IMPLEMENTATION AND VERIFICATION OF SAFETY CRITICAL SOFTWARE

Benedict Roberts
Supervisor: Dr Iestyn Pierce

School of Computer Science and Electronic Engineering

Introduction/Background

Programmable Logic Controllers have become an essential piece of equipment for automation in the manufacturing process. They have also been used at the heart of safety systems within all areas of industry due to their ability to work in environments where dirt, moisture, electromagnetism, and vibration are present, making them extremely reliable.

Safety is defined as being the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly or indirectly, because of damage to property or to the environment [1]. Functional safety is the part of safety that depends on equipment or a system operating correctly in response to its inputs [1]. It is the detection of a potentially dangerous condition resulting in the activation of a protective or corrective device or mechanism to prevent hazardous events occurring or providing mitigation to reduce the consequence of the hazardous event [1].

Aims and Objectives

1. Research into programmable logic controllers, the ideas around them, how they work and the programming method behind them.
   - 1.1. Gain a knowledge of the programming environment to allow myself to program the PLC to fit the purpose of my project.
   - 1.2. Gain hands on experience with the controllers in the department and to develop a personal understanding on programming the controllers.
   - 1.3. Learning the connections of the PLCs to gain the ability to create PLC circuits.
2. Understand the important ideas in functional safety verification to apply the ideas that I have learnt to the emulator I will hopefully create.
   - 2.1. Survey the literature of safety verification and then synthesise the information into a summary.
   - 2.2. Analyse the information I have gathered by looking for gaps in current knowledge and looking for areas of further research to deepen my knowledge.
   - 2.3. Present the information I have gathered in a professional and organised way.
3. Design and construct a PLC-based control system emulator by applying the knowledge I have gained in my previous aims.
4. Performing a functional safety verification of the PLC system.

Risk Graph

One of the techniques used to determine Safety Integrity Levels outlined by IEC 61508. It compares the severity of a potential incident with the probability. SILs are the relative levels for stating the safety integrity requirements provided by a safety system. Where 4 is the highest level and 1 is the lowest level of safety integrity.

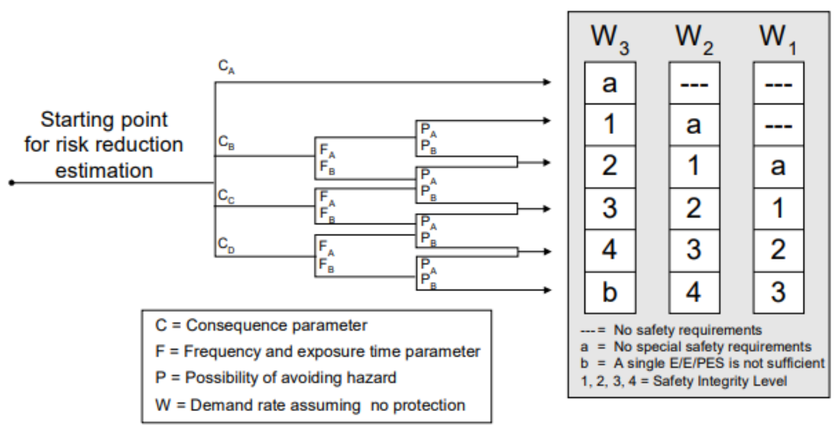

Figure 2: Risk Graph

Consequence	
C_A	Minor injury
C_B	0.01 to 0.1 probable fatalities per event
C_C	> 0.1 to 1.0 probable fatalities per event
C_D	> 1 probable fatalities per event
Exposure	
F_A	< 10% of time
F_B	≥ 10% of time
Avoidability / Unavoidability	
P_A	> 90% probability of avoiding hazard / < 10% probability hazard cannot be avoided
P_B	≤ 90% probability of avoiding hazard / ≥ 10% probability hazard cannot be avoided
Demand Rate	
W_1	< 1 in 30 years
W_2	1 in > 3 to 30 years
W_3	1 in > 0.3 to 3 years

Figure 3: Risk Graph Parameters

The required amount of risk reduction required is in correlation to the SIL target.

Future Work and Conclusions

Although no final results have been obtained I have the following:

- Now learnt a good depth of knowledge into how safety verification works.
- Can now look into performing Safety verification of a system and then assigning a safety system to it based on its Safety Integrity Level.
- This will look similar to the system seen in Figure 4 and will revolve around the use of a pressure sensor and thermal couple.

SIEMENS LOGO!

- The 'LOGO!' is a basic PLC that is perfect for learning the basics of Programmable Logic Controllers and can create basic control tasks using it. The PLC has 8 digital inputs and 4 relay outputs. It comes with an ethernet connection allowing monitoring and control via WLAN and the internet.

Figure 1: SIEMENS LOGO!

- Four digital inputs of the 'LOGO!' can be used as analogue inputs that can read a voltage range of 0-10v. These will then receive our input from the sensors and perform functions based around the readings received.

Pressure Sensors

- Pressure sensors work by converting the pressure received into an electrical signal which is then transmitted. They are also called pressure transmitters and they send one of two common signals. This is either a 4-20mA signal or 0-5v.
- The Pressure sensor being used gives a current loop output of 4-20mA and our 'LOGO!' PLC can only read an analogue input of 0-10v. To work around this problem, we can attach the sensor to the PLC input with a 500Ω load resistor in series to produce a voltage range of 2-10v (if zero, it would be an open sensor and would show integrity of system has been lost).

PLC Programming Using Ladder Logic

- In Figure 4, I have created a basic safety system that incorporates an analogue input received from a pressure sensor that will trigger a release valve when a certain voltage is read. The system has two valves, in the event one fails, and has a output relay symbolising an LED that will notify of a component failure.
- This demonstrates which types of safety requirements may be needed, because of the SIL level acquired.

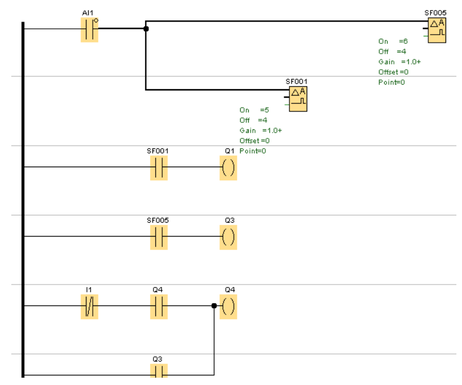

Figure 4: Ladder Logic Safety System

IEC Standards

- All work is done to fall inline with the rules and standards set out by the IEC
- This is due to the level at which the standards are used globally and so it only makes sense to ensure that this projects confines to those standards

[1] International Electrotechnical Commision. Functional Safety. IEC [Online]. Available : https://www.iec.ch/functionalsafety/explained/. [Accessed on: Nov. 19,20]

[Figure 1, Figure 2] wildeanalysis.co.uk/wp-content/uploads/2016/07/white_paper_methods_determining_safety_integrity_level_gulland_4sight_consulting.pdf

PRIFYSGOL BANGOR UNIVERSITY