IMPLEMENTATION OF INTEGRATED SYSTEMS MONITORING TOOL FOR POWER AND NETWORK APPLICATIONS IN EXPRESSWAYS

Nuwan Roshan Ediriweera Jayasuriya

(128810P)

Degree of Master of Science

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

May 2017

IMPLEMENTATION OF INTEGRATED SYSTEMS MONITORING TOOL FOR POWER AND NETWORK APPLICATIONS IN EXPRESSWAYS

Nuwan Roshan Ediriweera Jayasuriya

(128810P)

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

May 2017

DECLARATION

"I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

…… …….

N. R. E. Jayasuriya Date

The above candidate has carried out research for the Masters under my supervision.

…… …….

Dr. D. P. Chandima Date

…… ……..…

Prof. N.K. Wickramarachchi Date

Abstract

With the rapid development of information and communication technology, use of electronic and telecommunication based systems is increasing day by day. It improves the efficiency and quality of most of the traditional work flows by replacing fully manual operation procedures with fully automated systems or with partially automated operation with significantly reduced number of staff.

Operation of all modern expressways uses such systems significantly in large geographically distributed area for efficient management of expressways with enhanced comfortability and safety. But it will add an additional workload on expressway maintenance team to manage all critical systems in operation with minimum or theoretically with zero downtime.

System monitoring is one of the key elements in systems operation to verify its operation. It helps system maintenance team to identify the status of the system remotely. But, the system monitoring has not been implemented completely in Sri Lankan expressways, causing difficulties to identify system problems as soon as they occur. It was understood that this factor has major impact on the reliability of the systems operation by studying past operation experience. Hence the objective of the proposed design is to improve availability and reliability by minimizing down times of IT and Electronic systems in the expressway.

During the first part of this research, operation of two similar scenarios was selected for background study. Initial study is from Japan, which is a country of having more than fifty year experience of operation of the expressways and related facilities. Second study is performed with Sri Lanka Telecom, the pioneer of the telecommunication industry in Sri Lanka. Reliability analysis was performed based on the log book entries of the systems maintenance team of Southern and Outer Circular expressways in Sri Lanka, to identify possible scenarios of faults occurred and how its effect on the reliability of critical systems and challenges faced on identification of faults during corrective maintenance sessions.

As the second part, theoretical design of an integrated systems monitoring application was carried out by following modular design approach. The design is covered all critical functional blocks of a monitoring system with a model for performance analysis. The new design consists of several improvements over conventional monitoring systems to enhance the functionality. Several methods were designed for alarm optimization to reduce number of repetitive alarms. Heuristic knowledge base was linked with the designed monitoring system, hence it ensures that the maintenance personal is updated with its all history records before attending to the repair.

Implementation of designed monitoring system was completed in all critical components. It included both software and hardware module implementation and deployment. All modules use open protocols and open source software components.

Several functional tests were carried out with their performance values both in a test bed and in the production environment. The results indicated that the system is working as expected and help to improve the availability of systems through proposed methodologies.

Index Terms— Alarm optimization, Failure modes, Fault detection, Integrated monitoring, Protocols, Redundant systems, System reliability and availability

ACKNOWLEDGEMENT

This dissertation would not have been possible without the help from many individuals. I would like to acknowledge all of the people who in their own way helped and supported me with the present work.

I am truly indebted to my thesis supervisors, Prof. N. K. Wickramarachchi and Dr. D. P. Chandima, for the outstanding guidance, encouragement, and support, throughout the course of this work.

I would like to express my deep appreciation to my thesis committee members, Prof. Nalin Wicramarchchi, Prof. Sisil Kumarawadu, Dr. Jayathu Samarawickrama, for their insightful and constructive advice throughout my master degree.

I would like to thank my parents and my friend Thilini, for their continuing love and support during my master degree and also would like to thank for all the other gave me helping hand in many ways to complete my desertion.

I am much obliged to university of Moratuwa for offering timely valuable course modules under the Industrial Automation and it gave me helping hand in many ways to develop my theoretical and practical knowledge related to automation while it enhances my carrier as well.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation		Description
A/D	-	Analog to Digital
AC	-	Alternating Current
BPA	-	Batch Processing Application
CCA	-	Communication Control Application
CCTV	-	Closed Circuit Television
CUCCX	-	Cisco Unified Contact Center Express
CUCM	-	Cisco Unified Communications manager
DAQ	-	Data Acquisition
DC	-	Direct Current
EEPROM	-	Electrically Erasable Programmable Read Only Memory
ETC	-	Electronic Toll Collection
HMI	-	Human Machine Interface
HTML	-	Hypertext Markup Language
HTTP	-	Hyper Text Transfer Protocol
HTTPS	-	Hyper Text Transfer Protocol - Secure
IP	-	Internet Protocol
IT	-	Information Technology
ITS	-	Intelligent Transport System
LED	-	Light Emitting Diode
MCA	-	Monitoring Control Application
MCB	-	Miniature Circuit Breaker
MCU	-	Microcontroller
MDT	-	Mean Down Time
MTBF	-	Mean Time Between Failures
MTC	-	Manual Toll Collection
MUT	-	Mean Up Time
QM	-	Quality Management
RAM	-	Random Access Memory
RCD	-	Residual Current Circuit Breaker
REST	-	Representational State Transfer
RMS	-	Root Mean Squire
SFTP	-	Secure File Transfer Protocol
SMS	-	Short Message Service
SNMP	-	Simple Network management Protocol
TCP/IP	-	Transmission Control Protocol/Internet Protocol
UDP	-	User Datagram Protocol
UPS	-	Uninterruptable Power Supply
XML	-	Extensible Markup Language

LIST OF APPENDICES

Note: Appendix 02 to 06 is available on the provided compact disk (CD).

CHAPTER 1

INTRODUCTION

1.1 Use of Electronics and Information Technology Based Systems in Modern Expressway Operation

Modern road transportation methods use various electronic systems to increase the mobility, efficiency, safety and comfortability. To achieve this task, expressway operations use various electronic and IT based systems to gather information and process it to make control over the traffic flow, based on real time data. List of such IT infrastructure utilized internationally in operations are appended below.

(a) Toll collection system (manual and electronic methods)
(b) Closed Circuit Television (CCTV) system
(c) Voice communication including expressway emergency call center
(d) Intelligent transport systems
(e) Data communication network systems
(f) Management systems

Brief description of each systems listed above can be found from the section 1.1.1 onwards.

1.1.1 Toll collection systems

Toll collection in expressways are performed through either by manual toll collection (MTC) or electronic toll collection (ETC).

MTC is operated by a person at the toll gate (teller) by manual classification of the vehicle entered or through an automated vehicle classifier at the entrance. The teller issues an entry ticket or entry pass to the vehicle and the ticket must be returned at the exit point. Toll rate will be calculated by using distance travel data in an entrance ticket with pre-defined toll rate (fare) table. The exit location ticketing terminal issues an exit receipt after the teller collects the fare from the road user. The exit toll

barrier will be opened upon the successful transaction and will be closed by the teller on manual basis (or by the loop detector).

A semi-automated version of manual tolling named as “Touch and Go System (T&G)” is under operation in Japan and China. The process use fully automated vehicle type classifier (by detecting number of axels, height of the vehicle, length of the vehicle and weight) and issues entrance ticket automatically. The driver is required a stoppage at the entrance gate to collect the automatically printed entry ticket and at the exit, it has to be swiped or tapped at the exit point toll machine and the driver can be able to pay the toll by using his multimode travel pass.

ETC requires a special onboard unit, virtually linked with the vehicle number plate (including category of the vehicle), which is placed inside the vehicle’s front windshield. When such a vehicle passes through a toll gate, ETC antenna mounted on the gantry detects the ETC tag and transmits identification details to the associated server system. At the exit point, another ETC antenna will detect the onboard tag, will calculate the toll based on travel distance and debit the amount from the vehicle owner’s ETC account. Finally, the ETC barrier opens upon successful transaction or remains closed if otherwise. Figure 1.1 illustrates such ETC tag and ETC antenna used in Colombo-Katunayake expressway (E03).

Figure 1.1: ETC tag and ETC reader

Another improved version of ETC could be stated as “Multi-Lane Free Flow Tolling” (Figure 1.2) [1] which does not require a toll plaza, but needs a gantry mounted ETC readers and image recognition devices at a sufficient height over the road surface. Vehicles can travel at its full speed through the detection area and the

collection of toll will be performed with the combination of ETC technology and number plate recognition to improve detection accuracy.

Figure 1.2: Multi-lane free flow toll station

1.1.2 Closed Circuit Television(CCTV) system

Video surveillance system is one of the most effective tool to get real time view on the expressway operation through live video streams from the expressway control center. CCTV inspectors at the control center continuously monitor the events that take place along the expressway and utilize these information to make decisions. Much improved usage of CCTV, could be achieved by using advanced image processing techniques to detect abnormal incidents (shock wave propagations, traffic rule violations etc) through the CCTV videos automatically without human interaction and generate alarms to the control center staff.

1.1.3 Voice communication system

Voice communication and information passing is one of the most critical events in the expressway operation. It includes 24 x 7 expressway emergency call center and customer support center to connect with expressway users through the mobile phones or through public telephones by dialing a short code number to inform incidents and to obtain roadside assistance. It also facilitates a provision for providing real time traffic information to the public and media. The system operates with several wired or wireless internal communication channels for information passing with internal operational crews.

1.1.4 Intelligent transport systems (ITS)

This is one of the new concept utilized in modern expressway operations. It can be considered as a combination of services and methods for data collection, information processing and provision through the collection of several information from different platforms (CCTV cameras, vehicle detectors, vehicle counters, image analysis, manual data collection) and process them in a way to maximize the efficiency of the road network and dissemination to the public through various methods. (Variable message sign boards, Traffic information boards, ITS spots, online and broadcasted traffic data and traffic signal control) The road user can make use of this information to plan their travel routes in advance to their journey. Figure 1.3 illustrates such ITS structure using in Japanese expressway network. [2]

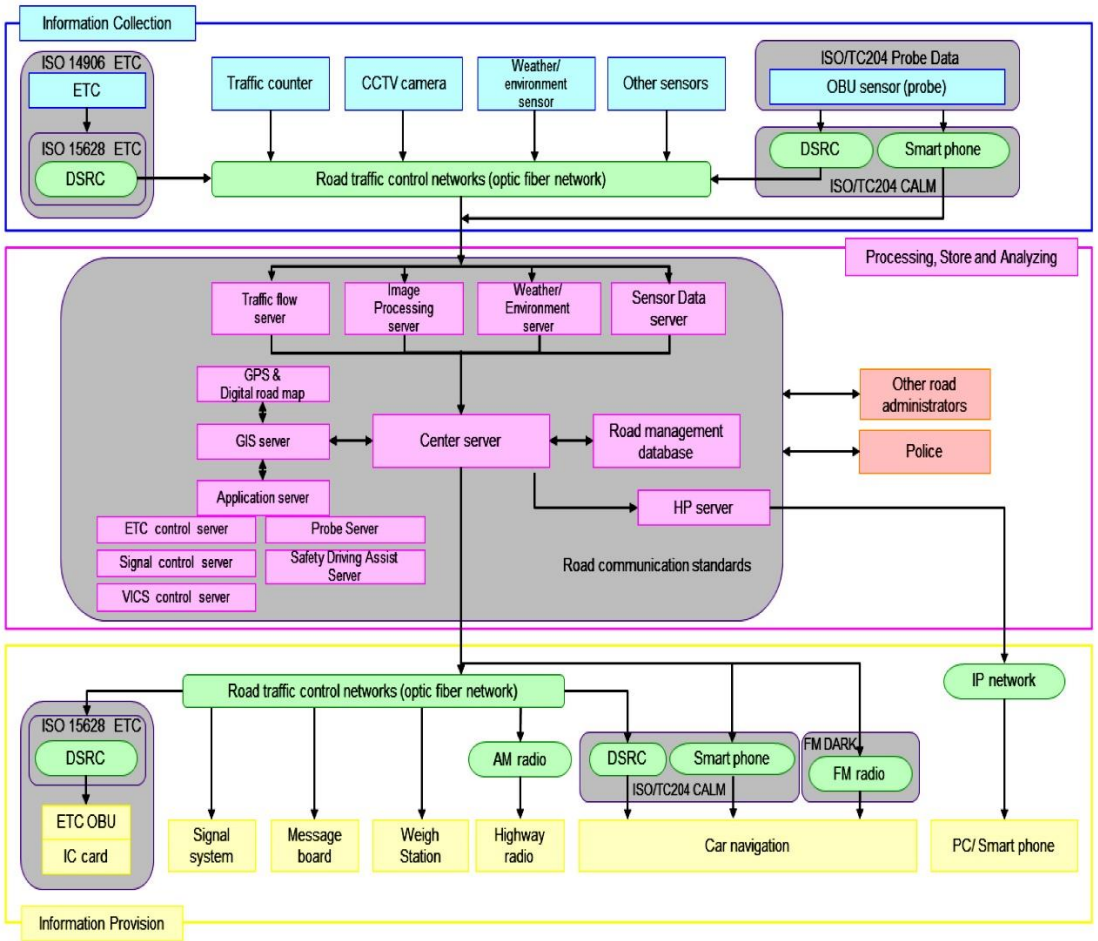


Figure 1.3: ITS structure in Japan

1.1.5 Data communication systems

All above systems require interconnections, via data communication system, distributed over the entire expressway network. The communication backbone between main centers is usually consists of the optical fibers and routers in fully redundant fashion, running routing algorithms for the interconnection. End nodes are usually consists with or without redundancy, through Transmission Control Protocol/Internet Protocol (TCP/IP) links.

1.1.6 Management systems

Several software based management systems have been utilized in the operations for proper management, work trace recording and work flow monitoring purposes. (Financial management system, Job record management, Asset management system etc.) Usually those IT systems function on internal server system based on the local data network.

1.2 Operation, Maintenance and Management Requirements of Systems in Expressways

Due to the nature of the expressway operation, a regular and uninterrupted operation is expected from all IT systems to impose highest comfortability and safety of the road user. It requires careful design and installation of systems, preventive and corrective maintenance sessions and system monitoring activities to be carried out during its life cycle.

1.2.1 Corrective and preventive systems maintenance

Corrective maintenance is required whenever the system stops its normal functionality. The interruption may take place either due to faulty in device or in component, causing complete loss of system functionality. The urgency of corrective maintenance is high. It has no pre-warnings before it occur. The incident recovery process might need expensive replacement of items, professional service support

from the vendors and can invoke extensive periods of down time, causing critical effects on the performance of the system and the operation.

Preventive maintenance sessions help to mitigate the occurrence of systems failures by detecting clauses for systems malfunction, before it grows to a catastrophic scale. Such preventive maintenance sessions will be carryout either periodic manner, decided by the systems vendor or by previous experience depending on the incidents. Table 1.1 illustrates the example of such preventive maintenance schedule applied in southern expressway in Sri Lanka.

Table 1.1: Preventive maintenance schedule on southern expressway CCTV system

No	Item	Schedule
1	Visual inspection on all equipment	Once per every 2 months
2	Cleaning of indoor cameras	Once per every 3 months
3	Cleaning of outdoor cameras	Once per every 2 months
4	Camera re-focusing	Once per every 6 months
5	Network recorder device service	Once per every 6 months
6	Check power and network surge arresters	Once per every month or after heavy lightening

1.2.2 Requirement of systems monitoring and management

Monitoring refers to the method of collection of regular data from the IT system infrastructure, to provide alerts on both of unplanned downtimes and resource saturation. It also makes operational practices auditable, which is useful in forensic investigations and for determining the root cause of errors. Monitoring provides the basis for the objective analysis of systems administration practices and IT systems.

Collection of data creates its own form of technical problems, and general purpose monitoring tools requires heavy customization and configuration for most cases. At the same time, most specialized monitoring tools are only collecting certain types of data from specified system and require system integration of different sub systems into general purpose monitoring platform. However, easy answers could not be found to these issues.

System faults detection is of great importance to ensure minimum downtime of systems by rectifying them at the outset. In some cases with redundant systems, identification of single device or single service failure is extremely difficult by observing service levels of the particular system, since the system has been designed to fail over to the high available node without service interruption. The failover happens silently and may continue to function until the point of second failure occurs at the high available node, causing complete service failure. Pre detection of problems may prevent secondary failures (problems created by the primary cause, with time) too.

The process of alarm optimisation of the monitoring system is required to avoid confusion of the technical crew by suppressing any repetitive alarms, which might be arising due to a single root failure. It requires an extensive analysis of collected data and unique methods of optimisation have to be adopted, more specific to the space to be monitored.

Faults or problem detection is insufficient itself and shall have a proper problem escalation path and methods, over the technical crew engaged in operations to attend and rectify the issues. Each fault rectification work requires different experts among the technical crew, hence the automated system shall be capable of escalating the problem information depending on the skill levels of each member. This will reduce the attention time and will guarantees the quality of the rectification work while managing human resources automatically by the system.

A good system administration refers not only the immediate faults rectification, but identification of present bottlenecks too, which will reduce the throughput of the system over the past period of time. A featured monitoring system needs record of past history data to generate system performance analysis reports by using the system operational history and which could be used to identify the bottlenecks to be improved.

Sometimes, system operation and maintenance records might require for forensic investigation purposes, in case of suspicious activities performed by humans

(intruders). As an example, one of the closed circuit television monitoring node can be interrupted by an intruder by manually turned off its power supply purposely to practice illegal operation during the down time. The automated systems monitoring could be used in such cases to retrieve date and time of the incident. Also the monitoring system database contents automatically generated status records of the sequence of event happened and related parameters, which could be trusted more accurate than the evidence presented by a human during the investigation.

Tack management and inventory management can be considered as extended application areas of integrated monitoring systems. For an example, the monitoring system could be programmed to allocate a job through the job record management system to rectify self-identified problems to a certain technician automatically, by considering his pre entered skills. Also it can keep track of utilisation of spare parts used for a certain rectification task, which is required to be claimed as inventory items through the main spare stores.

Finally, the monitoring system must be highly scalable. It shall be easily expandable over widely distributed systems and any later addition or change in the monitored space shall be incorporated without problems.

1.3 Problem Statement

Currently, the methods illustrated in table 1.2 used for the systems fault detection at all functional expressways (E01, E02 and E03) in Sri Lanka. All of them require individual attention to detect and isolate the fault and no provision provided for centralized or integrated faults monitoring facility.

Table 1.2: Present methods used for system fault detection

No	System	Sub System	Present Fault Detection Method
1	Electrical Systems	Power Generator	Visual Inspection or after service failure
		Automatic Transfer and main power distribution panel	Visual Inspection or after service failure
		Sub power distribution panels	Visual Inspection or after service failure
		Uninterruptable Power Supply units	Visual Inspection and Audible Alarms or after service failure
2	Communication System	Main Communication backbone and Related Sub Links	Network Management Application (Figure 1.4) through the monitor screen
		Extended Links through service provider's networks	After failure of the service, after reporting service failures by systems operators
3	Toll Collection System	Manual Toll System – server functions	After failure of both active and redundant systems, after reporting service failures by systems operators
		Manual Toll System – user interface	On failure of the service, after reporting service failures by systems operators
		Electronic Toll Collection System – Main functions	After failure of both active and redundant systems, after reporting service failures by systems operators
		Electronic Toll Collection System – Interface with bank	After failure of the service, after reporting service failures by systems operators

(Continued)

Table 1.2: Present methods used for system fault detection

No	System	Sub System	Present Fault Detection Method
4	CCTV System	Cameras	1. After failure of the service, after reporting service failures by systems operators 2. Alerts generated from video management software in the video client monitor
		Video Recording Server System	After failure of the service, after reporting service failures by systems operators
		Recording Management Server	After failure of the service, after reporting service failures by systems operators
5	IP based Communication System including Emergency Call Center	Communications Manager	On failure of both active and high available servers
		Contact Center	On failure of both active and high available servers
		Quality Management System including call recordings	After failure of the service, after reporting service failures by systems operators
		External links with call service providers	1. After failure of both active and high available links 2. After reporting service failure by systems operators

(Continued)

Table 1.2: Present methods used for system fault detection

No	System	Sub System	Present Fault Detection Method
6	Traffic counting, Variable Message Sign board system	Traffic Counting system	Through system generated error messages on operator screen (Figure1.5)
		Variable Message Signs	Through system generated error messages on operator screen
		Traffic Information Board	After reporting service failure by systems operators
		Rain Measurement System	Through system generated error messages on operator screen
7	Management Systems	Online Management System	After reporting service failure by systems operators

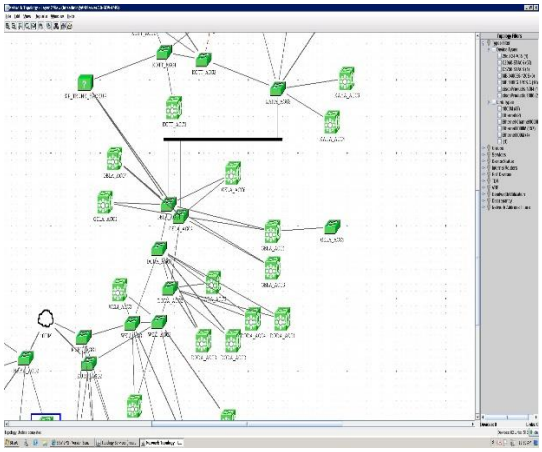


Figure 1.4 : Network management application

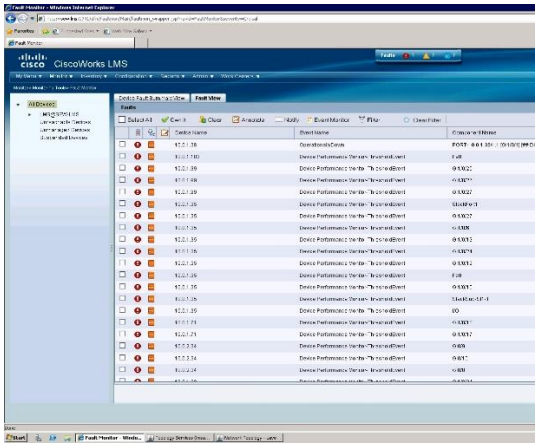


Figure 1.5: System generated error notifications

As mentioned in the table 1.2, status monitoring requires extensive attention on each and every individual system round the clock. Past operation experience indicates that the method of manual monitoring observed to be difficult due to its primitive nature and insufficiency and therefore to achieve required performance.

During the background study (Chapter 02), several gaps in the available monitoring methodologies were identified as listed below.

(a) Present methods are in lack of an integrated monitoring space, which is capable of monitoring various platforms through a single interface (system).
(b) Alarm optimisation of individual systems is not much successful due to lack of inter-system monitoring platform. It requires a human attention to filter out alarms which were generated due to secondary failures.
(c) Use of prop priory protocols limit the capability of continuous development of the monitoring system locally and causing increment of the maintenance budget for maintaining it. Also, it limits the possibility of integrating with different localized management system components for job management, skill based automated task allocation and system inventory management etc with the main monitoring system.

More information about the measures of systems reliability with present monitoring methodologies could be found in Chapter 03 of this document.

1.4 Aim And Objectives

The primary goal of this research is to develop an integrated, localized system, which includes monitoring, automated testing platforms and notification features to improve reliability of the operation of IT, communication, power and electronic systems use in functional expressways. The outcome must be complying with the following statements listed below for long term sustainability.

(a) **Expandability** – Since some of new expressways are under construction, the design shall be highly scalable.
(b) **Use of Open Protocols** – The development requires more general interfaces for long term sustainability.
(c) **Reliability** – The system require reliable platform for both software and hardware modules.
(d) **Simplicity** – Since the system will be design and developed by in house teams, further development and installation needed to be simple as much as

possible. Also system manageability should be simple as much as possible to the systems administrator.

(e) **Overhead** – The designed system should incur low overheads on system resources such as TCP/IP network.

By the end of project, it is expected to achieve following outcomes.

(a) Design verification and analysis framework to ensure how systems monitoring helps to improve system reliability.
(b) Localized design and hardware implementation to collect present status of the systems and equipment available in the expressway operation. (Data acquisition system architecture)
(c) Design and implementation of an integrated monitoring software system, which supports in monitoring, testing and notification features with improved detection methodologies.

1.5 Approach And Methodology In Brief

The approach and methodology mentioned below has been carried out to achieve the goals of this research.

(a) Performed background study to collect information about the available systems monitoring methodologies and their limitations from similar industries and commercially available systems.
(b) Identify the theoretical measures and parameters of the concept of systems reliability. Then it was performed reliability analysis on available systems by considering the past history without systems monitoring systems. Subsequently, the same could be repeated by assuming the availability of systems monitoring and recalculate empirical performance values to study how the systems monitoring affects on the system reliability.
(c) Study on available systems in expressway operation and their interface possibilities to use for monitoring. This study is much useful for the design of data acquisition platform in the monitoring system.

(d) Conceptual design of a monitoring system in block view with identification of all related functional blocks with its operation. This includes an analysis of empirical performance of the designed system.
(e) Data acquisition hardware and monitoring system software implementation in accordance with the designed system.
(f) Function tests with the implemented system in a test platform inclusive of real production environment.
(g) Results and discussion.

1.6 Structure Of This Document

Followings are the brief outline of the preceding chapters.

(a) **Chapter 02** – Background study had been performed in similar industries to identify available technologies used for systems monitoring methodologies. One study was performed with related to the Japanese expressways during the visit to the Japan. Second study was performed at Sri Lanka Telecom (a pioneer telecommunications service provider in Sri Lanka). Finally, it was reviewed other commercially available solutions for systems monitoring. These studies facilitated to identify the gaps to be filled during this study.
(b) **Chapter 03** – Study on theoretical background about systems reliability has been included in this chapter and contains a sample data analysis of the selected sample production system reliability and how it can be improved by introducing monitoring methodologies.
(c) **Chapter 04** – Information about theoretical design of the monitoring system contains in this chapter. The design phase includes modular approach to design each and every blocks of the main system. Later part of this chapter includes an empirical model for performance analysis of the designed system.
(d) **Chapter 05** – This chapter contains all necessary information about the implementation process of the monitoring system components designed under Chapter 04.

(e) **Chapter 06** – The implementation was tested in both test bed and with the production environment. All testing methodologies followed and their results with the acceptance were included.

(f) **Chapter 07** – General discussion about the outcomes from this research been included.

CHAPTER 2

LITERATURE REVIEW

A background study was carried out for two similar industries to identify key factors followed by them to manage operation maintenance of IT based systems including faults detection and systems monitoring techniques. One is the Sri Lanka Telecom, the pioneer telecommunication service provider in Sri Lanka and the other is Nippon Expressway Company Limited (NEXCO) from Japan, the pioneer for expressway systems operation in Japan. Information gathered from those studies is presented in the next section.

2.1 Systems Operation Study on Japanese Roads & Expressways

Japan International Cooperation Agency (JICA) offered a training session to visit Japan to study the operation of Japanese expressways and Intelligent Transport Systems with the collaboration of University of Tokyo, ITS Japan and Nippon Expressway Company Limited (NEXCO) in year 2015 July.

Japan has more than 50 years of experience with operation and maintenance of expressways, performed by the Japan Nippon Expressway Company Limited (NEXCO) with the collaboration of other several sub companies. A study was carried out in July 2015 with NEXCO for a period of one month regarding the operation and maintenance of expressways including systems maintenance with the aid of Japan International Corporation Agency (JICA) and University of Tokyo.

2.1.1 Systems operation and maintenance

Japanese expressway operation has included tolling, monitoring, communication and traffic control systems. Japan uses a system developed in-house with all necessary hardware and software. Some of their old equipment are still functioning normally for more than 30 years.

Separate centers called “Facility Control Center” are located in adjacent to the main control centers. Their main function is to monitor the systems used in expressway operation round a clock. Electrical systems, communication links, exhaust fans and other facilities at expressway tunnels, disaster recover options and data management of all maintenance activities are controlled and monitored by those centers. It uses different monitoring systems with different methodologies, either developed by the vendor or self-developed. Few of them are listed in table 2.1. An example procedure used for fault detection in Japanese traffic light system is presented in the Section 2.1.2.

Table 2.1: Systems monitoring methodologies in Japan

No	System	Method/Alarm types
1	Toll Collection systems	Through vendor-specific monitoring tools and log analysis
2	Monitoring Systems	Through vendor-specific monitoring tools
3	Communications Systems	Through vendor-specific monitoring tools and self-developed monitoring interfaces
4	Electrical and Power	System developed by OMRON® and self-developed hardware and software
5	ITS related facilities	Through vendor-specific monitoring tools and self-developed hardware and software
6	Traffic Light control	Through vendor-specific monitoring tools

2.1.2 Fault detection mechanism example in Japanese systems – Traffic light system

This system study was carried out at Nippon Signal Company Limited, Saitama-ken, Kuki-shi, Ezura in Tokyo prefecture Japan. The company is a major supplier of traffic management systems for local and global road networks from the year 1928.

Figure 2.1 illustrates simplified centralized traffic light control system [3] developed by Nippon Signal Company, used in Japan road network with all necessary fault detection methodologies. It is operated by using TCP/IP based platform, which connects the control center with all regional nodes. Traffic light control timing and

sequence is transferred by using the method called “next cycle table transmission” through their virtual private networks (VPN).

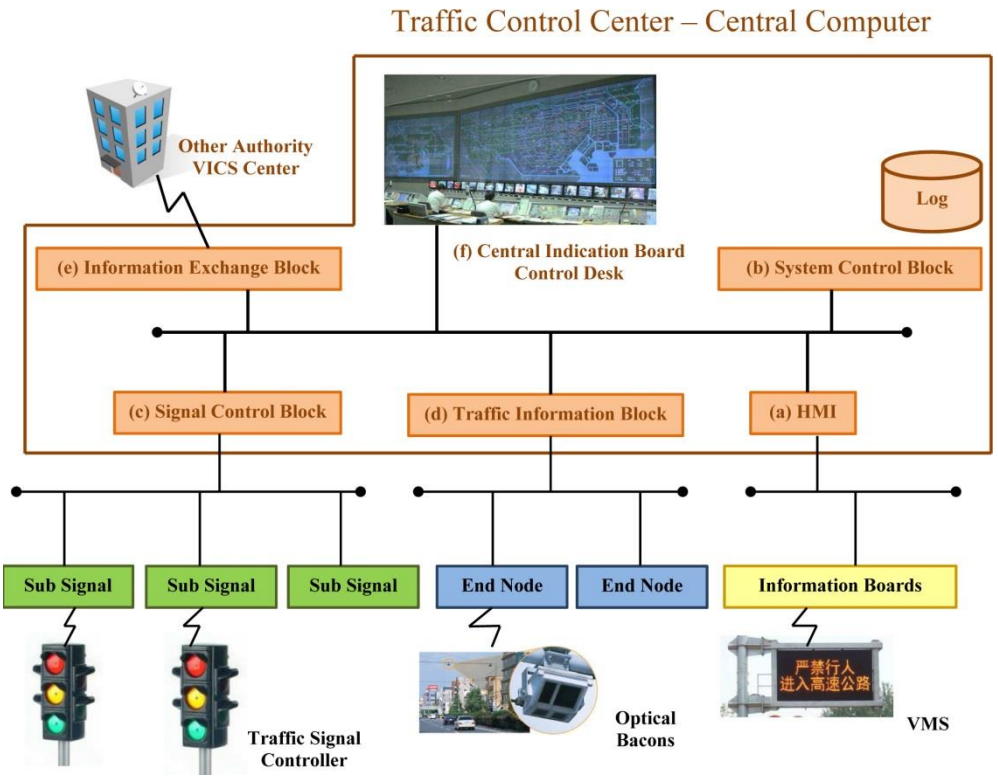


Figure 2.1: Simplified traffic light control system in Japan

Early faults detection is essential for such a system due to criticality of traffic control and signaling. Following is the description of the methodology used for faults detection and alarming in the system shown in figure 2.1.

(a) **Human Machine Interface (HMI)** – The HMI is a graphical user interface (GUI), which runs multiple instances on several computers in the traffic control center. This is based on the hypertext transfer protocol (HTTP) over TCP/IP through the local area network. The GUI consists of several sub modules such as for signal controlling, signal status monitoring, traffic status monitoring and system fault status alerts to perform the operation. The HTTP page is generated by the central system server through a web server. Detection of HMI based faults is much simple. The possible events are either inaccessibility to HMI from control center computers or not response of the

HMI instance. Both faults are directly visible to the HMI operators and special detection method is not required.

(b) **System Control Block** – This module handles all sub sections of the total traffic control system. The system control module is running on two high performance servers, in active-active configuration for maximum performance and redundancy. System control block is responsible for data collection, data processing, data storing, and information provision actions. The operation of the control block can be divided to several sub modules. Each sub module has different functions for processing, communication and error detection and generating alarms. As an example, data storage is performed by using Oracle data store, running on a cluster configuration and the system control data base driver collects all error information generated by the Oracle database engine and generates status report on HMI system alarm, as a database or cluster exception.

(c) **Signal Control Block** – This module handles communication with all sub signalling control blocks via extensible mark-up language (XML) through the TCP/IP network based on synchronous or asynchronous communication request passes. Signal control commands are generated as a XML data file to activate relevant signal control phases with different timings for each sub controlling stations. The transmitted control cycle will be activated from the next immediate cycle by the substation and will be kept in action until the next control command is received. The substation transmits an acknowledge message back to the signal control block upon successful execution or returns appropriate error code. Transmitted error codes have unique identification for wide variety of faults in the substation and it is sending to the system control block to activate fail over sequence and to the HMI as a system alarm. Signal control module is usually used polling to collect status data from substations. Substation identification is based on unique IP address and a block number. All communicated XML files are validated by using those parameters in both ends. Each individual substation can be accessed over secure shell (SSH)

protocol through the TCP/IP network and more detailed log files can be accessed to maintenance staff manually to identify faults.

(d) **Traffic Information Block** – This module has main two functions as to communicate with road side traffic data collection equipment and the communication with the system control module by using raw and human readable data formats. Data collection has been performed from the road side equipment by the system control module in every two minutes intervals. Data is encoded to vendor specific protocol format which can be decoded at the communication control module on the central server system. Same methodology has been used to communicate with the traffic information boards too. Unique sequence number has been used with each device for identification. Error codes transmission and alarm generation has been performed in a similar manner as the signal control block.

(e) **Information Exchange Block** – The traffic control system is connected with many other organisations to collect information. Vehicle information and communication system (VICS) center, Japan police information center, expressway control center are some of them. All collected data has been sent to the signal control block for automated sequence generation and to the traffic control center display wall for controller's manual signal control. Communication is based on the method of representational state transfer (REST) web service with web service security enables, over TCP/IP. Error detection mechanism is based on HTTP/1.1 (RFC 7231) [4] error codes.

(f) **Central Indication Board** – This is located at the traffic control center which displays all status in several screens in graphical form. It includes real time traffic information data, incident data, traffic control sequence information and system status indicators. If a fault has been detected from one sub component. It will pop up an alert on the defective component in the display and indicate the identified problem to the control room staff. Then this information will be passed to the maintenance team immediately to fix it.

In the study, it had been understood that the system is only providing scatted information on error detection. The detection mechanism is different from module to module. However, all modules are capable of generating log files for detailed description about errors and faults which are to be collected and analyzed manually by the systems maintenance staff. Power and electrical system alarms are collected either by equipment self-error detection methods (by uninterruptable power supply, equipment self-tests etc) or through the networked power status monitor equipment.

Another study had been performed with Sri Lanka Telecom, who is the pioneer telecommunication operator in Sri Lanka. More information is presented in the next section.

2.2 Systems Operation Study at Sri Lanka Telecom

Sri Lanka Telecom is one of the largest telecommunication service provider in Sri Lanka, who is managing fairly complex and large communication system environment. A study was had been carried out to collect information about their systems fault monitoring and work environment via the electronic mail (Appendix 01) from its one of the operational engineer. Important points gathered through this study are presented in Table 2.2.

Table 2.2: Systems monitoring methodologies practicing in Sri Lanka Telecom

No	Area	Details
1	Systems in use	Cisco, Huawei, Alcatel, Juniper routers Rectifiers, UPS systems
2	Monitoring Systems	Open NMS, Cacti Tool, Huawei U2000, Alcatel SAM 5620
3	Alarm Types	Critical – Node outage, Traffic congestions Major – issues in customer end equipment Minor – Temperature alarms, card CPU alarms

It had been observed that the Sri Lanka Telecom is using multiple primitive systems for systems monitoring, supplied either by the equipment vendor or by using an open

source software. All monitoring systems are functioned as scatted systems, resulting to assign dedicated monitoring teams for each and every system. None of them have facility for intelligent fault analysis.

2.3 Commercially Available Systems for Fault Detection & Monitoring

A brief review was carried out with selected commercially available systems monitoring tools and applications to identify their features. Table 2.3 illustrates such systems with their features, based on the information provided in their product manuals.

Table 2.3: Commercially available systems for systems monitoring

No	System and Manufacturer	Features	Observed Limitations
1	QRadar Platform [5] IBM Systems	• Integrating security information and event management (SIEM) • Systems log management • Anomaly detection • Incident forensics • Incident response • Configuration and vulnerability management	No provision for server power systems monitoring and integrated results analysis
2	Network Monitoring System [6] CISCO systems	• **Fault Management**—Detect, isolate, notify, and correct faults encountered in the network. • **Configuration Management**—Configuration aspects of network devices such as configuration file management, inventory management, and software management.	• No provision for server and network power systems monitoring and integrated results analysis • Use of Proprietary protocols

Continued

Table 2.3: Commercially available systems for systems monitoring

No	System and Manufacturer	Features	Observed Limitations
2	Network Monitoring System [6] CISCO systems	• **Performance Management**—Monitor and measure various aspects of performance so that overall performance can be maintained at an acceptable level. • **Security Management**—Provide access to network devices and corporate resources to authorized individuals. • **Accounting Management**—Usage information of network resources.	
3	Power System Monitoring and simulation software [7] Etap systems	• Multi-console with multi-screen monitoring • Graphical monitoring via ETAP one-line diagram • Visual monitoring via Man-Machine Interface (MMI) • Alarm warnings with graphical interface • Alert of equipment out-of-range violations • Monitoring of electrical & non-electrical parameters • Pseudo measurements (override measured data) • OPC interface layer • User-access levels	• No provision for server and network systems monitoring and integrated results analysis • Use of Proprietary protocols

Continued

Table 2.3: Commercially available systems for systems monitoring

No	System and Manufacturer	Features	Observed Limitations
3	Power System Monitoring and simulation software [7] Etap systems	• Continuous real-time monitoring • On-demand data retrieval • Data reconciliation & consistency check • Bad data detection & correction • Alarm management & processing • Energy cost monitoring & accounting • Real-time load forecasting & trending	

2.4 Observations on Systems Operation and Management

By considering the above studies, following observations can be made about the systems faults monitoring on commercial systems.

(a) Japan uses more advanced systems and self-developed fault monitoring procedures. However, scatted system monitoring tools for different platforms makes it difficult to manage work flows and repair records. Now they are starting to develop a system to centralize all.

(b) The systems used for expressways are not updated frequently unless they need new requirements, an important factor captured from Japan's operation. Hence, development of more specific monitoring system is acceptable to reduce design complexity than the method of designing them as more general interfaces to connect with different platforms.

(c) Sri Lanka Telecom uses vendor specific or open source scatted software tools for systems monitoring, requiring more human power for fault monitoring.

(d) None of the organisations use intelligent systems approaches to monitor faults, managing records or job records through a single system.

Hence the development of an integrated, customized fault monitoring and automated testing system for operational systems would be the best approach for Sri Lankan expressway network for easy monitoring and efficient management of corrective and preventive maintenance activities.

The basis about reliability of the systems in general will be discussed in Chapter 03.

CHAPTER 3

SYSTEM RELIABILITY ANALYSIS

3.1 Measures of Systems Reliability

With the demand of information and communication technology based services in expressway operation, maintaining systems availability and reliability is becoming a critical and hard work for systems maintenance teams. The service levels required to be satisfied by systems maintenance teams are defined and fixed in the expressway operation and maintenance guidelines manual [8] in year 2011 as mentioned in the table 3.1.

Table 3.1: Key performance levels [8]

No	Category	Details	Attention Time
1	P1	Major system failure, loss of more than 70% of the operation	within 15 minutes
2	P2	Major system failure, loss of more than 40% to 70% of the operation	within 2 hours
3	P3	Minor system failure, loss of more than 20% to 40% of the operation or total failure in high available node	within 6 hours
4	P4	Minor failures in end nodes or change/update requests	within 2 days or in next working day

3.1.1 Systems reliability parameters

Reliability, Availability and Serviceability (RAS) [9] can be considered as key indicators about operation of any kind of system. Reliability can be defined as the probability that the system produces correct output up to some given time ***t***. The corresponding system's reliability can be enhanced by adding features that help to avoid, detect and repair faults itself. Reliable systems will take an action promptly, if it detects an issue. It either reports the fault as error message, continue its functions with its redundant systems or simply halt the execution. Reliability can be

categorized in terms of mean time between failures (MTBF) [10] as shown in below.

$$\text{reliability} = e^{-t/MTBF} \qquad (1)$$

Availability refers to the probability of system running at the given time. It can be measured in terms of the ratio of actual operating time to the total time [9]. Availability features allow the system to operate continuously even when faults do occur. Deployment of a high available (HA) system will increase the performance and stability of the systems operation.

Serviceability or maintainability is the simplicity and speed with which a system can be repaired or maintained; if the time to repair a failed system increases, then availability will decrease [9]. Serviceability includes various methods of easily diagnosing the system when problems arise. Early detection of faults can decrease or avoid system downtime.

In practice, both terms; reliability and availability are much important. Loss of reliability may cause erroneous outputs from the system or loss of availability may cause access denied to the system operators.

System failures occur when a system is unable to provide its required functions [11]. It is governed by one or more individual errors, causing the system preventing work as expected. A good monitoring system will focus to detect individual or multiple errors, may cause system to functioning normally. System errors might be masked by the architecture (system design). High available arrangements may hide errors unless carefully monitored, causing loss of long term reliability in the system. Figure 3.1 illustrates such representation of failures, faults and errors [12].

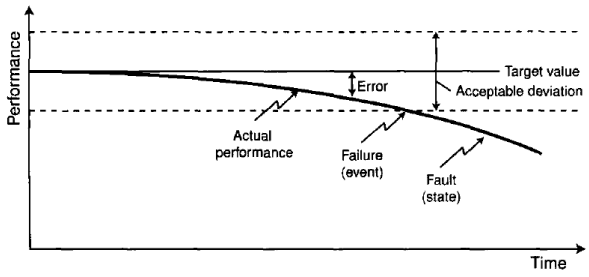


Figure 3.1: Failures, faults and errors [12]

In general, system downtimes may occur due to several reasons.

(a) A failure in the electrical system resulting power supply is not available to the system. This will reduce system availability.
(b) Failures in communication channels will make the functional block(s) isolating from the cluster system. This may cause either system availability or systems reliability due to lack of inputs.
(c) Hardware or software failures in individual or multiple functional blocks usually cause loss of system availability or reliability.
(d) Unauthorized third party involvements on systems, sometimes may define as viruses, cyber-attacks may resulting loss of all parameters, reliability, availability and serviceability.

3.1.2 Failure models and responses

Mainly, there are two types of faults can be occur in the system.

(a) **Permanent faults** – Due to permanent error in the system. Easy to detect even by following manual fault locating procedures.
(b) **Temporary faults** – Due to transient or intermittent errors. Most of the time they are difficult to detect without careful analysis about systems behaviour. Recovering from such faults might take time resulting reduction of serviceability.

The need of a monitoring system is critical for early detection of both types of failures. As an example, temporary faults happened due to transient conditions in the power supply might result in random activation of particular circuit breaker, result in loss of power to the system components which can be easily detected by using real time power system monitoring.

To prove the above concept, a study had been done with various theoretical aspects of systems reliability.

The state of equipment at time t may be described by $x(t)$;

$$x(t) = \begin{cases} 1 \text{ if the equipment is functioning at time } t \\ 0 \text{ if the equipment is not functioning at time } t \end{cases}$$

Time to failure, *T* [12] measures the time period to first failure from the beginning of the operation. ($t = 0$) Figure 3.2 illustrates such relationship between *T* and $x(t)$. The measures of time t always not in calendar dates, but may be depend on several indirect measures as system load, number of cycles the equipment can perform, number of repetitive surge currents which can be held on device etc. So the definition of mean time between failures (MTBF) defined by the systems manufacturer may differ than the actual scenario in the production environment.

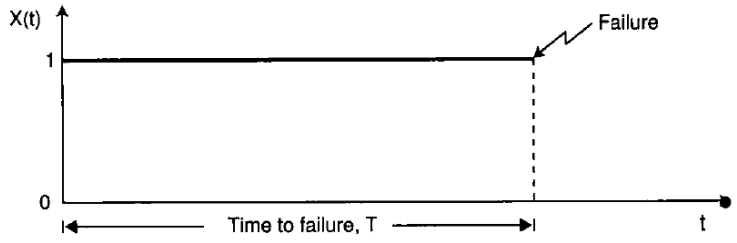


Figure 3.2: State of the equipment at time *t* vs. time to failure [12]

Time to failure, *T* is a discrete variable, however can be approximated to a continuous variable. It is assumed that the *T* is continuously distributed on probability density function *f(t)* and then the distribution function *F(T)* can be written as; [12]

$$F(t) = \text{Probability}(T \leq t) = \int_0^t f(u)du \text{ for } t > 0 \qquad (2)$$

Hence, with the time *t* passes, the chance of getting failure will be higher and higher. System reliability *R(t)* can be defined as; [12]

$$R(t) = 1 - F(t) = \Pr(T > t) \text{ where } t > 0 \qquad (3)$$

Since the reliability of the equipment *R(t)* decreases to zero after some operational

period expires. Figure 3.3 and Figure 3.4 illustrate the behavior of *f(t)*, *F(t)* and *R(t)* with time *t*. [12]

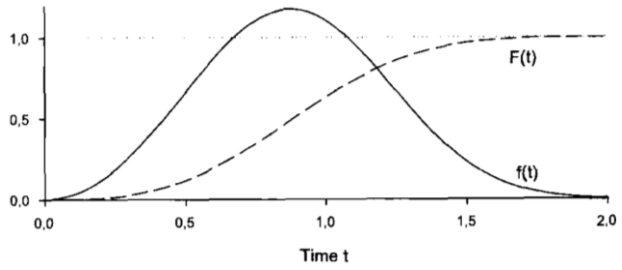


Figure 3.3: Distribution function *F(t)* and probability density function *f(t)* [12]

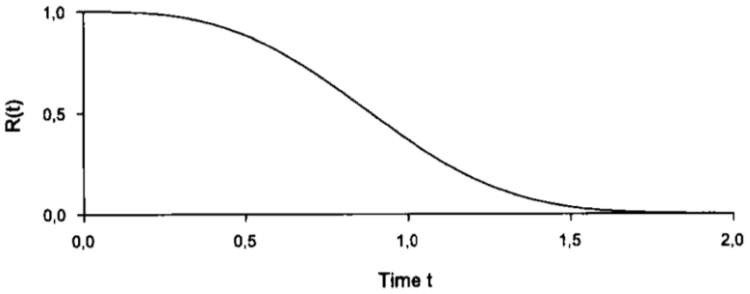


Figure 3.4: Reliability function *R(t)* [12]

When a system in operation, it might fail unexpectedly or might have planned outage for maintenance, causing loss of availability (*Av)* at the given time *t*. Mean down time *MDT* can be defined as the sum of mean time for problem detection *MTD*, mean time to repair *MTTR* and mean post recovery time *MPRT* including time to reboot the system [13].

$$MDT = MTD + MTTR + MPRT \qquad (4)$$

Availability of the system can be defined as a ratio of mean up time *MUT* to mean down time *MDT*, shown in equation (5) [12].

$$Av = \frac{\mathrm{MUT}}{MUT+MDT} \qquad (5)$$

Properly designed monitoring system can reduce *MDT*, hence it can help to reduce *MDT* and increases availability. However if the monitoring system can detect and make early warnings before the equipment fails (detect and warn stresses beyond limits, early power fail warnings etc) will reduce mean time between failures (MTBF) if the maintenance crew attend on time before it escalates in to a fault condition. These approaches will increase system availability, hence overall reliability.

In production environment, systems can be connected in parallel to gain high availability [14]. Industrial installations are consists with combined systems having both parallel nodes on weakest blocks and series nodes on comparatively reliable components. Figure 3.5 illustrates such simple combined system with two redundant inputs.

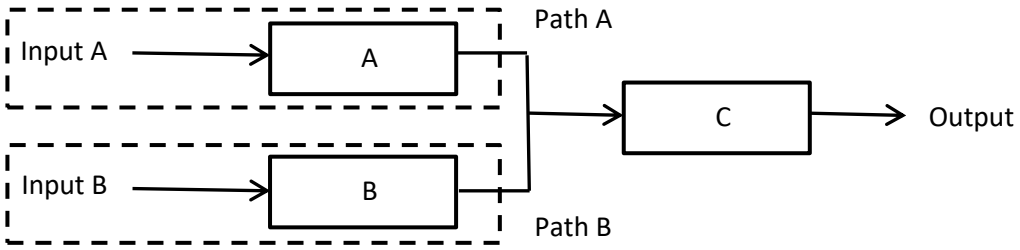


Figure 3.5: Availability of series and parallel redundant systems

The availability of individual components can be used to calculate overall system availability. This method allows a comprehensive analysis of availability of a system which utilizes scatted resources for the operation of individual blocks. The system shown in figure 3.5 is having two redundant inputs with redundant input processing blocks A and B. The output of the system is available either through path A or path B. The availability of the system (*Av)* can be defined in terms of availability of path A, (*Ava)* and availability of path B, (*Avb)* and availability of node C, (*Avc)* as [12];

$$Av = (Avc) \cap (Ava \cup Avb) \qquad (6)$$

Note that the system can perform expected output even if the path A or path B is completely down, due to redundancy arrangement. This means, there is no visible

indication to the systems operator in terms of systems availability due to one redundant node fail. This behaviour causes to hide errors to the maintenance team until the other working node also gets fail with the time. The need of careful systems monitoring becomes important in this case for early detection of redundant node failure. [15]

A study had been done by using realistic data collected from the systems operation team of the Southern Expressway emergency call system. The results of the study are presented from the next section.

3.2 Reliability Analysis of Southern Expressway Emergency Call System

Expressway operation centre is operating a short code number 1969 throughout the year for collecting emergency information, providing general details to the public. The system consists of two redundant SIP connections with the service provider, a voice gateway router, One Cisco Unified Communications Manager (CUCM) publisher, one CUCM subscriber, One Cisco Unified Contact Center Express (CUCCX) publisher, one CUCCX subscriber, one Quality Management (QM) and recording server and set of IP soft and hard phones with power over Ethernet supported data network. The same system is managing the communication with the other call center located at the Colombo Katunayake expressway, all IP communication in the entire region of Southern expressway and Outer Circular expressway.

3.2.1 Method of data collection and analysis

The aim of this study is to identify practical aspects affected on systems reliability in terms of avoidable instances through the systems monitoring and non-avoidable terms and calculate systems availability without systems monitor and with systems monitor.

Data had been collected from the systems maintenance team from year 2015 operation event log books, related to the Gelenigama Interchange, where the expressway control center is located. It is assumed that the team followed operation

and maintenance guidelines [8], instructions to complete event log books with accurate data in every incident in the critical system components. To fill up log entries, it had being reviewed automatic system logs related to the event, Network Management System (NMS) logs, Uninterruptable Power Supply (UPS) shutdown control card logs, equipment individual event logs and the records written by the end users (call operators). Since all system clocks are synchronized with local network time server, it is assumed that the automatic and manual log entries are accurate to nearest minute.

By considering the systems architecture, it had been able to calculate availability of each component of the system, hence total availability of the 1969 call system. Annex 01 consists of extracted few entries from the year 2015 technical incident log book (for reference only), which has some recorded information about systems breakdowns. The systems power supply arrangement to the system components with its calculated availability, based on the log book incident records can be found in annex 02 and network setup in annex 03 in form of single line diagrams.

It had being reviewed the full log book entries to identify root causes to the breakdowns. Table 3.2 illustrates summarized data by root cause for all systems based on Gelenigama Interchange. Table 3.3 consists of such availability values of each component, affected to the operation of emergency call system excluding planned outages of the system by using single line diagrams in annex 02 & 03 and complete log book entries and using equation *(5).* It is assumed that the sum of up time *MUT* and down time *MDT* is equals to the total time in minutes per 365 days. The device notation is based on the single line diagrams in annex 02 and annex 03.

Table 3.2: Reasons for systems breakdowns at year 2015

No	Reason	Number of incidents
1	Power related issues	17
2	Hardware problems	3
3	Software related issues	2
4	Human errors	5
5	Environmental related	8

Table 3.3: Actual availability of equipment at year 2015

No	Device Name (Based on annex 03 diagram)	Notation	Total Down Time in year 2015 (Minutes)	Availability (%)
1	GELA_AGG_01	L3SW1	105	99.98
2	GELA_AGG_02	L3SW2	105	99.98
3	GELA_VG_01	VG	210	99.96
4	SLT_SIP_TR_01	SIP_TR 1	30670	94.16
5	SLT_SIP_TR_02	SIP_TR 2	24280	95.38
6	CUCM_01		895	99.83
7	CUCCX_01		895	99.83
8	CUCM_02		1150	99.78
9	CUCCX_02		3625	99.31
10	GELA_ACC_04	L2SW3	160	99.97
11	GELA_ACC_06	L2SW4	315	99.94
12	Call Agent 01		840	99.84
13	Call Agent 02		995	99.81
14	Call Agent 03		420	99.92
15	Call Agent 04		365	99.93
16	Call Agent 05		365	99.93

Figure 3.6 illustrates the single line diagram, compiled to calculate total system availability using table 3.3 values. The calculation is based on the equation (6), for parallel and series systems. This is actual availability of expressway emergency call system for year 2015, by performing general procedures for inspection without having integrated systems monitoring.

It is observed that the some of the faults can be eliminated or down time due to faults can be reduced if an integrated monitoring system presents. Annex 04 consists of such analysis (for reference) made from the original reference document in annex 01.

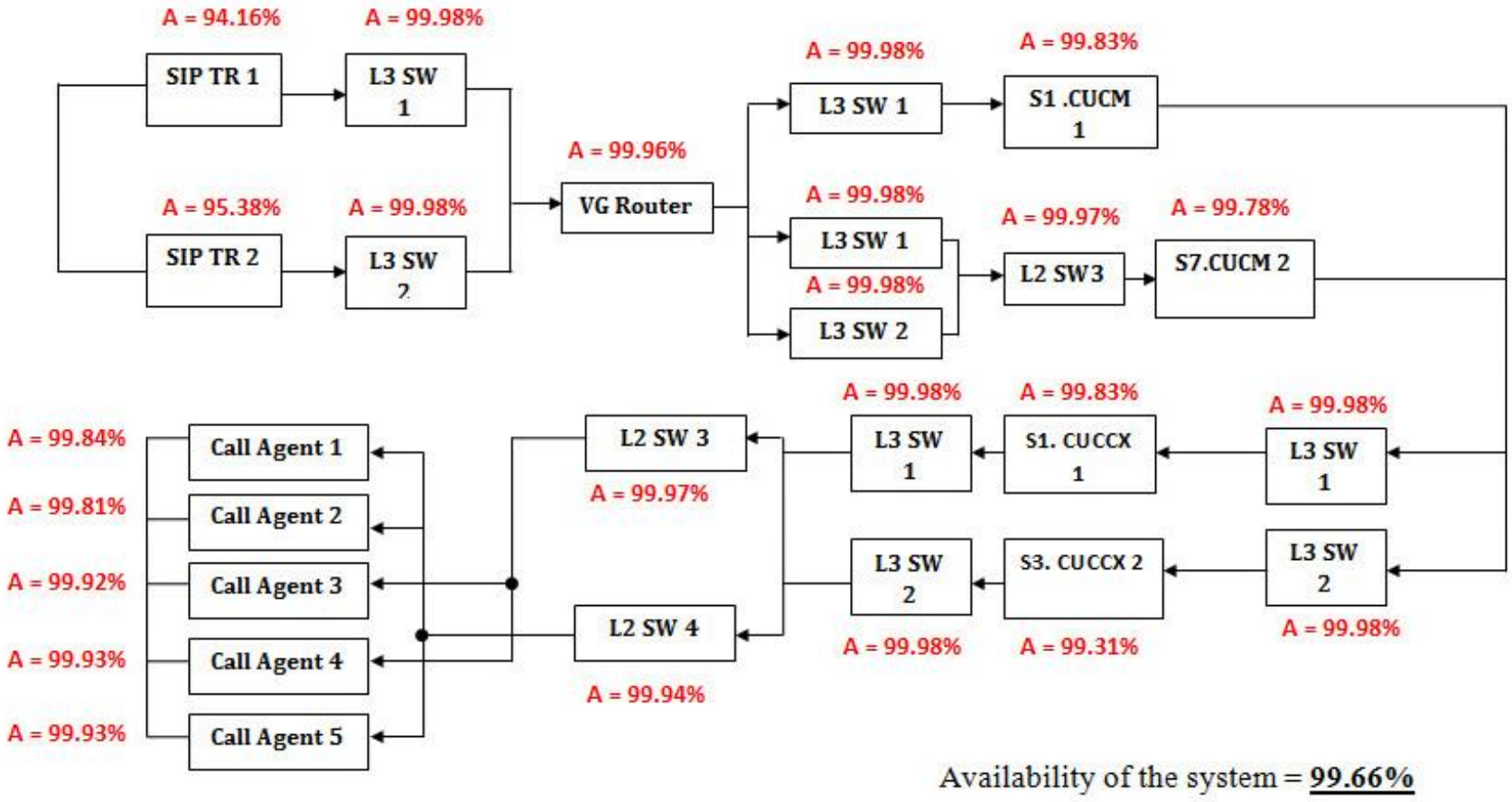


Figure 3.6: Actual systems availability for 1969 emergency call system in year 2015

Table 3.4 consists of recalculation of device availability after eliminating such incidents from all set of records. It is assumed that the detection time and attention time of an integrated monitoring system is 15 minutes and the crew is attending within specified time mentioned above. All other assumptions based to this calculation are stated in the annex 04.

Table 3.4: Predicted availability of equipment with an integrated monitoring for year 2015

No	Device Name (Based on annex 03 diagram)	Notation	Predicted Down Time for year 2015 (Minutes)	Availability (%)
1	GELA_AGG_01	L3SW1	0	100
2	GELA_AGG_02	L3SW2	0	100
3	GELA_VG_01	VG	0.5	100
4	SLT_SIP_TR_01	SIP_TR 1	12720	97.58
5	SLT_SIP_TR_02	SIP_TR 2	4730	99.10
6	CUCM_01		25	99.99
7	CUCCX_01		45	99.99
8	CUCM_02		40	99.99
9	CUCCX_02		50	99.99
10	GELA_ACC_04	L2SW3	0	100
11	GELA_ACC_06	L2SW4	0	100
12	Call Agent 01		25	99.99
13	Call Agent 02		165	99.97
14	Call Agent 03		0	100
15	Call Agent 04		35	99.99
16	Call Agent 05		35	99.99

Figure 3.7 illustrate predicted system availability with an integrated systems monitoring as stated above.

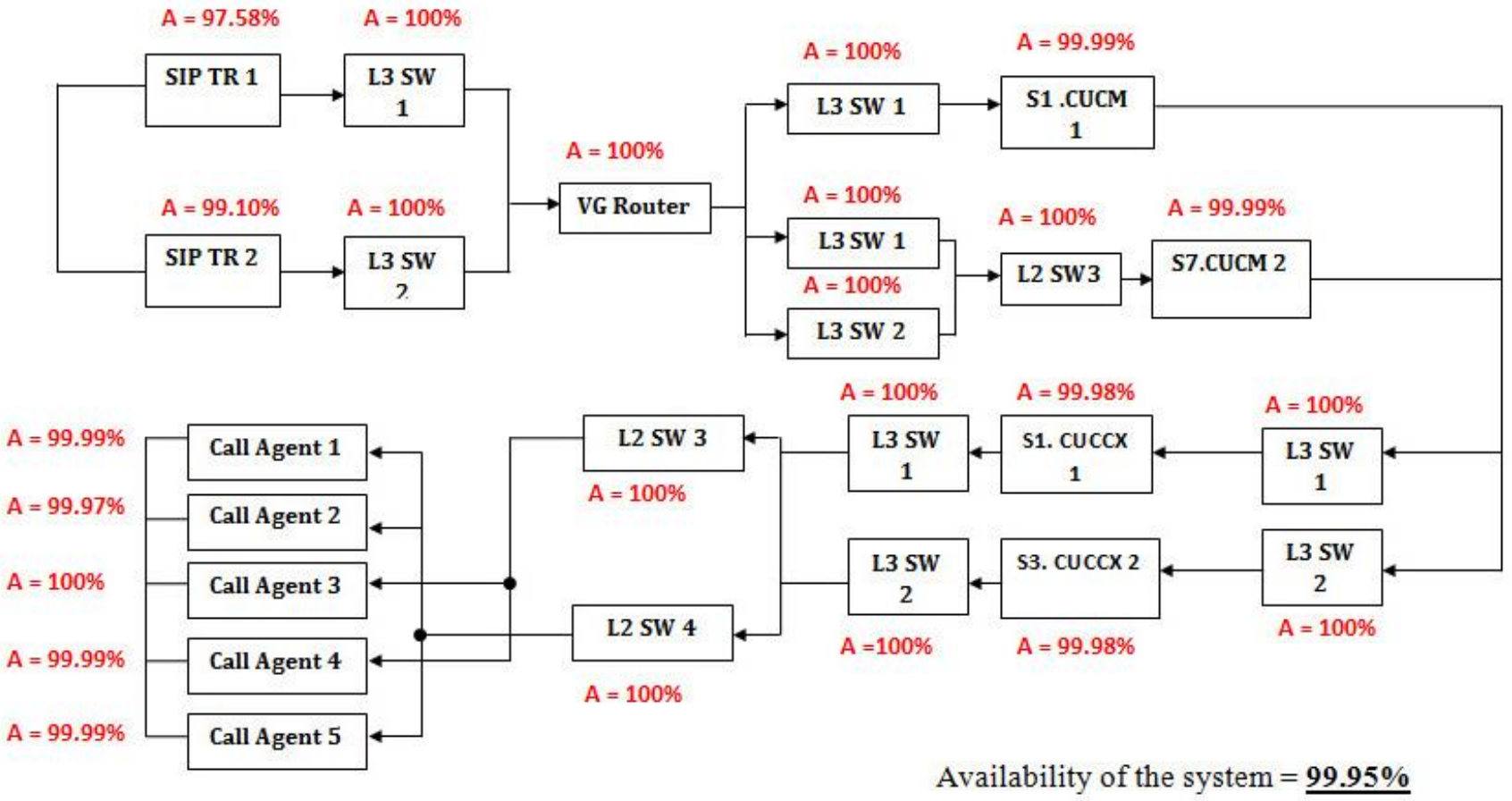


Figure 3.7: Predicted systems availability for 1969 emergency call system for year 2015

3.2.2 Observations and conclusions from reliability analysis of emergency call system

Following observations can be made through the above analysis.

(a) **Incident detection and reporting–** Due to lack of an integrated monitoring system, detection of event is difficult unless it affects or is visible as a disturbance to the operation. Detection of failures in the high available systems is difficult unless it is checked and observed by manually due lack of indication or until both active and high available nodes get failed.

(b) **Message passing procedure to technical crews –** Passing messages from the observer (call system operator) to technical crew introduces delays in early rectification.

(c) **Human factors –** Technical crews may forget the passed information to attend on breakdowns. Their performance evaluation is difficult due to loss of activity monitoring system.

(d) **Secondary failures –** As an example, if the input circuit breaker of uninterruptable power supply system is restored before the batteries getting dead, sudden system server shutdown events can be prevented. Restoring the operation will take time to reboot the servers, protocol negotiations etc.

(e) **Evaluation of systems performance/Identification of bottlenecks –** Post analysis of root causes for system non-availability is difficult due to lack of information.

(f) **Reference to historical reports –** Knowledge on repair history of equipment is not easy to access and the repair time can be optimized by using knowledge base of historical records.

Design of an integrated systems monitoring application to overcome above problems is described in the section 4.

CHAPTER 04

MONITORING SYSTEM – CONCEPTUAL DESIGN

The primary objective of an integrated systems monitor is to collect status from various sub systems, process the information to much simpler form and provide the processed information to the systems maintenance team in arranged manner. Design of such monitoring system requires careful analysis of each and every available system interface to collect data, identify data flows, processing elements and methods of data provision. This chapter presents the conceptual design of the integrated systems monitor.

4.1 Monitoring System – Main Functional Blocks

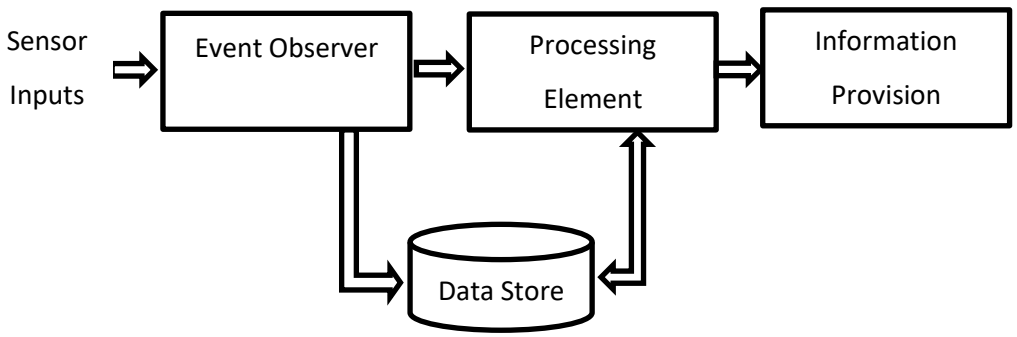


Figure 4.1: Simplified architecture of monitoring system

A typical monitoring system requires three main functional blocks as shown in figure 4.1 for operation. Additionally it requires data storage to store information for further analysis.

The process of designing a monitoring system requires an identification of key state variables to be monitored, designing of sensing methodologies, communication methodologies, cleanse and normalizing the collected data, data provision [16] as major steps. Those steps are described in detail in section 4.2.

4.2 Conceptual Design of Communication Control Application and Event Observer

Communication control application and the event observer of a monitoring system are responsible of collecting data about the status of the particular system to be monitored. The method of data collection and form of acquired data is depending on the monitored system and is highly varied from one to other. All available systems are described with the method used to acquire data for systems monitoring in section 4.2.1.

4.2.1 Interfaces available for data collection

Since the proposed monitoring system design is connected to monitor various types of communication, data processing and power related equipment, the event observer shall be capable of handling different input output methods as categorized in table 4.1.

Table 4.1: Types of interfaces available for systems monitoring

No	System	Sub System	Available monitoring Interface(s)	Form of Data
1	Electrical and power distribution	Backup Generator	Direct	Analog/Digital
		Master Distribution Panel including all breakers	Direct	Analog/Digital
		Automatic Transfer Switch	Direct	Analog/Digital
		Sub Distribution Panels	Direct	Analog/Digital
		Uninterruptable Power Supply Systems (UPS)	Direct/UPS monitoring card interfaces	Analog/Digital
2	Server and service status	Cisco systems – Voice and call center servers	Log files/Automatic e-mail alerts	text
		Bosch CCTV system	Log files	text
		Management Servers	Log files	text

(Continued)

Table 4.1: Types of interfaces available for systems monitoring

No	System	Sub System	Available monitoring Interface(s)	Form of Data
3	Network	Network Nodes	SNMP	SNMP responses
4	Data Links	Link status	SNMP	SNMP responses
5	Extended data links	Links to outsiders networks	Operation Logs of particular system	text
6	Environmental Conditions	Server room temperature, intruder alarms	Direct	Analog/Digital
7	Control Interfaces	• Generation of automated test signals • Generation of control signals	Direct	Digital output

4.2.2 Methods of data collection

It had been decided to use methodologies stated in table 4.2 to collect data from above listed interfaces by studying the operation of systems monitoring techniques in the background study in Chapter 01 and Chapter 02. However, a localized design is required to match with the available system interfaces.

Table 4.2: Data collection from system interfaces

No	Interface type	Method of data collection	Parameters
1	Direct (Digital I/O or analog inputs)	Through custom developed hardware module. Data can be sent through available TCP/IP network to minimize requirements of additional cabling	• Voltage measurement • Current measurement • Monitoring the status of the breaker or digital control signal (ON/OFF) • Generate digital outputs through the monitoring system to make controls • Status of an intruder or fire alarms • Measurement of room temperature in server rooms
2	System logs	Through log processing	• Server status • Service status
3	SNMP monitoring	Through SNMP polling application	• Node status • Link status
4	Special Interfaces	Through serial or USB interfaces	• UPS status

4.3 Conceptual Design of Batch Processing Application and Data Store

Data processing element of a monitoring system basically performs data comparison process of the collected data with pre-defined status values. Additionally it performs analysis to identify root cause of a series of events whenever possible. Build of data

storage is useful to generate system reports and to build knowledge base of events and repairs.

Processing of data may not require being in real time for some incidents. Generating analyzed reports, optimization of data store or alarm optimization process can be run later or as soon as the data collection cycle is completed. Sometimes, the system requires batch processing to optimize the fault alarms by removing duplicated records or erroneous alarms generated from the monitored end nodes by the root failure point. However, high priority alarms configured by the system could be able to process real time and forward them to the human machine interface without delay. Designed conceptual model of a data processing element is shown in figure 4.2.

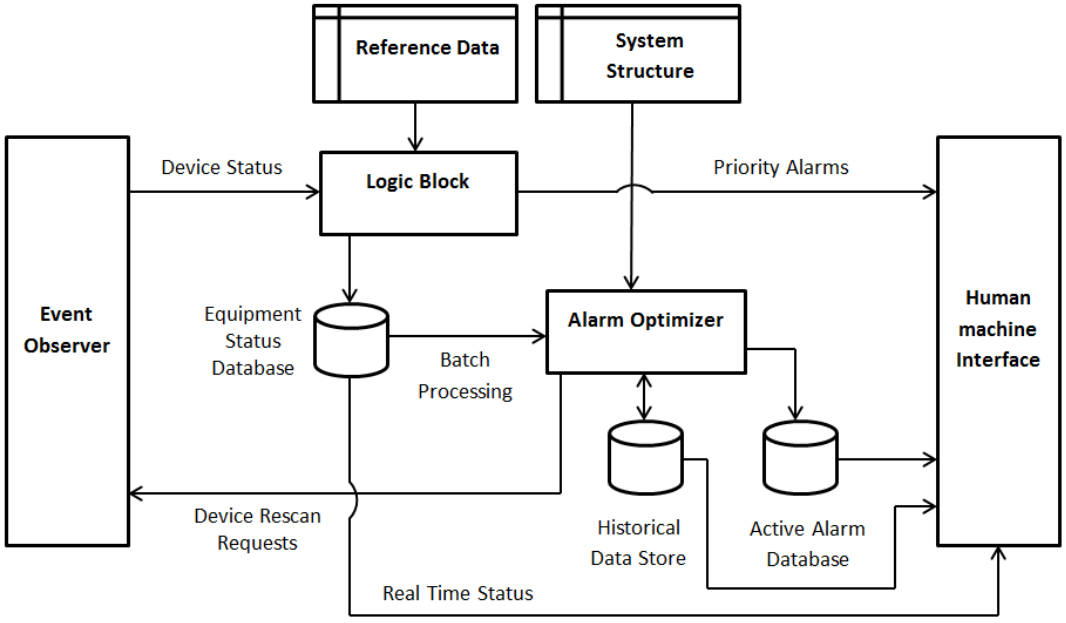


Figure 4.2: Simplified architecture of data processing element

4.3.1 Forms of collected and reference data and method of basic fault detection

Data collection and processing is based on the type of interface as shown in table 4.2 and the reference inputs need to be in similar manner. Defining the reference values has to be performed during installation procedure and can be kept unchanged during the operation. However, the system administrator can change the values of

references at any time and the updated reference values are considered to the logic processing from the next immediate processing cycle.

Basic faults can be identified by comparing the collected data value with the given reference. Table 4.3 lists the defined types of references and method of detection used for this design. Those results are raw data and will be used for the alarm optimizer.

Table 4.3: Type of references and method of detection

No	Parameter	Type of reference (*r*)	Method of Detection
1	Analog input	02 analog variables defining maximum and minimum	min < *r* < max or otherwise error
2	Digital input	02 status variables as *X1* and *X2*	If *X1* = *X2* ; *r* == *X1* or otherwise error If *X1* != *X2* ; status of digital input is ignored
3	Log file text	Keyword search by using method of text processing	If keyword matches; copy the line to database for further processing
4	Network device and link availability	Definition for SNMP timeout (*t*) and retry count (*Rc*)	If the device didn't response within time "*t*" and exceeds retry count *Rc* ; device or link has error

4.3.2 Procedure for fault alarms optimization and the concept of integrated fault monitoring

Consider the network architecture shown in figure 4.3. Assume that the monitoring system had been configured to monitor status of router R1, R2, R3 and the status of the end node. The central server which is running the monitoring application is

connected to the router R1. If the data link 1 fails, all nodes (R2, R3 including end device) beyond link 1 will report timeout errors even the failure is in link 1.

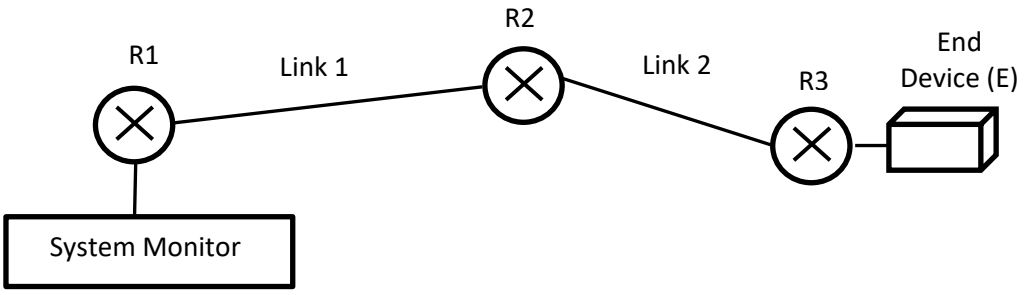


Figure 4.3: Alarm optimization for series systems

Alarm optimizer comes into action in such incidents to avoid projection of erroneous alarms. The system architecture shown in figure 4.3 can be programmed to the configuration database of the monitoring system which defines the node connectivity order. When the monitoring system detects an outage of end nodes (R2, R3 and end device E in this case) it can be automated to check its parent node availability in recursive manner, so the monitoring system faults finder ends up in router R2 (or Link 1 if configured) which is the closest guess for the fault.

During the data collection cycle, all node status will be collected and stored in the database. Then the alarm optimizer runs through the results for optimization. However, following extreme case is possible to happen, which also makes erroneous results during batch processing. Assume that the monitoring system checks the availability of router R2 at $t = 0$. When $t = T_1$ $(T_1 > 0)$, link 1 is down. Then the monitoring system checks availability of router R2 in $t = T_2$ where $T_1 < T_2$ and the system marks faults in the devices R3 and end device due to loss of connectivity. When the batch processor runs, it will end up in router R3 as the faulty device, but not in R2 which is the closest guess. To prevent such incidents, node rescan has to be performed before marking the R3 as the faulty node.

Consider the example system architecture shown in figure 4.4.

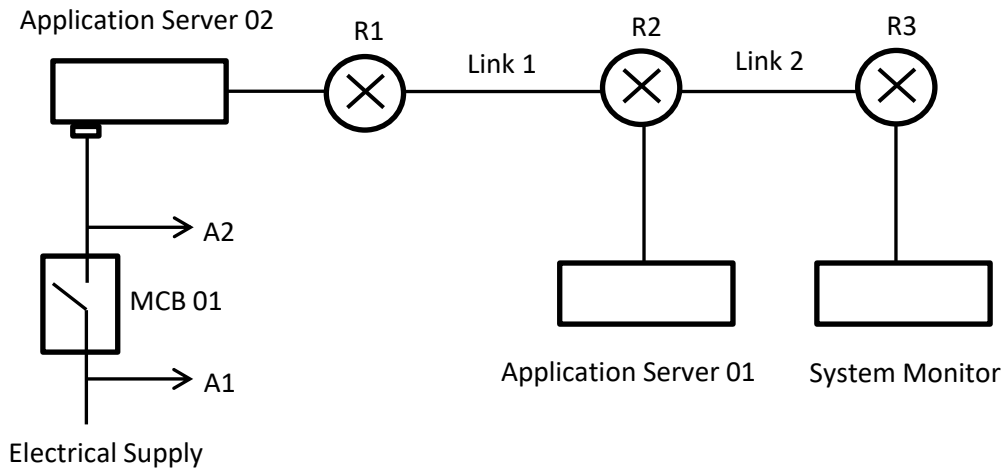


Figure 4.4: Alarm optimization method for complex systems

Assume that the monitoring space consists of R1, R2, R3 routers, link 1 and link 2 data links, application server 1 & 2 which run user program in cluster configuration. Also assume that the server 2 has only one power supply unit (no power redundancy) through MCB 01 and A1 and A2 are connected as analog inputs to the monitoring system through a data collection device. Consider the following cases.

(a) **Link 1 failure** – The system monitor detects R1 and application server 02 outage from SNMP monitoring responses. In the same time, application server 01 generates an error condition in its cluster demon process due to loss of connection with its paired device, server 02. The monitoring system detects the generated error through the log processing of server 01. Since the monitoring space structure is pre-programmed to the monitoring system, it will alarm the problems of link 1 or router R1 only and not the cluster demon error. How to define the system structure for such system is described in Section 5.3.2 of Chapter 05.

(b) **MCB 01 OFF** – This is an example for use of the new concept "integrated fault monitoring" to optimize alarms. If MCB 01 trips OFF due to some reason, the monitoring system will detect outage of application server 02 through SNMP and through log processing of server 01. From the network side, the monitoring system can alarm about outage of server 02, either due to

server problem or the patch code problem which connects server 02 with the router R1. Server properties can be bound with its electrical supply monitoring parameter A2 from the system structure definition, and then the monitoring system will converge to MCB 01, by considering A1 and A2 signals. The final alarm will generated only for MCB 01 in this way.

4.3.3 Alarm processing with heuristic knowledge base

Occurring repetitive failures in the same device or sub system is not rare case with continuously operated systems during its life span. The monitoring system design could use this scenario to identify following events.

(a) Repetitive failures could happen due to weakest device/link in the system. Monitoring system could store all events for every device that is configured for monitoring and could use the stored information to generate reports about weakest points in a particular system. Those data can be used to improve system design for maximum availability.
(b) Recovery procedures for a particular incident may be similar. The integrated monitoring system can be programmed to automatically link with similar type of previous repair incident reports for a particular fault. It can provide recovery procedures followed in previous incidents to the technical crew and will help to reduce the mean time to repair.

4.3.4 Data storage design

Data storage can be easily manageable through centralized data access software. The software shall provide secured and fast access to the database with pre-defined functions. MYSQL data base engine had been selected for this implementation due to its open source license and best past experience. Data base design for this system and driver class implementation details are given in Chapter 05.

4.4 Conceptual Design of Monitoring Control Application

Data provision of this monitoring system is based on following two methods.

(a) **Through web console** – The systems operator can monitor the systems status using web based user interface from the operator computer. Alarm monitoring and processing, system status monitoring, browsing through heuristic knowledge base, task browser activity and system configuration can be done through the same web interface.

(b) **Through short message service (SMS)** - Short Message Service (SMS) can be used to communicate information about abnormal status in the systems to the technical crews through the monitoring system.

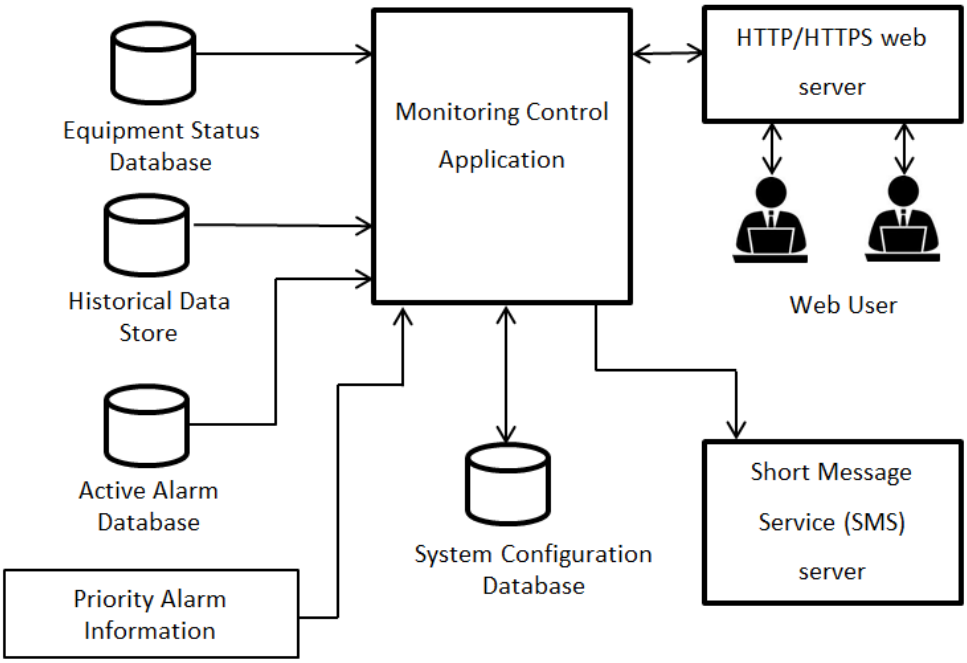


Figure 4.5: Data provision system design

The main functionality of the data provision is handled by the monitoring control application of the designed monitoring system. It collects information from the data storage and sends them to the web interface and SMS server accordingly. Figure 4.5 illustrates the simplified architecture of the data provision element.

4.5 Total System Architecture

The full system architecture can be considered as the combination of all communication control application, batch processing application, monitoring control application and the data base. Figure 4.6 illustrates such combined system architecture and more information is given in the Chapter 05 regarding the implementation.

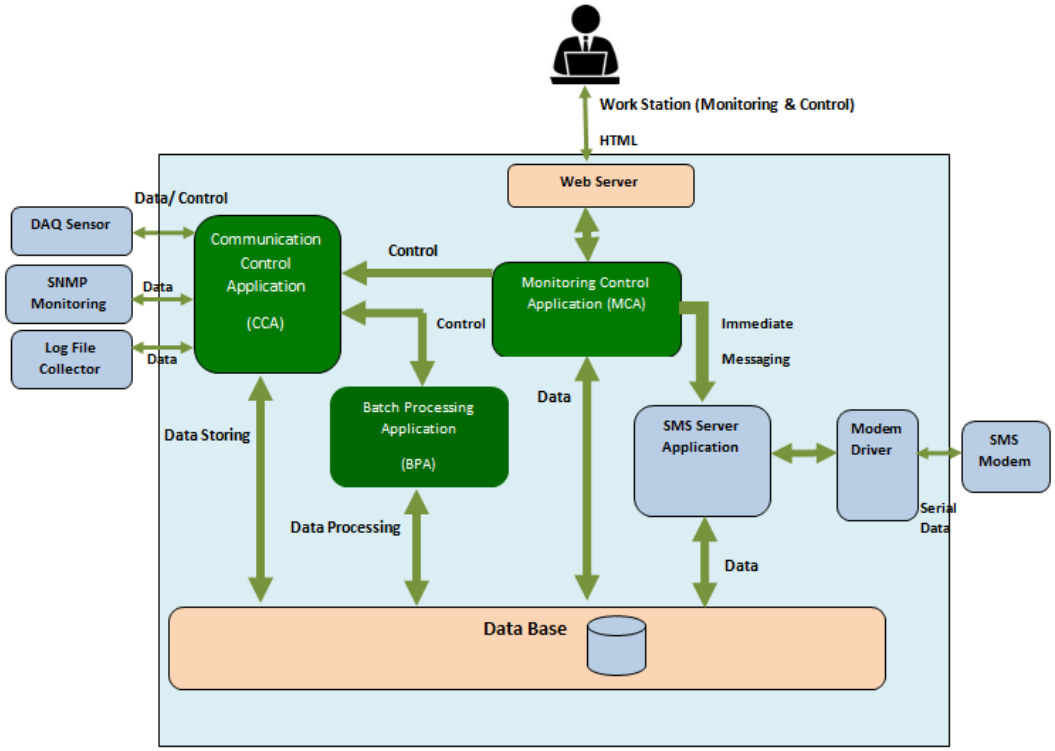


Figure 4.6: Total system architecture

Getting an idea about the empirical performance of the designed system would be useful before implementing the system. Section 4.6 describes such analysis on the system.

4.6 Theoretical Models for Empirical Performance Analysis

The designed monitoring system requires analysis about various key performance factors before making it practically. Time to detection, reliability of system alarms and stability of the system are some of them. [16]

4.6.1 Time to detection

The designed monitoring system can be modeled by using time delay blocks as shown in figure 4.7. Each functional blocks (time to detect the problem, time to communicate the results with the next stage or the processing time inside the block) of the system can be denoted with equivalent delay blocks. Table 4.4 describes the notation used in the figure 4.7 and their usual empirical values per one item/data process.

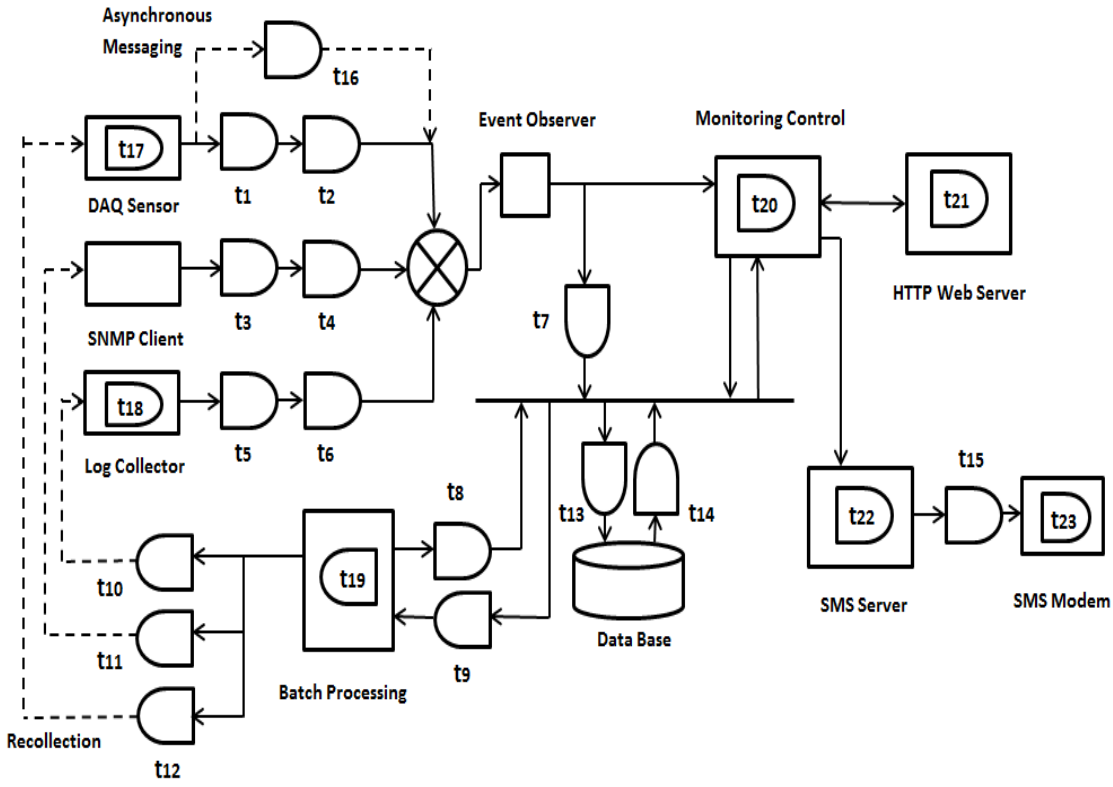


Figure 4.7: Detection and processing delays

Table 4.4: Empirical time delays

No	Parameter	Description	Empirical Value
1	t_1	Propagation delay in DAQ unit	50 ms
2	t_2	Communication channel delay and data read delay by the event observer	20 ms
3	t_3	SNMP response time	5 ms
4	t_4	Communication channel delay and data read delay by the event observer	20 ms
5	t_5	ftp of SSH response time by the server	50 ms
6	t_6	Communication channel delay and data read delay by the event observer	200 ms
7	t_7	Event observer data base access time and data write time	20 ms
8	t_8	Batch processor data base access time and data write time	100 ms
9	t_9	Batch processor data base access time and data read time	100 ms
10	t_{10}	Log file recollection request time	500 ms
11	t_{11}	SNMP recheck request time	20 ms
12	t_{12}	DAQ module data recollection request time	80 ms
13	t_{13}	Database access – Write time	50 ms
14	t_{14}	Database access – Read time	100 ms
15	t_{15}	Serial data – communication time delay	50 ms
16	t_{16}	Asynchronous Messaging – Link delay and data read delay by the event observer	50 ms
17	t_{17}	Time to complete one cycle scan of its parameters by the DAQ device	4000 ms
18	t_{18}	Log file creation time by the monitored service/software	1000 ms

Continued

Table 4.4: Empirical time delays

No	Parameter	Description	Empirical Value
19	t_{19}	Data processing time by the batch processor	100 ms
20	t_{20}	Data processing time by the monitoring control application	100 ms
21	t_{21}	Response time by the http web server	200 ms
22	t_{22}	Processing delay by the short message service server	1000 ms
23	t_{23}	Message send time taken by the SMS modem	2000 ms

The event observer in the communication control block handles all communication with available interfaces by using sequential polling. Each interface has different propagation and communication channel delays. Consider following operating conditions. Also note that the order of parameters is arranged according to the direction of data flow in the diagram.

(a) Consider the system collects data from single data acquisition sensor (DAQ) module with all parameters are normal. Then the operation cycle takes following time delay to complete one cycle of data collection and processing.

$$T_a = t_1 + t_2 + t_7 + t_{13} + t_{14} + t_9 + t_{19} + t_8 + t_{13} \quad (7)$$

(b) Consider the system collects data from single data acquisition sensor (DAQ) module with one or more parameters are not normal. In such incidents the event observer is programmed to recollect data from the DAQ module. Then the operation cycle takes following time delay to complete one cycle of data collection and processing.

$$T_b = t_1 + t_2 + t_7 + t_{13} + t_{14} + t_9 + t_{19} + t_8 + t_{13} + t_{12} + t_{17} \quad (8)$$

To reduce the recollection time delay, asynchronous message path can be configured to the monitoring system. In that case, DAQ sensor node will send asynchronous stream of data to the event observer regarding its abnormal parameter(s) irrespective of the polling request. Then the complete monitoring cycle will be completed in timeT_{b1} where $T_{b1} < T_b$, which is an improvement of the performance.

$$T_{b1} = t_{16} + t_7 + t_{13} + t_{14} + t_9 + t_{19} + t_8 + t_{13} \quad (9)$$

(c) When the DAQ module is offline or disconnected from the system, response time t_1 becomes to infinite. The internal timeout setting in the event observer software will be required to exit from the read thread, waiting for response in such situations. However, the internal timeout setting shall be more than usual values of $t_{17} + t_1 + t_2$ to avoid erroneous time outs. `

(d) When the monitoring system have *n* number of DAQ sensors, having total processing time for one cycle in normal condition will be;

$$T_{AQ(a)} = \sum_{k=1}^{n} T_{a(k)} \quad (10)$$

And, when it is having a problem in one or more parameter;

$$T_{AQ(b1)} = \sum_{k=1}^{n} T_{b1(k)} \quad (11)$$

With *n* increases, total time taken to complete one cycle will be higher, resulting performance of the monitoring system. Multiple threads of event observer can be run in the monitoring system server as a solution to this problem. Assume that the monitoring server is configured to run *m* number of parallel processes of event observer, total cycle time will be for an erroneous condition will be;

$$T_{AQ(b1)} = \frac{\sum_{k=1}^{n} T_{b1(k)}}{m} \quad (12)$$

Then the error will be displayed in the web user console in;

$$T_{web} = \frac{\sum_{k=1}^{n} T_{b1(k)}}{m} + t_{14} + t_{20} + t_{21} \quad (13)$$

And a SMS message will be sent regarding the detected error within;

$$T_{SMS} = \frac{\sum_{k=1}^{n} T_{b1(k)}}{m} + t_{14} + t_{22} + t_{15} + t_{23} \quad (14)$$

Usually, the value of *m* will be depending on the performance of the server system running the monitoring software and data link bandwidth. However, with mid-range performance server with 1Gbps link capacity, the value of *m* can be high as 50 to 100 without much issue. However, the limitation happens in the section of SMS server which can send only one SMS message at a time. For an instance, having *p* numbers of independent incidents to *y* number of subscribers the last SMS message will be sent after;

$$T_{SMS} = \frac{\sum_{k=1}^{n} T_{b1(k)}}{m} + y\left(\sum_{q=1}^{p} (t_{14} + t_{22} + t_{15} + t_{23}) \right) \quad (15)$$

However, it is possible to improve this scenario either by subscribing with multiple SMS gateways or connecting to the bulk SMS services provided by 3rd party organizations. This enhancement had been not considered for the scope of this research.

Assume that the system has 1000 DAQ nodes to monitor. With the empirical values from table 4.4, it can be estimated the time taken by the monitoring system to

indicate a fault at monitored system by using equation 13 and 14. Assume that the value of *m* is set to 50 and all processing steps for 1000 nodes take identical delays.

T_{web} = (1090 * 1000)/50 + 400 = 22.2 seconds

T_{SMS} = (1090 * 1000)/50 + 3150 = 24.95 Seconds

However, T_{SMS} does not include the time to deliver SMS to the subscriber's mobile. The delivery time will be decided by the mobile operator's conditions and coverage.

(e) SNMP agent will take following time delay to complete scan of single node when the node is active.

$$T_e = t_3 + t_4 + t_7 + t_{13} + t_{14} + t_9 + t_{19} + t_8 + t_{13} \quad (16)$$

(f) When the particular node is down or disconnected, the response time t_3 becomes infinite. The SNMP monitor software exits from the node scan after time out setting of T_o with *K* number of retries performed. Then the system performs recursive scan on the same path to find last active node. Assume that the N^{th} node has *N*-1 number of failed parent node scan tries from the last active node as shown in figure 4.8. During this process, all *N*-2 node status will be marked as "unknown" and the node just after the last active node will be marked as "offline".

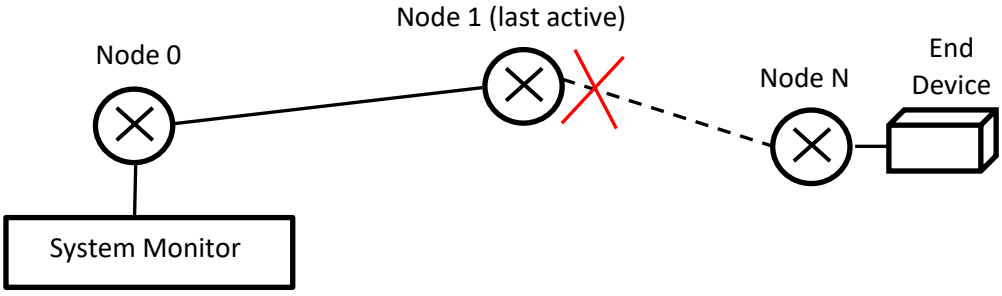


Figure 4.8: Detection of network node outage

The detection time of the failure point from the recursive check and update results to the database will be;

$$T_f = (N-1)\,(T_0\ K + t_8 + t_9 + t_{13} + t_{14} + t_{19} + t_{11}) + t_3 + t_4 + t_7 + t_9 + t_{19} + t_8 + t_{13} \quad (17)$$

When the monitoring system has fairly large number of nodes, the detection time will be higher. To overcome this problem, the event observer can be configured to run multiple instances, similar to the DAQ sensor scenario.

4.6.2 Monitoring system reliability and stability

Even though the monitoring system reports its monitored system status in normal conditions, the process of monitoring or the method of data collection may encounter problems. This might result in erroneous alarms or nothing will be reported at all. A faulty sensor, loss of communication with the sensor or monitored device, fault in the protocol negotiation, corrupted data or wrong logic due to system programming error or may be the monitoring system server problems can lead to faulty alarms or no alarms at all.

The system design must ensure the self-check or fault tolerant approach as much as possible for high reliable monitoring. Consider the following scenarios.

(a) **Use of redundant check points** – The system can be configured to have check points as shown in the example monitored system in figure 4.9 to improve reliability. In this approach, the parent node status sensors, S1 and S2 can be configured to check with its end node sensor S3 results. For an example, when 230V supply is present and all breakers are ON, if S1 indicates the analog measurement of 0 volts or S2 indicates 0 due to a sensor error, can be detected by the S3 reading or from the operation status (if any) monitoring on TCP/IP network. However, this approach is only possible for higher level nodes and the base level sensor results cannot be validated.

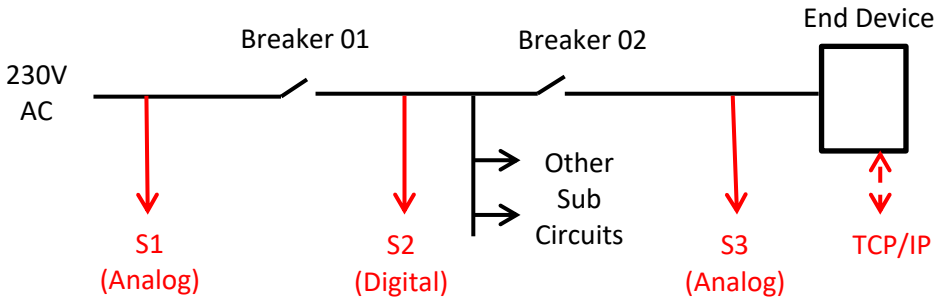


Figure 4.9: Method of redundant check points

(b) **Adding redundant Nodes** – The monitoring system components can be duplicated to get redundant functionality. For an example, setting up two short message service (SMS) modems in active-active configuration will improve the throughput and the reliability.

(c) **Loss of connectivity with the sensor module** – If the connectivity losses with one or more sensors or with the monitored system, it will reduce the stability of the monitoring system. In such cases, the monitoring application designed to perform recursive backward scan until it find the last active node and will report the outage. If the outage is present in the sensor module itself, then the monitoring system will produce sensor offline error message.

(d) **Corrupted sensor data/log file** – Corrupted data can cause errors in log or data processing in the monitoring system. However, self-error detecting mechanism required to be integrated with the sensor module communication design and more information are presented in the Chapter 05.

The implementation of the monitoring system is discussed in detail in Chapter 05, by considering design facts discussed so far.

CHAPTER 05

MONITORING SYSTEM – IMPLIMENTATION

Usual method of implementing any type of relatively complex systems is to breakdown of the total system to several sub modules. Each sub module can be tested and validated its transfer characteristics and finally they can be integrated to one system. The modular approach allows greatest flexibility for system changes, enhancements without changing much during its operation [17]. Similar approach is used to develop the monitoring system by dividing the system into following sub modules.

(a) Data acquisition sensor module development
(b) Communications control application development
(c) Data base system development
(d) Batch processing application development
(e) Monitoring control and web user interface development
(f) Short message service server implementation
(g) System integration and setting up initial configuration in the database

Implementation details on those modules are described from the next sub section onwards.

5.1 Method of Interconnection

All modules and devices associated with the monitoring system require interconnection for data communication. The systems at the expressways consist of Transmission Control Protocol/Internet Protocol (TCP/IP) based network with 1Gbps links. IP address assignment is based on static IP configuration. The same structure can be used easily for the implementation of the monitoring system interconnections. And it had been decided to use open protocols for all system

interconnections. Figure 5.1 illustrate such information about communication methods planned to use for this implementation.

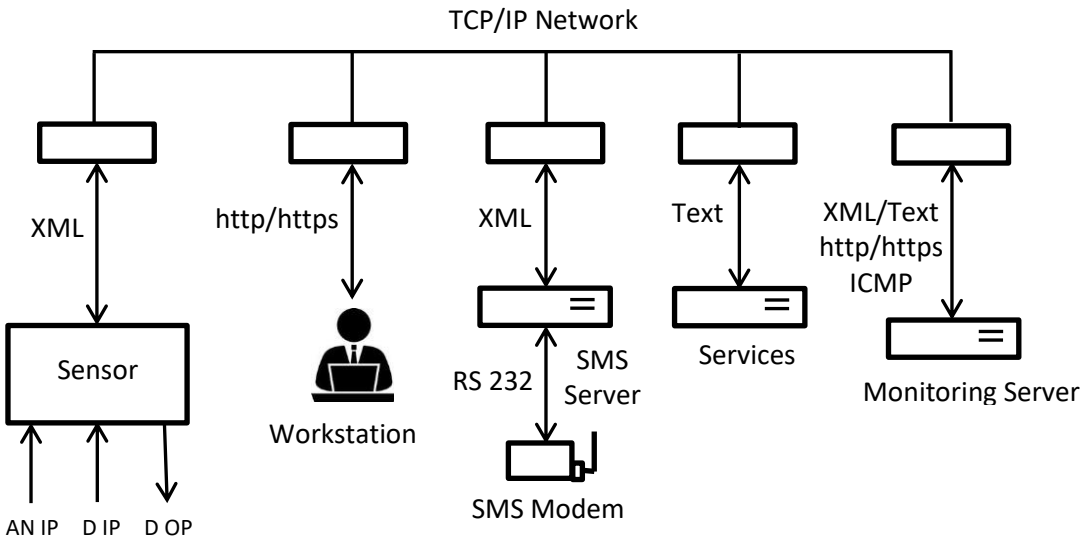


Figure 5.1: Method of interconnection

Extensible Markup Language (XML) [18] is selected due to its simplicity and open source technology to design most of the internal communications. And it is well supported for the TCP/IP network based communications.

5.2 Data Acquisition Sensor Module Implementation

The monitoring system requires collecting status data from various electrical, power and environmental condition readings. Collecting such data parameters requires passing through various steps like acquiring parameter data through sensors, analog to digital conversion, basic comparison with internal references and transmission of acquired data to the monitoring application server. The device must support to connect with TCP/IP links. Hence the module has to be designed with a network interface.

Figure 5.2 illustrates the tropology of one such sensor module designed in this project. One main sensor module can support up to six (06) sub modules. One such sub module can handle 06 analog voltage inputs, 12 digital inputs, 01 temperature measurement and 03 digital outputs totaling 36 analog inputs, 72 digital inputs, 18

digital outputs and 6 points for temperature measurement. All can be monitored/controlled through a single IP address assigned to the main module through XML data and Representational State Transfer (REST) [19][20] web service running in the main module. This allows monitoring large number of parameters through one IP address assignment.

All sub modules are powered through the main module power. Sub modules are also equipped with own controller chip for data acquisition and processing to increase the performance of the sensor system. They can be connected with 4-wire cable, carrying serial transmit and receive data with two other power supply wires with the main module. Number of connected sub modules can be set from one to six depending on the requirement at the site. Also number of parameters of particular sub module can be changed to lesser numbers by the design.

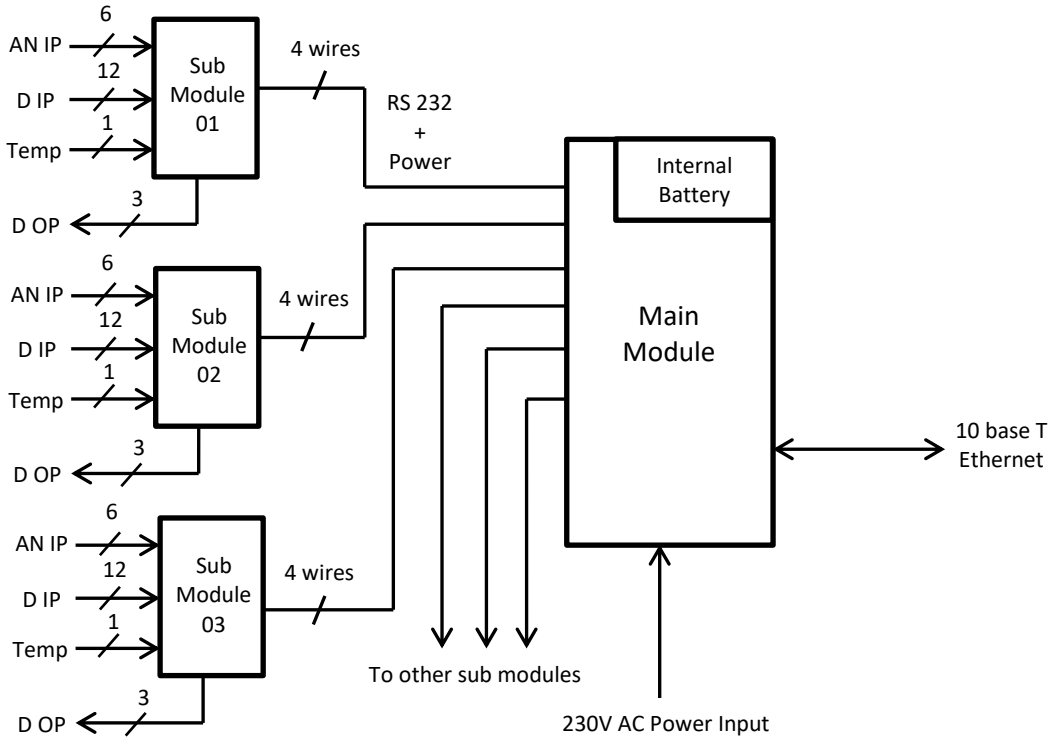


Figure 5.2: Sensor module tropology

Main module has no direct interfaces to acquire data. It is designed to act as a hub for sub modules. Main module is designed to support auto negotiation with sub modules. Also it monitors its link performance with sub modules. Rechargeable Li-

Iron battery pack is included to provide backup power to entire system during power failures. 10 base T half duplex interface allows to access the device through the network. Those units can be duplicated without limit (theoretically) over the monitoring space, allows flexible data acquiring sensor network. More detailed information about the design of this is presented in the next section of this document.

5.2.1 Detailed implementation details of main module

The main module has following functionalities required for the operation.

(a) The module requires Ethernet interface to connect with the network.
(b) Module firmware must support REST web service for GET requests.
(c) The output of the module is in the form of XML data.
(d) Web service security is essential to prevent intruder access and have embedded to the module firmware.
(e) Main module must support up to 6 sub module interfaces with power supply. Communication links will be RS 232. Sub modules shall be auto detected and configured by the main module automatically.
(f) All communication requires error detection mechanism to prevent forwarding incorrect results.
(g) Main module power extensions to sub modules require short circuit protection feature to prevent damages possibly happening to the module power supply unit.
(h) The total system should be operating continuously when the main supply fails through internal rechargeable battery pack. The batteries must be recharged automatically by the module and should provide protection against deep discharge.
(i) Sensor module must have self-error detection mechanism. All system status data must be available to access via the network interface.
(j) Main module parameters shall be configurable easily according to the site installation.

Figure 5.3 illustrates the block view of designed hardware for the main module with 04 interfaces available for sub sensor units. The constructed hardware based on this design is shown in figure 5.4.

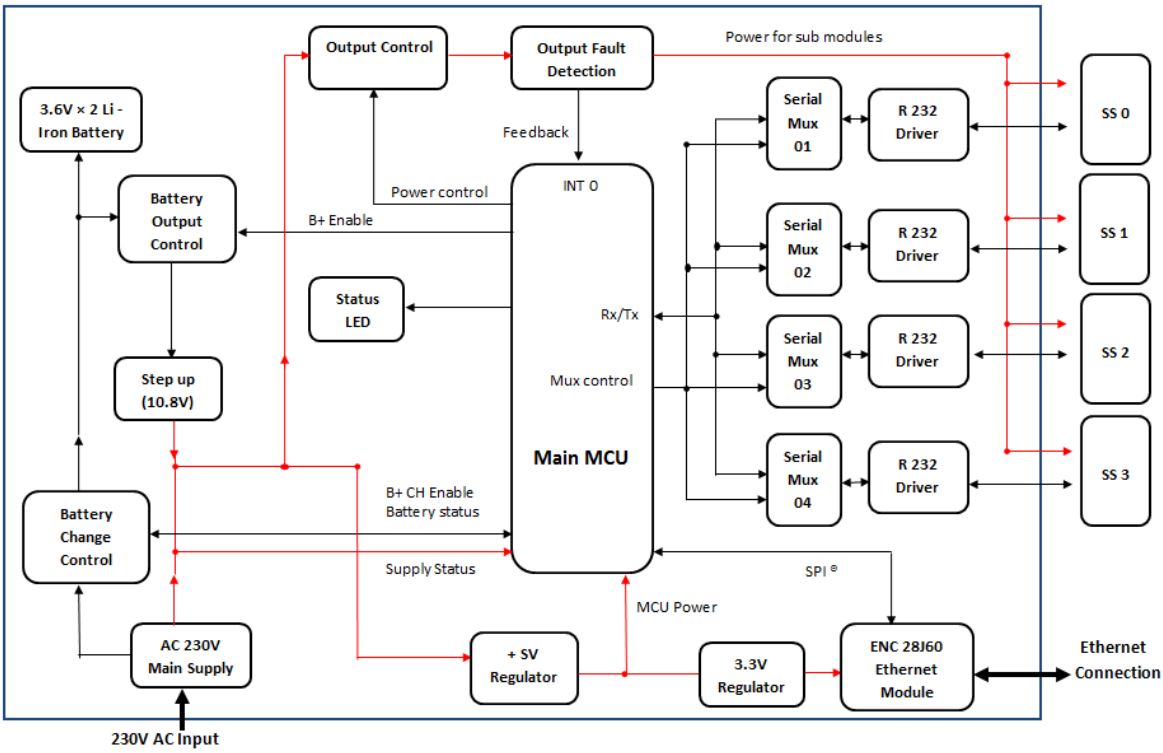


Figure 5.3: Block view of main module hardware

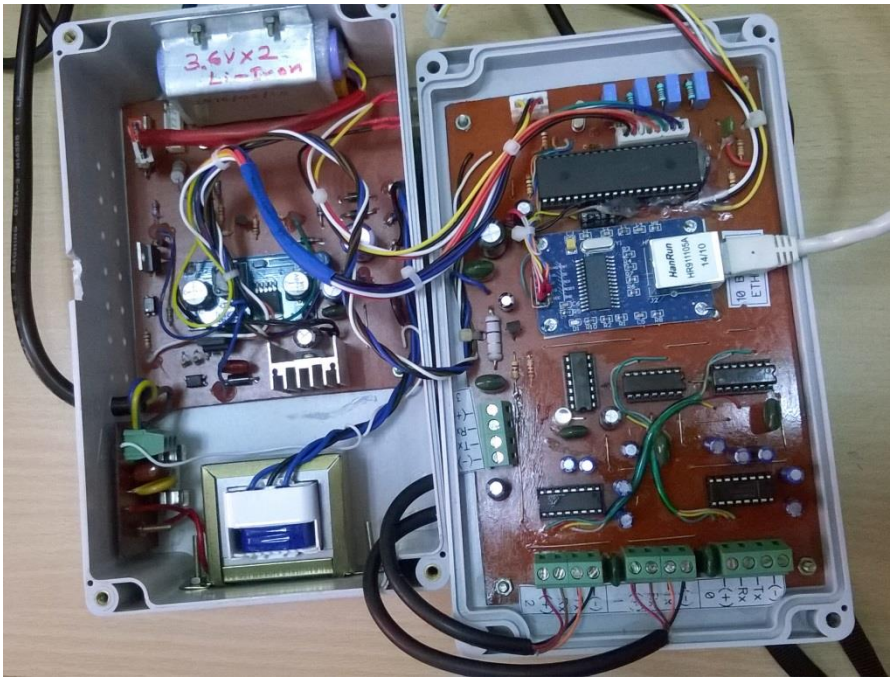


Figure 5.4: Prototype main module

PIC18F4520 running at its full speed of 40MHz is used as the main control MCU in this design. It is responsible of handling all input/output data through the module. One of the main functionality of the MCU is to control ENC28J60 SPI Ethernet module [21] through its serial peripheral interface (SPI). The MCU firmware acts as synchronous REST web service functions, which is necessary to communicate with the monitoring system server over TCP/IP network.

However, all incoming requests are filtered with an IP address filter written in the device firmware to restrict access on the device to ensure web service security. Only two IP addresses are allowed (monitoring system server and maintenance computer) to connect with the module and they can be configured through device configuration page of the module. All other requests coming from unauthorized IP addresses are replied with http/1.1 status message 403 – Forbidden. [22]

Asynchronous transmission mode is set on the master module to inform abnormalities in the measured parameters to the system server without waiting for polling requests, reducing the fault detection time. In this method, if one or more measured parameters are deviated 10% off from the defined range, it will generate asynchronous UDP request to the system server UDP application by the MCU.

REST service interface, response XML data formats, http status responses, UDP communication and information on device configuration interfaces are available in annex 05 in this document.

Other main duty of the micro controller is to communicate with sub modules attached to the main module. It is achieved through its multiplexed serial interface, communicating with one sub sensor at a time.

When the system is powered up, the main module sends “configuration request” command to all its sub models through serial interface. Then sub sensors will reply with its configuration data as a 10 byte string. If one or more sub sensor interfaces does not respond (either they are not connected or not working), the main module marks them as inactive. However, the availability checking will be continued once per 10 scan cycles in all inactive interfaces, allowing to detect sub modules (hot pluggable) even during the operation.

After auto negotiation, main module will send “data request” commands embedded with digital output control information to sub modules. Then the sub module

activates received digital output control data to its digital outputs and sends full set of collected data parameters as a 10 byte string to the main module. If one or more modules fail to send serial data (disconnected or faulty during operation) the main module marks it as inactive. All serial communications had been designed to run as interrupts to the MCU.

All serial link protocols are designed to having parity check to detect any possibility of data corruption. The link performance can be monitored through main module configuration page or, through the monitoring system automatically. This feature allows identifying possible interferences happened on serial communication links, which will reduce the performance of the system. The link performance can be measured in the form of number of parity errors detected in particular channel. In practice, shielded cables (up to 10m) are used for the device testing and none of errors had been observed.

More information about the main module to sub module communication are presented in the Section 5.2.2 and serial communication protocols designed and used for this implementation are presented in Annex 06.

Serial links with sub modules have possibility of happening short circuit or over current conditions, either due to damaged cables or due to electronics failures in sub modules. This can lead damages to the power supply section of the main module, if it continuous. Such overload conditions on sub module supply will be immediately detected by the output fault detection system of the main module and the system shuts down the power flow to the outside sub sensors. The total outside sub module current is limited to 0.8A. However, transient conditions (due to filter capacitor charging in sub modules during power reset etc) will be filtered out with low pass filter, containing RC time constant value of 0.22 seconds to prevent such transients triggering the MCU erroneously.

To simplify the requirement of having uninterruptable power supply source to the module, the main unit is designed to equip with a rechargeable battery, which delivers power to the main module and all sub modules in case of module main power failure. Hence the battery charging and management functions needed to be included to the design. The battery pack consists of 02 cells of Li-Iron, 2800mAh capacity each; 3.7V connects in series, producing 7.4V output. However, the module

main DC bus voltage is designed to be kept at 10.8V; step up converter module based on LM2576 chip is used to boost the battery voltage. The MCU is responsible for all controls of the battery such as safe charging, output enable and preventing it from deep discharge. Simple current-voltage detection method [23] is used to detect full charge condition during the charging cycle. Annex 07 illustrates the power management cycle, performed through the MCU functions.

The main module operation can be monitored visually by using two status LEDs (Green and Red) provided with this design. Table 5.1 illustrates the designed combinations of LED indications, controlled by the MCU firmware.

Table 5.1: LED indications

No	Green LED	Red LED	Message
1	Steady ON	OFF	Operation initializing
2	Fast blink	OFF	In operation – from mains power
3	Slow blink	OFF	In operation – from Internal battery
4	Steady ON	ON	Sub module power turned OFF due to detection of short circuit
5	Blink	ON	Link performance with sub modules is low
6	OFF	ON	Internal system error

Each master module is designed to hold a 16 bit unique identification number, assigned by the monitoring system server during initial configuration. 16 bit allows the system to be expandable up to 65535 master modules as one system. It saves in EEPROM of the MCU and must remain unchanged during the operation. The number is the primary key of the master module configuration database in the system server.

Complete logic flow of firmware routines and the circuit diagram is given in Annex 08. The printed circuit board layouts and device firmware (written in MikroC® version 6.6.0) of the main module are given in Appendix 02. (Included in CD)

5.2.2 Implementation details of sub modules

Sub modules can be designed and customized according to the requirement of the site, but limited to having maximum of 06 analog voltage inputs, 12 digital inputs, 01 temperature measurement and 03 digital outputs per device. Also it can be developed to obtain power from the main module or, self-powered options with isolated serial bus for special purposes. Following main functions were identified and added to the design flow of the sub sensor.

(a) Perform analog to digital conversion to measure the analog voltages
(b) Detection of status of the voltages in form of digital values
(c) Read environmental temperature
(d) Produce remotely operated (through the web service) digital outputs

The sub sensor is communicating with the master module by using RS 232 communication at 9.6kbps rate. In the initial power ON state, the master requests the configuration descriptor (10 data bytes contains the sub module parameter arrangement) from the sub module. Then in each collection cycle, sub module transmits 10 bytes of data containing measured parameter values with its parity byte to the main module. Detailed description of the designed communication protocol is given in Annex 06 in this document.

Two methods of power supply methods are possible to implement with the sub sensors depending on the requirement. Figure 5.5 illustrates the usual method of power supply, which acquires power from the main module. In this case, all measurement signals and all digital outputs require isolation with the sub sensor control board. Additional layer of isolation will affect to the performance of measurement, especially for analog measurements. But the measurement or control commands can be connected from different power supply and control schemes without considering the common plane due to the isolation layer.

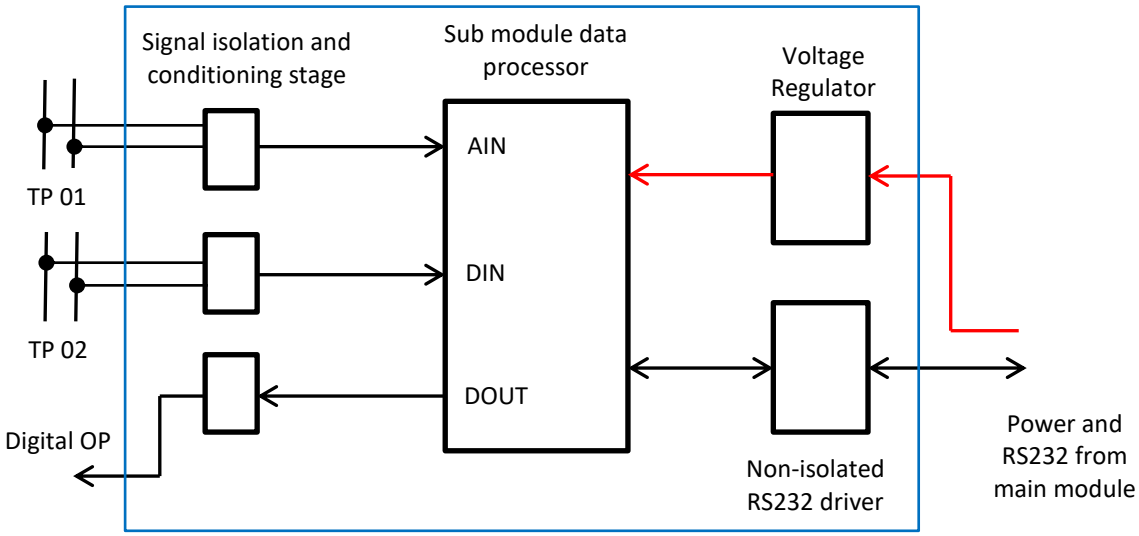


Figure 5.5: Power supply arrangement from main module

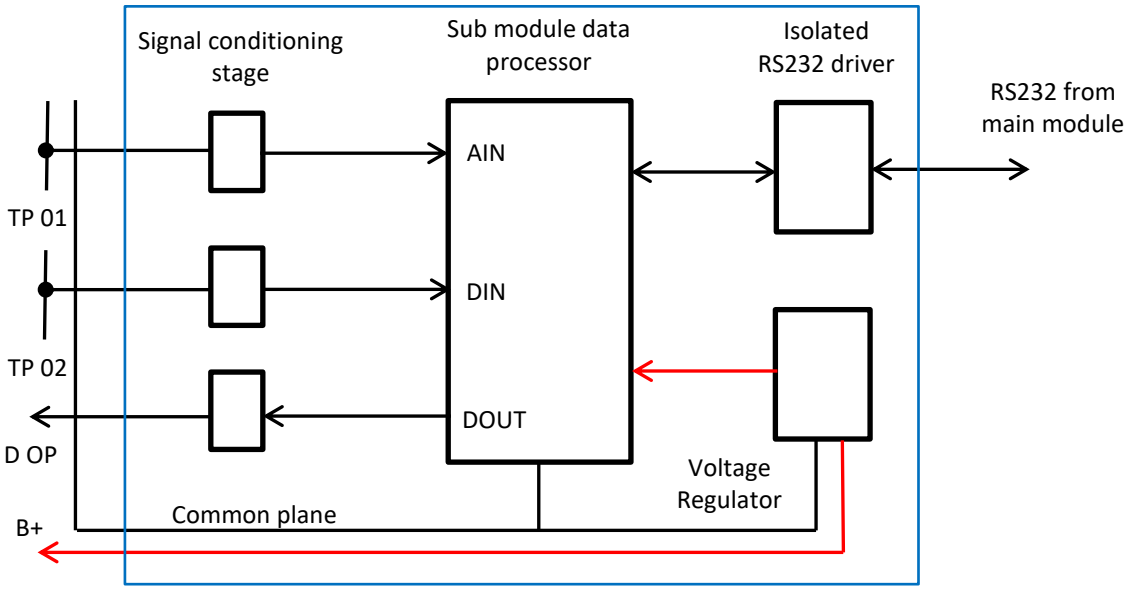


Figure 5.6: Self powered arrangement with common plane for signals

Figure 5.6 illustrates the other possible method of operating the sub sensor, which acquire power from the 3rd party system. The advantage of this method is the simplicity of the design due to not requiring isolation for each individual channels. However, the common for all measurements/outputs have to be same. Also the data acquisition stage can be directly coupled with the measured system. This method is more suitable for measuring small scale DC or AC voltages, states of digital control signals and driving logic control signals etc. Requirement of providing power from

the measured system power source, lack of backup power through the main module and requirement of having isolated RS 232 driver are the disadvantages of this method.

Practical implementation had been carried out to monitor/control with power distribution panels and automatic transfer switch panels; sub sensor was designed to obtain power from the main module with isolated signal inputs/outputs as in figure 5.5.

Figure 5.7 illustrates the block view of the designed sub module device to measure 230V AC analog readings and 230V AC as digital signals to observe presence or absence of the power. Figure 5.8 illustrates the photos of constructed prototype hardware.

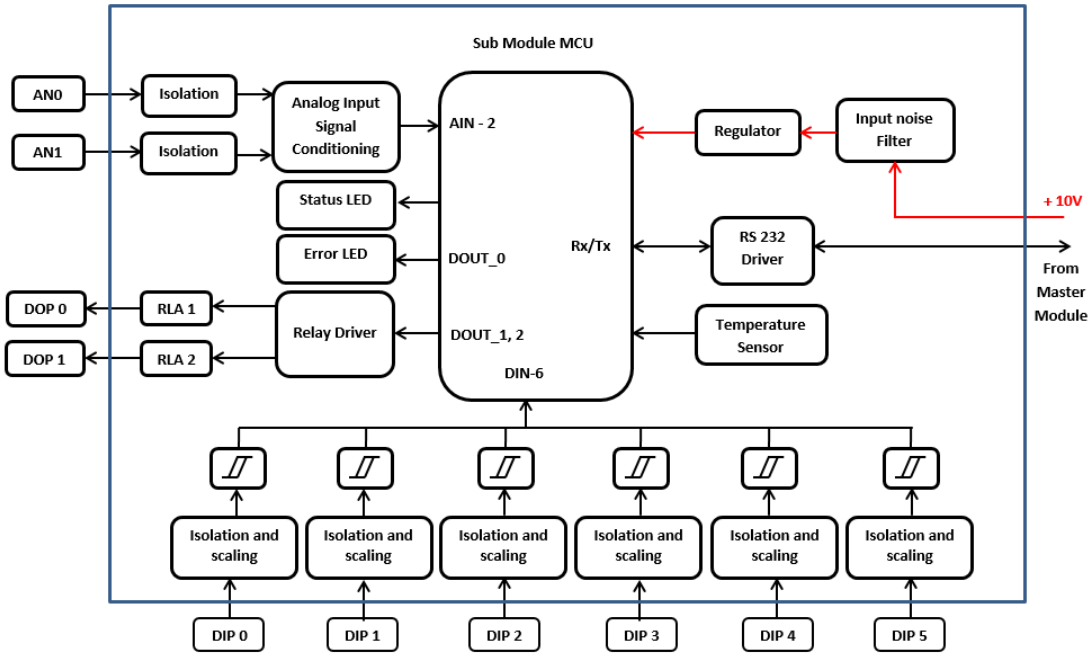


Figure 5.7: Block view of sub module design

All the individual sub modules were designed to equip with own micro controller (MCU) for data acquisition and processing. The local MCU (PIC18F2520) is converted the acquired data in according to the format of 10 bytes string (refer annex 06) and will sent them upon the collection request coming from the main nodule through the serial interface.

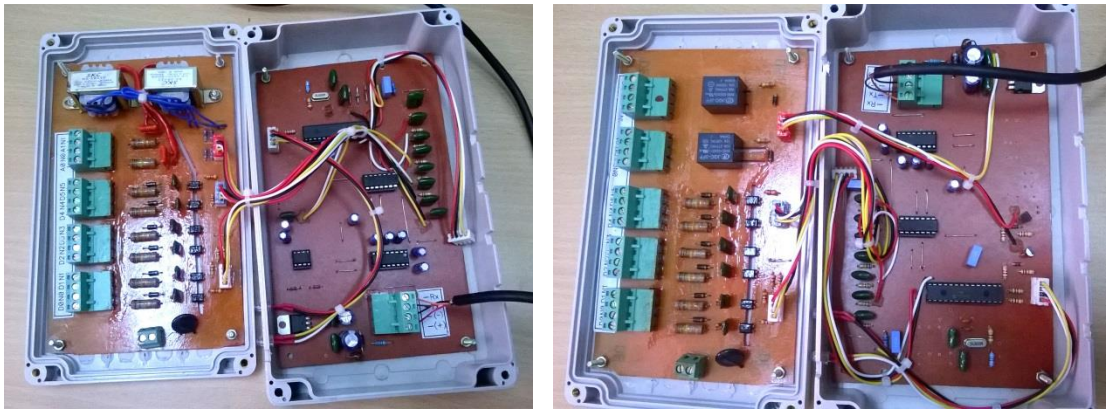


Figure 5.8: Prototype sub module hardware

One digital output of the sub module had been designed to drive a red colour LED indicator (DOUT_0), so it can be controlled by the monitoring system server. This LED can be used to indicate parameter error in the measured system. For an example, when the monitored location (ex: master power distribution panel) has multiple sub modules connected and when some fault is detected, the monitoring system server will turn ON this red LED indicator of the sub module, which connects with the most closest guess of the detected fault. This makes easier to the technical crew to isolate fault location.

Device configuration descriptor is hard coded (embedded) to the firmware, which describes the available interfaces in the sub module. For an example, device configuration descriptor for a sub module, which is acquiring power from the master module, with 02 analog inputs (A0 and A1), 06 digital inputs (D0 to D5), and 03 digital inputs with temperature sensor (similar to the figure 5.7) will be [0x01][0x01][0][0][0][0] [0x70][0x3F][0x81][0x40] according to the format given in Annex 06. This string of configuration will be collected by the master module during initial negotiation process and transmit them to the monitoring system server for module parameter identification. The system server uses this information to generate mapping between monitoring parameter table with a particular sub module device.

Analog inputs to the device are conditioned with a 230V to 6V step down isolation transformer as a voltage transformer, a voltage divider, AC waveform shift circuit [24] and anti-aliasing filter before the analog to digital converter module input. The

A/D module produces around 2000 samples per second per channel and calculates true root means squire (RMS) value as at 8 bit output.

All digital inputs have input range of 170V AC to 250V AC (tested value) with an opto-isolator and resister divider before connecting to the MCU. Two relay switches were used to produce two digital outputs from the sub module. DS18B20 temperature sensor is used for temperature measurements.

Complete logic flow of firmware routines and the circuit diagram are given in Annex 09. The printed circuit board layouts and device firmware (written in MikroC® version 6.6.0) of the sub module are presented in Appendix 03. (Included in CD)

5.3 Monitoring System Application Software Development

The monitoring system server application is the key element of the system which performs data collection, processing and providing human machine interface. The application requires server with several dependencies to run. Figure 5.9 illustrates the designed comprehensive block view of the main software system with all main functional blocks. However, the practical implementation had been not covered with all functionalities due to the limitation of the time.

Hypertext Preprocessor (PHP) language [25] version 5.6 used for the development of majority of the software system due to its flexibility for rapid development, open source technology and easiness. However, JAVA technology [26] is also used for some sections of the development where the sections of PHP cannot handle. The system require graphical user interface to access the system through the *http* web browser. However, the development of comprehensive user interface is a time consuming task and it is considered as out of scope for this implementation. So it had been decided to use *html* templates from Admin LTE [27] which is open source *html* theme template for development of control and monitoring dashboards. It can be easily integrated with PHP as well and much suitable for development of this implementation. MYSQL [28] version 5.7 was selected as the data base engine due to high performance and stable open source technology for data management. Several other dependencies were used for specific tasks and the more information is given in the next sub sections.

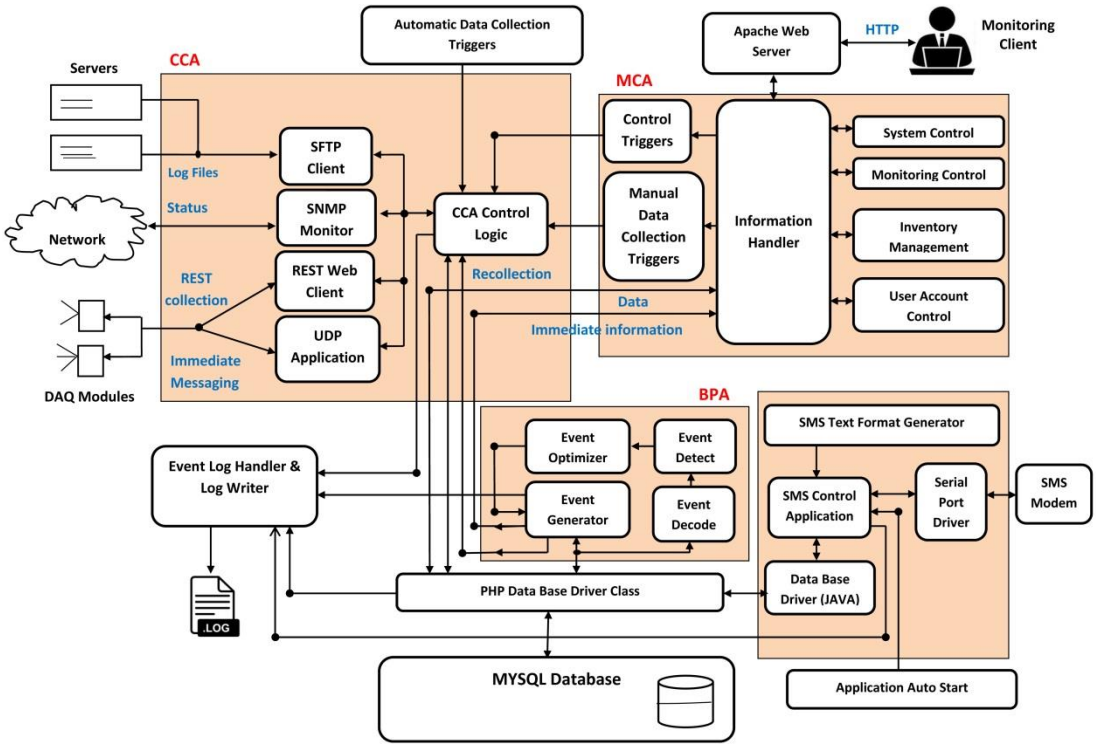


Figure 5.9: Monitoring system application

As discussed in the Section 04 in this document, the monitoring system consists of main three operational modules called Communication Control Application (CCA), Batch Processing Application (BPA) and Monitoring Control Application (MCA) with several other sub modules. The method of modular design had been used for the entire development process. Key factors of the practical implementation of each module are available in the next sub section.

5.3.1 Implementation of communications control application

The system requires connecting with the monitored systems and devices to acquire data. It is performed by the communication control application.

The data collection process is programmed to initialize through as a scheduled task (cron job scheduler) from the Linux operating system in the monitoring server. The collection script is executed once per every 04 minutes which guarantees the previous data collection is fully completed before initiating new cycle. Also the

collection process can be initialized manually through the web interface or, triggered by the batch processing application, whenever the data recollection of a certain part is required.

CCA has main 04 interfaces for data collection. Secure file transfer protocol (SFTP) module is used to connect with monitored servers through secure shell (SSH) protocol to collect log files. However, this method requires additional configuration on the monitored server(s) and its firewall(s) to allow SSH access on target log files from the monitoring server. The implementation uses pthreads [29], a multithreading approach for PHP and SFTP client of the PHP.

Simple Network management Protocol (SNMP) client collects the information about monitored network devices. The monitoring space should be pre-defined with IP address of each component and its location in the network. Automatic network discovery features are not implemented and considered as out of scope due to limitation of the time. This module also uses multi-threading approach.

Representational State Transfer (REST) web client module is responsible of collecting data from data acquisition sensor network. It sends HTTP GET request over the network to the particular sensor and collects sensor response XML message, through multi-threaded application interface.

User Datagram Protocol (UDP) is used for the asynchronous communication with data acquisition sensor modules to reduce latency for immediate messages. When a DAQ module requires transmitting its state change in the monitored environment, it will send UDP request to the monitoring server including its module identification number. Then the monitoring server performs special data collection cycle from the module and processes accordingly. Annex 10 illustrates the steps of synchronous and asynchronous data collection procedures performed by this design.

All collected data is saved to the database for batch processing. More details of the batch processing are presented in the next sub section of this document.

All errors generated in the run time will be written to a log file by the CCA. This will be helpful to identify system operation and hence improvements can be suggested by reviewing log entries. CCA control logic handles all requests and responses of all outside communication. Generalized form of operation routines of the CCA module is given in Annex 11 of this document.

5.3.2 Implementation of batch processing application

This section of the application performs all necessary processing for event detection, alarm processing and optimization, management of data storage, data recollection and status report generation. Batch processor is running after ending of each data collection cycle performed by the communication control application.

Event decoder is designed to check validity of collected data and then extract them to a simplified discrete form. Data collection procedures are not unique, so the decoding is performed with several methodologies. Table 5.2 lists all methodologies used by the event decoder.

Table 5.2: Method of decoding raw data

No	Form of data	Pre requests	Method of decode
1	XML data	XML string must be valid	Performs XML parser to decode parameters
2	Log collector data	File must be having alpha numeric data	Extract line by line and pass to the event detector for key word searching.
3	SNMP data	SNMP object must be not empty	Using PHP SNMP functions
4	UDP requests	UDP request must have special termination character "#" at the end.	UDP request format is pre-defined, extract module identification number.

Event detection and alarm optimization is performed according to the methods described in Section 4.3.1, 4.3.2 and 4.3.3 in this document. Event optimization requires carefully designed system architecture data, which has to be input during the

application setup. Defining system architecture is performed by including neighbor identification data to a particular entity (object). Figure 5.10 illustrates such entity-relationship model used to hold the system architecture data in the database. Note that these method uses parent ID as zero (0) for top level devices. This information is required by the alarms optimizer to navigate to a root cause of the detected fault.

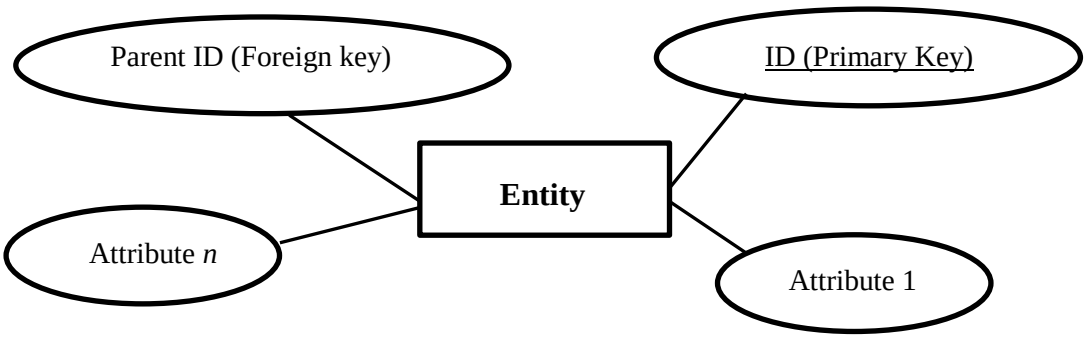


Figure 5.10: Defining parent ID with an entity

The event generator had been programmed to save all alarms and status to the database, generated by the event optimizer. Also it processes data recollection triggers whenever requested by the optimizer. It is possible to send immediate alarm information to the monitoring control application by the method of server send events (SSE) [30], but it was not implemented under this scope of work. Writing event logs are performed according to the information generated by the optimizer for debug purposes.

5.3.3 Implementation of monitoring control application

This is the human machine interface of the total system, which generates HTML format web documents to access through the monitoring client workstation. Apache server version 2.2 is used as the web server to test the application. As stated before, open source HTML template called Admin LTE had been used as the base of user interface. However, the admin LTE template uses several online dependencies during the page loading and those resources must be saved to the local web server and targets must be pointing to the local web server before using this template in the

local area connection environment.

Simple Model-View-Controller (MVC) approach [31] with PHP language had been selected for the implementation due to its simplicity and flexibility of expansion in the future. Table 5.3 contents all major sub modules required for complete user interface. However, some sections were not completely implemented under this implementation due to limited time frame.

Table 5.3: Human machine interface – function list

No	Main function	Sub functions
1	User management	o User login/logout o Session management o Add/edit/delete user o User privileges definition
2	Inventory management	o Add/edit/delete devices o Add/edit/delete sections o Define architecture
3	Monitoring control	o Define monitoring space o Mark scheduled maintenance o Error threshold level definition o Notification control o Monitoring space status display
4	System control	o Define default system parameters and values o Set global parameters

Figure 5.11 shows a screenshot of the monitoring application.

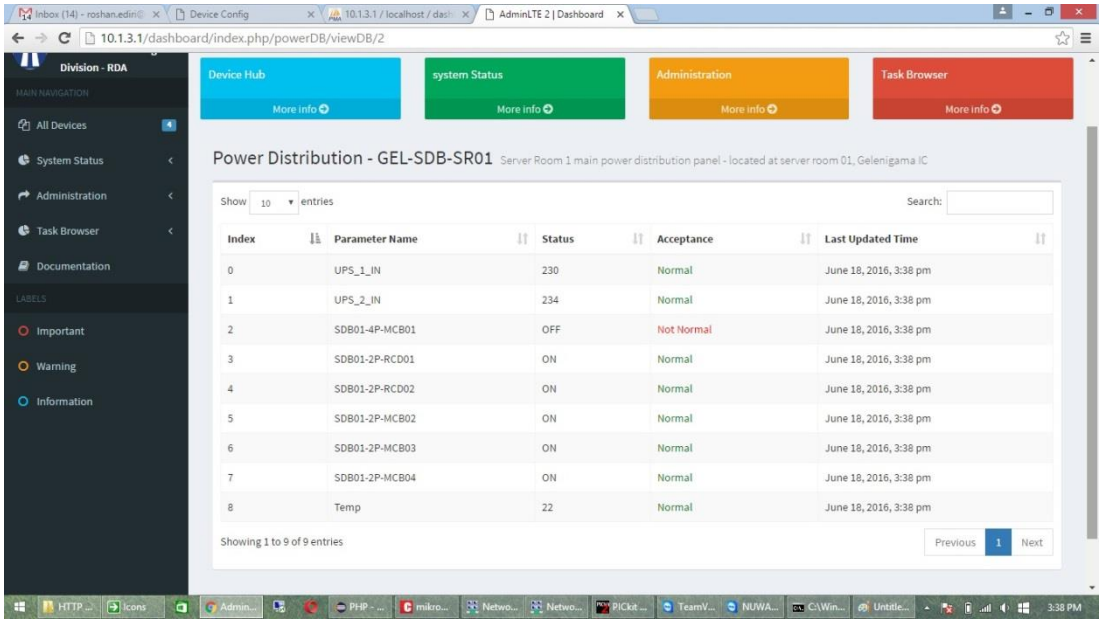


Figure 5.11: Screenshot of the monitoring application

5.3.4 Database system implementation

Database system is one of the most important factors for system operation and hence careful design is essential. The design procedures include both database structure design and data access and management software design process.

Designing of an entity-relationship model [32][33] for the database structure had been performed initially and then the normalization [34] procedures had been followed. Normalized ER model of the database is available in the Annex 12 of this document for reference.

Accessing the database is performed through only the database class. The data access class is basically a database connection manager. Additionally, it is designed to have protection against SQL injection attacks by filtering the incoming SQL queries for vulnerable character combinations.

All PHP code base and database structure (in form of SQL statements) with initial test data set is available in Appendix 04 (included in CD) for reference.

5.3.5 Additional security measures added to the system design

The monitoring system software must be secured enough against attacks and hence following list of security enhancements had been designed with the system software.

(a) Server file system is made write protected to outsiders.
(b) Database access class was enhanced to prevent SQL injection attacks.
(c) URL address directives are filtered with vulnerable characters before accessing the server web root directory.
(d) All software modules can be only accessed through the session manager application, direct access is not allowed.
(e) User login session data is maintained in both server memory and the database.
(f) User login information is saved in the database.
(g) Log file writing is enabled in the system.

5.4 Short Message Service (SMS) Server Application Development

Sending event notifications through a short message service (SMS) is identified as an efficient method of communication. It allows easy and fast information passing method among all relative groups of system maintenance crew to get aware of the status of the system.

Sending SMS had been designed to perform by using serial SMS modem based on SIM 900A module. The module has a RS 232 interface with an inbuilt slot for SIM card insertion. Separate 12V power supply is required to provide for the operation. Figure 5.12 illustrates the SMS device. The module requires a positive going pulse to start the operation after the power ON reset, and it is achieved by using a simple micro controller circuit. According to the datasheet, the module's RS 232 interface is having auto baud rate identification feature and it can be operated from 9600bps to 115200bps speeds.

Send SMS software is developed by using JAVA technology. It has major two (02) parts; one is to read from the database while other part is communicating with the SMS modem through the serial interface. Serial communication application is having

two threads for serial transmitting and serial receiving to ensure nothing is lost from the communication. The serial port driver is based on RxTx JAVA [35] implementation.

The application path is included to the startup script of the monitoring server's Linux operating system; hence the application starts automatically with the server boot up sequence.

Annex 13 consists of the flow chart which illustrates the operation of the software application and Appendix 05 consists of the JAVA application code base with RxTx driver (64 bit version), data sheet of the SIM 900 module and micro controller firmware (written in MikroC® version 6.6) for reference.

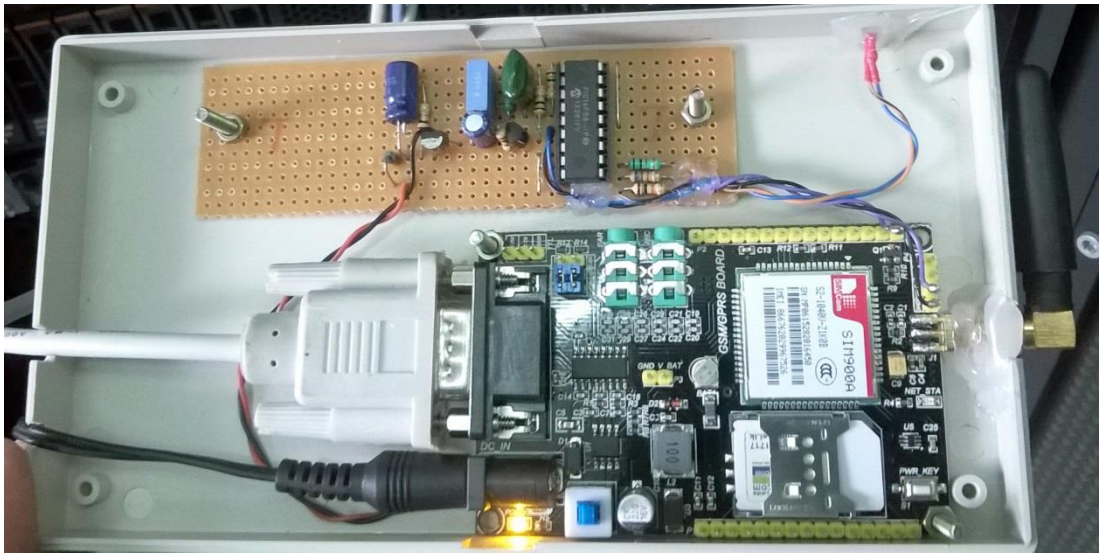


Figure 5.12: SMS modem unit

Several tests were performed to verify the operation of the developed system and such tests and corresponding results are given in the next chapter.

CHAPTER 06

SYSTEM TESTING AND RESULTS

The monitoring system implementation had been fairly complex and time consuming task, which requires implementations on hundreds of individual hardware and software components. Since the system design had been carried out by following modular design approach, all individual software components were tested individually in parallel with the implementation. However, the system requires an integration function test to study about the performance improvements in system operation achieved in monitored (targeted) systems. This chapter describes the information regarding such testing methodologies and their results.

6.1 System Function Tests and Results

The system had been initially tested for its basic functions in a test setup as shown in the figure 6.1, which is very similar to the architecture available in production environment. The monitoring application was deployed on a virtual machine, having 8GB RAM with dual core 2.8GHZ processor configuration with Ubuntu 14.04 LTS 64 bit server version in an IBM server.

Two layer 02 network switches were configured and connected with a core router to test network status monitoring and the power supply is arranged through three miniature circuit breakers. TP1 is connected with an analog input-0 measurement of the sub DAQ module and TP2, TP3 and TP3 were connected with D0, D1 and D2 digital inputs. System architecture was defined such that, TP 1 to TP3 in descending order, TP1 as the level zero (0) node. Two layer 2 switch IP addresses added to the monitoring system with parameter association of TEMP_01 to TP2 and TEMP_02 to TP3. Core router, monitoring workstation and monitoring server have separate power supply system, which is independent to the test setup. DAQ module is powered from TP4, but the main module has its own back up supply for operation during power failures. The system response time had been measured by using a hand stop watch.

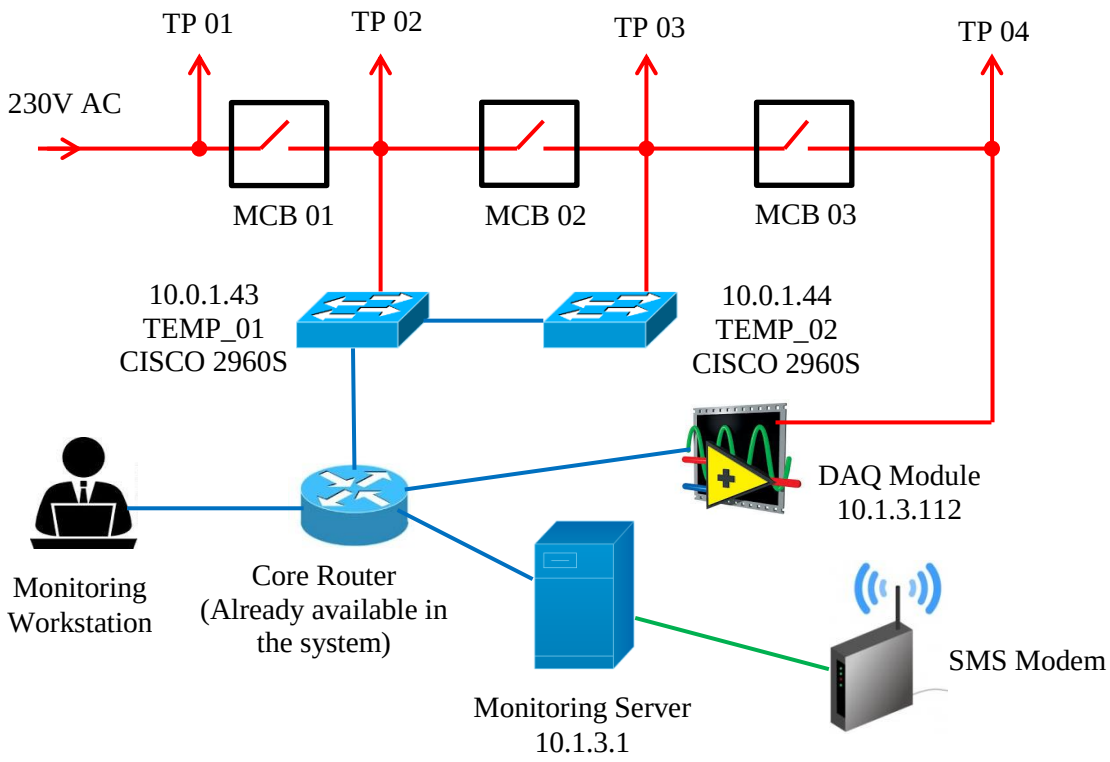


Figure 6.1: System test setup-01

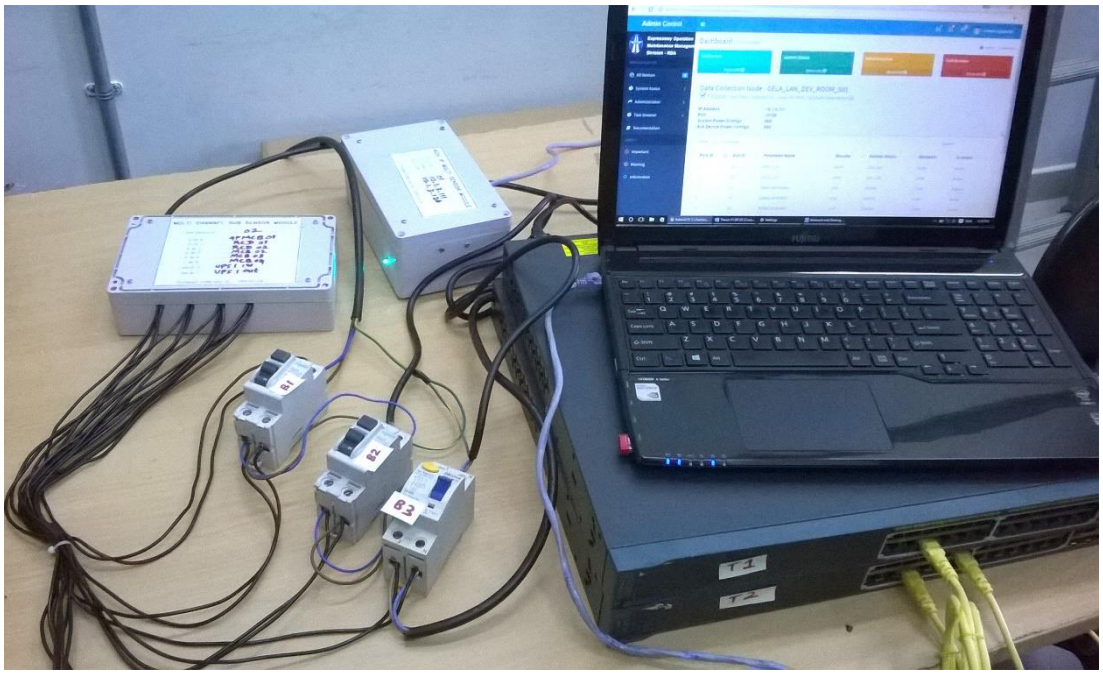


Figure 6.2: Practical test setup-01

Figure 6.2 shows practical equipment setup used for data collection, according to the test setup diagram shown in figure 6.1.

Table 6.1 illustrates tests and corresponding results generated from the monitoring system.

Table 6.1: Basic tests and results – Setup-01

Test No	**Test method**	**Physical observation**	**Response from the monitoring system**	**Acceptance**
1	Turn OFF MCB 03	Power to the DAQ cuts, DAQ operates from its battery power	Monitoring system indicates MCB 03 as OFF after 2.4 seconds and the SMS was delivered after 10.8 seconds.	Correctly detected.
2	Disconnect the data link between TEMP_01 and TEMP_02	Connected network interfaces inactivated in both switches.	Monitoring system indicates TEMP_02 as offline after 2.6 seconds and the SMS was delivered after 15.7 seconds.	Correctly detected.
3	Turn OFF MCB 02	Power to the DAQ cuts, DAQ operates from its battery power and TEMP_02 switch turns OFF.	Monitoring system indicates MCB 02 in OFF position after 2.8 seconds and SMS was delivered after 9.4 seconds.	Correctly detected about the power failure other than network outage as the root cause.

Continued

Table 6.1: Basic tests and results – Setup-01

Test No	Test method	Physical observation	Response from the monitoring system	Acceptance
4	Disconnect the data link between router and TEMP_01	Connected network interfaces inactivated in both switches.	Monitoring system indicates TEMP_01 as offline and TEMP_02 status unknown after 3.1 seconds and the SMS was delivered after 16.5 seconds.	Correctly detected the last active node.
5	Turn OFF MCB 01	Power to the DAQ cuts, DAQ operates from its battery power and TEMP_01 and TEMP_02 switches turns OFF.	Monitoring system indicates MCB 01 in OFF position after 2.1 seconds and SMS was delivered after 11.5 seconds.	Correctly detected about the power failure point other than network outage as the root cause.
6	Remove data link with DAQ module	Connected network interfaces inactivated in both switches.	Monitoring system indicates link error with the DAQ module.	Monitoring system had previous knowledge that the DAQ has an internal battery and its last charge status. So it is converged to a link error.

It had been observed that the monitoring system produced correct output in all cases. Now the test system had changed such that the DAQ node network connection was

connected to TEMP_02 and DAQ's parent node was defined as TEMP_2 as shown in figure 6.3. This setup is equal to the operation of the DAQ module in end nodes, where the network connection is likely to be non-available during failures.

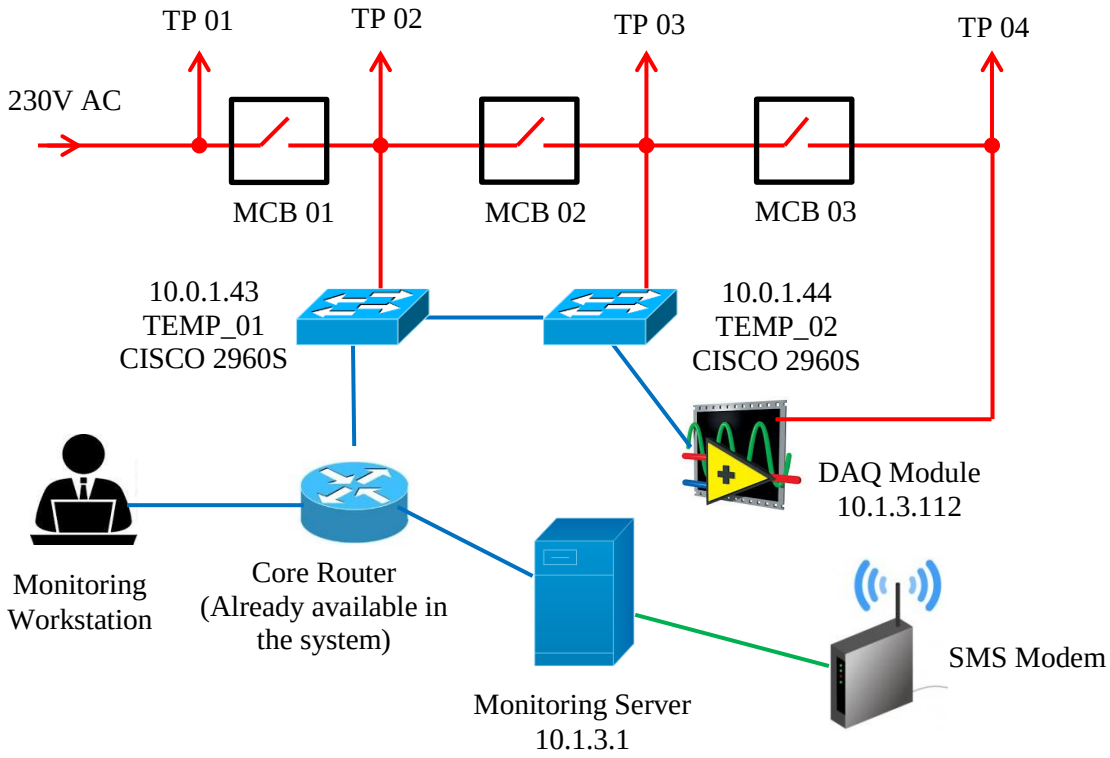


Figure 6.3: System test setup 02

Table 6.2 illustrates the tests and results with new setup. It is observed that the monitoring system is not accurate enough in this scenario. However, the fault optimizer gave logically correct answers with the information available. DAQ setup can be upgraded to have associative parameter measurements with secondary DAQ module, where the secondary DAQ unit is connected through separate network path in order to achieve most accurate results. The system has provision to add associative DAQ parameters (similar to the scenario of adding TEMP_01 to TP 02) with its design, but was not implemented under this scope.

Table 6.2: Basic tests and results – Setup-02

Test No	Test method	Physical observation	Response from the monitoring system	Acceptance
1	Disconnect the data link between TEMP_01 and TEMP_02	Connected network interfaces inactivated in both switches.	Monitoring system indicates TEMP_02 is Offline after 2.1 seconds and the SMS was delivered after 12.4 seconds.	Best possible guess is produced. Monitoring system knows the parent node of DAQ (TEMP_02) is offline and hence not attempt to connect with DAQ module.
2	Turn OFF MCB 02	Power to the DAQ cuts, DAQ operates from its battery power and TEMP_02 switch turns OFF.	Monitoring system indicates TEMP_02 as offline after 2.6 seconds and the SMS was delivered after 9.7 seconds.	Best possible guess is produced. Monitoring system knows the parent node of DAQ (TEMP_02) is offline and hence not attempt to connect with DAQ module.
3	Turn OFF MCB 01	Power to the DAQ cuts, DAQ operates from its battery power and TEMP_01 and TEMP_02 switches turns OFF.	Monitoring system indicates TEMP_01 as offline after 2.2 seconds and the SMS was delivered after 15.4 seconds.	Best possible guess is produced. Monitoring system knows the parent node of DAQ (TEMP_02, hence TEMP_01) is offline and hence not attempt to connect with DAQ module.

6.2 Integration Setup and Tests

The monitoring system is finally connected with the actual production system environment at Gelenigama interchange for final verification.

6.2.1 Integration with power distribution panel in main server room

The system is connected with part of the power distribution system inside the server room 01 at Gelenigama interchange as shown in the Annex 13 and tests were performed as shown in the table 6.3. The video evidence for this test is given in Appendix 06. (Included in CD)

Table 6.3: Monitoring tests on the server room power distribution panel

Test No	Test method	Physical observation	Response from the monitoring system	Acceptance
1	Turn OFF 4P-MCB-01	System power disconnected. Monitoring sub sensor red error indicator lights up	Monitoring system indicates 4P-MCB 01 in OFF state after 2.8 seconds (Red LED ON) and the SMS was delivered after 9.5 seconds.	Accepted
2	Turn ON 4P-MCB-01	Operation restored.	Monitoring system indicates 4P-MCB 01 in ON state after 2.6 seconds (Red LED OFF) and the SMS was delivered after 8.7 seconds.	Accepted

Continued

Table 6.3: Monitoring tests on the server room power distribution panel

Test No	Test method	Physical observation	Response from the monitoring system	Acceptance
3	Turn OFF 2P-RCD-01	System power disconnected. Monitoring sub sensor red error indicator lights up	Monitoring system indicates 2P-RCD-01in OFF state after 3.1 seconds (Red LED ON) and the SMS was delivered after 15.2 seconds.	Accepted
4	Turn ON 2P-RCD-01	Operation restored.	Monitoring system indicates 4P-MCB 01 in ON state after 2.8 seconds (Red LED OFF) and the SMS was delivered after 13.5 seconds.	Accepted

Also note that the Red LED in this sub module is connected to digital output 0 in the device and it had been accessed through the monitoring system to indicate measured parameter deviation (fault). So the time between event happening (t=0) and red LED On or OFF event can be considered as approximate event detection time of the monitoring system. But the actual time of detection must be lower than this value.
SMS delivery time had deviations due to the congestions in the service provider's network, which is beyond control of the client.

6.2.2 Link state monitoring through SNMP - Tests and results

Consider the actual system setup of 1969 emergency call system including systems monitor as shown in Annex 15. The monitoring system is configured to monitor network device status of GELA_ACC_STA 01, GELA_AGG 01, GELA_AGG 02, GELA_ACC 04, GELA_ACC 06 and SIP_TR 01 links and devices and following tests were performed as shown in the table 6.4. Since this system is in production state, the spaces to perform tests were limited and all of below tests were performed during off peak time with prior approval. Also note that the subsequent tests were performed after restoring previous change to normal state.

Table 6.4: Monitoring tests on the call system network

Test No	**Test method**	**Physical observation**	**Response from the monitoring system**	**Acceptance**
1	Administratively down Gi1/0/1 and Gi1/0/24 of GELA_ACC_STA 01	Corresponding links go inactive	Monitoring system indicates GELA_AGG 01 and GELA_AGG 02 as offline and SMS was delivered after 10.2 seconds.	Accepted
2	Administratively down Gi0/12 of GELA_AGG 01	Corresponding links go inactive	Monitoring system indicates SIP TR 01 link is inactive and SMS was delivered after 9.5 seconds.	Accepted
3	Administratively down Gi1/0/25 and Gi1/0/26 of GELA_ACC 04	Corresponding links go inactive, call agent CIPC went offline.	Monitoring system indicates GELA_ACC 04 link is inactive and SMS was delivered after 13 seconds.	Accepted

6.2.3 Tests on service status monitoring through log analysis

Consider the setup in Annex 15 again. It contents test service server, which is running windows WAMP based apache web server. The monitoring system is configured to read apache error log through SFTP to verify the functionality of the implemented system.

The apache web server error log (C:\wamp\logs\apache_error.log) has different levels of notifications including "*:error*" tag in the corresponding log entry and the monitoring system log analyzer is programmed to catch same keyword during log processing.

Private zone in the apache web server was manually created by editing its configuration and tried to access it from the HTTP, throws following error message which was written on the apache error log file.

"[Sat Jun 11 05:47:41.512121 2016] [authz_core:error] [pid 5620:tid 828] [client ::1:2945] AH01630: client denied by server configuration: C:/SVN/Software Dev/test/"

This error had been detected by the monitoring system log analyzer, approximately after 21 seconds. The same methodology can be followed for any kind of service state monitoring through automated log file analysis and the functionality was successfully verified.

6.2.4 Tests on automation scripts support

The monitoring system is designed to have a support on automation scripts. The system had been connected with the automatic transfer switch panel at Gelenigama interchange to control power generator through a simple automation script as a test.

Figure 5.4 illustrates the simple test setup. The DAQ system sub sensor D0-1 digital output relay, normally close (N/C) contacts was connected through one of the input phase of the phase fault relay, digital output DO-2 relay normally open (N/O)

contacts was connected with generator start signal and digital input DI-0 was connected with one of the output phase of the power generator. The automation script is illustrated in figure 6.5 and the practical setup is illustrated in figure 6.6.

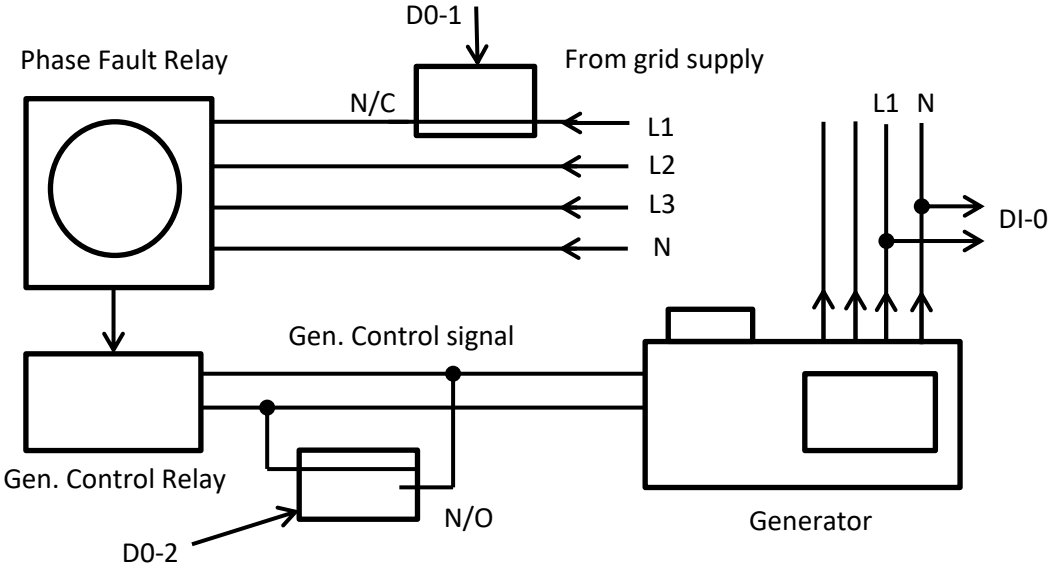


Figure 6.4: Test setup for automation script

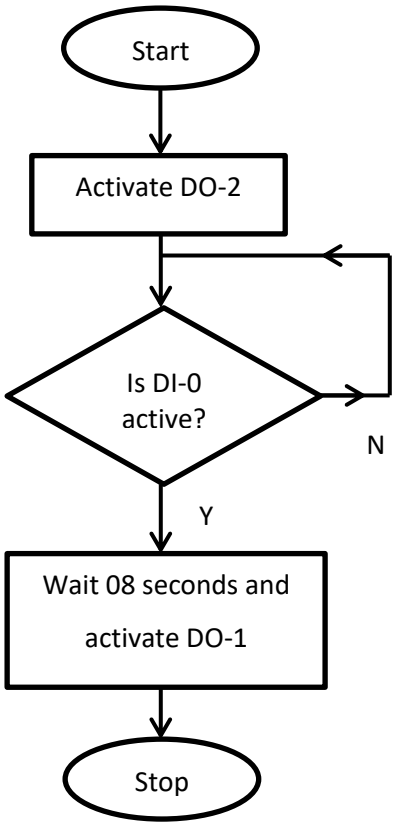


Figure 6.5: Flow of simple test automation script

This simple script will start the power generator and transfer system load to the generator after 8 seconds, even if the grid power is available.

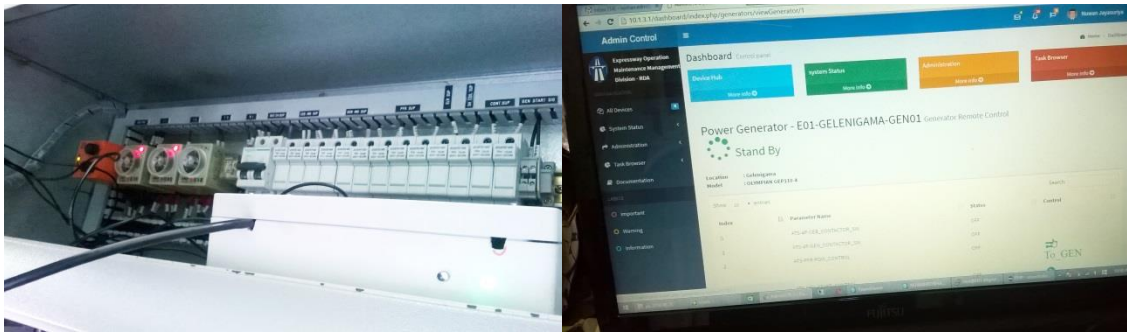


Figure 6.6: Automation test setup

The test had been successfully carried out.

6.3 Summery of Test Results

Table 6.5 shows the summery of all performed tests and their results. The average detection periods are based on the records in year 2015 maintenance log book (Annex 02), without the monitoring system.

Table 6.5: Summery of basic tests and results comparison

No	Scenario	Average detection time with monitoring system	Average detection time without monitoring system	Improvement
1	Detection of electrical problem (Ex: turn OFF some breaker)	Less than 20 seconds	20 minutes	60 times faster
2	Detection of network node outage	Less than 20 seconds	30 minutes	90 times faster
3	Detection of link outage	Less than 20 seconds	2.5 days	More than 10,000 times
4	Detection of service status	Less than 25 seconds	5.4 days	More than 18,000 times

Following indirect improvements are available with the implementation of the monitoring system.

(a) Incidents can be detected earlier and prevent secondary incidents as shown in the annex 05. It will improve the availability and reduce any possibilities of occurring secondary incidents. (Ex: crash of server disks after UPS battery power grains out)
(b) Repair will be easy for technical staff due to optimized faults alarms, pointing to the fault location.
(c) Corrective maintenance records are complete due to automated data collection.
(d) Possibility of execution of automation scripts can be used for simple level automation required in the production environment through the same monitoring system is added advantage.

There are several other advantages and disadvantages in the design which are discussed in the next chapter of this document.

CHAPTER 07

CONCLUSIONS AND RECOMMENDATIONS

Under the scope of this research, design and development of an integrated fault monitoring system for power systems and communication systems had been carried out. The design of system architecture was performed after studying the architecture of available monitoring systems and methodologies used in several commercial systems in both locally and internationally.

Log book data analysis and theoretical design approaches were used to develop the system components by using modular design approach. Finally, the implementation had been tested both in a test setup and in the real production environment. Followings are the summary of results and deliverables by the end.

7.1 Improvements Added Over the Conventional Monitoring Systems

Several improvements added and tested to the new system in addition to the conventional features of commercially available monitoring systems.

(a) The designed system can monitor both power systems and network and communication systems through a single interface.
(b) System fault alarms are optimized by implementing optimization procedures as described in Chapter 04 of this document.
(c) Integrated monitoring allows concentrating on the real fault location, by considering both power and networking link status.
(d) The system software allows connecting with heuristic knowledge base, which contents previous repair information. It will be helpful for the technical crew for the recovery process.
(e) The new system supports on ground level automation scripts.
(f) All modules use open protocols and open source software. Hence the system can be changed or developed without restrictions.

7.2 Disadvantages in this design

Following disadvantages were identified with this design.

(a) Initial system setup and configuration requires expert knowledge and time consuming process. However, if the target systems are not updating rapidly, this would not be a problem as much.
(b) Later changes in the target systems require reconfiguration of architecture and parameters in the monitoring system configuration database.
(c) Development processes of modules require skilled people.

7.3 Identified Fields of Improvements and Recommendations

Following improvements had been identified during the implementation and testing process.

(a) User interface based configurator is required to draw system architecture by using graphical tools during initial configuration process to simplify complexity.
(b) Base system implementation shall be done by JAVA technology other than PHP to improve multi-threaded execution performance.
(c) DAQ module implementation requires more powerful main processor than PIC micro controller.

In general, the main target of this research was achieved by having locally developed integrated monitoring system with enhanced features. It was tested and proved that the system can increase availability of target systems by having fast detection, support on maintenance crew through optimized alarms with heuristic records and fast information provision through a SMS. Also the system can cover both power and communication systems, makes it easier to reduce number of systems monitoring staff as well. The implementation is cost effective over the commercially available systems through an open source technology. This basic implementation and methodologies can be developed to the commercial level through the local people easily.

Annex 01: Part of Gelenigama Interchange Event Log Book Entries in Year 2015 (for reference only)

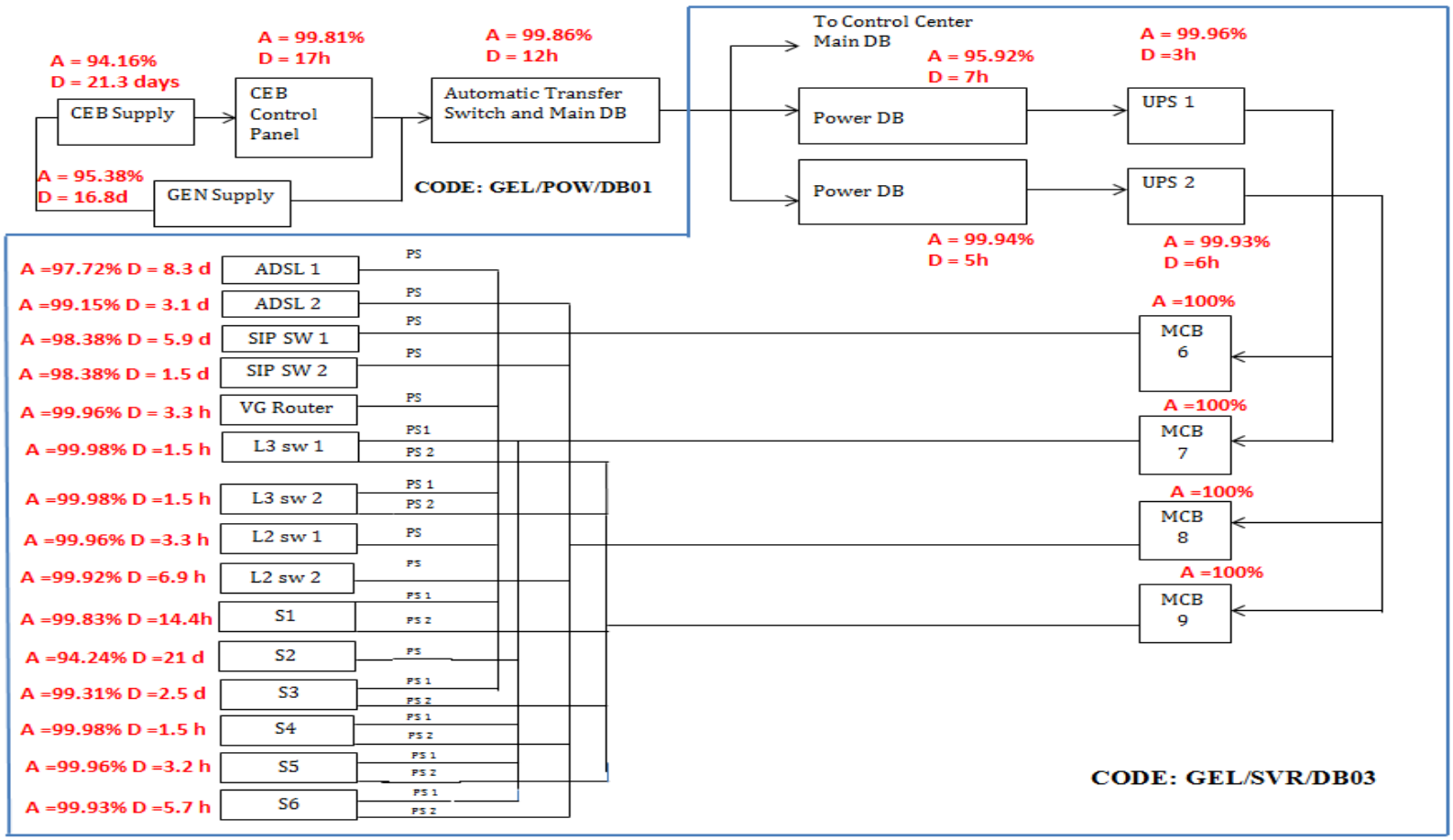
No	Case/Reported Issue	System/ Location	Date/time occurred	Reported date/time	Problems & Action	Recovered date/time	Comments
1	DB-03 RCD 02 trip OFF, Connection with NMS is lost on monitor screen	GEL-SVR/Main	2015/01/14 13:12	2015/01/14 14:04	S01-UPS02 turns OFF due to low battery, Turn it ON	2015/01/14 14:20	Due to lightening
2	DB-03 MCB 03 OFF, System failover notification at the operator screen	GEL-SVR/ITS	2015/01/14 22:40	2015/01/14 22:55	S01-UPS03 turns OFF due to low battery, Turn it ON	2015/01/14 23:15	Unknown Reason
3	DB-04 RCD 03 OFF, Power loss to the control center	GEL-CC/ALL	2015/01/19 18:05	2015/01/19 18:10	S02-UPS03 output turns OFF from output RCD	2015/01/19 18:25	Due to lightening
4	DB-01 ATS RLA01 not forwarding power, All power to Gelenigama IC lost	GEL-POW/ALL	2015/02/25 11:20	2015/02/25 11:25	Emergency switch pressed at the master panel by electrical section labor, S01-UPS01 at 11:30, S01-UPS02 at 11:30, S02-UPS01 at 11.30, S02-UPS02 at 11:30, S02-UPS04 at 11:50 turns OFF	2015/02/25 12:05	Detection of problem was difficult. No indication about emergency switch press event.

Continued

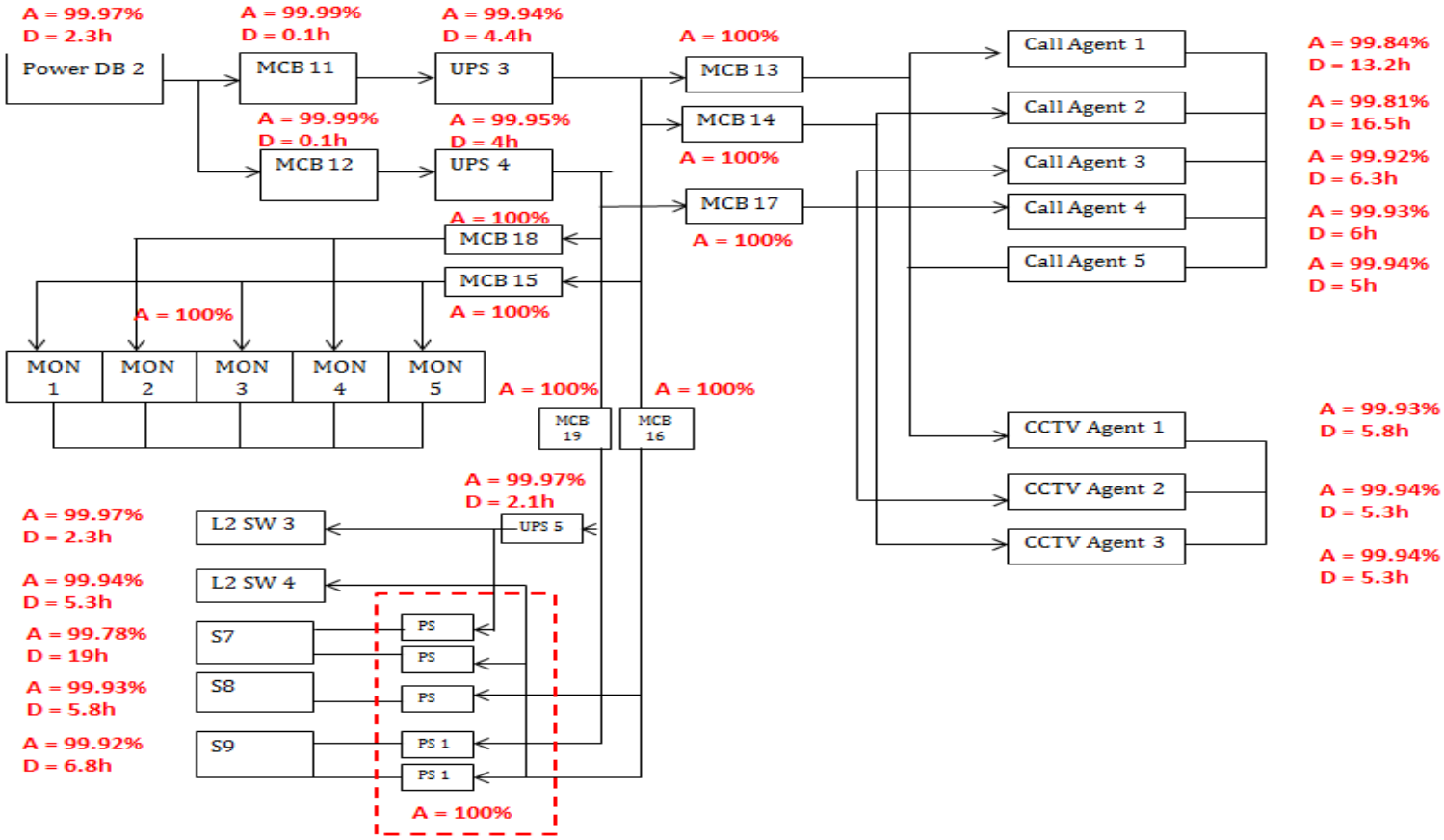
Annex 01: Part of Gelenigama Interchange Event Log Book Entries in Year 2015 (for reference only)

No	Case/Reported Issue	System/ Location	Date/time occurred	Reported date/time	Problems & Action	Recovered date/time	Comments
5	CUCCX01 Database Failover to UCCX02, Adding new configuration failed on UCCX system	GEL SVR/Call	2015/03/14 23:40	2015/03/19 10:05	Restarting UCCX01, Restarting UCCX02, Retrigger for database forced synchronization through web interface	2015/03/19 23:15	This is an error in UCCX 8.5 version. Upgrade recommended.
6	DB-03 RCD 01 trip OFF, Internet connection problem	GEL-SVR/Main	2015/03/22 17:30	2015/03/22 18:10	S01-UPS01 and all connected devices turns OFF due to low battery, Turn it ON	2015/03/22 18:20	Due to lightening
7	DB-03 RCD 02 trip OFF, Connection with NMS is lost on monitor screen	GEL-SVR/Main	2015/03/24 16:40	2015/03/24 18:15	S01-UPS02 and connected devices turns OFF due to low battery, Turn it ON	2015/03/24 18:35	Due to lightening
8	SLT SIP TR 01 (Kottawa route) not working, Called number transformations send as only “0000” to outside.	GEL-SVR & KOT/SVR	2015/03/24 17:00	2015/03/25 07:30	SIP Trunk terminator indicates connection lost. Informed SLT Enterprise Customer service. Complain Number :XXXXXX	2015/03/25 14:20	Due to a problem in SLT M-SAN

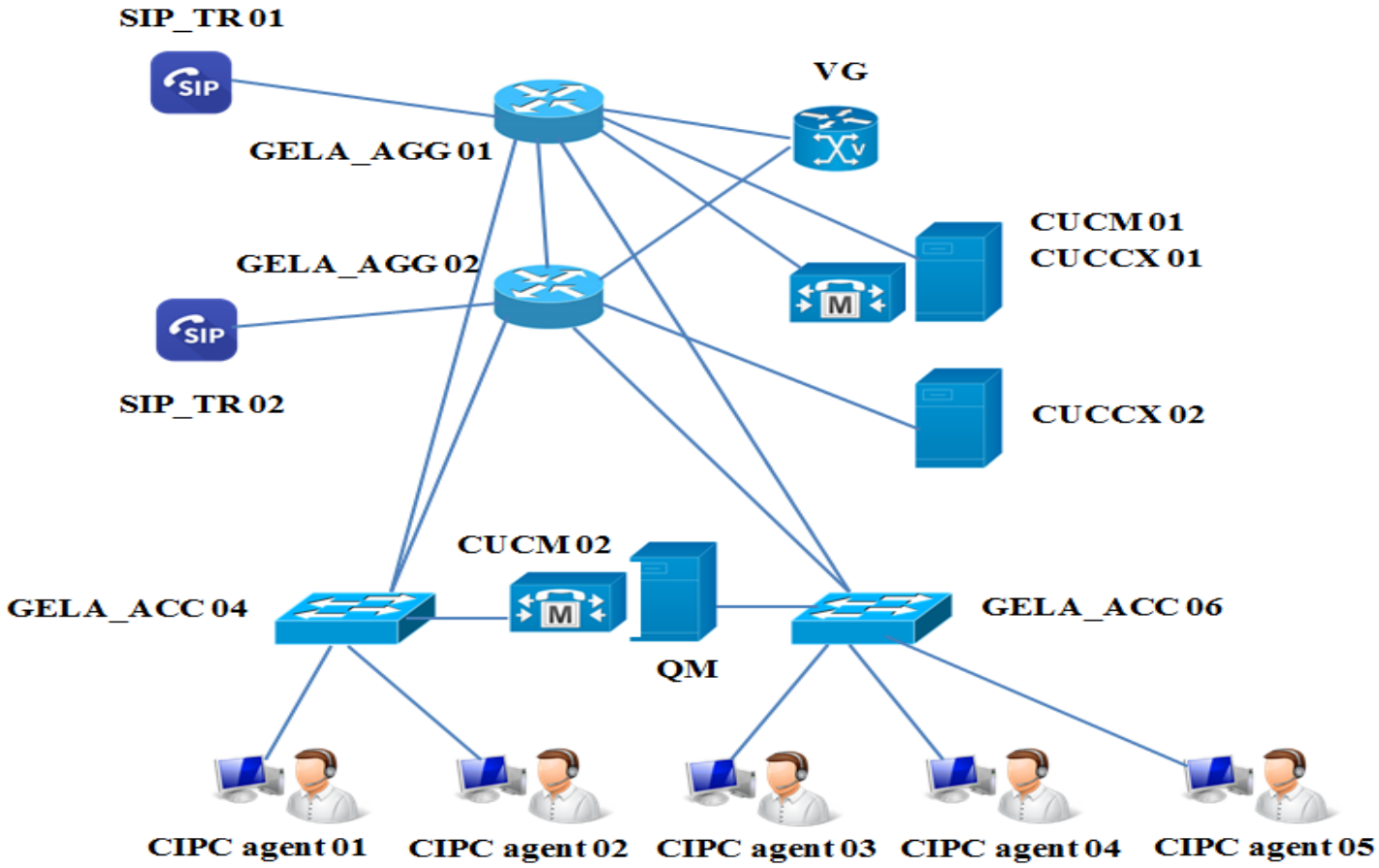
Annex 02 – Power Supply Arrangement (Server Room 01)



Annex 02 – Power Supply Arrangement (Control Center 01)



Annex 03 – Network Diagram of 1969 Emergency Call System



Annex 04: Recompiled Part of Gelenigama Interchange Event Log Book - Assuming with Monitoring System Available (for reference only)

No	Case/Reported Issue	System/ Location	Comment	Expected down time
1	DB-03 RCD 02 trip OFF, Connection with NMS is lost on monitor screen	GEL-SVR/Main	This incident can prevent if the RCD 02 is restored before the UPS batteries become fully dead. Monitoring system can help to detect RCD action immediately.	0 min
2	DB-03 MCB 03 OFF, System failover notification at the operator screen	GEL-SVR/ITS	This incident can prevent if the MCB 03 is restored before the UPS batteries become fully dead. Monitoring system can help to detect MCB action immediately.	0 min
3	DB-04 RCD 03 OFF, Power loss to the control center	GEL-CC/ALL	This incident can prevent if the RCD 03 is restored before the UPS batteries become fully dead. Monitoring system can help to detect RCD action immediately.	0 min
4	DB-01 ATS RLA01 not forwarding power, All power to Gelenigama IC lost	GEL-POW/ALL	This incident can prevent if the action of emergency shutdown switch identified before the UPS batteries become fully dead. Monitoring system can help to detect such actions immediately.	0 min

Continued

Annex 04: Recompiled Part of Gelenigama Interchange Event Log Book - Assuming with Monitoring System Available (for reference only)

No	Case/Reported Issue	System/ Location	Comment	Expected down time
5	CUCCX01 Database Failover to UCCX02, Adding new configuration failed on UCCX system	GEL SVR/Call	Monitoring system can detect database failover to subscriber immediately. But, the recovery procedure takes time.	20 min
6	DB-03 RCD 01 trip OFF, Internet connection problem	GEL-SVR/Main	This incident can prevent if the RCD 01 is restored before the UPS batteries become fully dead. Monitoring system can help to detect RCD action immediately.	0 min
7	DB-03 RCD 02 trip OFF, Connection with NMS is lost on monitor screen	GEL-SVR/Main	This incident can prevent if the RCD 02 is restored before the UPS batteries become fully dead. Monitoring system can help to detect RCD action immediately.	0 min
8	SLT SIP TR 01 (Kottawa route) not working, Called number transformations send as only “0000” to outside.	GEL-SVR & KOT/SVR	Monitoring system can detect such outages immediately. But a recovery procedure takes time.	345 min

Annex 05 - REST service interface, response XML data formats, http status responses and information on device configuration interfaces

(a) REST service interface

No	Call URL	Description
1	http://<DEV_IP>:10130/gd	Get XML data containing measured parameter information (format A)
2	http://< DEV_IP >:10130/dc	Get XML data containing device configuration (format B)
3	http://<DEV_IP>:10130/control /<SSID>/ /<p0>/<p1>/<p2>#	Set digital outputs of sub sensor denoted by SSID (Ex: /control/1/0/1/0# sets sub module 1 digital output 0 -> 0, 1 ->1, 2 -> 0)
4	http://<DEV_IP>:10130/re	Reset error counts in the main module
5	http://<DEV_IP>:10130/rst	Reboot main module
6	http://<DEV_IP>:10130/config	Returns module configuration page in form of HTML

(b) XML data formats

- **Data request (/gd) returns following string of XML data**

Content-Type: application/xml; charset=utf-8 <?xml version="1.0" <SensorData><info><sensor xmlns:xlink="http://SENSOR_IP: PORT/gd"><Sensor_ID> SENSOR_ID</Sensor_ID><Response_ID> SEQUENTIAL_ID</Response_ID><Status> STATUS_CODE</Status></info><data><S 0> ANO~ AN1~ AN2~ AN3~ AN4~ AN5~ REF 01 [D_LSB]~ REF 02~ [Even_Parity-1~8]~

E_MSB~ E_LSB</S 0><S 1> DATA STREAM 01 </S 1><S 2> DATA STREAM 02 </S 2><S 3> DATA STREAM 03 </S 3><S 4> DATA STREAM 04 </S 4><S 5> DATA STREAM 05
</S5></data></SensorData>

Please note that the XML string contents all data for 06 sub sensor modules (denotes as <S 0> to <S 5>) even if the corresponding sub sensor is not connected with the interface. Following is the brief description of data fields available in the XML.

- SENSOR_IP – IP address of the main module
- PORT - Connection port, usually it is 10130
- SENSOR_ID - Module ID of the module
- SEQUENTIAL_ID - Response ID of the corresponding GET request
- STATUS_CODE - Status code generates as follows;

B7	B6	B5	B4	B3	B2	B1	B0

MSB LSB

"<b6 - error in system voltage> <b5 - battery power status while discharging> <b4, b3 - Battery charging status> <b2 - outside Power Supply Error> <bit1 - Battery Status> < Bit 0 – Power Mode >"

Bit 7 : No use, always 0
Bit 6 : 0 - No error detected; 1 - Error in main system voltage
Bit 5 : 0 - Battery having power; 1 - battery discharge critically
Bit 4:b3 : 00 - charger disabled due to battery error; 01 - charging enabled; 10-tricle charge enabled; 11 - battery full
Bit 2 : 0 - outside power OK; 1 - outside power disabled due to detection of short circuit or over load event
Bit 1: 0 - Battery Not Detected; 1 - Online
Bit 0 : 0 – Battery; 1:AC

- AN 0 to AN 5 - Analog input value readings

- REF 01 - This data byte contents following information;

"[PS_Mode_1+D_OP_3+D_MSB_4]"

B7	B6	B5	B4	B3	B2	B1	B0
MSB							LSB

Bit 7 : Power supply mode of the sub module; 0 - Main module powered; 1 – self

Bit 6 to 4 : Status of digital outputs (B6 – DOP 2, B5 – DOP 1, B4 – DOP 0)

Bit 3 to 0 : MSB of the digital inputs (B3 – DIP 11 to B0 – DIP 8)

- D_LSB - LSB of the digital input (B7 – DIP 7 to B0 – DIP 0)
- REF 2 - This data byte contents following information;

"[T_Parity_1+T_7]"

B7	B6	B5	B4	B3	B2	B1	B0
MSB							LSB

Bit 7 : Parity bit for bits 0 to 6

Bit 6 to 0 : Value of temperature sensor reading of sub module

- EVEN_PARITY 1-8 - Even parity bits of data byte 1 to 8 of the string (LSB first order)
- E_MSB - Error count MSB (for link performance analysis)
- E_LSB - Error count LSB (for link performance analysis)

This stream of data will repeat for all other sub modules denoted as "DATA STREAM X" in the same way.

- **Configuration request (/dc) returns following string of XML data**

Content-Type: application/xml; charset=utf-8 <?xml version="1.0" <SensorData><info><sensor xmlns:xlink="http:// SENSOR_IP: PORT/dc"><Sensor_ID> SENSOR_ID </Sensor_ID><Response_ID> SEQUENTIAL_ID </Response_ID></info><config><main> STATUS_CODE #<S 0> REF 01~ REF 02~ REF 03</S 0>#<S 1> DATA STREAM 01 </S 1>#<S 2> DATA STREAM 02 </S 2>#<S 3> DATA STREAM 03 </S 3>#<S 4> DATA STREAM 04 </S 4>#<S 5> DATA STREAM 05 </S 5></main></config></SensorData>

- SENSOR_IP – IP address of the main module
- PORT - Connection port, usually it is 10130
- SENSOR_ID - Module ID of the module
- SEQUENTIAL_ID - Response ID of the corresponding GET request
- STATUS_CODE - Status code generates as in the previous case
- REF 01 - This data byte contents following information;

"[PW_Mode_1+AN_Status]"

B7	B6	B5	B4	B3	B2	B1	B0
MSB							LSB

Bit 7 : Not implemented

Bit 6 : Sub module power mode; 0 - Main module powered; 1 – self

Bit 5 to 0 : Presence of analog input channels; (Bit X -> 1, channel X presents or otherwise 0)

- REF 02 - This data byte contents following information;

"[Temp_Sensor_Presence_1+DOP_3 +D_IP_MSB_4]"

B7	B6	B5	B4	B3	B2	B1	B0
MSB							LSB

Bit 7 : Presence of Temperature sensor; 0 – not presents or 1 otherwise

Bit 6 to 4 : Presenc

e of digital outputs; 0 not presents or 1 otherwise and B6 -> DO 2, B5 -> DO 1 and B4 -> DO 0

Bit 3 to 0 : Presence of digital inputs MSB; 0 not presents or 1 otherwise and B3 -> DI 11, B2 -> DI 10 so on

- REF 03 - This data byte contents following information;

"[D_IP_LSB]"

Bit 7 to 0 : Presence of digital inputs LSB; 0 not presents or 1 otherwise and B7 -> DI 7, B6 -> DI 6 so on

This stream of data will repeat for all other sub modules denoted as "DATA STREAM X" in the same way.

(c) HTTP status response codes

- **The main module sends following format of XML data for success cases**

"Content-Type: text/xml; charset=utf-8 <?xml version="1.0" <SensorMsg><Success><Code>HTTP_CODE</Code><Message>STATUS_MSG</Message></Success></SensorMsg>"

- **The main module sends following format of XML data for unsuccessful cases**

  “Content-Type: text/xml; charset=utf-8 <?xml version="1.0" <SensorMsg><Error><Code> HTTP_CODE </Code><Message> STATUS_MSG </Message></Error></SensorMsg>”

  Note : All status codes are based on HTTP/1.1 default status codes and messages.

- The device configuration page of the master module can be accessed through the network by visiting “http://<device_IP>:10130/config” from the access allowed (permitted) IP address. This page is useful to configure the master module according to the site requirements.

Annex 06 – Serial Communication Protocol (Between Master Module and Sub Modules)

The communication between main module and sub modules is based on RS 232, 8 data bits and no parity bit format.

(a) From main module to sub module

Main module sends following command byte to the sub module. Note that the data byte is divided to upper nibble and lower nibble where the upper nibble is the inverse of lower nibble. This arrangement makes data byte has self-error checking mechanism at the sub module end.

B7	B6	B5	B4	B3	B2	B1	B0
MSB							LSB

Bit 7 to 4 : Inverse of bits 3 to 0

Bit 3 : Type of request; 0 for configuration request or 1 for data request

Bit 2 to 0 : Set values of digital outputs (Those bits will set on available digital outputs directly as B2 -> DO 2, B1 -> DO 1 and B0 -> DO 0)

(b) From sub module to main module

The response message from the sub module is contents of 10 byte string as shown below.

- **For data requests;**

[AN0][AN1][AN2][AN3][AN4][AN5][PS_Mode_1+D_OP_3+D_MSB_4][D_LSB][T_Parity_1+T_7][Even_Parity-1~8]

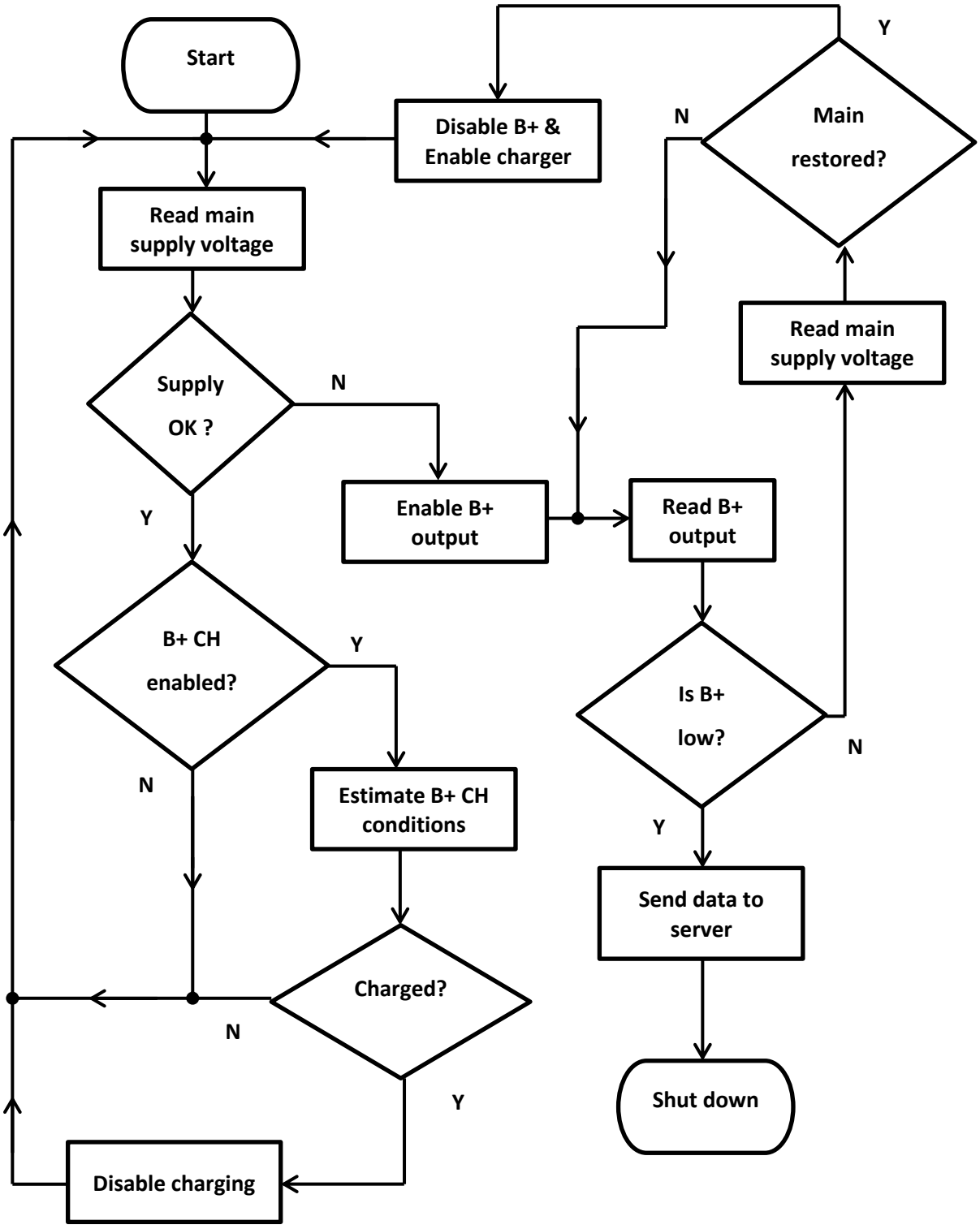
AN0 to AN5 will be filled with the readings from corresponding analog input channel. [PS_Mode_1+D_OP_3+D_MSB_4][D_LSB] two data bytes have usual meanings as described earlier in Annex 05. [T_Parity_1+T_7] byte contents 7 data bits for temperature measurement and 7th bit for even parity of 0 to 6 data bits in the same byte. The last byte contains even parity bits for the bytes 0 to 8 of the string.

- **For configuration requests;**

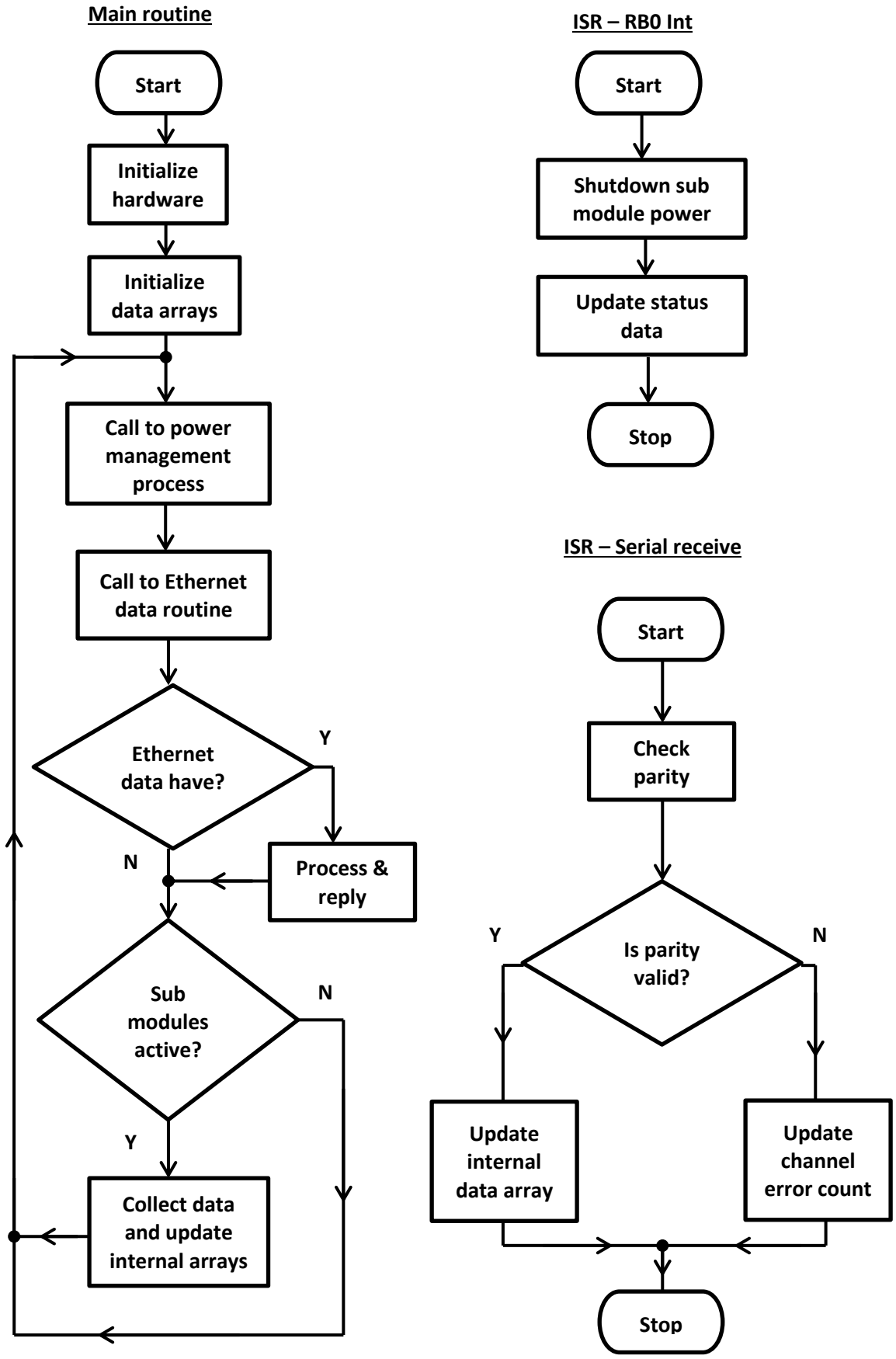
[AN0][AN1][AN2][AN3][AN4][AN5][PS_Mode_1+D_OP_3+D_MSB_4][D_LSB][T_Parity_1+000000+T_1][Even_Parity-1~8]

AN0 to AN5 will be filled with either 0 or 1 depending on the availability of corresponding analog input channel where 1 for available or 0 for otherwise. [PS_Mode_1+D_OP_3+D_MSB_4][D_LSB] two data bytes have usual meanings as described earlier in annex 05 of configuration. [T_Parity_1+000000+T_1] byte LSB bit indicates the presence of temperature sensor filled with leading 6 zeros and 7th bit for even parity. The last byte contains even parity bits for the bytes 0 to 8 of the string.

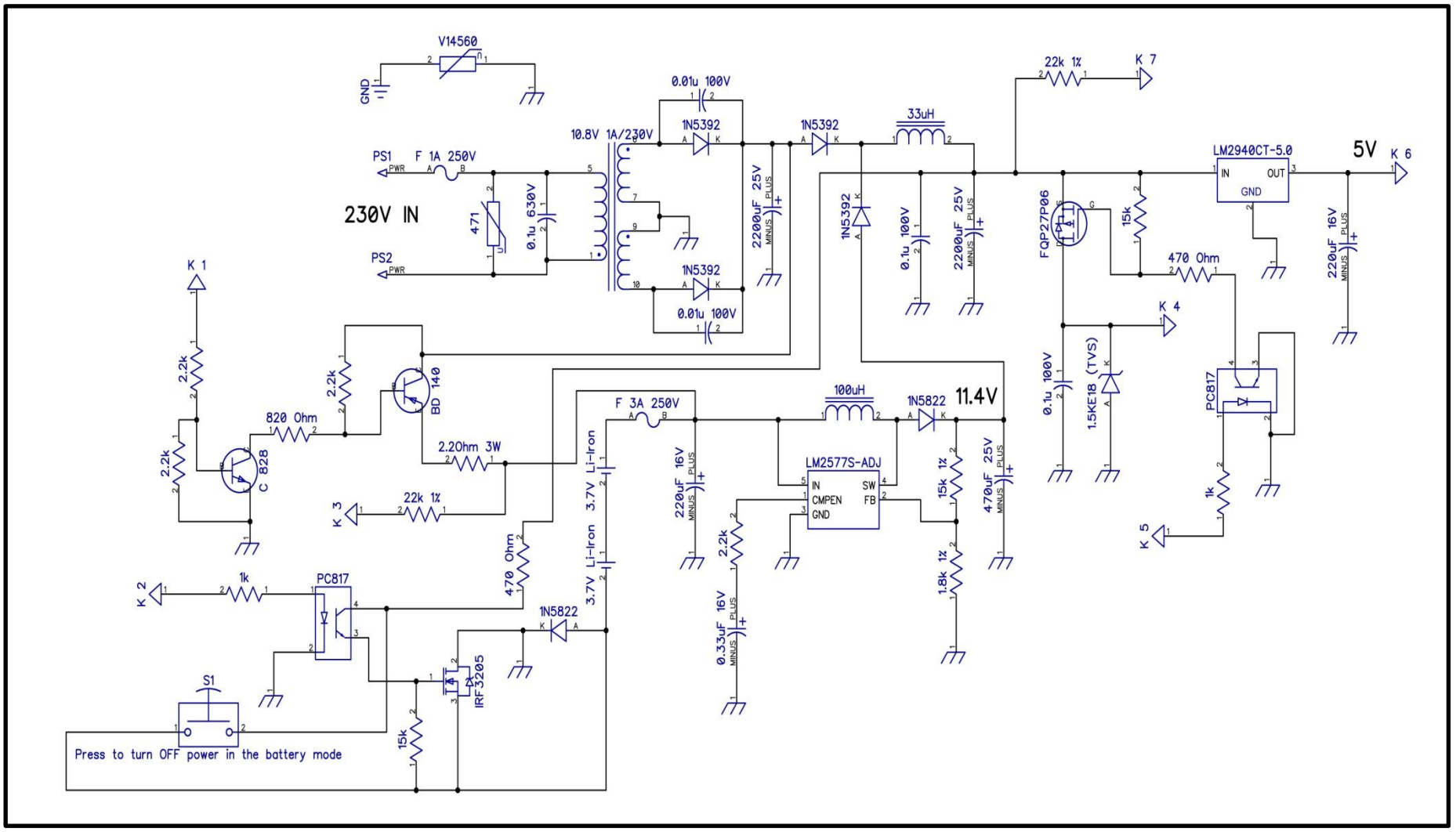
Annex 07 – Simplified Main Module Power Management Cycle



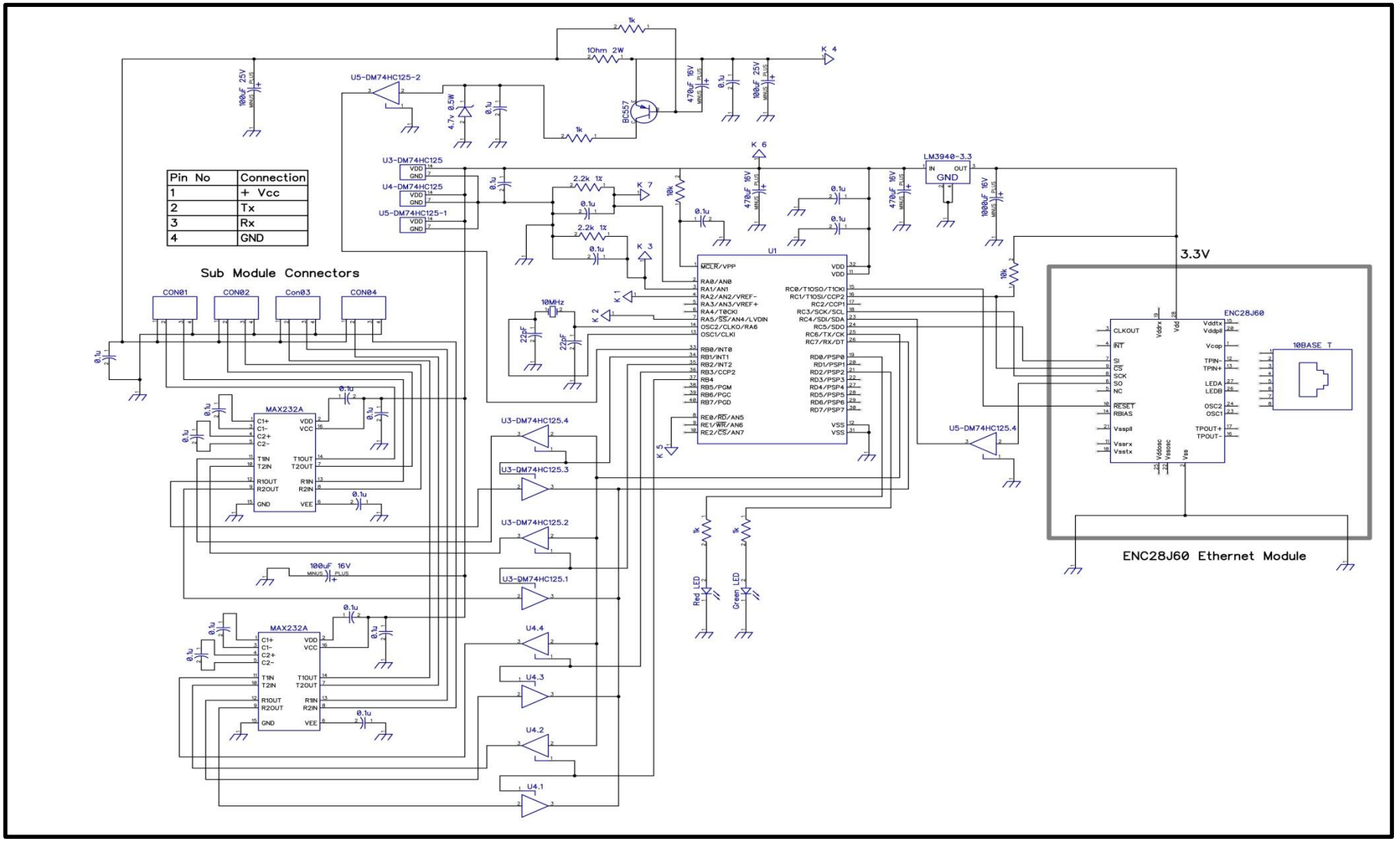
Annex 08 – (A) Simplified Main Module Firmware Cycles



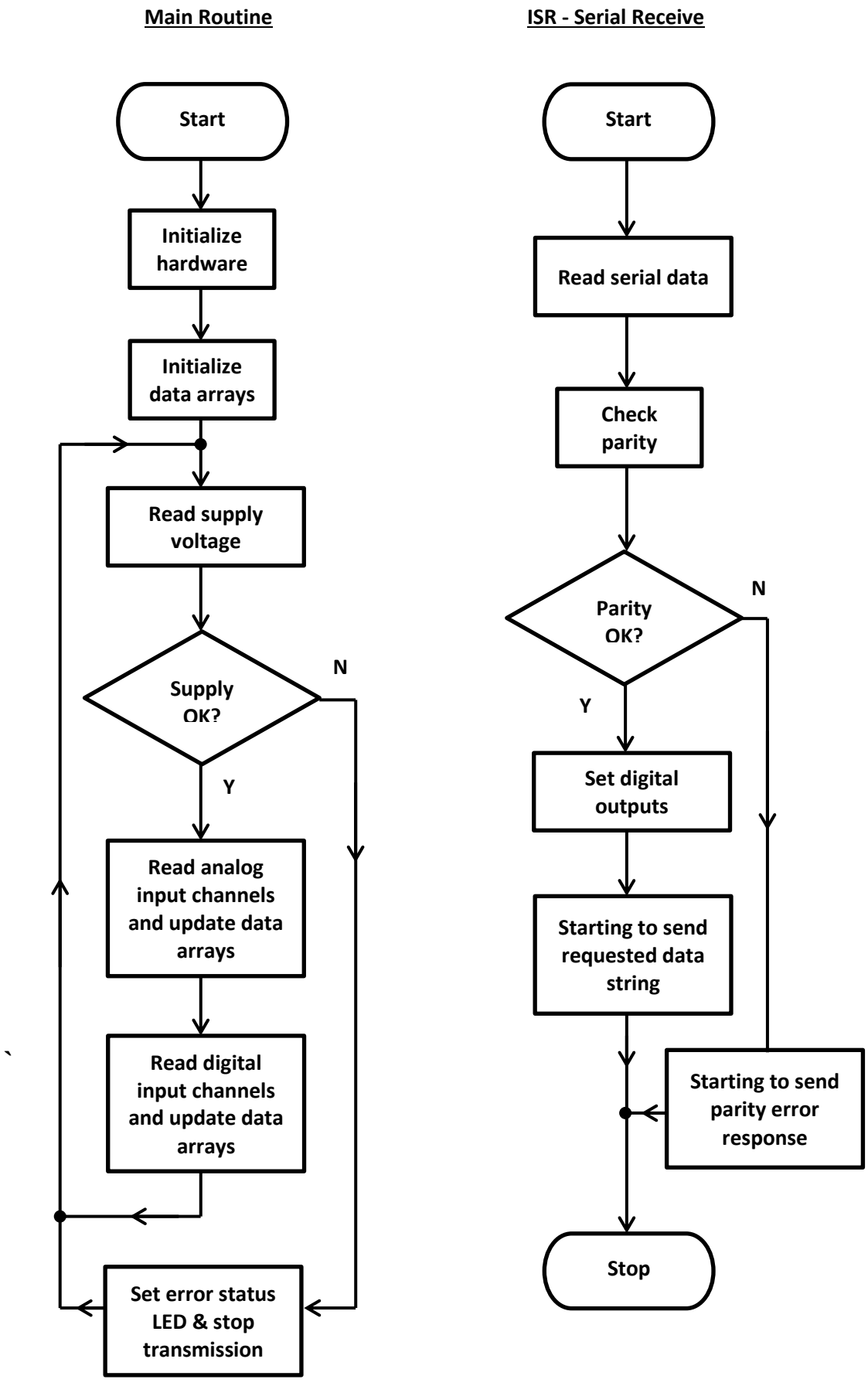
Annex 08 – (B) Circuit Diagram of the Main Module – Power



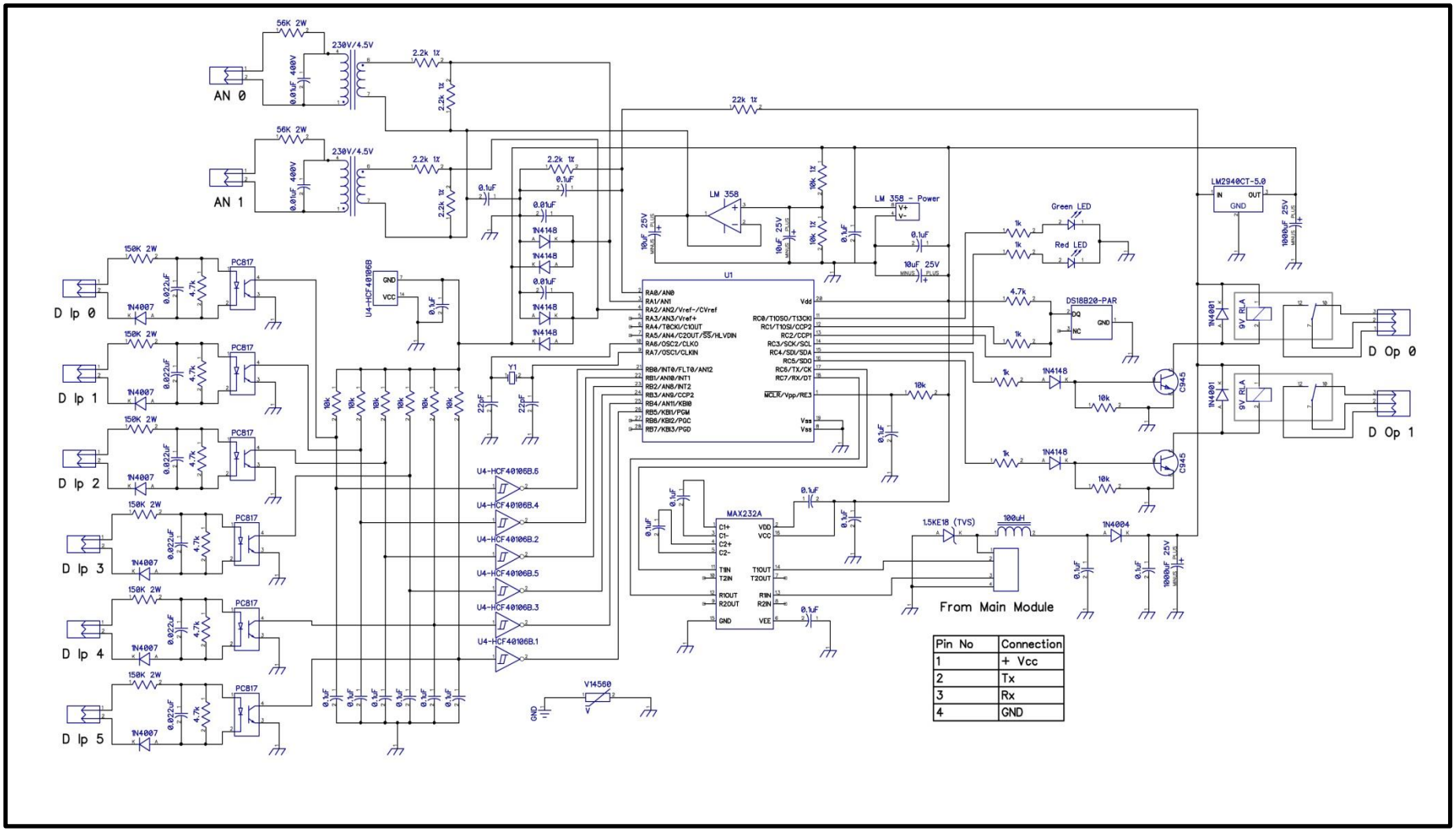
Annex 08 – (B) Circuit Diagram of the Main Module – Control and Communication



Annex 09 – (A) Simplified Sub Module Firmware Cycles



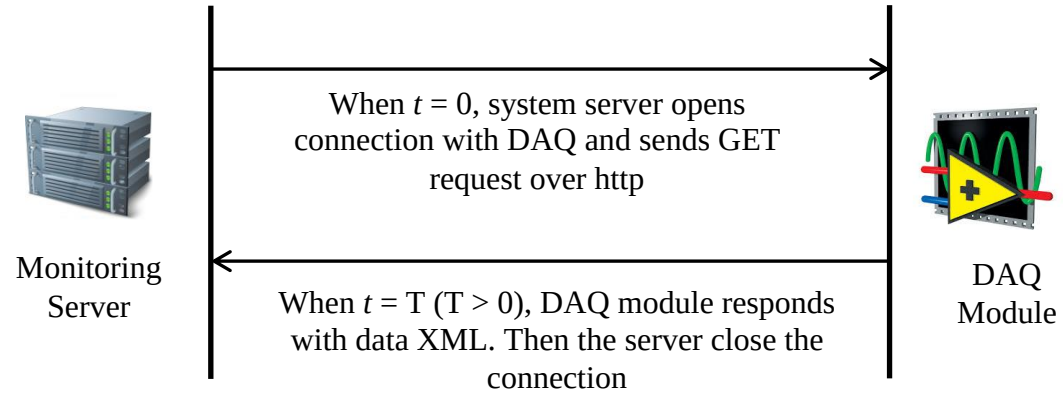
Annex 09 – (B) Circuit Diagram of Sub Module



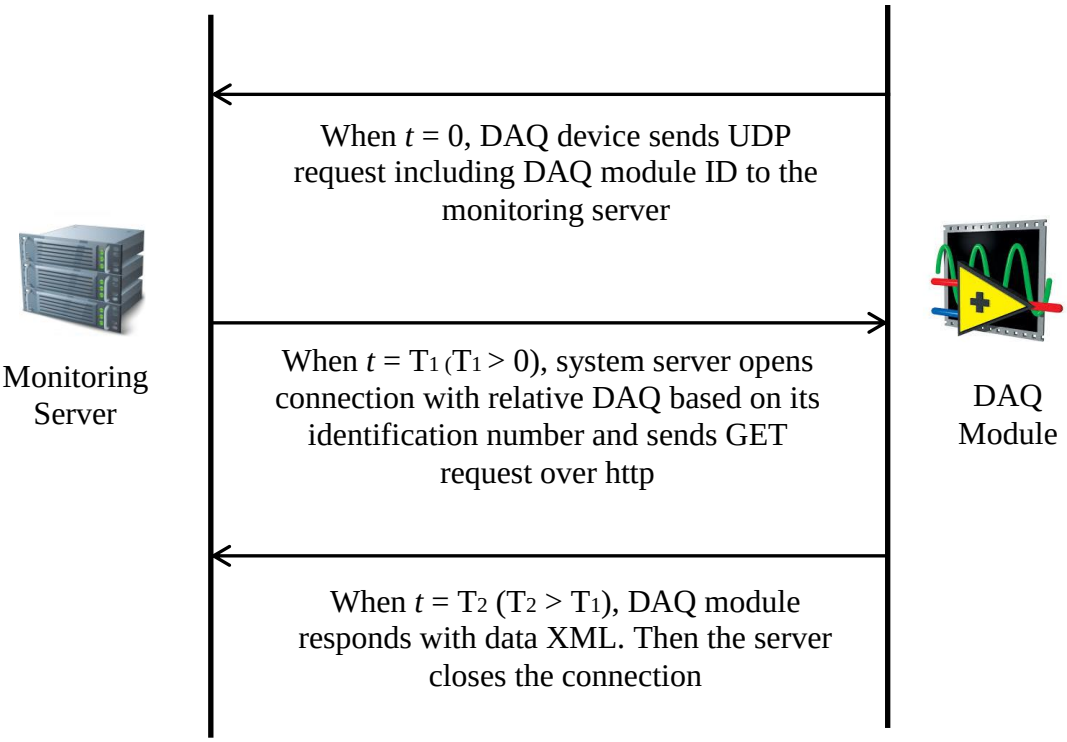
Pin No	Connection
1	+ Vcc
2	Tx
3	Rx
4	GND

Annex 10 – Synchronous and Asynchronous communication with DAQ modules

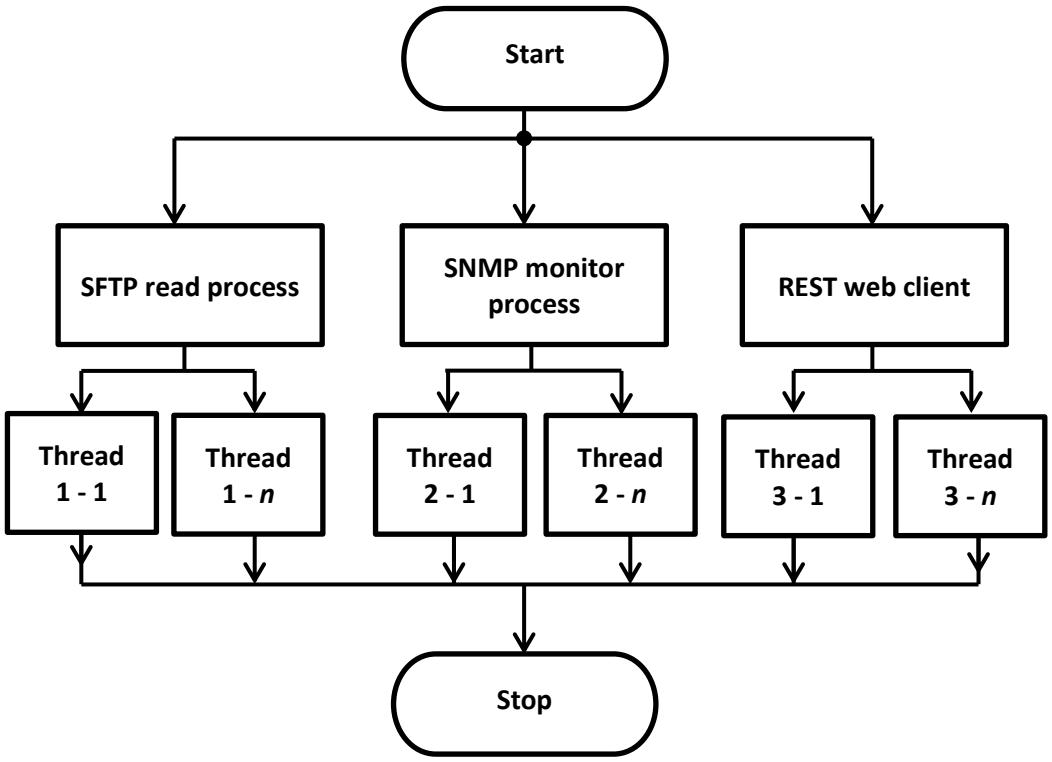
(a) Synchronous method – Initiated by general routine of data collection from the monitoring server.



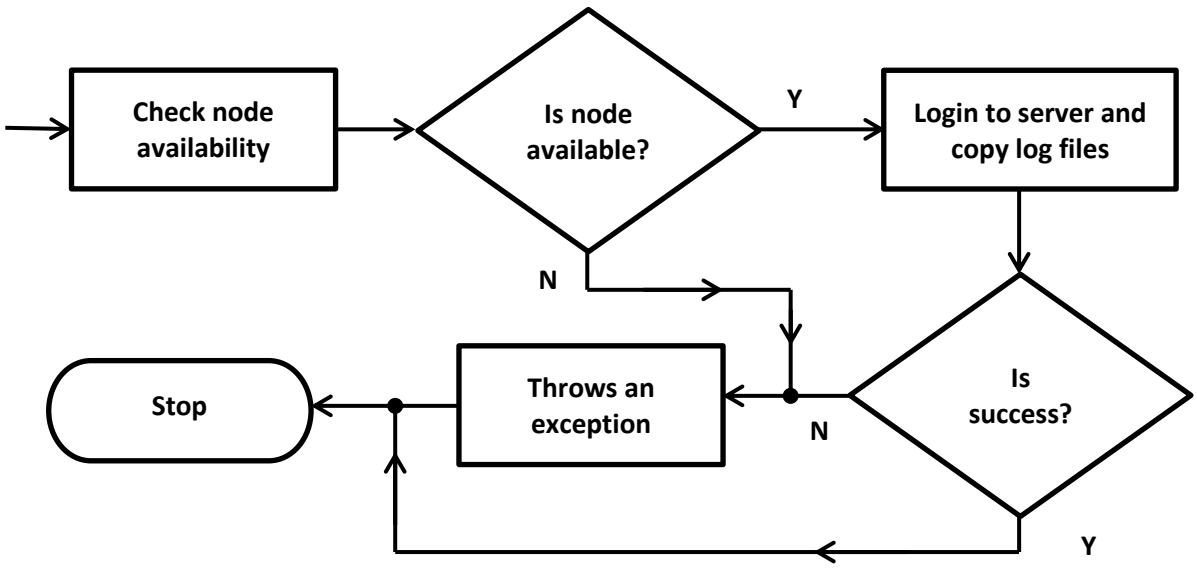
(b) Asynchronous method – Initiated by the DAQ module during abnormal parameter status of the monitored system.



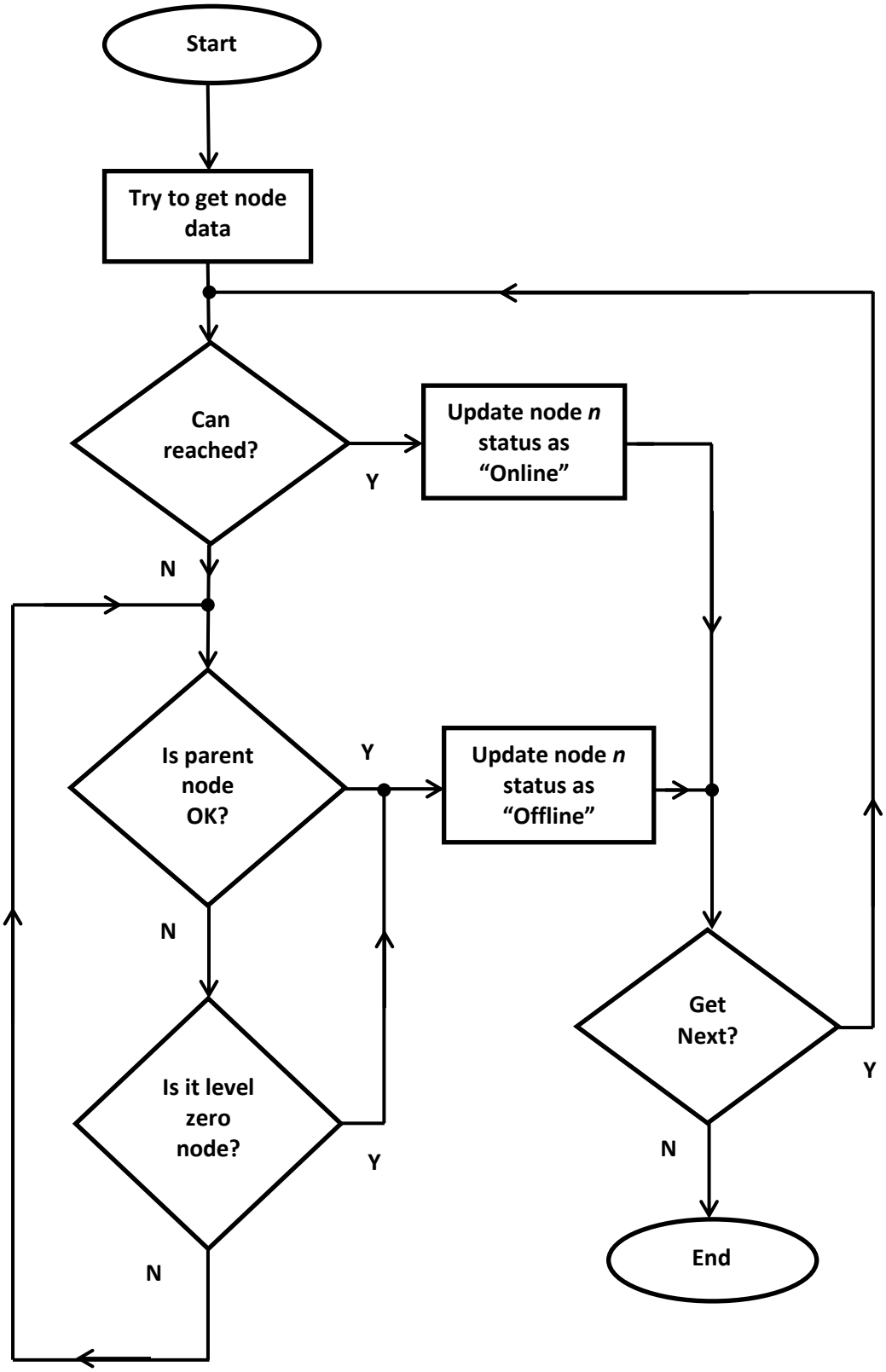
Annex 11 – Operation Cycle of CCA



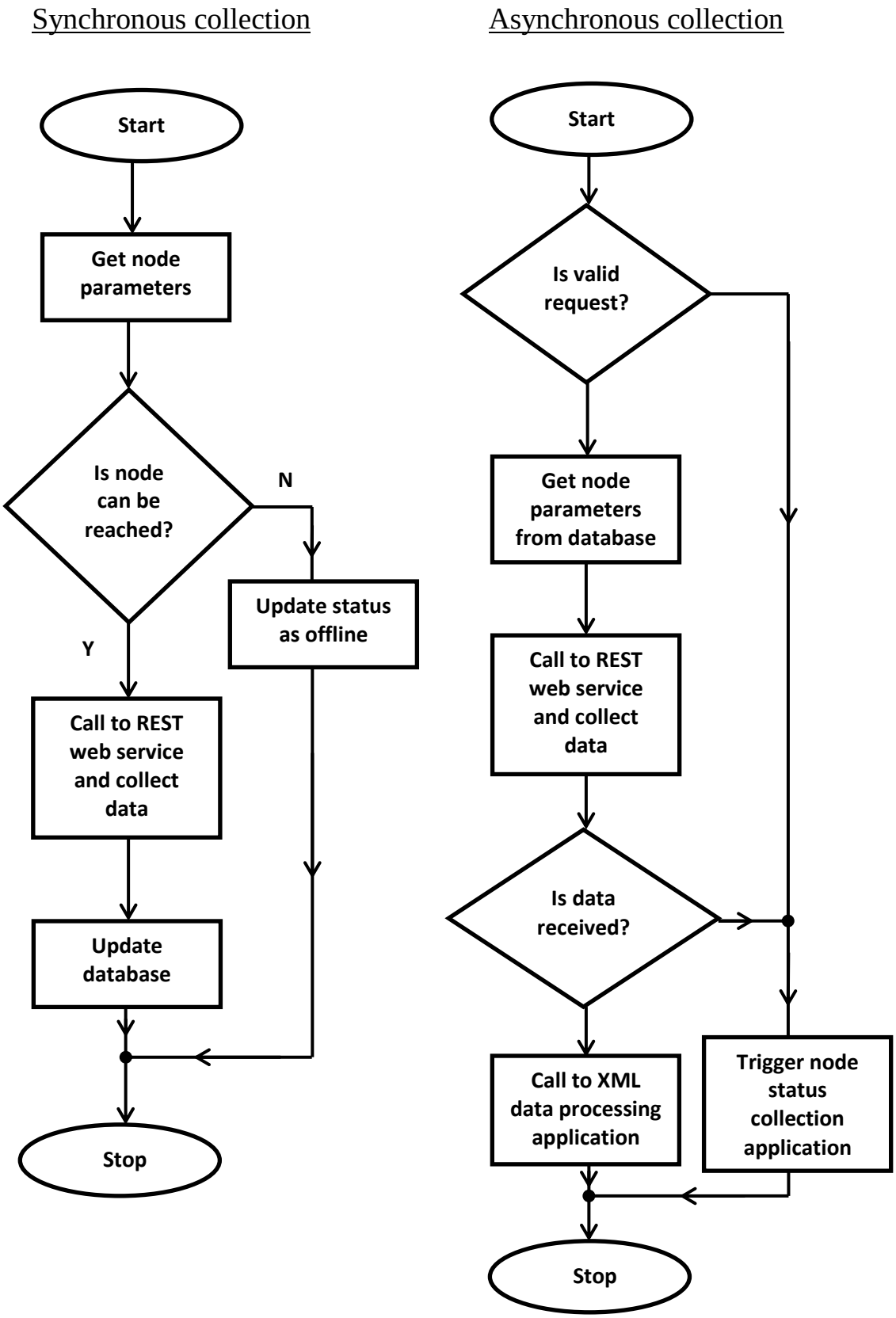
1. Thread (1 – *n*) – SFTP read process, used to read log files from the server



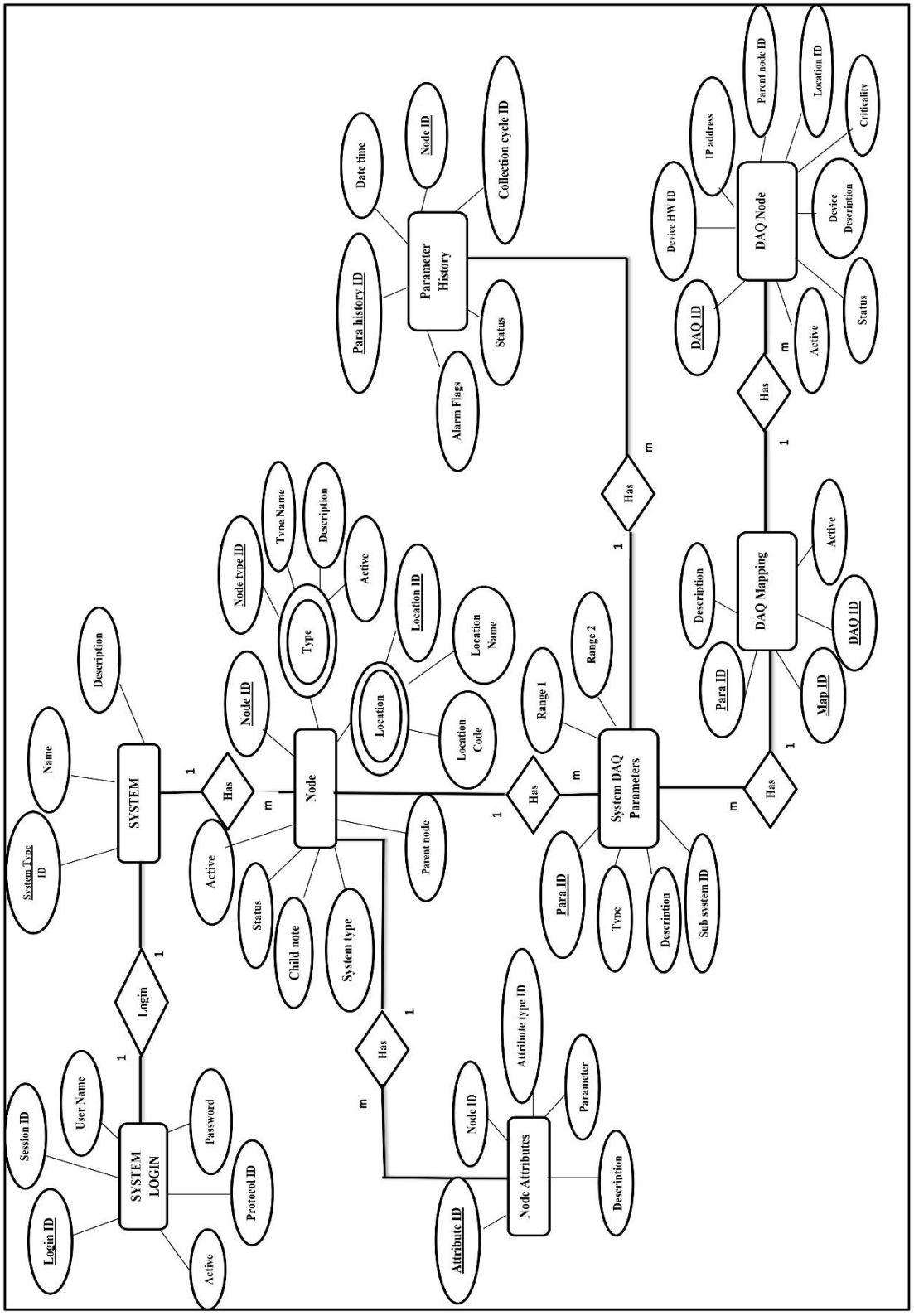
2. Thread (2 – *n*) – SNMP data collection process



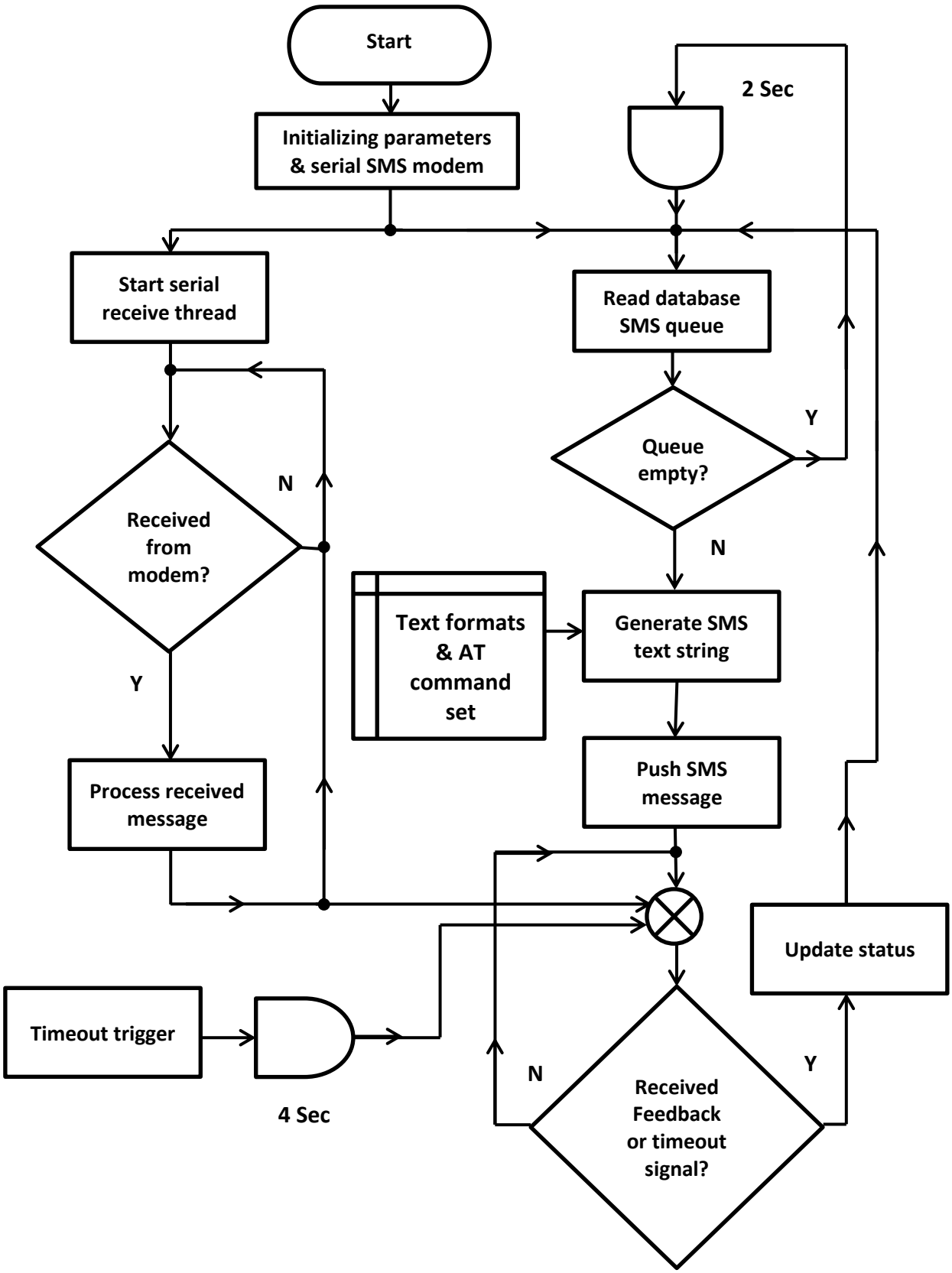
3. Thread (3 – *n*) – REST web service data collection process



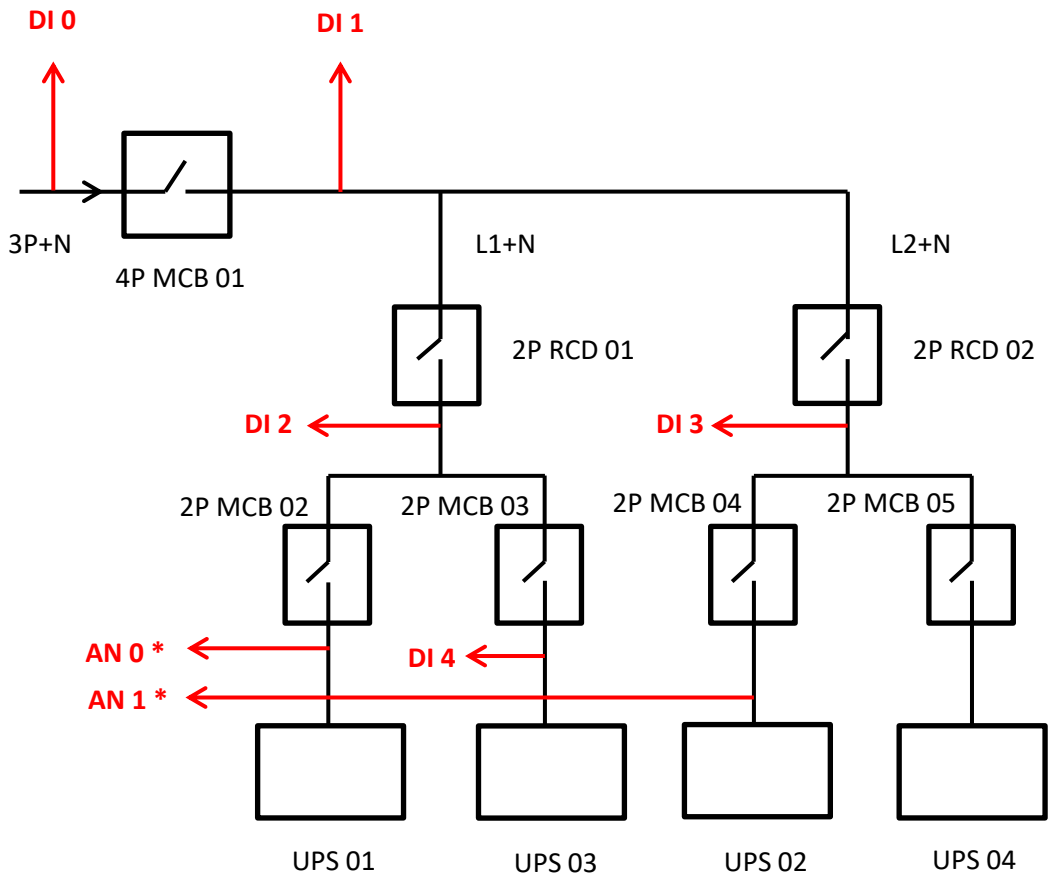
Annex 12 – Simplified ER Diagram for the Database System



Annex 13 – Simplified Operation Cycle of the SMS Server Software

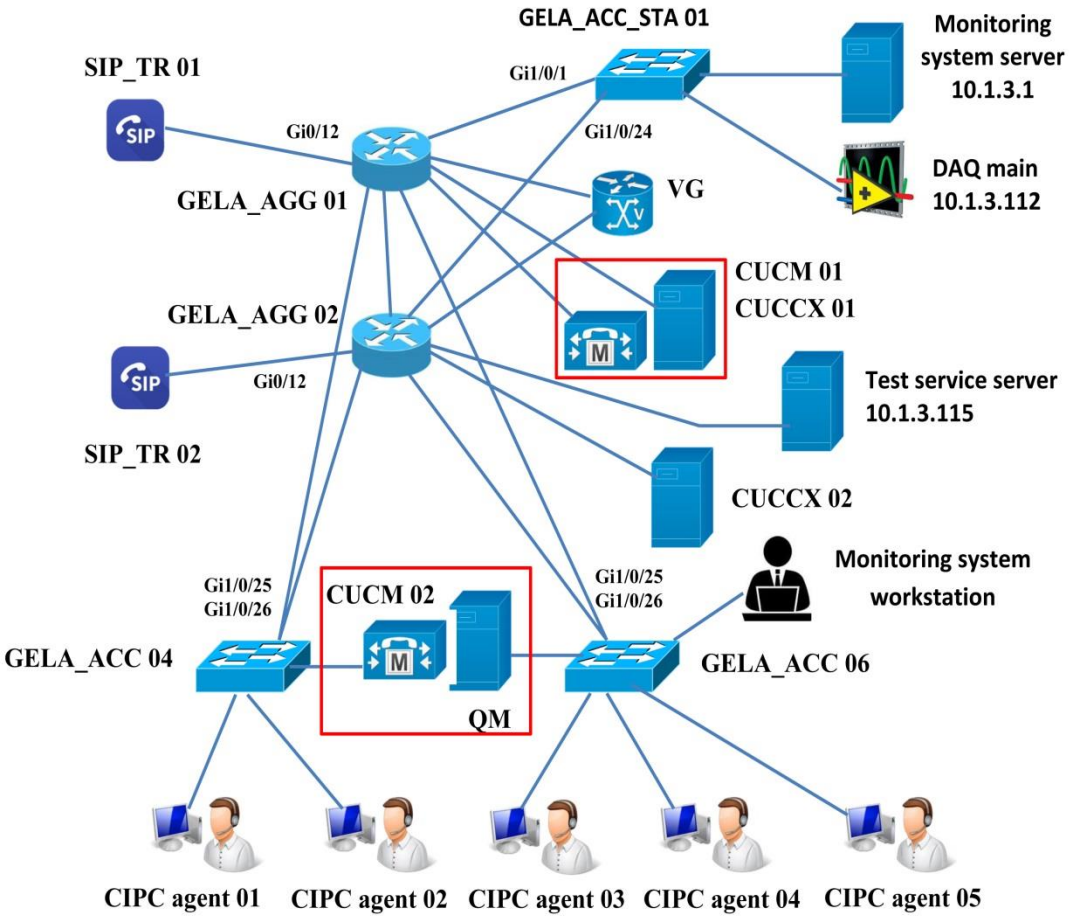


Annex 14 – Setup to Monitor the Operation in Sub Power Distribution Panel at the Server Room 01



Note * - AN 0 and AN 1 are also associated with MCB 2 and MCB 4 status parameters

Annex 15 – 1969 Call System Setup Including System Monitoring



References

[1] Mitshubishi Heavy Industries, "Multi Lane Free Flow (MLFF)" [online]. Available at: http://www.mhims.co.jp/en/products/its/mlff/index.html [Accessed 15 August 2016]

[2] Dr. S.Kamijo., "Intelligent Transport Systems Overview", University of Tokyo, Japan, July 2015, pp.18-36. [training material]

[3] Nippon Signal Company Limited, "Real time Traffic Signal Control", Marunouchi Chiyoda-ku, Tokyo, Japan, July 2015. Reference to their homepage: http://www.signal.co.jp/english/ [training material]

[4] W3 Org, "Status codes definitions" [online]. Available at: https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html [Accessed 12 August 2016]

[5] IBM Systems., "QRadar Platform" [online]. Available at: http://www-03.ibm.com/software/products/en/qradar [Accessed 18 August 2016]

[6] Cisco Systems., "Network Management System", details available at: http://www.cisco.com/c/en/us/support/docs/availability/high-availability/15114-NMS-bestpractice.html [Accessed 19 August 2016]

[7] etap™ Systems, "Power System Monitoring & Simulation - PSMS™", details available at : http://etap.com/power-system-monitoring-simulation/power-management-system-software.htm [Accessed 31 August 2016]

[8] Road Development Authority Sri Lanka, Expressway Operation Maintenance and Management Division, "Expressway Maintenance Manual", 1st Edition May 2011, pp.101-108.

[9] David Lie, "Reliability, availability and serviceability" [online]. Available at: http://cva.stanford.edu/classes/ee482a/scribed/lect16.pdf [Accessed 20 August 2016]

[10] E.J. McClusky & S. Mitra (2004). "Fault Tolerance in Computer Science Handbook" 2nd edition, pp.201-218.

[11] A. Goodloe, “Monitoring Distributed Real-Time Systems: A Survey and Future Directions”, National Aeronautics and Space Administration (NASA), February 2010.

[12] M. Rausand, A. Hsyland, “Systems Reliability Theory”, 2nd Edition, John Wiley & Sons, Inc. 2004, pp.81-106.

[13] EventHelix.com, “System Reliability and Availability”, [online]. Available at: http://www.eventhelix.com/RealtimeMantra/FaultHandling/system_reliability_availability.htm#.V8uAnVt97iw [Accessed 20 September 2016]

[14] Hoang Pham (PHD), “Handbook of Reliability Engineering”, Springer-Verlag London Limited 2003, ISBN 1- 85233-453-3, pp.120-156.

[15] J. Gray and Daniel P. Siewiorek, “High Availability Computer Systems”, Digital Equipment Corporation, San Francisco, CA. 94105

[16] R. Rajagopal, “Large Monitoring Systems: Data Analysis, Design and Deployment”, Electrical Engineering and Computer Sciences, University of California, Berkeley, 2009.

[17] Prof. Clarkson Fall, “Modular Design” [online]. Available at: https://www.cs.cornell.edu/courses/cs3110/2015fa/l/10-design/lec.pdf [Accessed 2 September 2016]

[18] Erik T. Ray, “Learning XML,” First Edition, January 2001, ISBN: 0-59600-046-4[online].
Available at:
http://ait.upct.es/asignaturas/ad/libros/OReilly%20Learning%20XML.pdf
[Accessed 5 September 2016]

[19] Leonard Richardson & Sam Ruby, “RESTful Web Services,” 1st Edition, O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA, ISBN:978-0-596-52926-0[online]
Available at: https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf [Accessed 10 Septemebr 2016]

[20] F.R.Thomas, “Architectural Styles and the Design of Network-based Software Architectures”, University of California, Irvine, 2000 [online]. Available at: http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm [Accessed 18 September 2016]

[21] Microchip, “ENC28J60 data sheet” [online], Available at: http://ww1.microchip.com/downloads/en/devicedoc/39662a.pdf [Accessed 25 September 2016]

[22] REST API tutorial, “List of HTTP status codes” [online]. Available at: http://www.restapitutorial.com/httpstatuscodes.html [Accessed 27 September 2016]

[23] The Battery University, “Charging Lithium-Ion Batteries” [online], Available at: http://batteryuniversity.com/learn/article/charging_lithium_ion_batteries [Accessed 22 September 2016].

[24] The Open Energy Monitor, “Measuring AC voltages with an AC to AC power adapter”[online] Available at: https://openenergymonitor.org/emon/buildingblocks/measuring-voltage-with-an-acac-power-adapter [Accessed 26 September 2016]

[25] PHP, “What is PHP” [online]. Available at: http://php.net/manual/en/intro-whatis.php [Accessed 10 October 2016]

[26] Oracle Corporation, “What is JAVA and why do I need it” [online]. Available at: https://java.com/en/download/faq/whatis_java.xml [Accessed 10 October 2016]

[27] Almsaeed Studio, “Admin LTE Control Panel Template” [online] Available at: https://almsaeedstudio.com/ [Accessed 18 October 2016]

[28] Mysql Documentation, “Mysql 5.7 Reference manual” [online] Available at: http://dev.mysql.com/doc/refman/5.7/en/ [Accessed 20 October 2016]

[29] PHP documentation, "pthreads introduction" [online] Available at: http://php.net/manual/en/intro.pthreads.php [Accessed 2 November 2016]

[30] W3Schools, "Server Send Events" [online]. Available at: http://www.w3schools.com/html/html5_serversentevents.asp [Accessed 5 November 2016]

[31] Microsoft Developer Network, "Model-view-controller" [online] . Available at: https://msdn.microsoft.com/en-us/library/ff649643.aspx [Accessed 8 November 2016]

[32] Tutorialspoint, "ER Model - Basic concepts" [online]. Available at: https://www.tutorialspoint.com/dbms/er_model_basic_concepts.htm [Accessed 12 November 2016]

[33] R. Ramakrishnan, J. Gehrke , "Database Management Systems", 2nd Edition [online].
Available at:
http://sirpabs.ilahas.com/ebooks/Computer%20&%20Technology/Database%20Mgmt/Database.Management.Systems.2nd.Edition.pdf
[Accessed 15 November 2016]

[34] Gavin Powel, "Beginning Database Design", Wiley Publishing, Inc., Indianapolis, Indiana, ISBN-13: 978-0-7645-7490-0, pp.271-290.

[35] RxTx Home Page, "RxTx Wiki" [online]. Available at: http://rxtx.qbang.org/wiki/index.php/Main_Page [Accessed 18 November 2016]

Appendix 01 – Information received from Sri Lanka Telecom

Gmail

Roshan Ediriweera <roshan.ediri@gmail.com>

Require help for my MSc - Systems monitoring at SLT

Lalith Wasantha <wasantha@slt.com.lk> Wed, May 20, 2015 at 1:01 PM
To: "roshan.ediri@gmail.com" <roshan.ediri@gmail.com>
Cc: Kaushalya Ekanayake <kaushalyae@slt.com.lk>

Dear Nuwan,

Please find feedback received from Eng. Kaushalya.

Dear Kaushalya,

Thanks you for expending valuable time.

Regds,

Wasantha

From: Kaushalya Ekanayake
Sent: Wednesday, May 20, 2015 12:50 PM
To: Lalith Wasantha
Subject: RE: Require help for my MSc - Systems monitoring at SLT

1. Brief outline of the systems using

   In our section we are using Cisco , Juniper , Huawei & Alcatel Routers.

2. Main categories of sub systems (communication, power etc)

   Power – Rectifiers , UPSs...etc

3. What are the common types of faults?

   Critical - Node Outages due to Power Issues , Fiber Link cut ..etc , Traffic Congestion Issues.

   Major – Customer Port down due to Fiber , Copper fault or Customer Premises equipment issues.

   Minor – Temperature alarms , Card CPU alarms... etc

4. How the staff do systems monitoring and what are the types/product names of the system or monitoring software tools using at the present ?

OpenNMS – Open source(Free) Linux based NMS system to monitor Network Nodes and their Faults (Cisco & Juniper)

Cacti Tool – Linux based Traffic Monitoring tool for Network Tarffic analyzing. (Free)

Huawei U2000 – EMS system for all Huawei Nodes.

Alcatel SAM 5620 – EMS system for all Alcatel Nodes.

5. Is there any intelligent (or artificial intelligence/machine learning) techniques use for systems monitoring ?

No. Only SNMP trap based systems.

6. Is there any identified drawbacks with the present systems monitoring methods?

There should be dedicated teams to monitor the tools and their alarms to rectify issues. If there is a single Monitoring system , number of users can be reduced.

7. What are the future plans of SLT to improve effectiveness of systems monitoring process ?

There is a plan to deploy End- to End service assurance system to interconnect all kind of Monitoring systems.

It would be great help to me, if you could reply me with answers. Simple short forms would be enough.

Regards,

Kaushalya Ekanayake BSc.Eng(Hons), AMIE(SL)

Engineer IP MPLS Network Operation I

Sri Lanka Telecom.

8th Floor, OTS Building

Lotus road, Colombo 01

Sri Lanka.

Tel: +94 11 2433713 (Voice) / +94 71 4345359 (Mobile)