

# PLC project

Gergely Feher, Craig Lennox, Daniel Rippe, and Murray Lyons

**Abstract.** This paper presents a system designed for Linamar Co.'s CAMTAC plant. The system is designed to allow their PLCs to send messages to a host PC running the Machine Status Monitor (MSM). The MSM is the client designed to receive and handle these messages. The system utilizes the existing DH+ network, ControlLogix Gateway and e-mail server. Each component of our design was individually tested and deemed operational.

**Key words:** Communication System, DDE, DH+ Network, PLC, RSLinx

## 1 Introduction

Linamar Corporation's CAMTAC plant is a modern manufacturing facility that specializes in the production and machining of engine components. CAMTAC has identified the need to implement a PLC communication system. This communication system will be developed in order to facilitate and assist in the maintenance of the CNC machines in order to keep production at a profitable rate by allowing technicians to recognize problems soon after an error occurs. The aim of this project is to design a communication system that will facilitate the maintenance of the working line by informing relevant personnel about errors in a timely and efficient manner.

## 2 Background

Some key components used by the system are:

**RSLinx:** Allen Bradley's proprietary software for managing PLC Networks. This software has to be installed in order for the MSM software suite to work. A key feature of RSLinx is its DDE (Dynamic Data Exchange) capabilities, enabling other software running on the same computer to listen in on messages sent by PLCs.

**ControlLogix Gateway:** Acts as a bridge between multiple DH+ networks, also making the network available through a remote connection using an Ethernet adapter card. To properly communicate from a DH+ network to a remote PC, the PC has to be assigned a node ID, which is linked to its IP address (Routing table). Without this routing table the gateway is unable to determine the destination of a message sent to the node identified as a PC.

**Mailsend:** An integral part of the software suite, this program handles the communication with the SMTP server. Mailsend is an open source program with GNU GPL created by Muhammad A Muquit. It has not been modified in any way to work with MSM. When MSM detects an error it calls Mailsend feeding it the proper arguments, to deliver the messages to the intended recipients.

The system at linamar's CAMTEC plant is set up as follows:

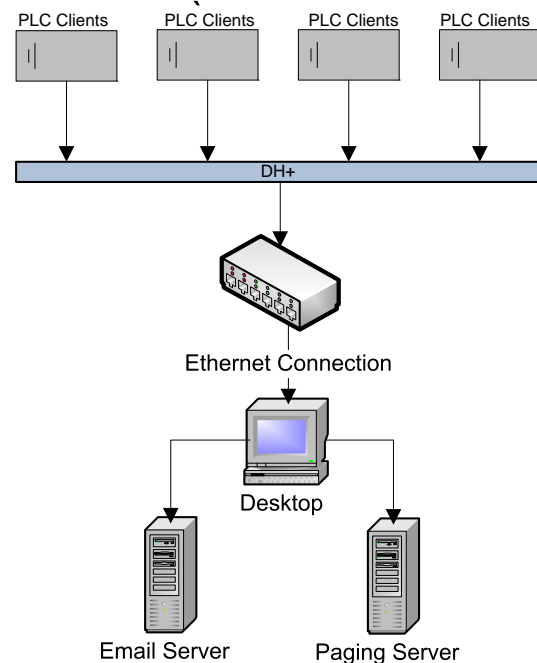


Fig. 1. System Environment

## 3 Requirements

### 3.1 Functional

- Interface must exist in order to allow for configuration.
- When a failure occurs, emails and pages must be sent to everyone on the distribution list.

### 3.2 Non-Functional

- Software must not interfere with already existing safety and functional protocols.

- Software must be a standalone application.

## 4 System Design

### 4.1 Design Overview

**PLC Program:** The ladder logic program was developed as a simple add on to existing PLC programs. As such it is very portable and minimal in size. It is simply 2 ladder rungs, the first a test condition to trigger the sending of the message. This is accomplished by initiating a timer, and triggering the message. When the timer reaches its preset it disables the msg rung.

**PC Host Program:** The software was developed for interfacing RSLinx to an email/pager server. This software is based on a Windows interprocess communication protocol known as DDE (Dynamic Data Exchange). RSLinx has the capability to communicate with this protocol. When a PLC sends data through the MSG command, directed at the node identified as a PC, RSLinx can link and place this data in a predefined Link Topic and Link Item. This link topic is then read in by the PC host program (Machine Status Monitor), then based on the number (PLC Node ID) sent, a file number offset is calculated, loading the email address list, and message belonging to that PLC node ID. Then through a shell command mailsend is invoked with the recipients and message body loaded into the arguments. A simple graphical user interface was developed to facilitate the usage of the software. The GUI gives the user the ability to monitor the current message from the PLC, also to interact with the dependency list, and another to easily modify the settings for the SMTP connection.

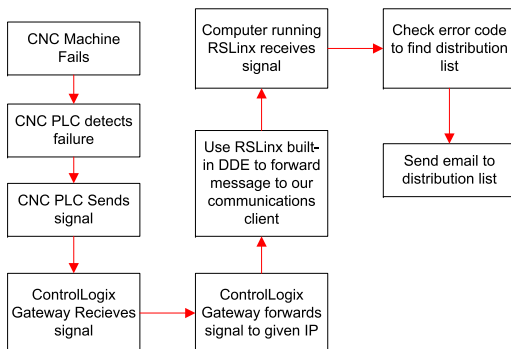


Fig. 2. Block Diagram of System

### 4.2 Mailsend

This program is an integral part of the software suite, this program handles the communication with the SMTP server.

## 5 Implementation

The Machine System Monitor has been designed and tested to run on a Windows XP operating system and RSLinx is to be running simultaneously. The chosen programming language was Visual Basic 6.0. SMTP is being used to distribute both e-mails and Pages.

## 6 Testing

The testing plan followed had three main stages; MSM standalone testing in a controlled environment, MSM testing linked to a single PLC, MSM testing linked to ControlLogix Gateway. The results of these test were for the most part satisfactory, with the exception of being unable to implement a routing table at the test location. The first two trials succeeded with the ability to send multiple or single recipients a customized message with the length of 25 Chars, based on the incoming message from the PLC. A notable area that the software suite lacks in is the ability to catch SMTP errors, if the message sending fails or a server error occurs.

## 7 Discussion

Overall this project was beneficial in providing a unique learning experience as it provided a real world problem that challenged our potential to collectively work as a team to solve. Most importantly we learned that creating a detailed technical outline of each system is crucial when approaching a software design problem.

## 8 Conclusion

The objective of this design project was to design and develop a plant-wide PLC Communication System. This system will assist in the maintenance of the CNC machines in order to keep production at a profitable rate by allowing technicians to recognize problems soon after an error occurs. The system has undergone some initial testing and the components in the system have been proven to work.