SEMARAK ILMU
PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

SEMARAK ILMU
Journal of
ADVANCED RESEARCH IN APPLIED SCIENCES AND ENGINEERING TECHNOLOGY
ISSN 2462 - 1943

A Low-Cost Automatic Tester of PLC Functions

Suvalee Chuvanich[1,*]

[1] Faculty of Industrial Technology, Songkhla Rajabhat University, Mueang Songkhla District, Songkhla 90000, Thailand

ARTICLE INFO

Article history:
Received 3 February 2024
Received in revised form 12 June 2024
Accepted 14 August 2024
Available online 2 September 2024

Keywords:

PLC testing; ATE; Capella; ARCADIA; COTS

ABSTRACT

In smart industries, a Programmable Logic Controller (PLC) is necessary to control system functions that confirm to work according to requirements. Furthermore, the testing of PLC functions should be carried out in a very short time with very good accuracy and high reliability, but at the lowest cost possible. This paper presents the Automatic Test Equipment (ATE) for PLC functions. The objective of this paper is to design low-cost ATE from commercially off-the-shelf (COTS) with ease of testing. A model-based systems engineering approach is used to design the tester. The ATE design uses the open-source Capella tool, which supports the Architecture Analysis and Design Integrated Approach (ARCADIA) method. The structural elements of the ATE system are the test controller, the switching relays, the system software, and the Human-Machine Interface (HMI), which is used to display the test results. The test controller executes the test program and controls the inputs of the PLC under test using the test patterns via switching relays. The responses from the outputs of the PLC are sent to the inputs of the test controller and compared with the expected responses. The results of PLC function testing are displayed on the HMI. The results show that the maximum speed of the ATE is 10 ms/pattern, and the total cost is under 140 USD.

1. Introduction

Industry 4.0 is the integration of digital technologies into manufacturing and industrial processes. This includes IoT, data analytics, artificial intelligence (AI), and automation, etc. [1,2]. The aim of Industry 4.0 is to enable smart factories and the creation of intelligent manufacturing [3]. Programmable Logic Controllers (PLCs) are essential in Industry 4.0 and Smart Factory. They are the main control for manufacturing machines and processes that receive real-time input from sensors and automatically control devices and machines. Figure 1 shows a basic PLC architecture as described in Bedi *et al.,* [4]. PLC programs are developed using specialized software and a structured approach. PLC programming software often includes debugging tools like online monitoring and simulation. Before deploying the program to the PLC, thoroughly testing it in a simulated or offline environment is required. PLC programs are tested by using various methods and tools to verify the functionality, performance, and reliability of the PLC logic and hardware.

[*] *Corresponding author.*
E-mail address: suvalee.ch@skru.ac.th

https://doi.org/10.37934/araset.51.1.5269

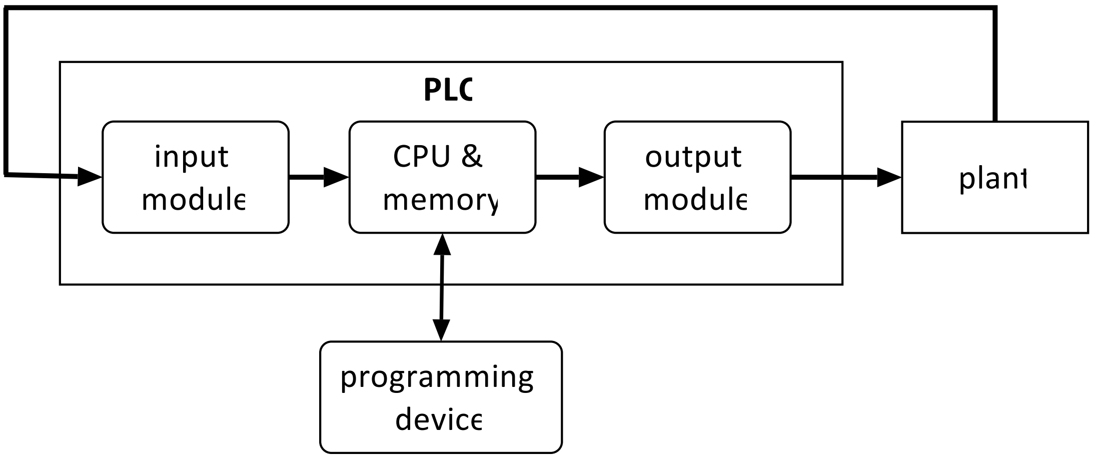


Fig. 1. Basic PLC Architecture

Testing helps ensure that the PLC operates as expected and meets the control requirements of the system. Testing PLC functions thoroughly helps ensure the reliability and safety of automated systems. Therefore, the testing of PLC functions is necessary before commissioning. It helps avoid the potential costs of system downtime, production losses, and emergency repairs. A functional tester generally tests the functions of a unit under test (UUT), which can be boards or systems. A functional tester evaluates the overall functionality of the UUT by applying input signals to the UUT and comparing the output signals from the UUT with the expected response using two common methods: stored response or signature analysis. With the stored response, the actual output response of the UUT is compared with the stored expected response. Any disparities between the stored and expected responses indicate a fault in the UUT. In signature analysis, the UUT's compressed output responses are used. During testing, the actual UUT signature is compared to the expected signature. Any discrepancies indicate a fault in the UUT.

Automated Test Equipment (ATE), as shown in Figure 2, can be a valuable tool for testing the functionality and performance of PLCs, especially for larger and more complex PLCs used in industrial automation and control systems. Automated PLC acceptance testing uses ATE to perform scalable and reproducible tests of the PLC design as part of the factory acceptance test [5]. ATE testing is a technique used to test complex real-time embedded systems, such as controllers, ICs, or sensors. ATE testing involves connecting the system under test to a simulation model that mimics the real-world conditions in which the system will operate [6]. A typical ATE architecture consists of four main elements:

i. the unit under test (UUT) and fixture
ii. the test programs
iii. the ATE test software
iv. the ATE test hardware, which consists of stimuli switching, stimuli devices, measurement devices, and measurement switching.

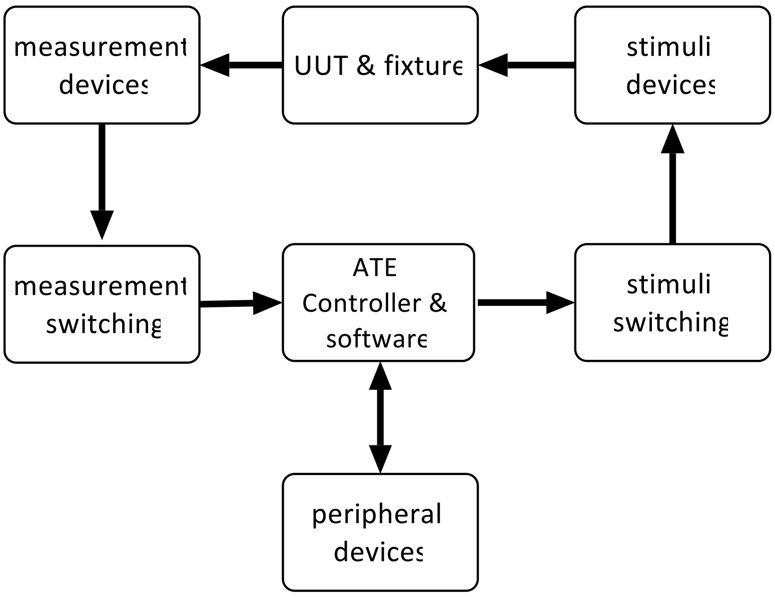


Fig. 2. Basic ATE Components

The test program can vary depending on the complexity and requirements of the UUT. Developing test programs to verify the functionality of PLC functions is a critical part of the PLC programming and commissioning process.

Even though ATE can be a powerful tool for testing PLC functions, small PLC designers may have valid reasons for not employing it, including cost constraints, scale of production, complexity, testing requirements, and the availability of alternative testing methods. They may rely on manual testing and quality control processes more suited to their needs and resources. Using a PLC as an ATE is possible, as shown in Figure 3, provided that this PLC hardware can facilitate all necessary functions during testing. Most modern PLCs have modules that can be programmed as the ATE. However, only high-end PLCs can program in high-level languages that are appropriate for ATE test functions.

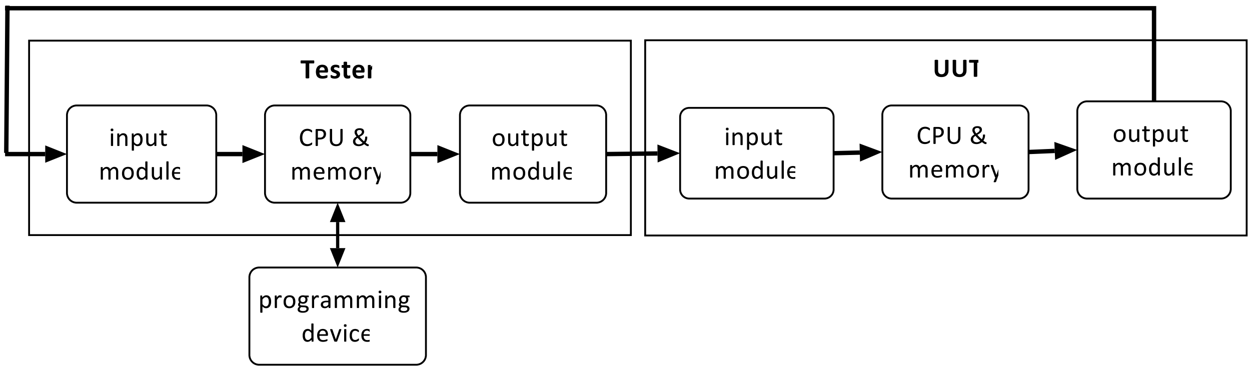


Fig. 3. Using the PLC as the tester

Commercial off-the-shelf (COTS) [7] parts are parts or components where the buyer imposes no special requirements. COTS components can offer robustness, reliability, and significantly accelerate tester development. Currently, COTS components are widely used. For example, a COTS unmanned rotorcraft was used to design brick wall construction [8] and miniature implantable wireless medical devices [9]. The COTS components available on the market provide less opportunity for customization. Therefore, in many instances, it can be challenging to find a COTS component that completely matches the requirements of a specific application.

The STM32 is a family of microcontrollers produced by STMicroelectronics. These microcontrollers are widely used in various applications, including industrial automation, robotics, and control systems. For example, STM32F1 is a physical hardware system for indoor environmental detection [10]. Laski *et al.,* [11] present that STM32F4 allows both rapid prototyping of complex control algorithms and solving kinematics equations and their hardware implementation. While the STM32 is not a Programmable Logic Controller (PLC) in the traditional sense, it can be used as a PLC with the appropriate software and hardware. There are several clone PLCs available in various capacities.

These clone PLCs are often marketed as a cheaper alternative to established brands such as Mitsubishi, as shown in Figure 4 for a FX3U-30MR clone.

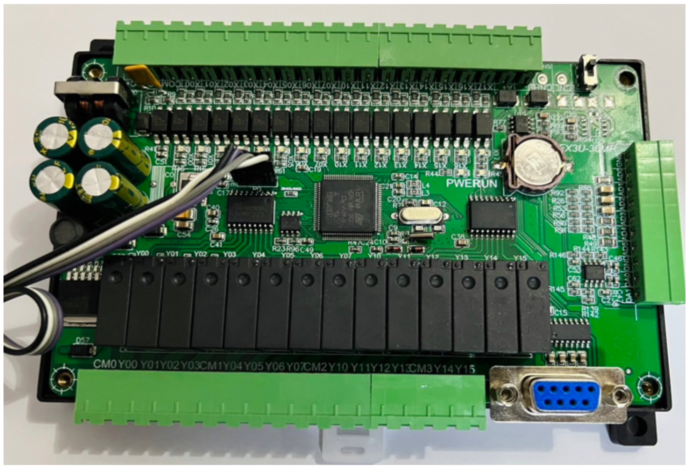


Fig. 4. Typical COTS hardware

Figure 5 shows the simplified internal architecture of this Mitsubishi clone, which is comparable to the basic ATE architecture in Figure 2. Repurposing COTS PLC hardware and software for ATE can be a viable and cost-effective solution for many applications, especially when flexibility, customization, and broad availability are required. However, it also requires careful planning, custom development, and thorough testing to ensure that the resulting ATE system meets the specific testing needs and performance requirements. The FX3U-30MR clone employs Cortex-M3, a low-cost and power-consuming 32-bit processor core. Despite the variety of user workloads and the wide range of needs, the STM32 series retains pin and peripherals configuration compatibility throughout the board [12]. The FX3U-30MR clone employs a STM32VET6 microcontroller based on an ARM Cortex-M3 core optimized for low-cost, low-power, and high-performance embedded applications [13].

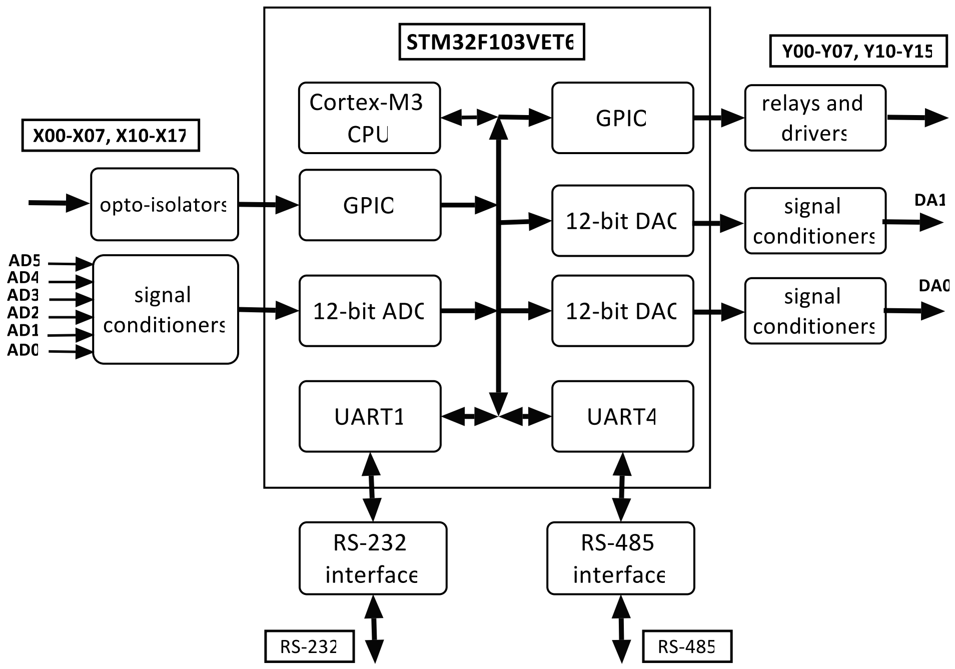


Fig. 5. COTS block diagram

In the industrial sector, inspection of PLC functions still requires experts and knowledge of programming to test PLC functions, which are difficult to use, such as programming with Structured Text, etc. Moreover, the tools used for testing on the market are expensive. Thonnessen *et al.,* [14] present an approach to the Hardware in the Loop (HiL) specification that allows PLC programmers to write PLC test cases in the Structured Text language specified by IEC 61131-3. However, for non-PLC programmers, this approach is nontrivial. Vince *et al.,* [15] implemented the Arduino-based PLC Universal HiL system, which provided basic plant simulations for a PLC system. The hardware consists of an Arduino Mega 2560 board and a custom-designed PCB that provides interfacing with industrial signals. This somewhat adds complexity and cost to the Universal HiL system. Therefore, this paper aims to design a low-cost PLC functional testing system using hardware and software components in the COTS components that are easy to use for non-PLC programmers.

The main contributions of this research are:

i. Low-cost and COTS: A low-cost PLC function testing development uses the COTS components. The hardware devices are inexpensive and readily available. The researcher uses the FX3U-30MR clone board as the tester. This board is repurposed by changing the usage from a PLC to the tester used to test the functionality of the PLC.
ii. Easy-to-use: The researcher provides a user-friendly development environment for test engineers. These environments do not require knowledge of PLC languages or computer programming languages. The researcher uses the Excel VBA program to provide test program development.

With the proper design, a COTS can be a good candidate for low-cost ATE for small firms. In Section 2, the proposed tester design is elaborated. To confirm that the tester design conforms to functional testing practices, ARCADIA/Capella is used as the method and tool. The results of the design are shown in Section 3. In Section 4, the outcomes of this research are concluded, and different perspectives are discussed.

2. Tool and Methodology

The testing method used in this research falls into the data-driven testing approach, as in Ramler *et al.,* [16]. Test engineers can formulate test programs based on natural language in Domain-Specific Language (DSL) similar to that described in Winkler *et al.,* [17] and Wiechowski *et al.,* [18] without knowing details on the underlying computer languages used by the development tools.

2.1 Requirements of the Low-Cost Tester

The requirements of the proposed low-cost tester are defined as:

i. The tester shall be employed to verify that the PLC performs the intended logical operations accurately.
ii. The tester shall assess the real-time performance of the PLC.
iii. The tester shall evaluate the robustness of the PLC when subjected to unexpected inputs or faults.
iv. The data shall be logged during testing, and it can be effectively stored and analysed.
v. In addition, the following attributes will be investigated:
vi. The tester should be scalable and adaptable.
vii. Cost-effectiveness and the cost implications should be maintained.
viii. A user-friendly interface for configuring and running tests is preferable.
ix. The tester should be seamlessly integrated into the PLC development cycle.
x. The tester should follow regulatory compliance.
xi. Improvement and expandability should be achieved.

2.2 ARCADIA and Capella

Over recent years, a model-based approach has been used in engineering and manufacturing instead of the traditional document-based method. Model-based systems engineering (MBSE) [19] is an approach used in engineering to design, simulate, and test systems using models before implementing them in hardware or software. It brings many benefits to the engineering and manufacturing sectors, such as improved communication between designers and customers. Model-based systems enhance the ability to capture, analyse, share, and manage the information associated with the complete specification of a product. ARCADIA [20] is a system engineering methodology for analysing, designing, and developing systems. It emphasizes architecture-centric engineering and model-based system development. ARCADIA [21] is becoming popular and increasingly expanding its application across several industrial fields to model various systems at different stages of their development and from other points of view, such as the automotive industry, aerospace systems, transportation, rail systems, etc. Capella [21], which supports the ARCADIA method, is a modelling and system engineering tool designed to develop complex systems and software. To ensure that the resulting ATE system meets the specific testing needs and performance requirements, the open-source Capella tool is used to design this low-cost automatic tester of PLC functions.

2.3 Hardware Design
2.3.1 Operational analysis

In this analysis, user requirements are analysed and documented. The operational actor is a test engineer who wants to write a test program, run the test program on the tester, and check the test results, as in Figure 6. The tester accepts the test program and executes the test program by sending the stimuli to the UUT to exercise the PLC functions. The responses from the UUT are sent to the tester, which checks them according to the comparison logic set in the test programs. The checked results are then sent back to the test engineer, who will verify the test results.

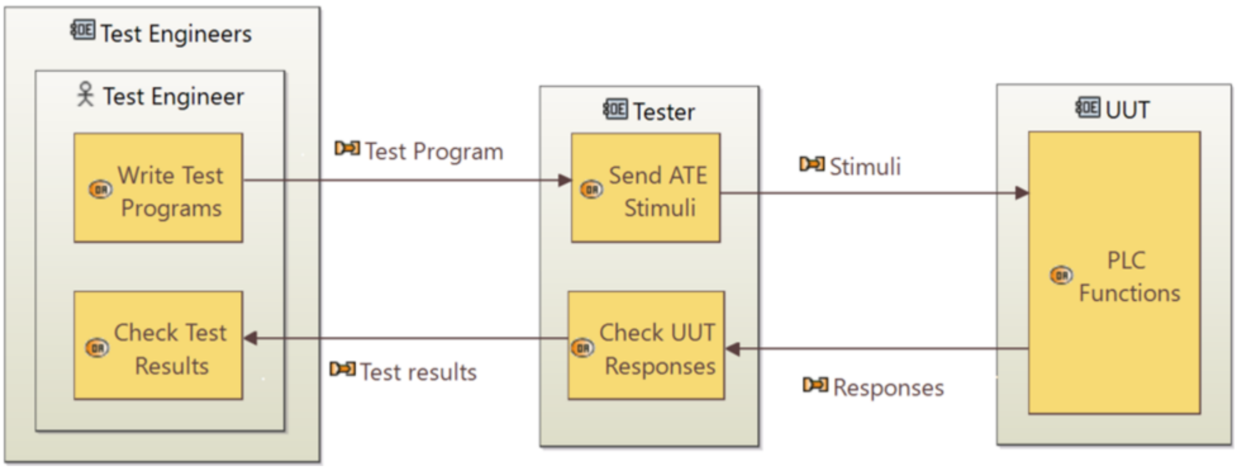


Fig. 6. Operational Architecture Diagram

2.3.2 System analysis

This analysis identifies the main functions of the systems (see Figure 7). The tester sends the test stimuli given by the test engineer in the test program, comparing responses from the PLC with the expected responses, and sending the test results to the test engineer. The engineer performs two main functions: writing the test program and verifying the test results.

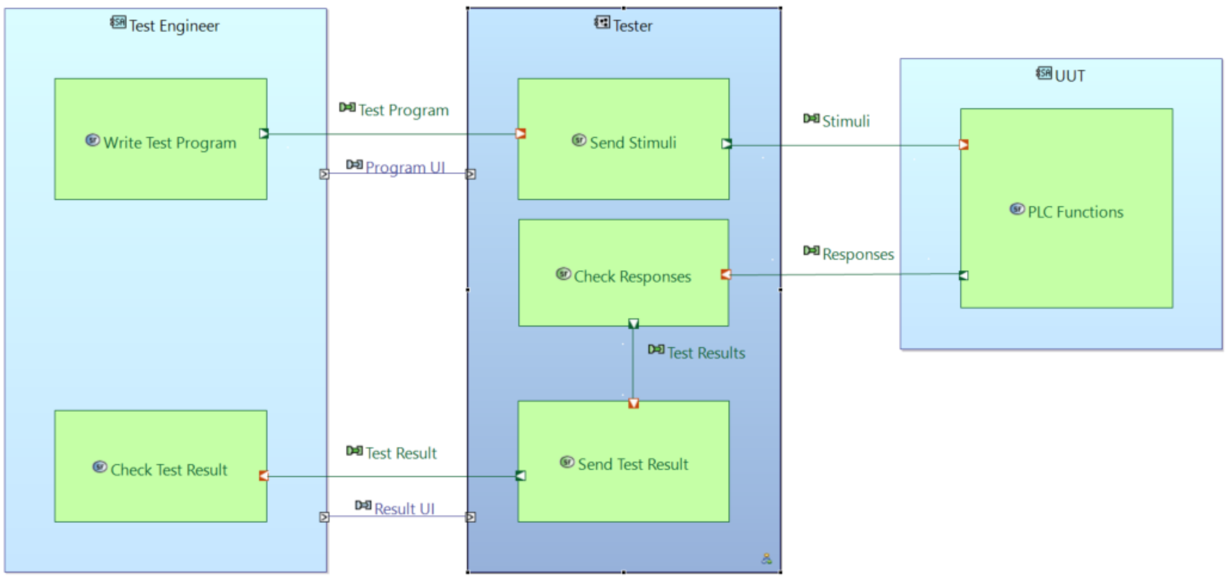


Fig. 7. System Architecture Diagram

2.3.3 Logical architecture

The logical architecture shows the allocation of functions to logical components, as shown in Figure 8. Each function is allocated to logical components. For example, the function “Check Response” is achieved by reading the response, checking responses, and notifying the test. Some non-functional requirements, signal isolation, and conditioning were added to complete the logical system.

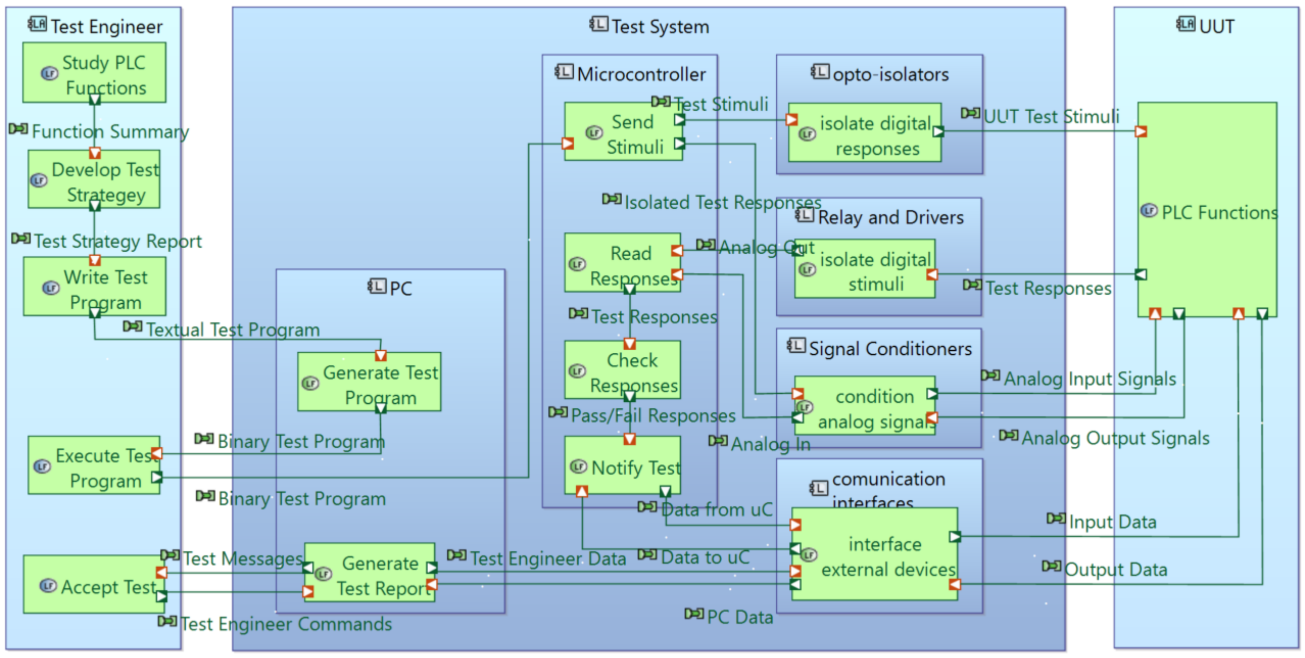


Fig. 8. Logical Architecture Diagram

2.3.4 Physical architecture

The physical architecture is the allocation of logical components and functions to physical components, as shown in Figure 9. The generation of physical links between components was included. For example, the logical “Generate Test Program” function is identified as running Excel VBA, and Arduino tools on a PC in the physical architecture. The PC is connected to the tester via a RS-232 cable, while the test is linked to the UUT via a RS-485 cable.

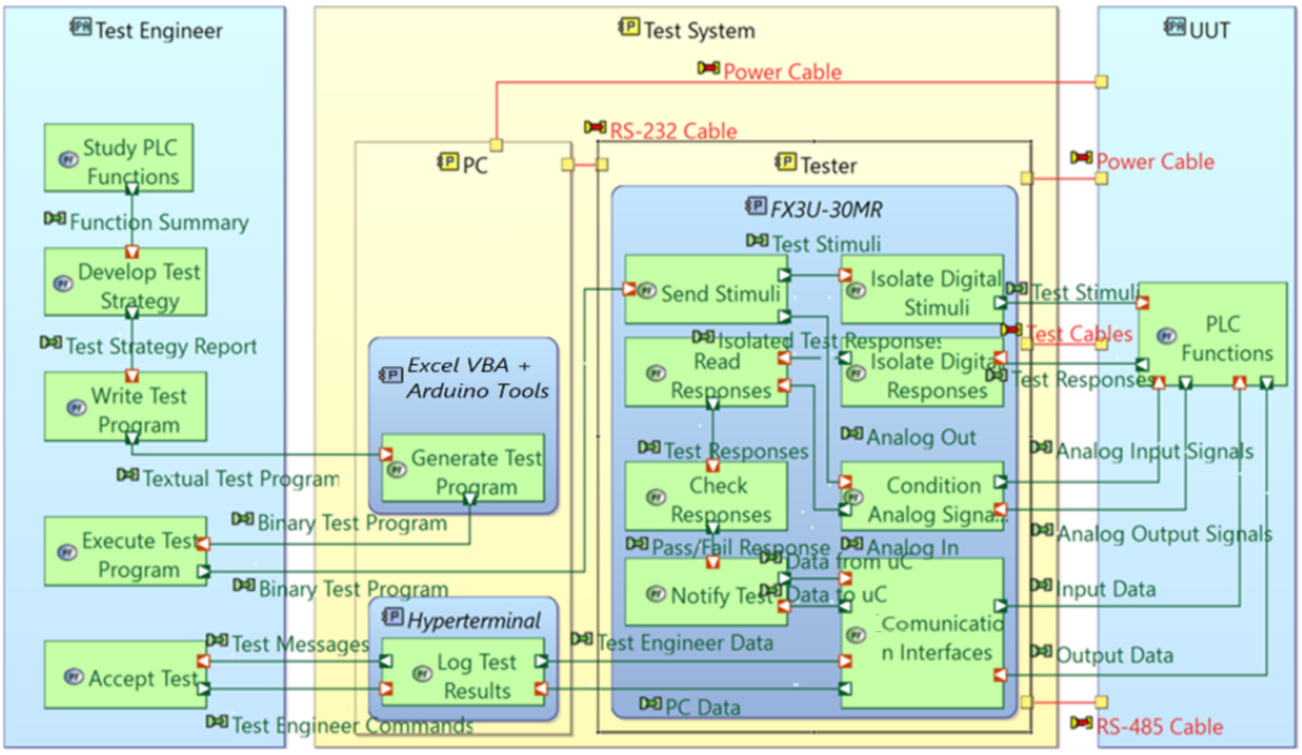


Fig. 9. Physical Architecture Diagram

2.4 Software design

The main concept of software design consists of two key points: open-source tools and ease of use. Most software tools used in this research are based on the open-source community, except Excel VBA, which is primarily available on most PCs. During the test program development phase, the test program is developed using the Excel environment. The test engineer can write the test program with a minimum learning curve since learning specific test languages is unnecessary. The test program development generates a test file in C (Tester.c), which is then included in the Arduino sketch. The Arduino commands are then used to compile the sketch into a tester executable file (Tester.hex), which is then loaded to the tester. In the execution phase, the CPU in the tester then executes its native codes, and the test results are then sent to the monitoring and logging program (Tera Term) for further investigation by the test engineer. In addition, the test program development flow is shown in Figure 10.

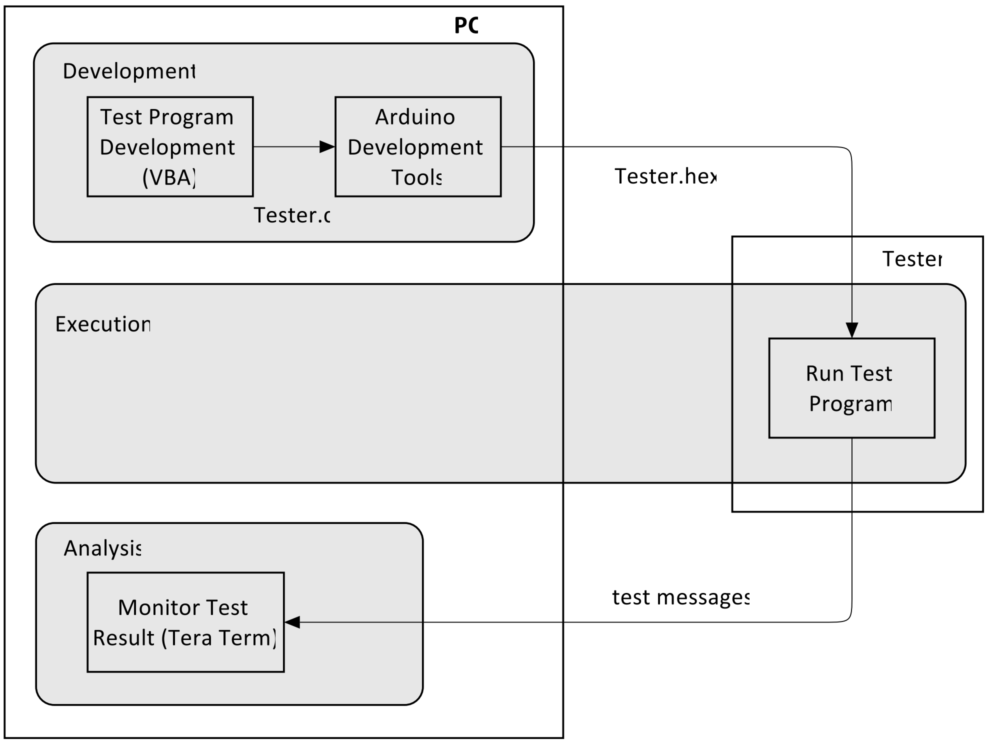

PC
Development
Test Program Development (VBA)
Arduino Development Tools
Tester.c
Tester.hex
Tester
Execution
Run Test Program
Analysis
Monitor Test Result (Tera Term)
test messages

Fig. 10. Test Program Development Flow

3. Results

This section presents the results of designing and developing the tester for PLC functions. It is structured as follows: Part 1 shows the hardware performance of the various items provided. It refers to the execution time of the STM32 microcontroller on a FX3U-30MR clone board. The details of the user interface are described in Part 2. Part 3 presents the sample results of its application in an air purge process, and Part 4 is the cost estimation of the tester. The final part is the comparison of this study with previous related research.

3.1 Hardware Performance

The hardware performance is obtained using C language test functions on a STM32 microcontroller. The C language test functions were written in the Arduino IDE. The execution time, measured by an oscilloscope, is shown in Table 1. For example, the execution time of “10 digital written statements” measured by an oscilloscope, 7.8 µs in Table 1, is shown in Figure 11(a). The details of the corresponding function, which consists of ten digital write statements as logic 1 and one digital write statement as logic 0, are shown in Figure 11(b).

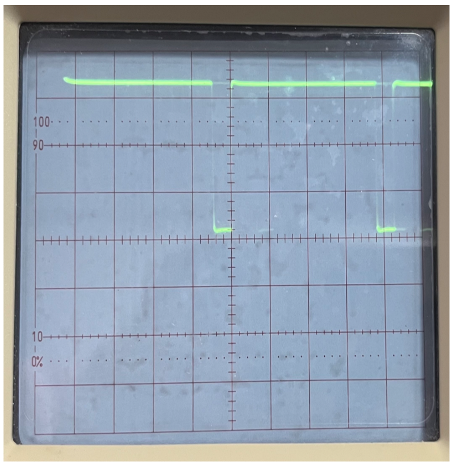
Table 1
Performance of the tester

Test Item	Execution Time (µs)
100 sequential order statements*	72
1000 sequential order statements*	720
2000 loops with a simple statement*	1550
150 conditional statements*	7.2
100 function call with one formal parameter and one return value*	76
10 analogue read statements	70

10 analogue write statements	26
10 digital read statements	8
10 digital written statements	7.8
Poll with Modbus protocol via RS232	2.3

Note The five items (*) are as follows: Jiang *et al.,* [22]

The test results shown in Table 1 are better than those provided by the Python test script used in Jiang *et al.,* [22]. For example, the "100 sequential order statement's execution times" of the C function and Python test scripts are 72 µs and 3.368 ms [22], respectively. A C function usually runs faster than a Python test script because C is a compiled language while Python is an interpreted one. In real-world applications, the maximum performance is 10 ms/pattern due to the tester's relay turn-on times. The switching time for mechanical relays can typically range from a few milliseconds to tens of milliseconds. Transistors can be used in place of relays to increase performance.



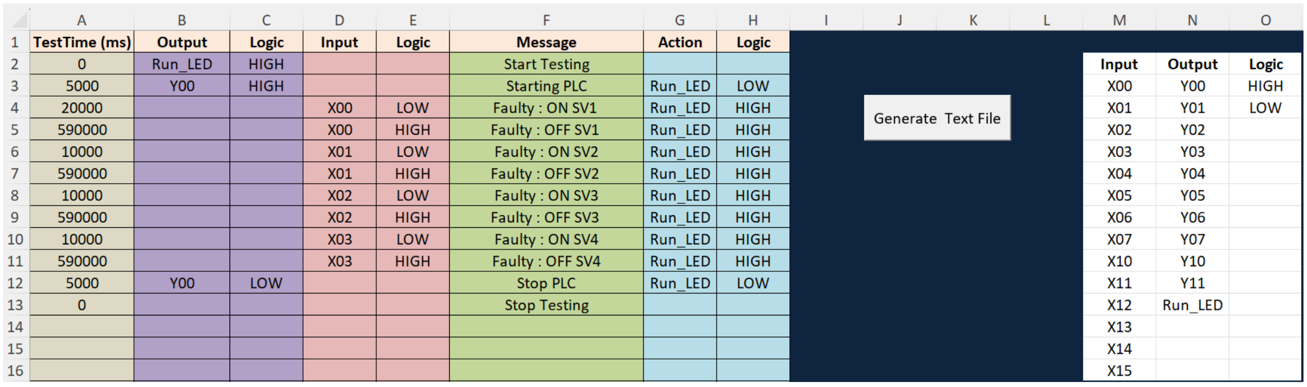
```
// loop of ten digital writes (logic 1) follow by
// one digital write (logic 0) which is used as a marker
void loop() {
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 1)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 2)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 3)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 4)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 5)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 6)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 7)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 8)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 9)
  digitalWrite(PD10, HIGH);   // turn PD10 on (No. 10)
  digitalWrite(PD10, LOW);    // turn PD10 off (zero marker)
}
```

(a) (b)

Fig. 11. The sample experimental information of hardware performance (a) The execution time (y-scale = 1V/division, x-scale = 2 us/division) (b) the C loop function of 10 digital written statements

3.2 User Interface Design and Software Development

Microsoft Excel is selected as a user interface for setting up the stimuli, expected responses, test time, and test messages. The user interface is shown in Figure 12. The user can select a setup value from dropdown lists in Excel.



	A	B	C	D	E	F	G	H
1	TestTime (ms)	Output	Logic	Input	Logic	Message	Action	Logic
2	0	Run_LED	HIGH			Start Testing		
3	5000	Y00	HIGH			Starting PLC	Run_LED	LOW
4	20000			X00	LOW	Faulty : ON SV1	Run_LED	HIGH
5	590000			X00	HIGH	Faulty : OFF SV1	Run_LED	HIGH
6	10000			X01	LOW	Faulty : ON SV2	Run_LED	HIGH
7	590000			X01	HIGH	Faulty : OFF SV2	Run_LED	HIGH
8	10000			X02	LOW	Faulty : ON SV3	Run_LED	HIGH
9	590000			X02	HIGH	Faulty : OFF SV3	Run_LED	HIGH
10	10000			X03	LOW	Faulty : ON SV4	Run_LED	HIGH
11	590000			X03	HIGH	Faulty : OFF SV4	Run_LED	HIGH
12	5000	Y00	LOW			Stop PLC	Run_LED	LOW
13	0					Stop Testing		
14								
15								
16								

M	N	O
Input	Output	Logic
X00	Y00	HIGH
X01	Y01	LOW
X02	Y02	
X03	Y03	
X04	Y04	
X05	Y05	
X06	Y06	
X07	Y07	
X10	Y10	
X11	Y11	
X12	Run_LED	
X13		
X14		
X15		

Fig. 12. Sample User Interface

The description of each field in Excel is in Table 2.

Table 2
The detail of the user interface

Field	Description
TestTime	A delay time that is used to control the timing of specific actions.
Output	The output channels (Y00-Y07, Y10-Y11, and Run-LED) of the FX3U-30MR clone board that control the input function of the tested PLC.
Output Logic	The logic (HIGH and LOW) used to control the behaviour of the tested PLC.
Input	The input channels (X00-X07 and X10-X15) of the FX3U-30MR clone board that receives responses from the tested PLC.
Input Logic	The expected logic (HIGH and LOW) used to compare with the responses from the Tested PLC.
Action	The output channels (Y00-Y07, Y10-Y11, and Run-LED) of the FX3U-30MR clone board that notifies the test engineer.
Action Logic	The logic (HIGH, and LOW) used to notify the test engineer.
Message	Messages are used to display information and error conditions.

3.3 Sample Test

In this part, the air purge process is used as an example to illustrate the automatic testing of PLC functions. The air purge process is used to remove non-condensable gases and primarily air from a refrigeration system. The example process consists of four solenoid valves that are controlled by a PLC. The control diagram of the air purge process is shown in Figure 13. The control system implements the following steps:

STEP 1: Send logic 1 to input X0 of the PLC. The PLC starts the air purge process.
STEP 2: The "SV.Airpurger No. 1" is ON and delayed for 600 seconds.
STEP 3: The "SV.Airpurger No. 2" is ON and delayed for 600 seconds.
STEP 4: The "SV.Airpurger No. 3" is ON and delayed for 600 seconds.
STEP 5: The "SV.Airpurger No. 4" is ON and delayed for 600 seconds.
STEP 6: repeats STEP 2-5
Note: If input X0 of the PLC is logic 0, the PLC stops the process (all solenoid valves are OFF).

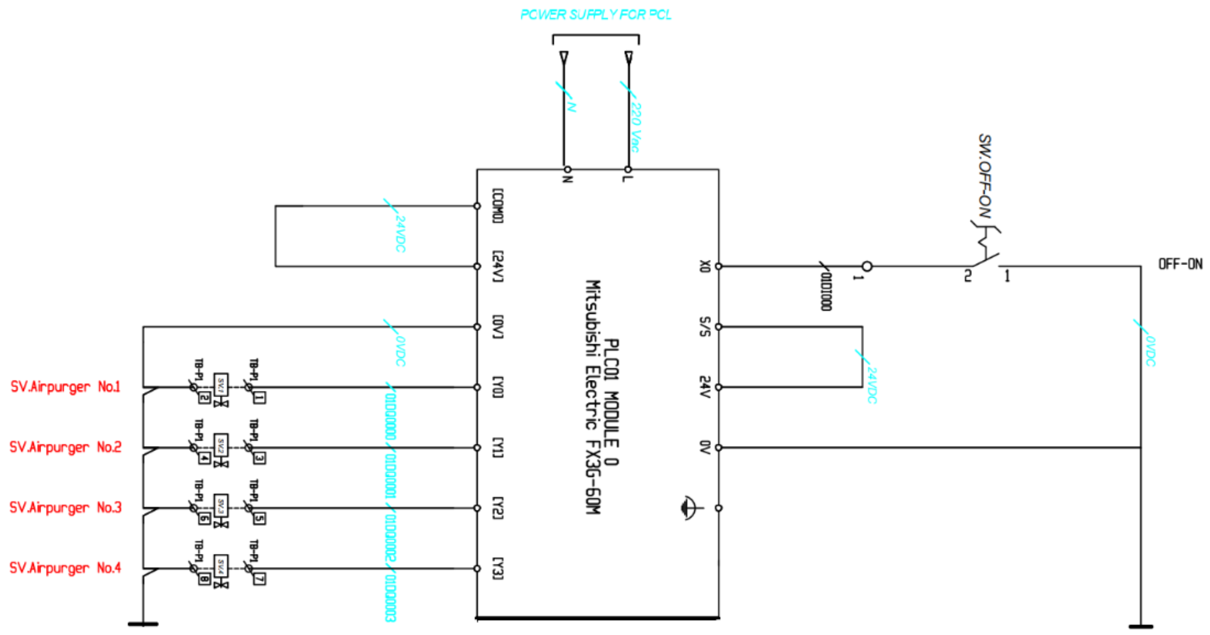


Fig. 13. Control Diagram of Air Purge Process

Figure 14 shows the PLC Test Bench. The hardware setup consists of the tester, a PLC under test (Mitsubishi FX3G-60MR), and a laptop computer.

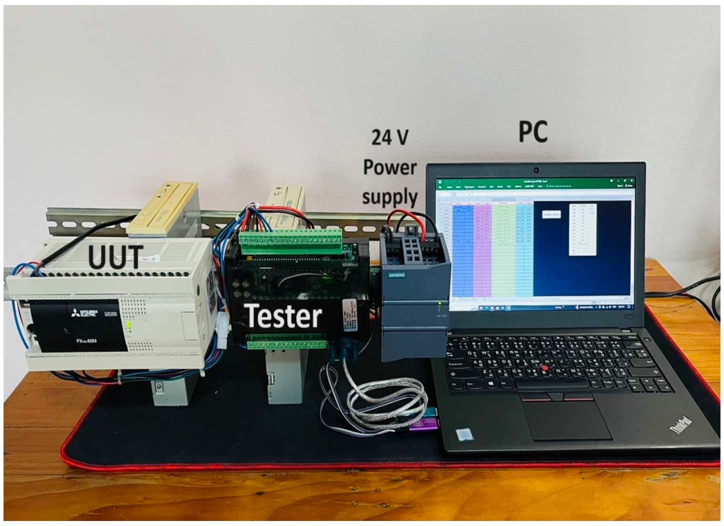


Fig. 14. PLC Test Bench

The wiring between the tester and PLC is shown in Figure 15.

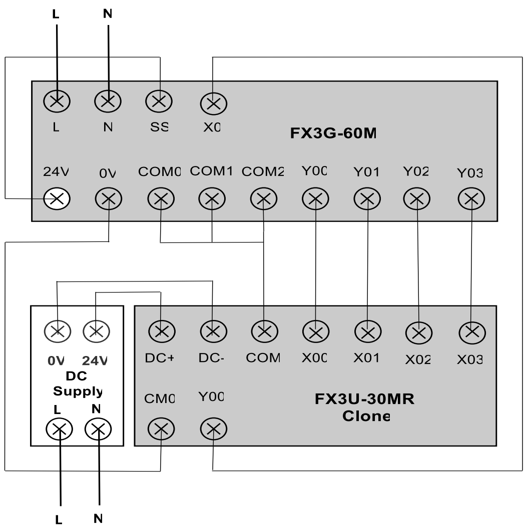
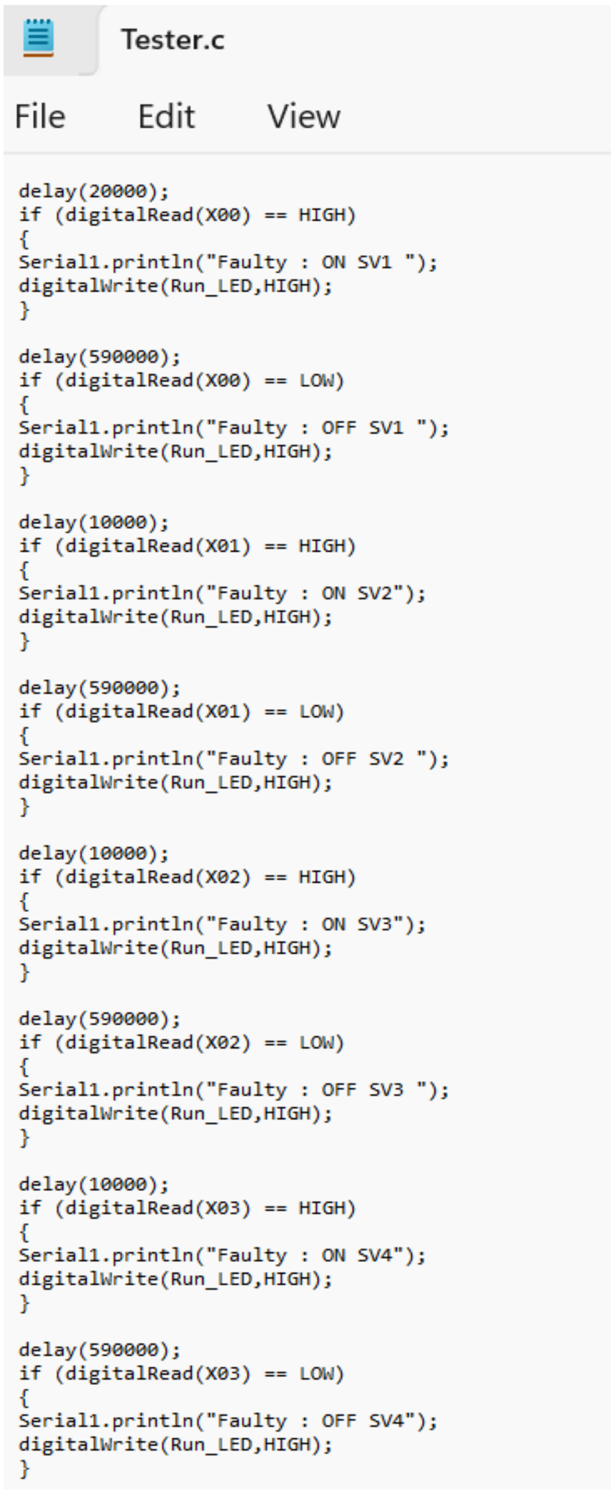


Fig. 15. Tester and PLC wiring

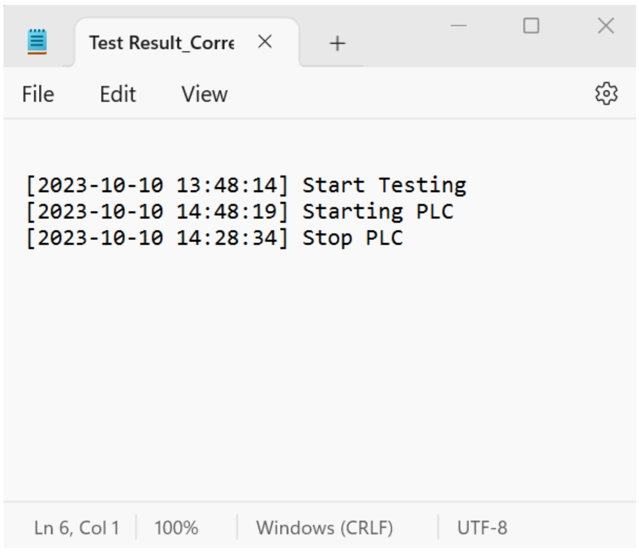
To illustrate, the X00 input channel of the FX3U-30MR clone board connects to the Y0 of the FX3G-60MR. The Y0 of the FX3G-60MR controls solenoid valve No.1. After the test engineer clicks the Generate Text File button in Excel to generate "Tester.c", a part of the C source file, as shown in Figure 16. Note that the opto-isolators block in Figure 5 reversed the input logic of the STM32 microcontroller on the FX3U-30MR clone board.



```
delay(20000);
if (digitalRead(X00) == HIGH)
{
Serial1.println("Faulty : ON SV1 ");
digitalWrite(Run_LED,HIGH);
}

delay(590000);
if (digitalRead(X00) == LOW)
{
Serial1.println("Faulty : OFF SV1 ");
digitalWrite(Run_LED,HIGH);
}

delay(10000);
if (digitalRead(X01) == HIGH)
{
Serial1.println("Faulty : ON SV2");
digitalWrite(Run_LED,HIGH);
}

delay(590000);
if (digitalRead(X01) == LOW)
{
Serial1.println("Faulty : OFF SV2 ");
digitalWrite(Run_LED,HIGH);
}

delay(10000);
if (digitalRead(X02) == HIGH)
{
Serial1.println("Faulty : ON SV3");
digitalWrite(Run_LED,HIGH);
}

delay(590000);
if (digitalRead(X02) == LOW)
{
Serial1.println("Faulty : OFF SV3 ");
digitalWrite(Run_LED,HIGH);
}

delay(10000);
if (digitalRead(X03) == HIGH)
{
Serial1.println("Faulty : ON SV4");
digitalWrite(Run_LED,HIGH);
}

delay(590000);
if (digitalRead(X03) == LOW)
{
Serial1.println("Faulty : OFF SV4");
digitalWrite(Run_LED,HIGH);
}
```

Fig. 16. Generated C Code

For example, the pass and fail of PLC function test results are shown in Figures 17 and 18, respectively. Fault insertions, such as shorted wires or broken wires, are widely used to check the correctness of test programs. Note that, in Figure 17, the correctness of the test program is validated by breaking wires (Y00-Y03). The notification messages are displayed according to settings in the message field in Excel.

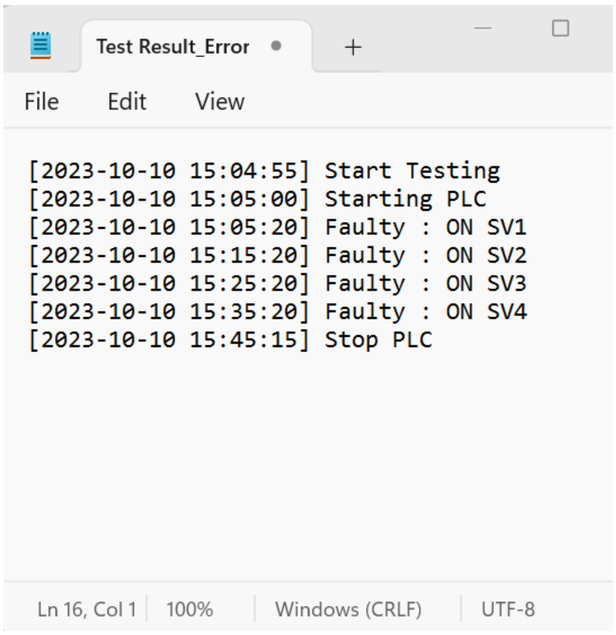


```
[2023-10-10 13:48:14] Start Testing
[2023-10-10 14:48:19] Starting PLC
[2023-10-10 14:28:34] Stop PLC
```

Fig. 17. The pass test response

3.4 Cost

The ATE test costs vary significantly based on many factors, such as the complexity of testing requirements, hardware, software, the signal type of system (analogue, digital, mixed signal, etc.), the testing environment, etc. The cost of the FX3U-30MR clone board and shipping is about 70 USD. About 40 USD is the cost of a DC Industrial DIN Rail 24 V power supply. A ST-Link and an USB to RS232 converter cost about 10 USD. The cost of the remaining accessories is about 20 USD. In this research, the total cost of the hardware, including the FX3U-30MR clone board, the 24 V DIN-Rail power supply, the ST-Link, and the USB to RS232 converter, is about 140 USD.



```
[2023-10-10 15:04:55] Start Testing
[2023-10-10 15:05:00] Starting PLC
[2023-10-10 15:05:20] Faulty : ON SV1
[2023-10-10 15:15:20] Faulty : ON SV2
[2023-10-10 15:25:20] Faulty : ON SV3
[2023-10-10 15:35:20] Faulty : ON SV4
[2023-10-10 15:45:15] Stop PLC
```

Fig. 18. The failure test response

3.5 Related Work

Based on the related work, Thonnessen *et al.,* [14] present PLC HiL testing using an extension of structured text, and Vince *et al.,* [15] implement the PLC universal HiL System based on ATmega2560. The comparison of this study with previous related research is shown in Table 3.

Table 3
Comparison of this study with related work

Criteria	Thonnessen *et al.,*	Vince *et al.,*	Present study
Application Area	Industry/ Education	Education	Industry/Education
Cost	Depend on PLC Brand	Low	Low
COTS	COTS	Semi-COTS	COTS
Test development languages	Structure text	C/C++	DSL and C/C++
Test Environment Package	PLC software tool	Arduino IDE	Microsoft Excel
Easy-to-use for non-PLC Programmer	No	---	Yes
Follow industry-specific regulatory	IEC 61131-3	No	No
Processing speed	Medium	High	High

4. Conclusions

Model-based approaches have already shown the benefits of systems engineering activities. This research's primary outcomes and conclusions are that the model-based tester design approach is a promising solution that can improve and address some of the limitations identified for system engineering design. The prototype tester was implemented by repurposing off-the-shelf PLCs that reduced the total cost. Compared to current testing approaches, where testing is based on high-cost testers, several benefits are possible:

i. For functionality testing, a COTS-based tester effectively simulates various input signals to test the PLC's functionality. Testing methods and algorithms that test engineers develop can be employed to verify that the PLC performs the intended logical operations accurately.
ii. For real-time performance evaluation, the tester can assess the real-time performance of the PLC, especially in critical applications. Benchmarking was used to measure the speed and responsiveness of the PLC under different scenarios.
iii. For fault tolerance and error handling, the tester can be programmed to simulate fault conditions and assess the PLC's ability to detect and handle errors. The tester can evaluate the robustness of the PLC when subjected to unexpected inputs or faults.
iv. For scalability and adaptability, the tester can be scaled to accommodate different PLC models and configurations. The tester can also accommodate changes in the PLC's programming or hardware.
v. For data logging and reporting, data can be logged during testing and effectively stored and analysed. The tester can generate comprehensive test reports, including pass/fail results and detailed performance data.
vi. For cost-effectiveness, the cost implications of using a COTS-based tester compared to other testing methods or ATE systems are less. In contrast, the tester does not impact the overall production process.

On the other hand, these preliminary evaluations have highlighted some limitations:

i. For the user interface, the tester cannot provide a user-friendly interface for configuring and running tests.
ii. For integration with PLC development, the tester cannot be seamlessly integrated into the PLC development cycle to facilitate rapid iteration and debugging. No tools or interfaces are needed to support this integration.
iii. In regulatory compliance, the tester does not meet industry-specific regulatory requirements for PLC testing.

In future research, instead of depending on VBA, a full open-source test software development environment should be developed using free software tools such as Python. In addition, the tester should be improved and expanded to accommodate evolving PLC technologies and testing needs, such as distributed testing. Finally, in order to achieve confidence testing, simulation tools such as Scilab should be considered.

In summary, the model-based testing approach is a promising solution that can improve the co-engineering activities. It provides a visual and interactive way to show system components and their interactions. However, applying the existing approach to repurposing commercially available PLCs reveals some limitations and possible improvements that need to be addressed by future studies.

Acknowledgment
The author would like to thank Sirayooth Company Limited (Thailand) for this research's great collaboration, a PLC, and an industrial-grade DC power supply.

References

[1] Rodzalan, Shazaitul Azreen, Noor Nazihah Mohd Noor, Maisarah Mohamed Saat, Nor Hazana Abdullah, Harcharanjit Singh, and Natrah Mohd Emran. "An Investigation of Present and Future Work Skills in Industry 4.0: Systematic Literature Review." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 28, no. 2 (2022): 356-371. https://doi.org/10.37934/araset.28.2.356371

[2] Mahariya, Satish Kumar, Awaneesh Kumar, Rajesh Singh, Anita Gehlot, Shaik Vaseem Akram, Bhekisipho Twala, Mohammed Ismail Iqbal, and Neeraj Priyadarshi. "Smart campus 4.0: Digitalization of university campus with assimilation of industry 4.0 for innovation and sustainability." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 32, no. 1 (2023): 120-138. https://doi.org/10.37934/araset.32.1.120138

[3] Soori, Mohsen, Behrooz Arezoo, and Roza Dastres. "Internet of things for smart factories in industry 4.0, a review." *Internet of Things and Cyber-Physical Systems* (2023). https://doi.org/10.1016/j.iotcps.2023.04.006

[4] Bedi, Harpreet Singh, Shekhar Verma, R. K. Sharma, and B. Singh. "The Concept of Programmable Logic Controllers and its role in Automation." *International Journal of Advanced Research in Computer and Communication Engineering* 4, no. 6 (2015).

[5] Stranges, Meredith KW, Saeed Ul Haq, Luis HA Teran, and William Veerkamp. "Factory acceptance tests: a comparative analysis to establish a baseline for testing data." *IEEE Industry Applications Magazine* 22, no. 2 (2016): 48-53. https://doi.org/10.1109/MIAS.2015.2459114

[6] Thönnessen, David. "Hardware-in-the-Loop testing of industrial automation systems using PLC languages." PhD diss., Dissertation, RWTH Aachen University, 2021, 2021.

[7] Minihan, John, Ed Schmidt, Greg Enserro, and Melissa Thompson. *Commercial Off-the-Shelf (COTS) Components and Enterprise Component Information System (eCIS)*. No. KCP-613-8421. Honeywell FM&T-Kansas City Plant, 2008. https://doi.org/10.2172/935135

[8] Sridhar, Nithya, Shubhankar Shobhit, Kaushik Das, and Debasish Ghose. "Model Based Control of Commercial-Off-TheShelf (COTS) Unmanned Rotorcraft for BrickWall Construction." *arXiv preprint arXiv:2103.12335* (2021).

[9] Khan, Sadeque Reza, Andrew J. Mugisha, Andreas Tsiamis, and Srinjoy Mitra. "Commercial off-the-shelf components (cots) in realizing miniature implantable wireless medical devices: A review." *Sensors* 22, no. 10 (2022): 3635. https://doi.org/10.3390/s22103635

[10] Ren, Luyong, and Xiaoyu Yu. "Hardware implementation of STM32 microcontroller-based indoor environment monitoring system." *Open Journal of Applied Sciences* 11, no. 9 (2021): 997-1008. https://doi.org/10.4236/ojapps.2021.119072

[11] Laski, Pawel Andrzej, and Mateusz Smykowski. "Using a development platform with an STM32 processor to prototype an inexpensive 4-DoF Delta parallel robot." *Sensors* 21, no. 23 (2021): 7962. https://doi.org/10.3390/s21237962
[12] Deng, Baoqing, Zhang Bo, Yuchuan Jia, Zengfa Gao, and Zhuoya Liu. "Research on STM32 Development Board Based on ARM Cortex-M3." In *2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT*, pp. 266-272. IEEE, 2020. https://doi.org/10.1109/ICCASIT50869.2020.9368860
[13] Al-Shareeda, Mahmood A., and Selvakumar Manickam. "IoT Technology and STM32 Microcontroller Based Design of Smart Suitcase."
[14] Thönnessen, David, Niklas Reinker, Stefan Rakel, and Stefan Kowalewski. "A concept for PLC hardware-in-the-loop testing using an extension of Structured Text." In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1-8. IEEE, 2017. https://doi.org/10.1109/ETFA.2017.8247580
[15] Vince, Tibor, Matej Bereš, Stanislav Makiš, and Dmytro Mamchur. "Plc universal hardware in the loop system based on atmega2560." In *2020 IEEE Problems of Automated Electrodrive. Theory and Practice (PAEP)*, pp. 1-4. IEEE, 2020. https://doi.org/10.1109/PAEP49887.2020.9240867
[16] Ramler, Rudolf, Werner Putschögl, and Dietmar Winkler. "Automated testing of industrial automation software: practical receipts and lessons learned." In *Proceedings of the 1st International Workshop on Modern Software Engineering Methods for Industrial Automation*, pp. 7-16. 2014. https://doi.org/10.1145/2593783.2593788
[17] Winkler, Dietmar, Kristof Meixner, and Stefan Biffl. "Towards flexible and automated testing in production systems engineering projects." In *2018 IEEE 23rd international conference on emerging technologies and factory automation (ETFA)*, vol. 1, pp. 169-176. IEEE, 2018. https://doi.org/10.1109/ETFA.2018.8502650
[18] Wiechowski, Norbert, Thomas Rambow, Rainer Busch, Alexander Kugler, Norman Hansen, and Stefan Kowalewski. *Arttest–a new test environment for model-based software development*. No. 2017-01-0004. SAE Technical Paper, 2017. https://doi.org/10.4271/2017-01-0004
[19] Beery, Paul, and Eugene Paulo. "Application of model-based systems engineering concepts to support mission engineering." *Systems* 7, no. 3 (2019): 44. https://doi.org/10.3390/systems7030044
[20] Voirin, Jean-Luc. *Model-based system and architecture engineering with the arcadia method*. Elsevier, 2017.
[21] Baron, Claude, Lorenzo Grenier, Vitalina Ostapenko, and Rui Xue. "Using the ARCADIA/Capella Systems Engineering Method and Tool to Design Manufacturing Systems—Case Study and Industrial Feedback." *Systems* 11, no. 8 (2023): 429. https://doi.org/10.3390/systems11080429
[22] Jiang, Chongwu, Bin Liu, Yongfeng Yin, and Chang Liu. "Study on real-time test script in automated test equipment." In *2009 8th International Conference on Reliability, Maintainability and Safety*, pp. 738-742. IEEE, 2009. https://doi.org/10.1109/ICRMS.2009.5270090