

Architecture of Product Digital Scales Modbus Protocol-Based

Esa Fauzi^a, M. Fajri^b, Yani Iriani^c, ^{a,b}Department of Informatics Engineering, Faculty of Engineering, ^cDepartment of Industrial Engineering, Universitas Widyatama, Bandung, Indonesia, Email: esa.fauzi@widyatama.ac.id, nurrohmanfajri90@gmail.com, yani.iriani@widyatama.ac.id

PT "X" is one of the manufacturers of cigarettes that have 3 production lines and 3 different products. During the master packing process, the three products will be weighed in 3 different lines, where each product has weight references (standard). This process is less effective when considering that a new product has a different weight are requires different production and manufacturing in turn. When the product does not comply with the standard, a rejector will work to separate the product, which will then be checked by the operator. This is usually done manually by using the manual scales 1. Based on this, digital scales software is needed. The research methods used in this research are library research method, field research method, and laboratory research method. This study aims to design and create a digital weighing software using the ModBus protocol to communicate data from the scales to the program and to read the barcode using a scanner through RS-232 serial communication. This new system will also use the jet printer to print the weight of the products with communication over UDP (User Datagram Protocol). The results of the design and testing that was carried out on more than 1000 products, is that the software was able to perform its function properly and give information about the weight of the product. After the testing phase, the program was installed in the production line using all the actual devices that were used for the production process.

Key words: *ModBus protocol, RS-232 serial, UDP protocol.*

Introduction

PT "X" is one of the manufacturers of cigarettes that have 3 production lines and 3 different products. During the master packing process, the three products will be weighed in 3 different lines, where each product has weight references (standard). This process is less effective when considering that a new product has a different weight are requires different production and manufacturing in turn. When the product does not comply with the standard, a rejector will work to separate the product, which will then be checked by the operator. This is usually done manually by using the manual scales.

Based on this problem, an equipment scale that can measure 3 different product weight references is needed. Programmable digital scales are one solution. Digital scales typically have many advantages such as: mass scales are lighter and the measured load measurement results are more accurate. In addition, this scale allows for better data processing and there are other interfaces that support the whole system so it can facilitate the operation of the instrument itself (Manege et al., 2017).

The purpose of this research is to design and build a weight gauge (scale) digitally based on the modbus protocol. Modbus protocol is a data communication that is widely used in industrial control systems (Tabaa, 2018). Modbus is an industrial protocol standard that has been used for many years. Modbus ASCII and Modbus RTU are relatively simple serial protocols that use EIA-232 (RS-232) or EIA-485 (RS-485) to transmit data packets while Modbus TCP exchanges data over Ethernet TCP/IP (Nguyen and Huynh, 2016); (Zhu, 2013). The Modbus protocol was created by a PLC company named Modicon in 1979 and until now it is one of the standard communication protocols used in Building Management, Industrial Process Automation etc (Song and Zhang, 2014).

In this new system, a barcode scanner will be installed that will communicate with the program via RS-232 serial communication. RS-232 is a serial interface in as single-ended standard which has much characteristic, such as more communication distance in a low-speed serial communications, moderate price and good practicality of the system (Han, and Kong, 2011). The output of the digital scale prototype will be equipped with a jet printer to print the weight of the product. The communication between the jet printer and the digital scale prototype is through UDP (User Datagram Protocol) (Postel, n.d).

Methodology

The method used in software development in this study is the Waterfall Development Model (Royce, 1987). In this paper the steps taken are: Problem Identification, Analysis of Old and New Systems, Making Block System Diagrams, State Diagrams, Communication Scales and making Communication Printers

1). Problem Identification

There are several problems that exist in this system, namely:

- a. There are 3 lines that have the same functions and devices
- b. This system is not scalable. Addition of every product means additional production lines and operators.

2). Analysis of Old and New Systems

In the old system each product is weighed with one special scale set in the upper and lower limit values. When the product is not within the limits, the rejector will work to separate the product, which will then be checked by the operator. There are 3 production lines with the same function and the same device. The difference in each line is only the upper and lower limit value of the scales. In the new system there is an important main process, that is, the product selection process. The new system must be able to decide what product is being weighed and then choose the weight limit that is in accordance with the product being weighed.

Each product has a unique barcode that is different for each product variant. So this barcode can be used as a selection process and because the limits on the scales cannot be set when the scales are running, the decision-making process about the weight of the product is appropriate for the program. Therefore, the program must communicate with I / O devices so that when the product weight is not appropriate, the program sends a reject signal to I / O then the rejector will separate the product to a different line.

To read the barcode, a barcode scanner will be installed that will communicate with the program via RS-232 serial communication. Likewise communication with I / O, will be done via RS-232 serial. As for data communication from the scales to the program, it will use the ModBus protocol. The new system will also print out to print the weight of the product being weighed. The type of printer that will be used is a jet printer with communication through UDP (User Datagram Protocol).

Fig. 1. Production Line On The Old System

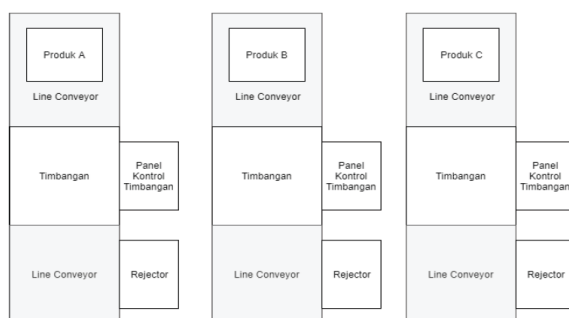
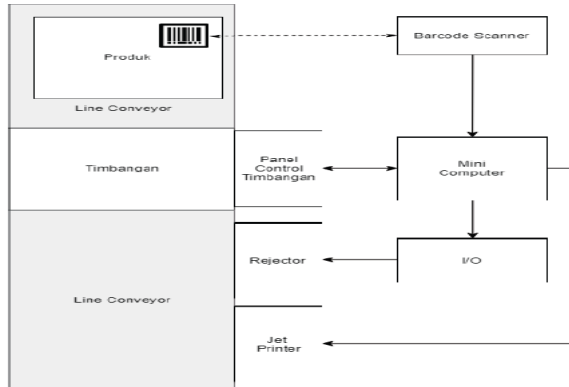


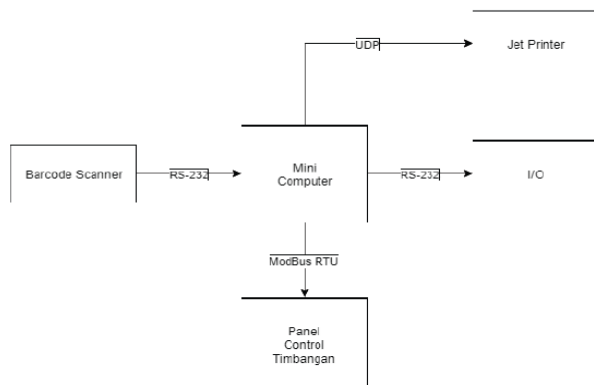
Fig. 2. New System Process Diagram



3). Block Diagram System

The following is a block diagram design of the system that will be created along with the communication protocol used between devices.

Fig. 3. Block Diagram Sistem

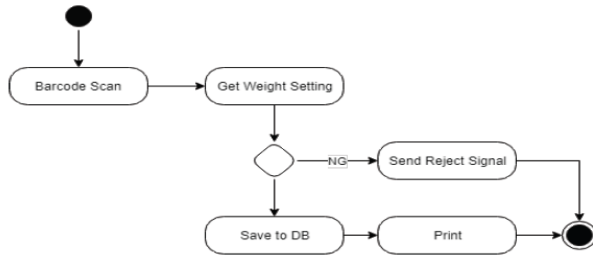


This program is connected with 3 types of communication interfaces namely UDP, RS232 and ModBus RTU. The printer uses the UDP protocol, the scales use ModBus communication while the barcode scanner and I / O use the RS232 communication protocol.

4): State Diagram

The following is a state diagram that describes the steps from the start of the product and scanned by the barcode scanner until the process is complete.

Fig. 4. State Diagram System



When the product on the conveyor passes through the barcode scanner, the barcode scanner will read the identification number of the product. After that, the program will retrieve the set point weight data previously inputted by the production operator. After the product passes the conveyor weighing scale and the machine weighs the product data, the data is sent to the program then the program will compare the data set point with the actual weight data. If the weight is not suitable, the program will send a reject signal to I / O and the product will be released from the conveyor by the rejector. If the product weight is appropriate, then the actual weight of the product will be stored in the database and then the weight will be printed on the product packaging box.

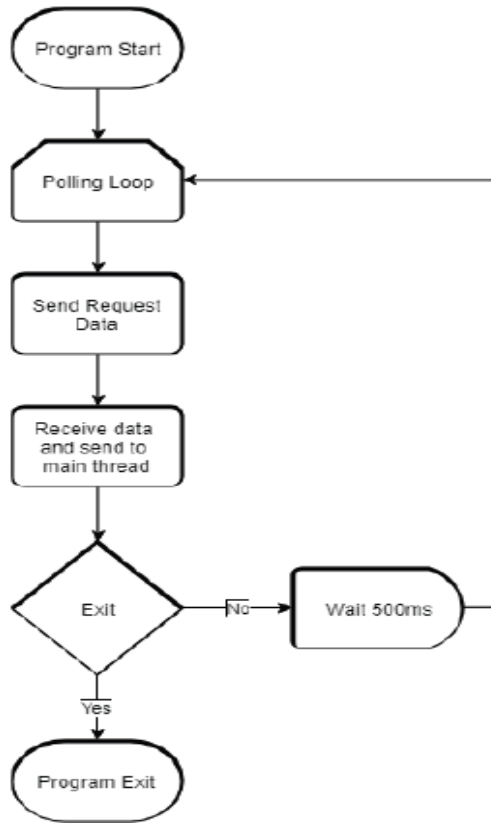
3). Communication Scales

The modbus protocol is used to retrieve the product weight value. The parameters used to request heavy data are as follows:

- Register: 3001
- Address: 5

After the scales receive a request from the program, the scales will respond to the heavy scores with the Big-Endian format. Following is the process of polling data from the program to the scales.

Fig. 5. Scales Data Flowchart Polling



4). Printer Communication

There are 4 commands sent by the program to the printer using the UDP protocol, namely:

a. Start printing

0x1B 0x02 0x11 0x1B 0x03 0xEA

b. Stop printing

0x1B 0x02 0x12 0x1B 0x03 0xE9

c. Delete data

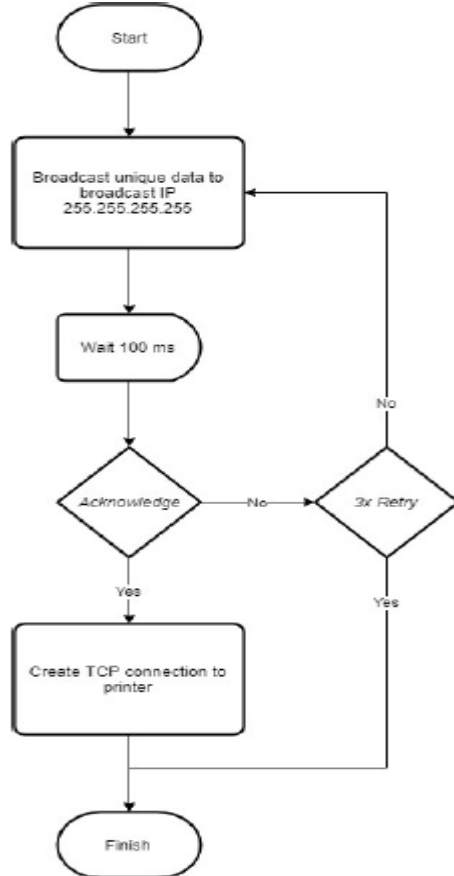
0x1B 0x02 0x1D 0x00 0x00 0x1B 0x03 0xDE

1). Send data

0x1B 0x02 0x1D <Data Length> 0x00 0x1B 0x03 <Data>
<Check Sum> 0xDE

In addition to the four commands, to begin connecting the program with the printer discovery process is needed. This process functions to find the IP of a printer on a LAN network. The following is a flowchart from the discovery process of the printer.

Fig. 6. Flowchart Printer Discovery



The discovery process begins with the program sending unique commands that are recognized by the printer to IP broadcast. Broadcast IP depends on the network, on the network here IP broadcast is 255.255.255.255. After that the program will wait for a reply from the printer for 100ms. If there is no reply, the program will send the data broadcast again and the process will repeat 3 times. If after 3 times there is still no reply, the program will declare the printer not found. If the printer in the network receives data broadcast, the printer will send an acknowledgment signal that will be received by the program.

When the program receives this signal, the program records the IP of the signal sender. After that, the program will establish a TCP connection to the printer which will then be used for data communication with the printer.

Results and Discussion

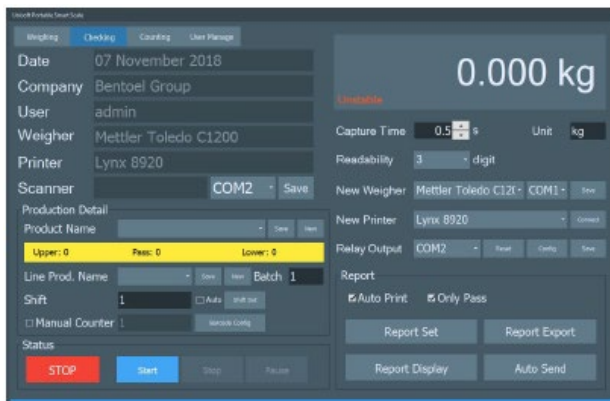
A. Implementation

System implementation is the stage of system implementation that will be carried out if the system is approved including programs that have been made at the system design stage to be ready for operation. Implementation for this system will use the Object Pascal programming language with IDE Delphi 2010. The program will run on a mini computer with the Windows 10 operating system.

1). Interface

Implementation of the interface is a stage in meeting user needs, in interacting with computers. A good interface facility helps users to understand the processes being carried out by the system and can improve system performance. The following is the display interface created for this system.

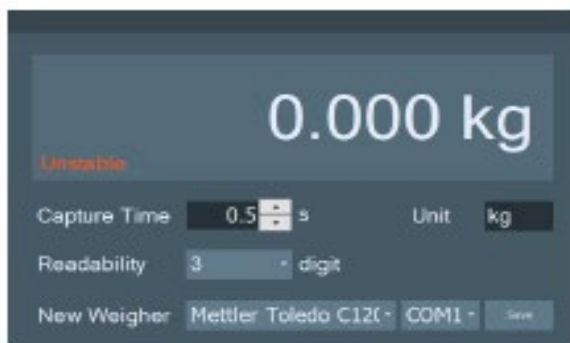
Fig. 7. Program Interface Flowchart



2). Device Communication

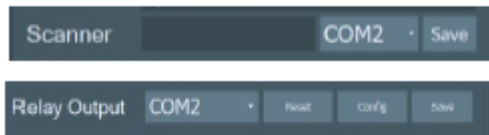
Scales do not actively provide data to the program, so the polling process is carried out. The polls conducted have an interval of 0.5 seconds. Polls can be set by users. In addition, settings for the number of comma numbers are added and the RS232 or com port is chosen for communication with the printer.

Fig. 8. User Interface and setting of scales



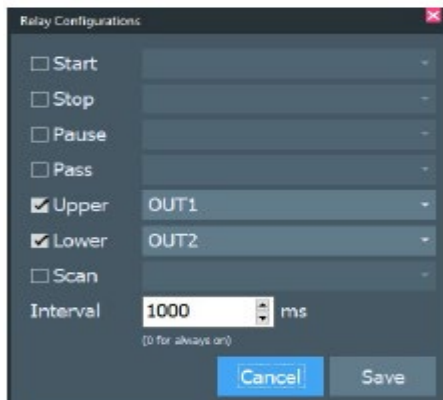
For barcode scanner and relay output communication such as the scale communication, a setting is added to select the com port to be used.

Fig. 9. UI For Barcode Scanner Settings and Relay Output



For setting the relay output itself, a special configuration button configuration is added to change the relay channel for each function. In addition, a reset button is added, to disable all channel relays

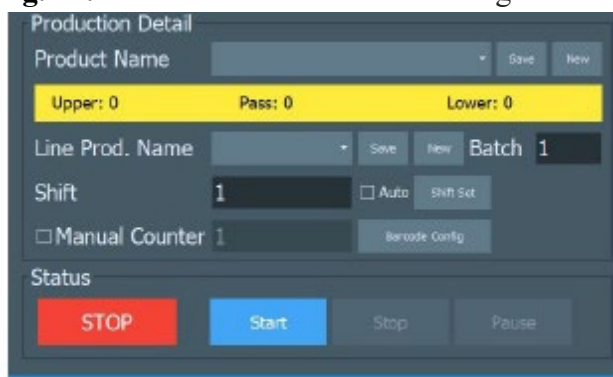
Fig. 10. UI Dialog for Relay Output Function Settings



3). Software

In the software, some product data is added to facilitate data collection on weighing results. The start, stop and pause buttons are added to start the weighing process by the production operator.

Fig. 11. Production Data Form and Program Status



For scales, barcode scanners and I / O that have the RS232 protocol are required to choose the port used. As for the printer, the connect button is added to connect the program to the printer.

When the connect button is pressed, the first program will perform a discovery process. This process is done by sending unique data to IP broadcast. IP broadcast is a special address used to send data to all IPs registered in the network. If the printer on the network gets this data, then the printer will send acknowledgment data to the IP program. That way the program can find out the printer's IP without needing to be set beforehand by the user

B. System Testing

The printer is tested using products manufactured from main line. There are 3 products that will pass this weighing process.

TABLE I: PRODUCT LIST, BARCODE CODE AND WEIGHT

No	Product Name	Barcode Code	Weight
1.	Dunhill Filter	8998130184015	17.6 kg
2	Dunhill Mild	8998127514123	16.4 kg
3.	Dunhill Light	8998159439394	17.1 kg

Product data is entered through the program. The program can correctly choose the product to be weighed and equalize the actual product weight with the reference data that has been input (Prakash et al., 2019; Yazici, 2018). Following are the test scenarios along with the results performed.

TABLE I: SOFTWARE TESTING

N o.	Testing Scenario	Expected Output	Test Result	Status
1.	Displays the actual weight of the scale	The weight displayed is in accordance with	Actual Weight: 1kg Display Weight: 1kg	Valid

		the weight weighed		
2	Read barcode data	Barcode data that is read must be in accordance with the label on the product barcode	Barcode Data: 8998130184015 Read Data : 8998130184015	Valid
3.	Detects the IP of the printer	The program must be able to detect the printer's IP through a discovery process	IP Printer 192.168.10.4 IP Detected: 192.168.10.4	Valid
4.	Filter the appropriate product weight	The program must pass the product of the appropriate weight	Product Weight: 17.6kg Range valid: 17.4 - 17.8 kg Program Result: Pass	Valid
5.	Filter product weight that is not suitable	The program must be able to filter products	Product Weight: 16.2kg Range valid: 17.4 -	Valid

		with an inappropriate weight	17.8 kg Program Result: Fail	
--	--	------------------------------	------------------------------------	--

Based on the results of testing, on approximately 1000 products where the program works perfectly, it can be deemed that it performs its function properly. After the testing phase, the program is installed in the production line utilising all actual devices for the production process.

Fig. 12.Implementation of the Production Line.



Conclusion

From the results of the design, manufacture and testing of digital scale measuring devices based on the Modbus Protocol:

- a. Product identification to distinguish weight as a reference for each product is completed by barcode identification using a barcode scanner. Scales can be connected to the program using the modbus protocol.
- b. The process of checking the weight is done by comparing the data input into the database with the actual data scales read by the program. If there is inappropriate data, the program will send reject signals through I / O so that products that do not match will be filtered.
- c. The results obtained are in accordance with what was planned. The production process runs smoothly even though it only uses 1 line so the company in fact, can make cost savings

REFERENCES

- Han, X. and Kong, X. (2011). The designing of serial communication based on RS232. Proc. - 2010 1st ACIS Int. Symp. Cryptogr. Netw. Secur. Data Min. Knowl. Discov. E-Commerce Its Appl. Embed. Syst. CDEE 2010, pp. 382–384.
- Manege, P. M. N., Allo, E. K. & Bahrin, (2017). Digital Scales With Microcontroller Based 20Kg Capacity, 6(1): 57–62, 2017.
- Nguyen, T. S. & Huynh, T. H. (2016). Design and implementation of modbus slave based on ARM platform and FreeRTOS environment. Int. Conf. Adv. Technol. Commun., 2016-Janua(November): 462–467.
- Postel, J. (n.d). User Datagram Protocol.
- Prakash, G., Darbandi, M., Gafar, N., Jabarullah, N.H. & Jalali, M.R. (2019). A New Design of 2-Bit Universal Shift Register Using Rotated Majority Gate Based on Quantum-Dot Cellular Automata Technology, International Journal of Theoretical Physics, <https://doi.org/10.1007/s10773-019-04181-w>.
- Royce, W. W. (1987). Managing the development of large software systems: concepts and techniques,” Proc. 9th Int. Conf. Softw. Eng., no. August, pp. 328–338.
- Song, L. & Zhang, H. (2014). Implementation of Modbus Communication Protocol Based on ARM Coretx-MO.
- Tabaa, M. (2018). Industrial Communication based on Modbus and Node-RED ScienceDirect ScienceDirect ScienceDirect Industrial Communication based on Modbus and Node-RED Industrial Communication based on Modbus and Node-RED. Procedia Comput. Sci., 130(September): 583–588.
- Yazici, O. (2018). Erosion Surfaces of the Land Between Iznik-Mekece (Turkey). International Journal of Geography and Geology, 7(1): 1-13.
- Zhu, S. (2013). Design of Monitoring Control System Based on MODBUS for Transformer Measurement Equipment. Proc. 2nd Int. Conf. Comput. Sci. Electron. Eng. (ICCSEE), no. Iccsee, pp. 304–307.