OFF	ON	OFF
24.5	OFF	ON	OFF
25	OFF	ON	OFF
25.5	OFF	ON	OFF
26	OFF	ON	OFF
26.5	OFF	ON	OFF
27	OFF	ON	OFF
27.5	OFF	ON	OFF
28	OFF	ON	OFF
28.5	OFF	ON	OFF
29	OFF	ON	OFF
29.5	OFF	ON	OFF
30	OFF	ON	OFF
30.5	OFF	ON	OFF

Fig. 9. Miscellaneous Table Sample

Figure 9 shows the “Miscellaneous Table” button when it is open. The true icon represents an ON status of the sensor while the false icon represents its OFF status. This table represents

the Belt, water, and brake sensors for each 0.5 second sample saved into the microcontroller’s EEPROM. The same method is used to show the Lights status when the “Lights Table” Button is pressed; however this table shows 8 lights sensors as shown in the figure 10.

Fig. 10. Lights Table Sample

The “Accident Graph” button, when pressed will open the form shown in the figure 11. This form shows 20 vehicle samples (after the accident’s samples) . On each vehicle 8 icons are placed exactly where the accident sensors were installed in the vehicle. The true icon represents ON status of the sensor while the false icon will represent an OFF status of the sensor. Note that only 20 samples were shown in this form since all the accident sensors are OFF before the accident.

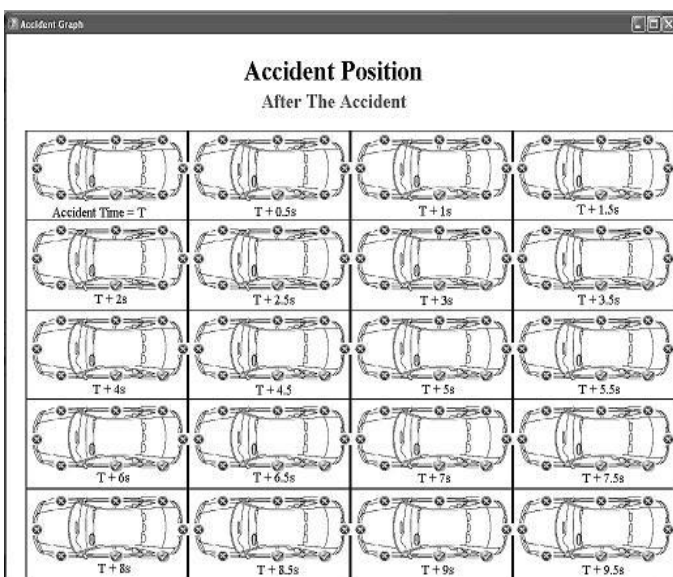


Fig. 11. Accident Graph sample

As expected, the forms showed in the last 4 figures matches exactly the description of the example given before displaying these forms.

The only button which remains unexplained is the Report button. Once pressed, it will send to a formal EXCEL report all the data received from the black box so they can be saved and printed. Additional information should be entered in this formal report by the user. This information includes Vehicle, type Driver name, and accident information.

IV. CONCLUSION

This paper has presented a new vision for the vehicles industry, which is the Black Box system used for vehicles. A full and detailed description was made for every part of this system. This paper has also offered a user friendly visual basic program to analyze the data of the accident. In addition, the transmission method between the two parts has been introduced and developed.

The Black Box system built can be implemented in any vehicle. As soon as the driver runs the motor, this system will begin saving the events of the corresponding vehicle. The last 21 seconds are always saved in the EEPROM of the Black Box, and in case of an accident, an additional 10 seconds of events after this accident will be saved. The data saved can be retrieved only after the accident for privacy purposes. Using serial transmission, a VB.NET program will read the data from the EEPROM and display it to the user in Graphical format. In addition, a detailed report will be given to the user containing all necessary information.

ACKNOWLEDGMENT

Authors would like to acknowledge the financial support from NDU, and the CAD tools from Microchip Company.

REFERENCES

- [1] G. Hayes, F. Blosser, “Motor Vehicle Crashes Claim More than a Million Lives Worldwide”, CDC Injury Center Media Relations, Press Release, April, 2004.
- [2] <http://www.airbagcrash.com> (General Motor Event Data Recorders)
- [3] Thomas K. Kowalick, “Black Boxes: Event Data Recorders”, MICAHA, summer 2005.
- [4] Thomas K. Kowalick, “Black Boxes: Event Data Recorder Rulemaking for Automobiles”, MICAHA, summer 2006.
- [5] Thomas K. Kowalick, “Fatal Exit: The Automotive Black Box Debate”, Wiley, IEEE Press, Feb. 2005.
- [6] J. Lawrence and T. Newbery, “Crash Data Recorders in Motor Vehicles”, MacInnis Engineering, Associates, 2003.
- [7] <http://www.hyelec.co.kr> (Inductive proximity sensor HYP-18RL8P)
- [8] Chidester, Augustus, et al, “Recording Automotive Crash Event Data”, International Symposium on Transportation Recorders, Arlington, VA, 1999.
- [9] L. Dae Geun, J. Se Myoung, L. Myoung Seob, “System on Chip design of Embedded Controller for Car Black Box”, Intelligent Vehicles Symposium IEEE 2007, pp. 1174-1177, 13 June 2007.