RESEARCH ARTICLE OPEN ACCESS

Error handling in programmable logic controller (PLC'S)

Eng. Abdulrahman Al Kandari*, Eng. Salem Menshawi **
**Public Authority for Applied Education and Training*
*** Public Authority for Applied Education and Training*
Corresponding Author: Eng. Abdulrahman Al Kandari

ABSTRACT
For a long time, the programmable logic controllers (PLC's) has become the main unit for industrial process control in various areas of industry. Due to its great advantages over those using traditional methods, it is used in large, medium and small industries by producing different types of controllers suitable for each size and type of industry. One of the important things in these programmable controllers is the Error handling or error diagnostics. Therefore, in this paper we deal with the types of faults whether in the controllers or in the programme and take as a study to deal with the faults in the programmable controller's plc_s5_100 and programmable controllers plc_s7_200.
Keywords – PLC, Diagnostics, Error handling, Special memory, I Stack

---

DATE OF SUBMISSION: 15-01-2020 DATE OF ACCEPTANCE: 31-01-2020

I. INTRODUCTION

In the industrial control field, there are many solutions using to control processes in industrial plants such, microprocessor-based systems, microcontrollers, relay circuits and PLC's. Every one of those are suitable for certain field but the best solution for the industrial field is the PLC's.

1.1 Microprocessor and Microcontroller

At the beginning of 70'S, two companies, Intel and Texas instruments produced microprocessors to change the future of the world. The term microprocessor, which is a very small electronic circuit fabricated by semiconductor material, contains arithmetic logic unit, control unit, general purpose registers and timing and control unit. The microprocessor controls all the activities of the system. To perform a specific job or task, the microprocessor executes a programme stored in memory. The programme consists of a set of instructions. It issues address, control signals , fetches instructions and data from memory. The instructions are executed one by one from internal processor and based on the result it takes its appropriate action. We can say that the microprocessor forms the core of microcontroller. The term microprocessor-based system is any system that contains a microprocessor. The term microcontroller, consists of microprocessor as CPU, memories like EPROM and RAM, input device, output device and interfacing devices.

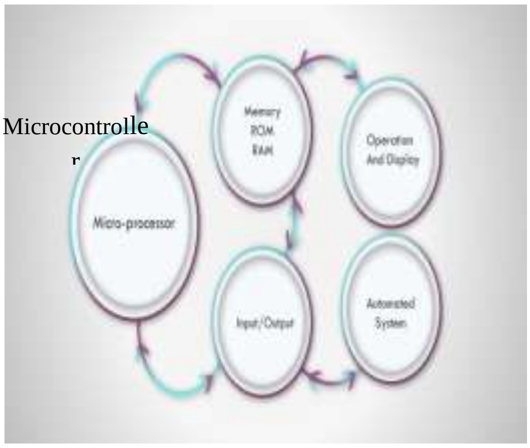

Fig.1 Microcontroller

To perform a specific job or task, read current data (water level, temperature, pressure, flow etc.), predefined data (level and temperature which should be reached for particular programme) and the processor which receives the current data from system (sensors or keyboard) via I/O component and kept it in the memory. The microcontroller reads the instructions of the programme and changes the state of the output unit (start/stop pump, on/off heater, open/close valve). The microcontroller systems are used in industrial processes, office administration and high speed performance, compact in size and cost less. The devices are programmable. There is flexibility to alter the system by changing the software only, and it also is more reliable. The Microprocessor based System has some disadvantages, limitations of I/O signals, limitations the size of data, applications are limited and analog signals cannot be processed directly. The

Programmable Logic Controller, or PLC, is an industrial computer control system. These controllers can automate a specific process, machine function, or production line. Any process can be greatly enhanced using this type of control system. There are a few features that set PLCs apart from industrial PCs, microcontrollers, other industrial control solutions, the large number of I/O modules and various types of signals (analog or digital). Input devices might include sensors, switches, and metres while outputs might include relays, lights, valves, and drives. In addition to input and output devices, a PLC might also need to connect with other kinds of systems; for example, users might want to export application data recorded by the PLC to a supervisory control and data acquisition (SCADA) system. PLCs offer various ports and communication protocols to ensure that the PLC can communicate with other systems. To interact with the PLC in real time, users need Human Machine Interface (HMI). It enables users to review and input information to the PLC in real time.

1.2 PLC's and Computers

A computer is an electronic device that manipulates information, or data. It can store, retrieve and process data. You may already know that you can use a computer to type documents, send email, play games, and browse the Web. You can also use it to edit or create spreadsheets, presentations, and even videos. The PC computer receives data and processes it into valuable information stored in different storage media. The programmable logical controller (in the simplest form) consists of input unit, output unit and central processing unit. Important characteristics of PLC systems for PC computers are designed for industrial and production processes. PLC fittings withstand electrical noise, vibration, temperature and humidity. PLC is programmed in several easy-to-use programming languages and is built into the language of the programmes embedded in memory and not on a drive, a screen or a keyboard. PLC systems rely on many input and output units and connection points. The PLC programme executes a single programme in a logical, sequential and organized manner. It is also designed in an easy way to install and maintain it. The input and output units are easily connected and replaced. The programmer can know the errors and fix them by messages and error indicators that appear on the screen. To write programme for PLC, we use application package installed in PC computer. By using the programming languages, we can write the programme through several languages (LDR-FBD-STL), save the programme, know the implementation of the process the monitoring of programme steps and the possibility of displaying the programme in several images (process - graphical representation) through the HMI screen and determining how to operate the system and receive alerts, alarms, values and trends. PLC controllers are produced for a wide variety of applications, not for a specific application. All logic circuits, memory functions, timers, and counters are set up by the manufacture inside the CPU and used by programming. PLC are available in different sizes according to the number of inputs, outputs, memory, counters, timers, flaps, speed of executing orders, types of operations on the programme. Large controllers are made in the form of cards, CPU cards, input cards, and output cards. Using these cards, the programmed control system can be designed and installed to make specific applications.

PLC's and relay systems

Increasing direction towards the operation of factories by automatic control necessitates a massive change in the rules of force in electronic and electrical control technology in recent years. Programmed control systems have become of great importance for control systems that use relay and contactors, in various fields the traditional control has been replaced by programed control systems. The fundamental difference between programmed control and conventional control can be summarized in the following; control functions do not use many wires but are replaced by the programme.

Programming is an easy way that uses simple and understandable symbols in the engineering field.

The growing set of jobs in industrial and production processes requires many more flexible and sophisticated applications. In PLC systems, the rationale and reasoning principle to a programme that performs the functions of the migration has been increased and implemented through the implementation of applications (signals - analog processing - comparison - timing - calculation - counting) and these advantages make it replace traditional control. Relays must be interlocked for one function and when changing the system, the relay wires must be changed, and control panels replaced. PLC systems have managed to get rid of most of the change in the system and the wires are attached to conventional control circuits, so we find many advantages in PLC systems. Reliability, the ability of PLC systems to complete the required task at a specific time, is one of the most important features of the programme. That helps to perform systems work, reduce the chances of failure and has been developed using the possibilities. When the programme is written and implemented, it can be tested, measured, and how effective it is. Many contact points, a single complement within the PLC controls gives hundreds of points of communication, whether closed or open, and the addition of contact points is done easily by adding a simple part to the programme. PLC communication is the ability to communicate with controllers and devices to perform the functions of controllers (graphical - supervisory - monitoring ...). Optional implementation of the programme is before installing the PLC controllers on the industrial equipment, the programme can be tested in the laboratory. Implementation can be carried out at the required location and add an update or amendment. Response, PLC systems operate at high speed and the programmable control unit works in real time. Industrial and production machines require fractions of a second to carry out the tasks, and through PLC systems that can respond effectively and quickly to implementation, make the programme more robust and easy. Flexibility, since the stored programme has become easier to create, changes and modifies the inputs and outputs in writing the programme. Therefore, we can add or change steps in control without resorting to making changes in the connections, whether the input or output, the user can make the appropriate update of the system, and provide security functions through keys or passwords in the programme. Error Correction, making changes or fixing errors in the case of relay control panels requires a long time, while using PLC controllers, the command changes in some commands in the programme. The errors and correction can be found through the display and implementation after modification. Expressive messages about errors that occur in machines appear during operation. In advanced PLC controllers, messages can be programmed to the machine operator for every possible error. The message appears on the connected PLC controllers screen. Low cost, with the introduction of modern technologies, it has been possible to reduce the cost of PLC controllers so significantly that in many cases they are more affordable than traditional methods of control.

1.3 Errors

Data processing and error handling are among the most important parts. In the process of digital data processing, there must be methods and mechanisms for handling, detecting and correcting errors, and the technique of error detection adopts data correction **.**Error handling forms occur in IP communications, chain communication, or memory data storage **.** And the error may be a difference between the data sent and the data received or in the case of storing data a difference between the data to be sent for writing and the data written on the media (memory). The process of detecting errors is a mechanism that depends on several steps, and once the error has occurred, it must be confirmed that there is an error and where the error occurred, and this contributes greatly to solving the errors received. Therefore, we rely on methods of detecting the error by the occurrence of the error, but we do not know the location of the error, the method of the error occurs, and we know the location of the error. The error is corrected in algorithm operations and fix the error bit to the normal bit any data bit error to the normal bit data. The error may be in one bit and it is the result of the change during the transfer of data from zero to one or vice versa, or an error in a group of bit as a result of the change in two or more values within one unit and thus changing their values from zero to one or the opposite and the reason is the speed of data transfer time that has been exposed. It has data to change, and one error is to add an additional bit to the data from the sender or receiver **.** Detecting and correcting an error is one of the most important things, and since there is no way to prevent errors, but it is possible to find different ways to prevent and solve the error, there are ways to discover and correct the error**:**

1. Receiving and receiving data**.**
2. Know and track the processed data**.**
3. Error detection by organizing the original data into a single structure**.**
4. Find errors in the data received, request for retransmission and error repair**.**

The error handling mechanism that the system uses:

1. ARQ (Automatic Repeat reQuest).
2. FEC (Forward Error Correction).
3. HARQ (Hybrid ARQ).

II. HANDLING ERRORS IN THE S7-200

The PLC_ S7-200 is small, compact systems and has several types of CPU. The PLC_ S7-200 classifies errors as either fatal errors or non-fatal errors.

2.1 The fatal errors

It contains several types as follow.

1. Hardware failure, requiring the PLC module to be repaired (you cannot view the PLC error code).
2. Hardware failure, such EEPROM, Memory cartridge (you can view the PLC error code).
3. Checksum error on user programme, configuration parameters, force data, default output table values and user data, DB1. (You can view the PLC error code).
4. Indirect addressing error, floating-point value error.

2.2 Handling fatal errors

The PLC performs the following tasks when a fatal error is occurring.

1. Entering to the stop mode
2. Light RED LED fault

3. Turns off the outputs

The table shown below containing some fatal errors and small description for it.

Table 1 fatal errors

Code	Description
0000	No fatal errors present
	...
0004	Internal EEPROM failed
0005	Internal EEPROM checksum error on user program
0006	Internal EEPROM checksum error on configuration
	...
000D	Memory cartridge checksum error on force data
000E	Memory cartridge checksum error on default output table values
000F	Memory cartridge checksum error on user data
	...
0011	Compare contact indirect addressing error
0012	Compare contact illegal floating-point value error

By STEP 7-Micro/WIN, select the menu command PLC > Information to display the error codes generated by the PLC as in the figure below.

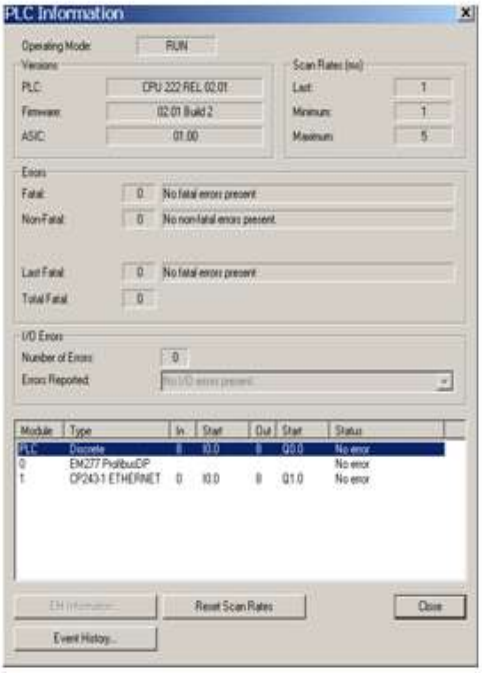

Fig.2 information

shows the PLC Information displays:

1. The error code and the description of the error.
2. The Last Fatal field shows the previous fatal error code generated by the S7-200.
3. I/O errors and the description of the error.

Once the error corrected, restart the PLC (the PLC has a fatal error and the fault light is on) by Power-cycle. Move the mode switch from TERM or RUN to the STOP position, then to RUN position. Restarting the PLC to clear the fatal error and cause power-up diagnostic testing. If another fatal error occurs, the PLC sets the System Fault LED again, otherwise, the PLC begins normal operation. The S7-200 error listed in codes, and the special memory (SM) bits used for monitoring errors.

2.3 The non-fatal errors

It may degrade some aspect of PLC performance, but do not render the PLC incapable of executing the user programme and updating I/O. The non-fatal errors indicate the errors in construction of the user programme, execution of instructions and the input/output module. There are three basic categories of non-fatal errors, Programme-compile errors, I/O errors and Programme execution errors.

2.4 Handling non-fatal errors

2.4.1Programme-compile errors

STEP 7-Micro/WIN compiles the programme (OB1, Subroutine and interrupt routines) before downloads. If the PLC_S7-200 detects that the programme incompatible with the compilation rule (The compiler checks for calls to non-existent subroutines, invalid calls to subroutines, illegal syntax for instruction operands and invalid network constructs), the download is aborted, and an error code is generated. After you correct your programme, you can download it again.
Use the PLC > Information… command to determine what type of errors has occurred.

2.4.2 I/O errors

Each I/O module has configuration, the PLC_S7_200 reads the configuration of modules at startup, stores this information in the SM memory, during normal operation, and the PLC_S7-200 periodically, checks the status of each module and compares it with the configuration at startup. If the PLC_S7_200 detects a difference, the PLC_S7_200 sets the configuration error bit in the module-error byte. The PLC_S7_200 does not read input data from or write output data to that module until the module configuration again matches the configuration at startup. The module status information is stored in special memory (SM) bits. The programme can monitor and evaluate these bits.

2.4.3 Programme execution errors

User programme can create errors while being executed. The causes of these errors may be from use improper instruction or from the invalid data by an instruction. Run-time programming errors are non-fatal error created by you or your programme while the programme is being executed, for example, an indirect-address pointer that was valid when the programme compiled could be modified during the execution of the programme to point to an out-of-range address. Use the PLC >

Information… command to determine what type of Run-time programming error has occurred. Special memory (SM) bit is set at occurrence of a run-time programming problem and remains set while the S7-200 is in RUN mode. Programme execution error information is stored in special memory (SM) bits, as in the flowing table.

Table 2 special memory (SM) bits

SM Bit	Description
SM0.2	Retentive data lost
SM1.3	Divide by zero error
SM3.0	Parity error
SM4.0	Comm. Interrupt queue overflow
SM4.1	Interrupt queue overflow
SMB9	Module 0 I/O error Byte
SMB11	Module 1 I/O error Byte
SMB13	Module 2 I/O error Byte
SMB15	Module 3 I/O error Byte
SMB17	Module 4 I/O error Byte

User programme can monitor and evaluate these bits. Information about the SM bits are used for reporting programme execution errors. The s7-200 does not change to stop mode when it detects a non-fatal error. It only changes state of SM memory and continues with the execution of your programme. However, you can design your programme to force the S7-200 to stop mode when a non-fatal error is detected. The following sample programme shows two of the global non-fatal error bits and changes the S7-200 to stop whenever either of these bits turns on.

```
LD SM 5.0
O SM 4.3
STOP
```

When an I/O error or a run time error occurs, go to stop mode.

III. HANDLING ERROR IN PLC_S5-100U

The industrial controller PLC_S5_100U has several types of CPU (Central Processing Unit) depending on the size of RAM, the number of inputs and outputs, counters, timers and the number of blocks that can be programmed and the execution time of commands. The PLC_S5-100U classifies errors as CPU errors, programme errors and I/O errors.

3.1 Central Processing Unit errors (CPU)

The PLC_S5-100U Handling errors, first by determine the location of the error is in the Central Processing Unit (CPU) or in the Input and Output units (I / O Modules) or in the programme. When the red LED lights up, it shows that the Central Processing Unit (CPU) is in stop mode, the error is either in the CPU or programme (when loading or scanning the Programme).

3.1.1 Read the relevant bits in I Stack

Table 3 bits in I Stack

Byte/ Bit	7	6	5	4	3	2	1	0	System Data Word(DS)
1			BST SCH	SCH TAE	ADR BAU				SD 5
2									
3	STO ZUS	STO ANZ	NEU STA		BAT PUF				SD 6
4						AF			
5			KOPF NI						SD 7
6	KEIN AS	SYN FEH	NINEU					UR LAD	

In I Stack(Interruption Stack is an internal memory area within the CPU and is used to store the causes of errors if errors occurs, the relevant bit is set), the value of relative bit is reset(bit = 0) in case of no error and in case of error is set (bit = 1) the contents of I stack can be read by byte through the menu within the programming unit, the PLC must be in the stop mode and there are some errors or messages can be read in run mode. The following table shows the bits for some errors as example.

Table 4 errors change PLC to stop mode

Error Massage In I Stack	Byte	Cause Of Error	Error Elimination
BAU	10	During automatic loading of the program -Battery missing/low and no valid program on memory submodule	Replace battery and generate or reload program
NAU	10	CPU power supply failure	
STOPS	9	Mode selector set to the STOP position	Set mode selector to RUN position
STS	9	- Software stopped by statement (STP) -STOP requested by programmer -STOP requested by SINEC L1 master (from CPU 102 onwards)	

3.2.2 Determining the causes of errors

In the case of programme loading and occurs error in the programme. The PLC enters the STOP mode and the red LED flashing. The following table can be used to determine the cause of the errors and How-to elimination the errors. When loading the programme, the red LED flashing, the following table shows some of these errors.

Table 5 errors change red LED flashing

Error Massage In I Stack	Cause Of Error	Error Elimination
ASPFA	- Program on memory submodule too long for PC program memory - Program on memory submodule contains invalid block number	Set mode selector from RUN to STOP
ASPFA	EEPROM submodule defective or too small for Program in PC memory	Replace memory submodule or use larger EEPROM submodule

I Stack contains a lot of information about the system and the errors occurring in the system, but the explanation of the abbreviation used in interruption stack (I Stack) put in tables in the manual of the PLC, these table explain every code of the errors, as in the following table.

Table 6 abbreviation of errors

Error Massage in I Stack	Byte	Explanation
BST SCH SCH TAE ADR BAU	1	Shift block Execute shift operation Structure address list
STO ANZ STO ZUS BAT PUF NEU SAT	3	PC in "STOP" mode Internal control bit for STOP/RUN change Battery backup working PC net pet in "RUN" mode after power ON (for cause see bytes 9/10)
AF	4	Interrupt enable/ enabling of time-driven OB 13 and Interrupt-driven OB2
KOPFNI	5	Program contains error(s) Block header cannot be interpreted

3.2 Programme Errors

3.2.1 Handling errors by using I Stack function

After downloading the programme and through executing the programme, occurring error as illegal statement, the step address counter in I Stack (byte 25, 26) contains the absolute address of the next STEP 5 statement before the CPU entered STOP mode.

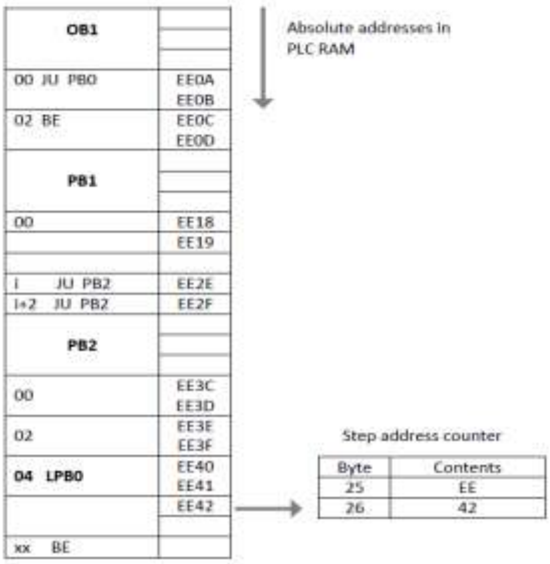

Fig.3 absolute address addresses in PLC RAM

In order to correct the error in the programme, it is necessary to have the address of the statement in the block (relative address), the block containing the error found by comparing the contents of the step address counter and the contents of the programmer function (start address of blocks, OB1,PB1,PB2), the relative address of the error is obtained from difference between contents of the step address counter and the block start address. Example:
If step address counter contents = EE42
PB2 start address = EE3C
Then:

Relative address of the error = EE42 – EE3C = 0006

"0006" is the address of the statement in PB2 before the CPU entering in the STOP mode. We can be using "SEARCH" function in PG programmer to display the statement containing the error.

3.2.2 Handling errors by using B Stack function

When occurring error, as illegal statement and the CPU entries stop mode, the B Stack contains, data block that was valid before exiting the block, the relative return addresses of the blocks called and the absolute return address in the programme memory at which programme scanning is resumed after return of control.

Table 7 B stack function

B Stack function		
Block	relative address	DB
PB 4	0010	5
PB 2	0004	5
OB 1	0006	5
Absolute return address	0505	

The programmer shows DB5 contains the path of the error (via OB1 then PB2 then PB4). By the return address in the PB4, we find the block containing error.

3.2.2what the difference between Handling errors by B Stack function and I Stack function

Table 8 DIR PC function

DIR PC function (I Stack)		
Block	start address of Blocks in CPU RAM	DB
PB 4	EE3C	NO
PB 2	EE18	NO
OB 1	EE0A	NO

In I Stack function, the step address counter contains the absolute address of the next statement in the programme memory, DIR PC function contains the start address of blocks, then the error address in the block = absolute address of programme memory - start address of the block.

In B Stack function, the stack contains all return addresses of the blocks and absolute return address then the error address in the block =the return address of the block -absolute address of programme memory.

IV. CONCLUSION

Finally, after study the types of error and how handling errors in PLC_S5_100U and PLC S7_200 We find that PLC S7_200 classified errors as fatal and non-fatal errors, then classified fatal error to hardware failure ((you cannot view the PLC error code), Hardware failure(you can view the PLC error code),checksum error on user programme, configuration parameters, force data, default output table values and user data(you can view the PLC error code) and indirect addressing error, floating-point value error. handling fatal errors in PLC S7_200 by change state of plc to stop mode, light red led and turn off the output. PLC S7_200 classified non-fatal errors to Programme-compile errors, I/O errors, Programme execution errors. Every type of this errors has a number and a description and we can see it in the PLC > Information… command. PLC S7_200 handling the Programme-compile errors by STEP 7-Micro/WIN according compilation rule, I/O errors, PLC S7_200 checks the status of I/O modules and compares it with the configuration at startup, if detects a difference, it sets the configuration error bit in the module-error byte. Programme execution errors, PLC S7_200 checks improper instruction or the invalid data and sets the special memory (SM) bits. The PLC_S5_100U classifies errors as CPU errors, programme errors and I/O errors. Handling errors first by determining the location of the error is in the Central Processing Unit (CPU), I/O modules or in the programme if the error in CPU or programme (when loading or scanning the Programme), red led lights up or flashing, then by using I Stack we can know the error and by table we can know the description of error and how eliminate it. If there are user programme error, we need some calculation to know the address of the statement in the programme which errors occur at it. The big advantage of s7_200 over s5_100u the size of memory using for storing errors and configuration, s5_cpu using 32 byte for errors but in s7_200 using 549 for error and configuration, it using 14 byte for I/O module and error register only, s5_100u using diagnostics module for monitoring the I/O bus but s7_200 not using any module for diagnostics, this indicates that s5_100u. We recommend more study for error handling in industrial controllers, especially large ones such as S7_300 and S7_400, so that we learn about the progress made in this area and this point.

REFERENCES

[1]. Programmable logic controllers, Fifth Edition Published by McGraw-Hill, Education by frank D. Petronella. (2017)

[2]. S5_100U Programmable Controller Manual Order Number 6ES5 998-0UB22.

[3]. S7-200 Programmable Controller System Manual. Order number: 6ES7298-8FA22-8BH0 Edition 04/2002.

[4]. A knowledge-based real-time diagnostic system for PLC controlled manufacturing systems, Published in: IEEE SMC'99 ConferenceProceedings.1999IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99 CH37028).

[5]. Introduction to microprocessors and microcontrollers, john crisp,2004

[6]. 8085 microcomputer training system by R. Adorf, H.Bax,W.Becker

[7]. Hu, W., Schroeder, M., Starr, A.G.: A Knowledge Based Real Time Diagnosis System for PLC controlled manufacturing Systems, City University London, EC1V 0HB, UK. Department of Computing. School of Engineering, University of Manchester, Manchester, UK

[8]. Qin S., Wang G. (2012) A Study of Fault Detection and Diagnosis for PLC Controlled

[9]. Manufacturing System. In: Xiao T., Zhang L., Ma S. (eds) System Simulation and Scientific Computing. ICSC 2012. Communications in Computer and Information Science, vol 326. Springer, Berlin, Heidelberg

[10]. Effects of diagnostic on the safety of a control system realized by safety PLC. Publisher: IEEE.DOI:10.1109/ELEKTRO.2016.7512118