Article can be accessed online at http: //www.publishingindia.com

An Experimental Safety Analysis using FTA for A Ball Position Control System

Kadupukotla Satish Kumar*, Panchumarthy Seetha Ramaiah**

Abstract

FTA is a top down, deductive failure analysis method in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events. It is widely used in the aerospace, automotive and other safety-critical intensive systems. This work addresses the use of FTA by using an experiment for safety-critical ball position control system. The work presented here provides a general example illustrating how FTA can be effectively applied to an 8-bit micro-controller (Chip 89S52) based computer control system having little or no hardware protection. The safety analysis reveals several design deficiencies and physical faults for which modifications are needed. This paper also found that, when properly implemented FTA at the right point in the Software Development Life cycle, it makes requirements, design and code reviews more effective. It also identifies single point failures due to software.

Keyword : Safety Critical Systems, Hazards, Safety Integrity Levels

1. Introduction

Software plays a primary and central role in the functioning of various kinds of systems like real-time computer control systems, embedded computer systems, etc. In fact, software is increasingly being used to handle safety-critical computer systems which were formerly controlled by humans or hardware [1]. With the increased dependency of systems on software and with its evolving size and complexity, it has now become the major contributor to system failures and a primary threat to system safety and reliability.

Weaknesses within the software portion of systems are of particular concern [8] [9] [10]. However as software increases in size and complexity it cannot typically be exhaustively verified. As a result, latent software faults have become a leading source of safety and quality issues for devices using software. Coming to medical devices, which are using software, according to the *Office of Science and Engineering Laboratories* (OSEL), USA's annual report 2011, over 20% of all medical device recalls in the United States were due to software failures [2].

FTA is one of the risk analysis method which is widely used to identify risk and possible failures [3] [5]. FTA was initially developed in 1962 at Bell Laboratories by H.A. Watson, for a U.S. Air Force Ballistics Systems Division project to evaluate the Minuteman I Intercontinental Ballistic Missile (ICBM) Launch Control System. The use of fault trees has since become very popular and it is now often used as a failure analysis tool by reliability experts. Following the first published use of FTA in the 1962 Minuteman I Launch Control Safety Study, Boeing expanded the use of FTA to the entire Minuteman II system in 1963-1964. FTA received extensive coverage at a 1965 System Safety Symposium in Seattle sponsored by Boeing and the University of Washington. Boeing began using FTA for civil aircraft design around 1966. FTA is carried out to

- To exhaustively identify the causes of a failure
- To identify weaknesses in a system
- To assess a proposed design for its reliability or safety

* Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India. Email: satishkmca@gmail.com

** Department of Computer Science and Systems Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India. Email: psrama@gmail.com

- To identify effects of human errors
- To prioritize contributors to failure
- To identify effective upgrades to a system
- To quantify the failure probability and contributors
- To optimize tests and maintenances

FTA methodology is one of the risk analysis techniques recommended by international standards [12] [13]. It is a sys- tematic process to identify possible failures in a system. FTA is mostly adapted for equipment and material failures, but in a broad sense, human error, performance and software errors can also be included. By applying the FTA methodology during the various phases of a product's life cycle, the methodology provides a systematic strategy for examining all the ways in which a product can fail. The results of FTA in turn affect the product design and process development. The benefits of using FTA are as follows:

- The fault tree explicitly shows all the different relationships that are necessary to result in the top event
- In constructing the fault tree, a thorough understanding is obtained of the logic and basic causes leading to the top event
- The fault tree is a tangible record of the systematic analysis of the logic and basic causes leading to the top event
- The fault tree provides a framework for thorough qualitative and quantitative evaluation of the top event

This paper describes an experimental safety analysis using FTA for Safety-Critical Computer Systems. When properly implemented FTA at the right point in the Software Development Life cycle, it makes requirements, design and code reviews more effective. It also identifies single point failures due to software.

The rest of the paper is organized as follows. The experimental setup and implementation is described in section 2. FTA with results described in Section 3 and section 4 gives conclusion.

2. Embedded Computer Based Ball Position Control System (BPCS)

The objective of this research work is to enlighten the use of FTA for an embedded control system "Ball in the Tube system" through the development of an experiment [4][6] [7] [10] [14]. The central objective of the BPCS system is to regulate the flow of air into a plastic tube so as to keep a small light weight ball suspended at a predetermined height called the set-point. Increasing the flow raises the ball and decreasing the flow lowers it. The BPCS experiment consists of: 3-foot long white plastic tube, light weight ball, DC motor fan, and ultrasonic sensor circuit and 89S52 micro controller.

The vertical 3-feet long clear plastic tube attached to a stand, which contains a light weight ball inside, a DC motor fan at the bottom to lift the ball, and an ultrasonic sensor at the top to sense the balls height. The tube is connected to the DC motor fan inlets via an input manifold which has an inlet at the bottom as indicated. There is an output manifold at the top of the plastic tube with an outlet as shown. The presence of the manifolds is a key part of the experiment. The ultrasonic sensor circuitry detects the position of the light weight ball and the micro-controller regulates the power supply applied to the DC motor fan so as to control the air flow into the white plastic tube, keeping the light weight ball at a predetermined height.

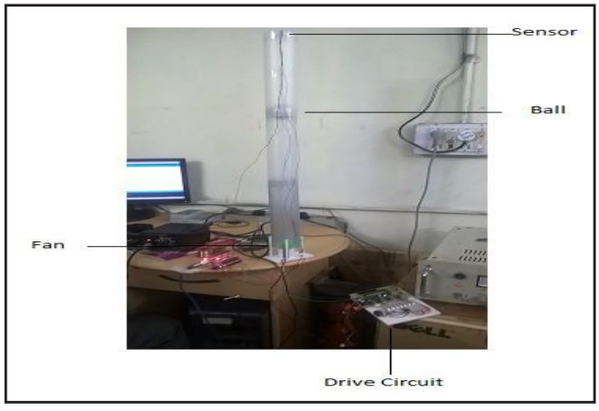

Fig. 1. Ball Position Control System (BPCS)

2.1 BPCS Explanation

The light weight ball position control system experiment is a system composed of five modules, where one of them has a DC motor fan to blow air into the white plastic tube moving a styrofoam light weight ball inside it as shown in figure 1. The modules are instrumented with an ultrasonic sensor to sense the light weight ball height. The air flux coming from the DC motor fan is transferred to the white plastic tube using a pressurization cylinder whose objective is to reduce the turbulence produced by the DC

motor fan, which delivers an almost laminar flux. Each module is coupled with the others by the operational objective of the system is to maintain the light weight ball at a predefined height in the plastic tube. The software in the micro-controller provides the control mechanism; it implements a proportional controller that is, the output signal data is proportional to the amount of error in the balls position relative to the set-point. The software is implemented in assembly language, which is assembled and downloaded into the micro-controller.

A Block diagram of the BPCS system is shown in figure

Each module is coupled with the others by a common manifold. The base box corresponds to the input manifold. The airflows into the manifold by the unique input located at the left side of the box. The air in the input manifold is distributed over each module in a parallel way. Depending on the power applied by the DC motor fan and the input size of the manifold, the air flux continues its trajectory moving the ball inside it. The air from the plastic tube is combined again in the output manifold and ejected through the output, in the right side of the box. The BPCS dynamical model comprises an energy transfer by airflow from the DC motor fan to the light weight ball. This transfer is typically nonlinear. The system has five main functional modules:

(i) Ball-position-controller
(ii) DC motor fan and drive circuit
(iii) Ultrasonic sensor
(iv) Display module
(v) USB-Interface.

The micro-controller generates a Pulse Width Modulation (PWM). It is a modulation technique used to encode a message into a pulsing signal. Although this modulation technique can be used to encode information for transmission, its main use is to allow the control of the power supplied to electrical devices, especially to inertial loads such as motors. PWM generates a control signal to regulate the speed of the DC motor fan. The higher the duty cycle of the PWM signal, the faster the fan runs. An output control circuit between the micro-controller and the DC motor fan provides hardware "stops that ensures that the PWM signal is within the specified "safe operating range.

If the Pulse Width Modulated cycle is out of the meticulous range, the fault pointer light is switched-on and the system is closed. This fault verification function ensures a safe functional range for the DC-motor even in the occurrence of a software malfunction. The altitude of the ball is determined by the sensor. The ultrasonic sensor generates high-frequency sound waves and evaluates the echo which is received back by the sensor, measuring the time interval between sending the signal and receiving the echo to determine the distance the ball is from the sensor. If the ball is nearer to the sensor, the time interval between the transmission of sound wave and reception of echo will be very small. The output power of the sensor is sampled by the converter Analog to Digital, which is embedded in the micro-controller. The inward power is transformed into a one-byte value in an ADC register, which demonstrates elevation of the ball. The ADC is standardized so that a 016 corresponds to the ball being at the base of the tube, and FF16 corresponds to the ball at apex.

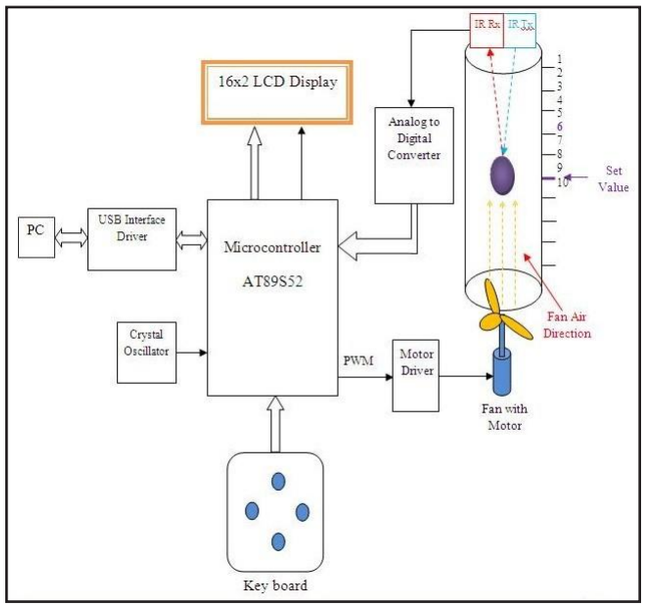

Fig. 2. Block Diagram of Ball Position Control System (BPCS)

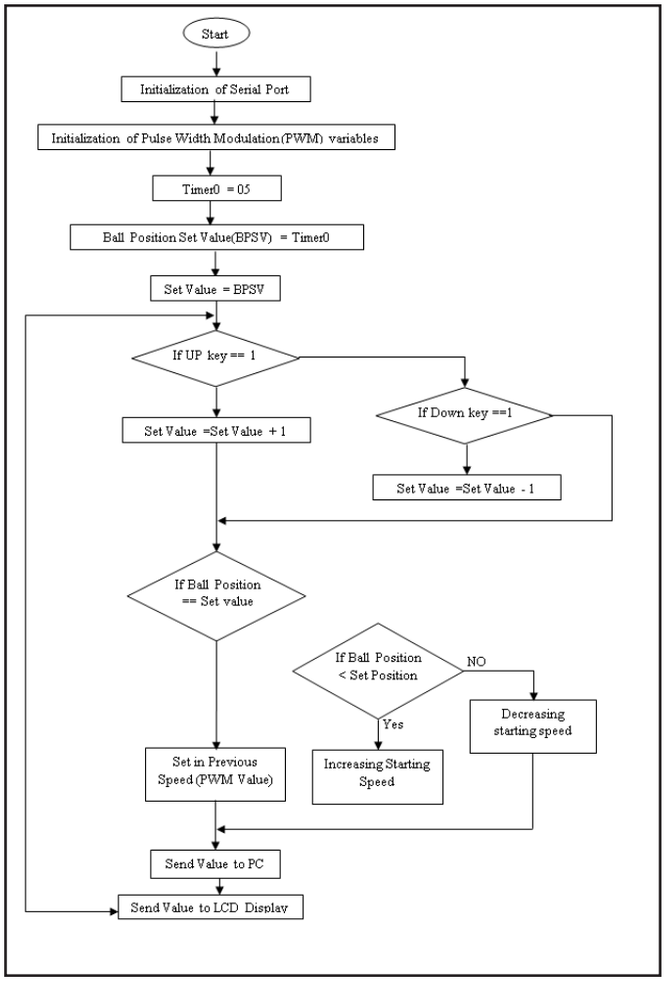

Fig. 3. Flow Chart for the Main Program of the BPCS

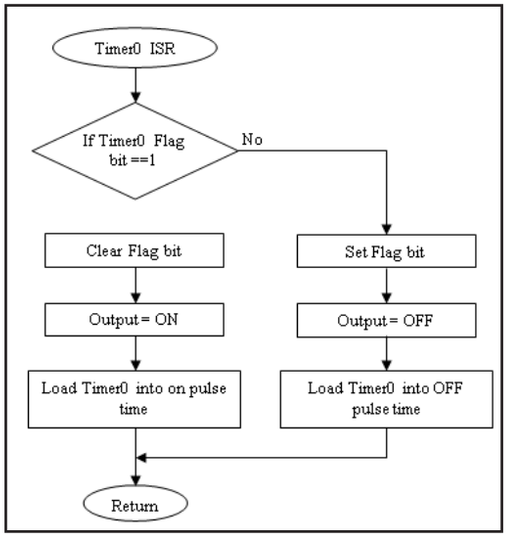

Fig. 4. Flow chart for the Interrupt Service Routine (ISR) of the BPCS

The error of the ball point is calculated by deducting the calibrated ADC value from the set-point value. The variation is multiplied by a constant to determine the rectification to the PWM signal. If the intensity of the error is high, then more correction is required.

If the calculated ADC value is below the set-point variable, the ball is above the set-point and the micro-controller enlarges the duty cycle to increase the fan speed and raise the ball. On the other hand, if it is below the set-point, the duty cycle is reduced to lower the ball. The input control circuit sandwiched between the ultrasonic sensor and the micro-controller make sure that the voltage at the input port is not more than 5 volts. The I/O control circuits decrease the risk of damage to the system from unforeseen ultrasonic sensor failures. For example, if the sensor shorts out and outputs an enormously high power (e.g. 10 V) to the micro-controller without a safety control circuit, then the sensor would be harshly smashed and would get damaged and would have to be replaced.

Figure 3 shows the program flow chart for the ball position control system. In the flow chart, first initialization of serial port and then initialization of PWM variable are done. Then the software enters an infinite loop in which the analogy voltage from the sensor is read, converted to a number, and stored in an ADC register. The duty cycle is then calculated and the output pin set, high or low, to achieve this cycle. The timer interrupt in the micro-controller triggers an Interrupt Service Routine (ISR) every 5 microseconds: a counter, which is initially set to 100 and is decremented by 1 on each interrupt. To achieve a given duty cycle (say 50%) for the PWM signal, the output pin is set low for each interrupt until the counter reaches the given value (i.e. 50) then it is set to high until the counter is decremented to 0. At 0 the counter is reset to 100 and the output pin to low and the cycle is repeated, thus producing a 0.5 ms (milliseconds) period with a 50% duty cycle.

3. FTA FOR BPCS

In the process of FTA, analysis of events is carried out to identify the causes for that event which can lead to errors or faults. A complete understanding of the system is required to carry out an FTA.

3.1 Fault Analysis

For the Ball in the Tube experiment, a careful study of the system resulted in the following fault trees along with their descriptions.

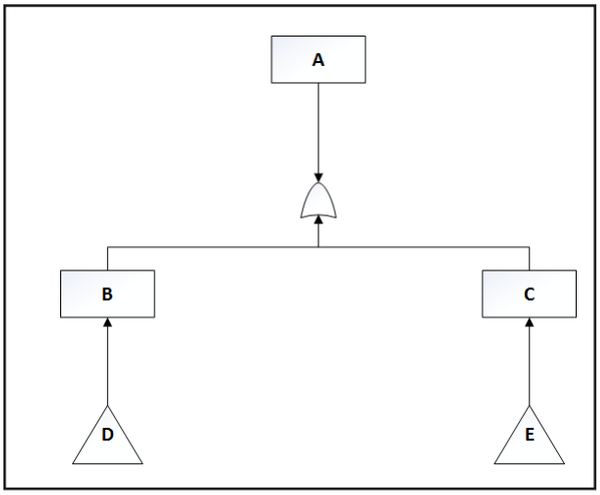

Fig. 5. Ball Not in Set Position

Figure 5. shows the FTA diagram for the event when the ball is not at the set position. It can be either at a lower position than the set position or at a position higher than the set position.

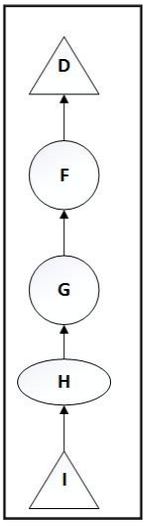

Fig. 6. Ball Higher than the Set Position

Figure 6. shows the events that can lead to a condition when the ball is at a higher position than the set position which will damage the sensors. For this to happen, the fan has to run very fast, which can happen when the output is high.

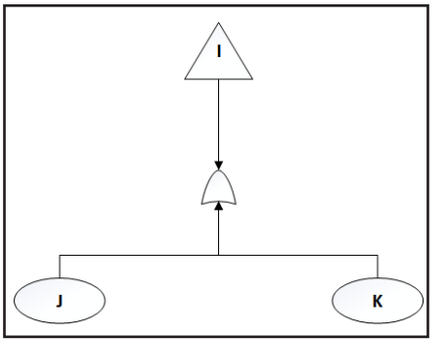

Fig. 7. Output Too High

Figure 7 shows the condition where the output can be high. It can happen in either of the two conditions that the output of the pin D4 is stuck high or input duty cycle to ISR stuck high.

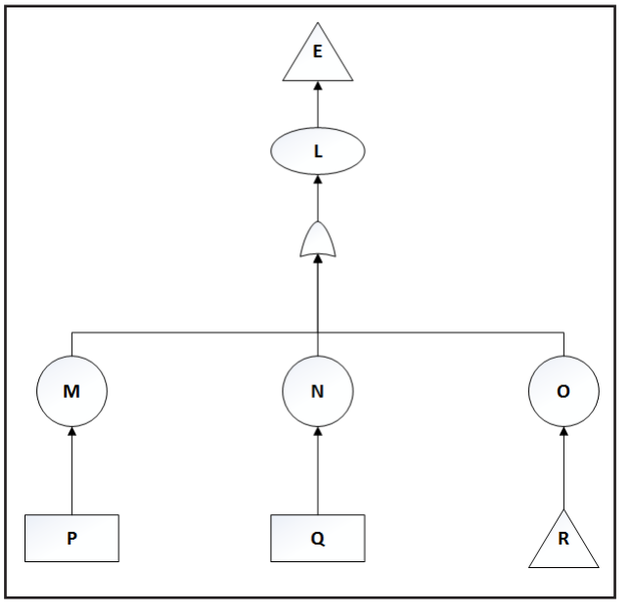

Fig. 8. Ball at Bottom

Figure 8 shows an event where the ball is stuck at the bottom of the tube. It can happen when either the system does not run or the fan runs too slow or the fan does not run. The system will not run if there is a loss of output signal to drive circuit. The fan will not run if the output is too low.

The nodes in the FTA diagrams have the following descrip- tions:

1. A: Ball not in set position
2. B: Ball higher than the set position

3. C: Ball lower than the set position
4. D: Connector
5. E: Connector
6. F: Ball at top of the tube/Sensor damaged
7. G: Fan runs too fast
8. H: Output signal too high
9. I: Connector
10. J: D4 pin output high
11. K: Input duty cycle to ISR stuck high
12. L: Ball at bottom of the tube
13. M: System does not run
14. N: Fan runs too slow
15. O: Fan does not run
16. P: Loss of output signal to drive circuit
17. Q: Output too low
18. R: Connector
19. S: Failure to initialize input register
20. T: Failure to initialize output register
21. U: Failure to initialize ISR
22. V: Loss of output signal to drive circuit

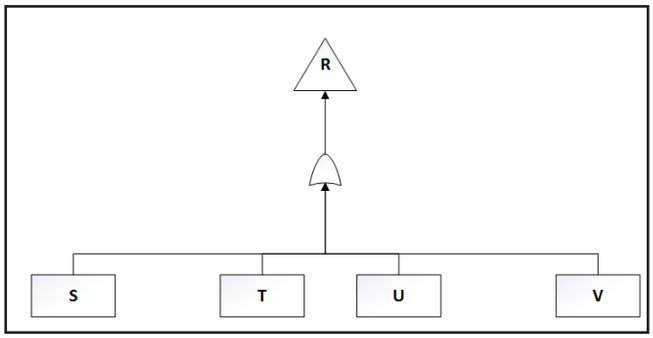

Fig. 9. Fan Does Not Run

Figure 9. shows the event that the fan does not run. It can happen in either of the following four events being true. Firstly, there is a failure to initialize input register. Secondly, there is a failure to initialize output register. Thirdly, there is a failure to initialize ISR and lastly there is a loss of output signal to drive circuit.

FTA helped identify the following hazardous events which can have catastrophic results:

1. Sensor getting damaged
2. Ball not raising from the bottom

Safety Measures

In case of the fan running too fast, there exists a danger that the ball will hit the sensor and the sensor getting damaged! A safety measure that can be taken in this scenario is to have an emergency shut down in the system. A point should be marked above the top most point that can be used as a set point, where if the ball reaches then the system should shut down automatically. After that a check should be carried out to figure out the reason for the output being so high to make the fun run very fast.

In case of the ball being at the bottom, a code review should be carried out to see if any variables have not been initialized. The variable/s should be identified, initialized accordingly and the system should be restarted. Though, it is highly recommended that during implementation stage itself care should be taken to ensure that no variables are left uninitialized as the saying goes “Precaution is better than Cure”.

In case the root cause of the failure is identified as a hardware fault, necessary steps should be taken to overcome the hardware fault.

4. Conclusion

This paper describes the importance of FTA by conducting an experiment on a micro-controller based control system having tiny or no hardware safeguard. The objective of this work is to regulate the flow of air into a plastic tube so as to keep a tiny less weight ball suspended at a predetermined height called the set-point. In FTA the faults in the system are regarded as an event and its corresponding causes are found. In this paper, FTA of Ball in the Tube system was carried out. All the faults were identified and their corresponding root causes were found. FTA not only helped identify the software faults, but also helped identify the hardware faults. We have also proposed the measures that should be taken in case of the faults. Incorporating the measures suggested will help the software developers to develop a system that does not fail due to software. Also in case of failure, it will help identify the root cause of the failure and rectify and start the system again in shorter possible time.

References

1. Knight, J. C. (2002). *Safety critical systems: Challenges and directions*. Proceedings of the 24th International Conference on Software Engineering (ICSE), Orlando, Florida.
2. Office of Science and Engineering Laboratories (OSEL) 2011 Annual Report.
3. https://www.hq.nasa.gov/office/codeq/risk/docs/ftacourse.pdf
4. Ziwei, O., Schnell, M., & Wei, K. (2007). *The experiment Ball- in-tube with Fuzzy-PID controller based on d-space*. IEEE International Conference on Systems, Man and Cybernetics, (pp: 877-881)
5. http://www2.warwick.ac.uk/fac/sci/wmg/ftmsc/modules/modulelist/peuss/ slides/section11b fta-lectureslidescompatibilitymode.pdf
6. http://www2.ece.ohio-state.edu/p̃assino/lab5 nonlinear ball tube ex.pdf
7. Pereira, J. S., & Bowles, J. B. (1996). *Comparing controllers with the ball in a tube experiment*. Proceedings of the 5th IEEE International Conference on Fuzzy Systems, (pp: 504-510).
8. Dunn, W. R. (2003). *Designing safety-critical computer systems*. IEEE Transactions on Computers, November, 36(11), 40-46.
9. Swarup, M. B., & Ramaiah, P. S. (2009). A software safety model for safety critical applications international. *Journal of Software Engineering and Its Applications*, October, 3(4), 21-32.
10. Douglass, B. P. (1999). *Safety critical systems design.* New York: Addison-Wesley
11. Konrad, S., & Cheng, B. (2002). *Requirements patterns for embedded systems*. In Proceedings of the IEEE Joint International Requirements Engineering Conference (RE02), (pp. 127-136).
12. Ostroff, J. S. (1992). Formal methods for the specification and design of real-time safety critical systems. *Journal of Systems and Software Archive,* April, 18(1), 33-60.
13. Bowen, J. (1993). Safety-critical systems, formal methods and standards. *Software Engineering Journal*, July, 8(4), 189-209.
14. Konrad, S., Cheng, B., & Campbell, L. (2004). *Object analysis patterns for embedded systems*. IEEE Transactions on Software Engineering, 30(12), 970-992.