

Development of Internet of Things (IoT) for Local SME (Software Development)

M.S.Osman¹, A. Z. Abdul Karim², M.A. Ali³

^{1,2,3}Section of Electrical Technology Universiti Kuala Lumpur - British Malaysian Institute Batu Caves, Selangor, Malaysia

shaiful@unikl.edu.my¹, ahmadzaki@unikl.edu.my², muhd.alif.ali@gmail.com³

Article Info

Volume 83

Page Number: 8050 - 8060

Publication Issue:

March - April 2020

Abstract

In the modern era, humans have developed various systems to automate their daily lives for the ease of community. Internet of Things (IoT) is the newly developed technology system that allows things in daily lives to be connected internet that can allow human to monitor and control their desired things through mobile devices. This project aims for the development of the Internet of Things (IoT) for local small-medium enterprise (SME). Its main objective is to allow communication between machines to achieve automation and monitoring system using Arduino as its main microcontroller to make it into an IoT system. Specifically, in this research project, the machine that is needed to be automated is an industrial stamping machine which is controlled by a programmable logic controller (PLC). There will be a system integration between Arduino and PLC to achieve item counting data collection. All the data that will be collected will be stored in IoT cloud storage where it that can be accessed by the user. This research project may help users to obtain and store counting data on the internet which could reduce human error and reduce costs in the manufacturing industry

Article History

Article Received: 24 July 2019

Revised: 12 September 2019

Accepted: 15 February 2020

Publication: 09 April 2020

Keywords; *Internet of Thing (IoT) for SME*

I. INTRODUCTION

In the modern and technological advance future nowadays, people who are involved with industry relies heavily on robotics and automation to run the machines. The use of industrial machine is widely applied in all industries to create their products. However, new technological advancement has always been studied by engineers around the globe to improve current technology and provide better living for people around the world. One of the current developing technology that are being studied nowadays is the Internet of Things (IoT)[1][2].

Internet of Things (IoT) is defined as a system that does not require human-to-human or human-to-computer interaction and has the ability to transfer data to the internet cloud through the network using interconnected computing devices that involves with daily lives item such as machines, animals, plants or

humans. In other words, it is a new technology that connects items in daily lives into the internet and display the desired output through electronic devices, according to the user's specific needs[3][4].

II. MATERIALS AND METHODS

A. Materials

Materials that need to be used to develop this project is tabulated in Table 1. The list of materials used covers the software part which is the IoT system and the hardware part which is the stamping machine.

TABLE I. LIST OF EQUIPMENT

No	Equipment	Quantity (unit)
1	NodeMCU ESP8266 Microcontroller	1
2	4 Channel Relay Module	1
3	Infrared Sensor	1
4	Jumper Wire	7
5	Programmable Logic Controller (OMRON CQM1H)	1
6	Air Tubing (6mmx1m)	3
7	SMC Solenoid Valve Power Cable	1
8	1/8 Cooper Exhaust Silencer	1
9	1/4 Copper Exhaust Silencer	1
10	Switch Socket	2
11	Plug Top	1
12	Vacuum Solenoid Valve (CONVUM CVA-15HS-24A)	1
13	USB to RS232 Serial Converter Cable	1
14	Magnetic Reed Sensor	7
15	Magnetic Reed Sensor Bracket	7
16	Vacuum Pad	1
17	Synchronous Motor	2
18	Conveyor Mechanism	2
19	24V _{DC} Power Supply Unit	1
20	5V _{DC} Power Supply Unit	1
21	Air Compressor	1
22	Solenoid Valve Manifold	7

23	Proximity Sensor	1
24	Busbar	4
25	Pneumatic Cylinder	1
26	Pneumatic Stamping Module	1
27	Pneumatic Pick & Place Module	1
28	Circuit Breaker (MCB & RCCB)	4
29	Push Buttons (Start, Stop, Emergency Stop)	3
30	Indicator Light (Red, Yellow, Green)	3

B. Block Diagram

The block diagram focuses on the communication between ArduinoNodeMCU (ESP8266) and programmable logic controller which controls the stamping machine through internet. The user also able to control the system through a mobile application (Blynk). As shown in the diagram, the wireless router is the main device to transmit and receive data. The NodeMCU (ESP8266) is microcontroller responsible for the IoT system. Two data signal is received by the microcontroller which is the infrared sensor on the stamping machine and another is the signal from user's Blynk application on their smartphone.

The counting mechanism starts from the detection of metal plate by the infrared sensor. The signal is then sent to microcontroller and microcontroller sends it to the Blynk server through internet connected wireless router. The Blynk server then sends the signal to user's Blynk smartphone application to display the counted value [5][6].

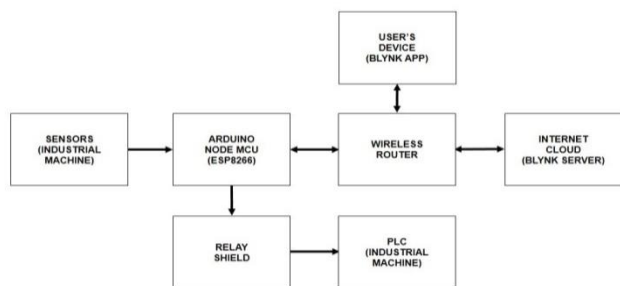


Fig. 1. Block diagram of the IoT system.

The process is similar when the user controls the stamping machine for the Blynk application on their smartphones. The user pressed a button on the Blynk interface application. Then Blynk application sends the signal to the Blynk server through internet connected wireless router. Next, the Blynk server sends the signal to a microcontroller which the microcontroller will then activate the corresponding relay to control the state of the machine such as start, stop or emergency stop function.

C. Project Model and Flow Chart System

This project uses pneumatic technology to implement the stamping of metal plate. As shown in Fig. 2, the product will include 2 conveyor, a pneumatic system to stamp the metal parts and other pneumatic arm system to pick and place the metal plate. For the metal plate counting mechanism, an infrared sensor that acts as the electronic counter is placed above one of the conveyor.

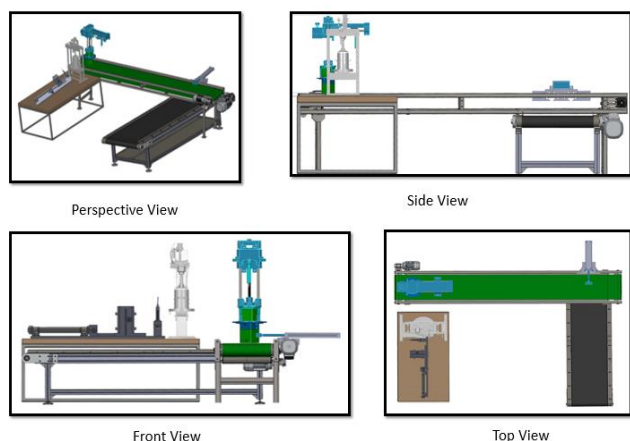


Fig. 2. Project model of the stamping machine.

The system starts with the user powering the machine and hardware so it can prepare and be ready. The user then opens up the Blynk application on their smartphones to access and control the system. The application will first send a signal to Blynk server through the internet to achieve a successful connection between user's device and the hardware.

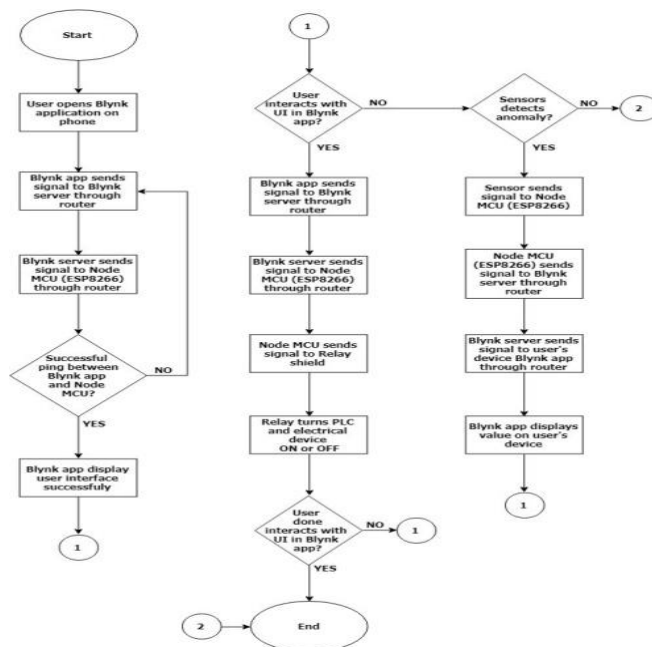


Fig. 3. Flowchart of the IoT system.

The Blynk application will send a ping to ensure connection and wait confirmation of connection from the server. If the connection is successful, the user will be able to view its project interface on their device. The next, user can start to interact with the interface to control the machine. There are several virtual buttons on the interface which the user can start or stops the machine. When the user pressed a button, the Blynk application will send the signal to the Blynk server and then transmit that signal to the hardware which is the relay. The switching of the relay will turn the hardware on and off[7][8].

Besides that, there is a virtual LCD screen on the Blynk application of the user interface where the user can view how many metal plate that has been counted. The counting is done by an infrared sensor which is connected to the microcontroller

NodeMCU ESP8266. When the sensor detects any anomaly, it will send the signal to the microcontroller. Then the microcontroller sends the signal to Blynk's server through the internet and Blynk server will send that signal to the user's device Blynk application where the user can view the amount of counted metal plates.

D. Circuit diagram and Programming

This project is an integration between an Arduino microcontroller and a Programmable Logic Controller (PLC) specifically NodeMCU ESP8266 and Omron CQM1H. The integration is done by adding a 4 channel relay board where the wiring of the system is done mostly at the start, stop and emergency stop button on the hardware. The wiring of components is also essential because the PLC has the input and output panel.

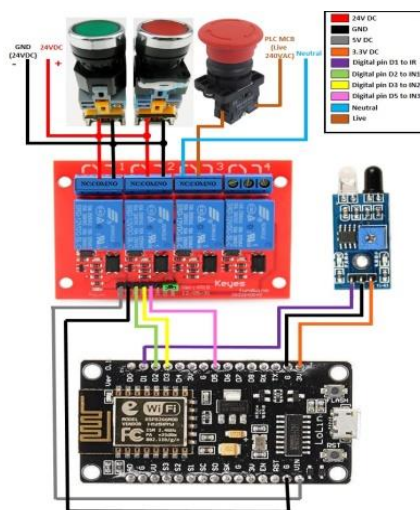


Fig. 4. Wiring diagram for IoT system.

The components used needs to be connected to the correct panels with the correct corresponding address which means that each component is identified as an input or output for the PLC together with its PLC address which need to be parallel with ladder diagram programming. The wiring on Fig. 4 shows the wiring for the NodeMCU ESP8266 microcontroller with the 4 channel relay module as well as infrared sensor and its integration to the hardware's start and stop push buttons and the emergency stop button. Firstly, the microcontroller

is connected to a 5V_{DC} supply to work. On the microcontroller, there is a V_{IN} and ground (GND) pin which is connected to the V_{CC} and GND pin of the 4 channel relay to power up the relay module. Next, the infrared sensor is connected to the microcontroller by connecting a 3.3V and GND pin to the infrared sensor to power it up. The signal pin of the infrared sensor is connected to the D1 pin on the microcontroller. This infrared sensor is used for the counting purpose. Then, the 4 channel relay module is controlled by the microcontroller by connecting pins D2, D3 and D5 on the microcontroller to the pins IN1, IN2, and IN3 on the relay module. These pins are used for the microcontroller to send signals and activate the relays. Each of the relay on the 4 channel relay module has 3 port which is COM, NO and NC. As discussed previously, the start and stop push buttons is powered by a 24V_{DC} supply. Relay 1 and relay 2 on the 4 channel relay module act as the start and stop push buttons for the IoT of this project. The connections for both relay is by having a parallel wiring connection with the corresponding push buttons. The COM port on the relays is connected to positive 24V_{DC} of the push buttons and the NO port on the relays is connected to negative 24V_{DC} of the push buttons.

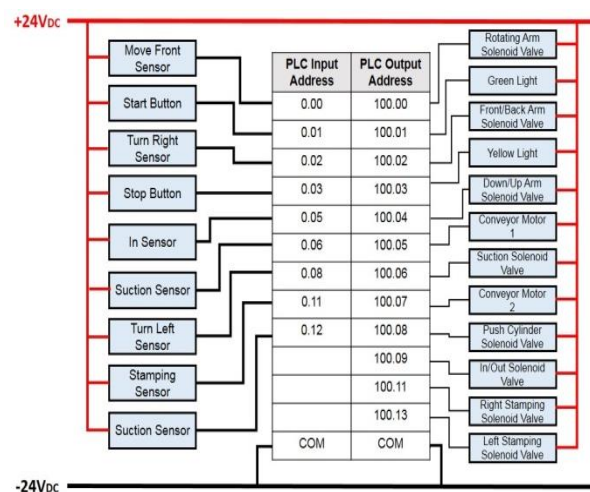


Fig. 5. Wiring diagram to PLC.

To establish the wiring of each component onto the PLC we firstly need to identify which components

act as an input and output for the PLC. For this project, the start and stop push buttons, proximity sensors and magnetic reed sensors are connected to the input panel of the PLC while the rest of the components are connected to the output panel of the PLC as shown in Fig. 5.

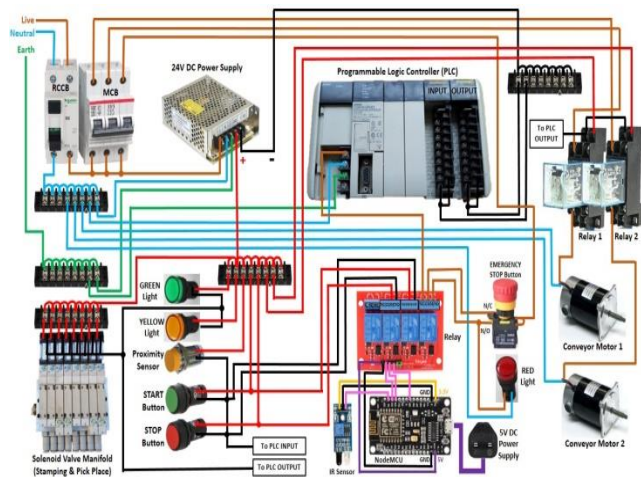


Fig. 6. Wiring diagram for IoT system.

The wiring of those components is by connecting a positive $24V_{DC}$ on one leg of the component and the other leg would be connected to the PLC input or output port according to the corresponding address of the component. From the PLC input and output panel there is a COM port which a wire is connected to the busbar of the negative $24V_{DC}$. From the Fig. 6, it can be observed that the system uses a $240V_{AC}$ supply and also a $24V_{DC}$ supply. Firstly, the $240V_{AC}$ is connected to a Residual Current Circuit Breaker (RCCB) where the live and neutral wire is connected. The earth wire is connected directly to a busbar. From the RCCB, the neutral wire is connected to a busbar while the live wire passes through a Miniature Circuit Breaker (MCB) and to the $24V_{DC}$ power supply. Next, the live wire on the MCB is looped into 3 sections. The first section is used to power the PLC while the other two is to give supply to both of the conveyor motor by connecting it the $24V_{DC}$ relays first. Before the live wire on the MCB is connected to the PLC, it is connected to an emergency stop button where the emergency stop button has a normally open (NO) and a normally closed (NC) slot. The live wire is connected to both

of the slots. The output of the emergency stop button which is the normally open (NO) output is connected to a red light and to one of the 4 channels normally open port while the normally closed output of the emergency button is connected to one of the COM (common) port on the 4 channel relay. From that relay, the normally closed (NC) port is connected to PLC.

Next, the other two live wires from the MCB is connected to a $24V_{DC}$ relay. From the $24V_{DC}$ relay, it is connected to the conveyor motor. The relay is powered/activated by a $24V_{DC}$ supply. From the RCCB, a live wire connects to a $24V_{DC}$ power supply to convert the $240V_{AC}$ into $24V_{DC}$. All of the other components uses $24V_{DC}$ to function. The positive and negative $24V_{DC}$ is connected to a busbar. In order for the PLC to control the switching of the components, the wiring of each component should be having a wire from the positive busbar connected to the component. Next, a negative pin from the components is connected to the PLC, according to its corresponding address. At the input and output panels of the PLC, there is a COM port where the port is connected to the negative busbar of the $24V_{DC}$ supply. The wiring concept for the $24V_{DC}$ components is applied to the green and yellow light indicator, start and stop push buttons, proximity sensor and the solenoid valve manifold which is to control the pneumatic cylinder.

The Arduino coding is the program that is related to the IoT function of this project. This programming is done on the NodeMCU microcontroller through Arduino IDE software. The Arduino coding for this project is divided into 4 sections which are the hardware initialization, reset button, system initialization and counting[9][10].

```

1 #include <ESP8266WiFi.h>
2 #include <BlynkSimpleEsp8266.h>
3
4 // #define sensor D1
5 #define BLYNK_PRINT Serial
6
7 char auth[] = "4a394e9a551c48ae8a95c8e05d514fbf";
8 char ssid[] = "YARLEP";
9 char pass[] = "yarlep94";
10
11 int sensor = 5;
12 int counter = 0;
13

```

Fig. 7. Arduino coding for hardware initialisation.

For the first section of the coding, it involves with the declaration of library used for this system. As shown in Fig. 7, on line 1 and 2 this program includes a library of 'ESP8266Wifi.h' and 'BlynkSimpleEsp8266.h'. Both libraries are required because the microcontroller that is used is NodeMCU ESP8266. Next, on line 5 the 'BLYNK_PRINT Serial' is defined for the sending data and displaying on Blynk application protocol. Then, on line 7, 8, and 9 is about the connection address declaration between hardware and user Blynk application. Every interface in a user's Blynk application on the smartphone has their own unique authentication token. This token is placed in the coding on line 7. As for line 8 and 9, the 'ssid' and 'pass' is the Wi-Fi network credentials that the microcontroller NodeMCU needed to connect to send data to the internet. Next, the coding in line 11 and 12 is about the declaration of hardware used that is connected to the microcontroller. The 'int sensor' is the infrared sensor declaration and it is connected to pin 5 on the microcontroller. The 'int counter' is the variable name to store the value of the counter and its initial value is set to 0.

```

14 BLYNK_WRITE(V0)
15 {
16   int Value = param.asInt();
17   if (Value == 1)
18   {
19     counter = 0;
20     Blynk.virtualWrite(V1, counter);
21     Serial.println("--Counter value is RESET-- ");
22     Serial.print("Counter : ");
23     Serial.println(counter);
24   }
25 }

```

Fig. 8. Arduino coding for reset button.

In the second section, this code does the sending of data from Blynk application to the hardware for reset counter function. As shown in Fig. 8, this whole process is set under program 'BLYNK_WRITE(V0)'. V0 is used because in the user Blynk interface, the reset button is set on pin V0. In this program, it starts with reading if the button in Blynk application is pressed as shown in line 16. The code in line 17 checks if the button in the user interface is pressed. If the button is pressed the counter reset program begins. In the counter reset program, it start with changing the counter value to 0 and store that value in variable name 'counter'. Next, the variable name 'counter' is send to Blynk by using coding line 20 which is 'Blynk.virtualWrite'. Note that there declaration of 'V1'. The V1 is the virtual pin for LCD that has been set in the Blynk application on user interface. The three next line which is line 21, 22, and 23 is to display the counter value at the serial monitor in the Arduino IDE.

```

27 void setup()
28 {
29   pinMode(sensor, INPUT);
30   Serial.begin(9600);
31   Blynk.begin(auth, ssid, pass);
32 }
33
34 void loop()
35 {
36   sendCount();
37   Blynk.run();
38 }

```

Fig. 9. Arduino coding for system initialization.

For the third section, the coding does the system initialization. The system initialization has two programs which are named as 'void setup' and 'void loop' as shown in Fig. 9. In void setup, the coding line 29 is to initialize the infrared sensor connected to pin 5 on the microcontroller.

Next, coding line 30 is to initialize for serial monitor on Arduino IDE. The channel chosen to display is set at baud rate of 9600. Then, the coding line 31 is to initialize an internet connection of the microcontroller to the internet using the credentials

of the auth token, ssid and pass as password. The second program is void loop.

In void loop, the IoT system for this project begins. The coding line 36 is a call function named 'sendCount' which is a program to send and display counted value to user interface on the Blynk application. The next coding on line 37 is to begin Blynk application on the users smartphone when the system has been initialized.

```
40 void sendCount()
41 {
42   int state = digitalRead(sensor);
43   if(state == 0)
44   {
45     counter++;
46     Blynk.virtualWrite(V1, counter);
47     Serial.print("Counter : ");
48     Serial.println(counter);
49     delay(1000);
50   }
51 }
```

Fig. 10. Arduino coding for system counting.

In this final section of the coding, it is about the counting mechanism which uses the infrared sensor. As shown in Fig. 10, in line 42 the program reads the condition of the infrared sensor and store the read value in a variable name called 'state'. Next, in line 43 the program checks the condition by checking the value stored in 'state'. The value turns to 0 if the infrared detects the metal plate.

When the condition is true, counting program begins. Line 45 increased the value stored in variable name 'counter' by one increment. Next, line 46 sends the new counter value to Blynk application to be displayed on the user interface LCD which has been given address 'V1'. The next coding line 47 and 48 takes the new counter value to be displayed on the serial monitor of the Arduino IDE. The delay function in coding line 49 is set to 1000milliseconds where the sending of data will occur every 1 seconds.

III. RESULT AND DISCUSSION

The IoT system for this project uses the Blynk technology which is related to Arduino. The interface for user's control of this IoT stamping machine is designed fully on the Blynk application that is installed on the user's smartphone.

The main user interface on Blynk application for this project consist of a virtual LCD as shown in Figure 11. The purpose of this LCD display is to show the counted number of metal plate that has been successfully stamped by the machine. This is done by the infrared sensor where, when the sensor detects a metal plate the microcontroller will increase the counter and display it on the virtual LCD on the Blynk application.



Fig. 11. GUI in Blynk application.



Fig. 12. Start button on GUI in Blynk application.

As shown in Fig. 12, the start button on the Blynk application is set as a push button which is similar to the hardware push button. The virtual start button

will send signals to the microcontroller when it is pressed and stop sending signals to microcontroller when released. This start button is set on pin D2 on the microcontroller through coding in the Arduino IDE. The settings for this button is that whenever the button is pressed it will send high '1' signal but inside the coding the signal is reversed into low '0' because the 4 channel relay module connected to the microcontroller is active low. Pressing this button will activate one of the relay on the relay module which is connected to the start button on the hardware which will send a signal to the PLC to start the machine.



Fig. 13. Stop button on GUI in Blynk application.

In Fig. 13, the stop button on the Blynk application is also set as a push button which is similar to the start button as discussed before. The virtual stop button will send a signal to the microcontroller when it is pressed and stop sending a signal to microcontroller when released. This start button is set on pin D3 on the microcontroller through coding in the Arduino IDE. The settings for this button is similar to the virtual start button that whenever the button is pressed it will send high '1' signal but inside the coding the signal is reversed into low '0' because the 4 channel relay module connected to the microcontroller is active low. Pressing this button will activate one of the relays on the relay module which is connected to the stop button on the hardware which will send a signal to the PLC to stop the machine.



Fig. 14. Reset button on GUI in Blynk application.

The purpose of this virtual reset button shown in Fig. 14 on the user interface is to reset the counter of the metal plate to 0. Unlike the start and stop button, this reset button is not connected to any pin on the microcontroller. This reset button is assigned into virtual pin 0 (V0) through the coding. This button is set as active high which means when it is pressed it will send a 'high' signal to the Blynk server. From there, Blynk server will identify the signal to reset the counter value and thus the counted value of metal plate is set to 0 which will then be displayed on the virtual LCD screen on user's interface.



Fig. 15. Emergency button on GUI in Blynk application.

This emergency stop button shown in Fig. 15 acts similar to the emergency stop button on the hardware. The relay for this virtual emergency stop button is connected in series with the emergency stop button on the hardware. So, if there is no internet connection to activate the emergency stop button through the application, the user can still

press the emergency stop button on the hardware side to stop the machine. This virtual emergency stop button is connected to pin D5 on the microcontroller and it is set as a switch which means that the user need to press the button to activate and send the signal and press it again to deactivate or stop sending signal to the microcontroller.



Fig. 16. Send report button on GUI in Blynk application.

Shown in Fig. 16, the purpose of send counter report button on the user interface is for the user to store the amount of counting metal plate that has been detected by the system. When the user press the button, the application will send a report in a Microsoft Excel format to the user's email. From there, users can open and analyze the result of the amount of stamped metal plate at which time and date.

This project was created to implement IoT system on the existing SME industrial machine. Since almost all controller used in industrial machine is PLC, thus an integration between IoT systems with PLC should be achieved. This was done by integrating Arduino components with PLC. There are several recommendations that could be added to enhance this project functions and to make it become more reliable to use for industrial purpose. Any future student or future innovator can improve the technology of this project to provide better reliability and more functions.

The first recommendation would be to install a sensor before the push mechanism. The purpose is so that it can detect the condition of the metal plate after the stamping. The sensor will measure the shape and condition of the metal plate. If the metal plate does not achieve the accurate actual measurement, shape then the product will not be counted and will be rejected. An additional sensor should also be placed all over the machine which purpose is to detect problems and any malfunction that occurs on the machine and would notify the user in case the sensors detects anomalous. Next improvement would be to install a camera that can integrate with the Internet of Things (IoT) interface so that it can help to enable real-time monitoring through the mobile phone. The reason to install the real-time monitoring camera is to monitor the working progress at every section of the machine without being near the machine. So that if any problem occurs in the machine, users can monitor and review footage that can help solve the problem.

Lastly, the IoT data storage should be improved for the system. Currently the data storage only allows numbers of counting the value of metal plate to be send to user email. An improvement of this feature would be the documents that are send to the user includes timestamp and also with the addition of monitoring camera, screenshots of the machine in progress should be included in the report. Moreover, in case of any problems that occurs on the machine, the system would automatically send diagnostic reports to the user's email addressing the problems it detected.

IV. CONCLUSION

Based on the outcomes of this project, all of the objectives created regarding the establishment of this project has been achieved. The project titled Development of Internet of Things (IoT) for Local Small-Medium Enterprise (Software Development) achieved its first objective which is the studies and implementation of IoT technology on SME industry was a success.

Besides that, a system was successfully developed that allows controlling the electronic and electrical devices through the internet which is the second objective of this project. This was achieved by using an IoT platform called Blynk. Blynk is a phone application that allows users to design the interface to fit the user's project preferences. All of the data transfer and communication is through Blynk server and the establishment of the Blynk into the project's hardware is by Arduino programming and Arduino boards since the Blynk application only works with Arduino products.

Moreover, through the Blynk application the third objective was achieved, which is creating a monitoring system for SME industrial machines that allows supervisions of product's output through internet. For this project, a metal plate counter was the purpose of monitoring. An infrared sensor was used to act as a counter for this monitoring system.

The infrared sensor detects the metal plate and sends signal to the microcontroller which through programming increased the counting value. Then, the counted value is send to the Blynk application where user can view the amount of counting metal plate.

Next, the final objective was also achieved, which is the storage of product-related data to the internet. Through the Blynk application interface, user can store the counted data and the report will be sent directly to the user's emails with a click of a button. In conclusion, the developed system of the Internet of Things (IoT) is also simple that the process of production can become easier and efficient to the Small-Medium Enterprise users.

This system is also affordable for local Small-Medium Enterprise user since it can help the Small-Medium Enterprise reduces the cost of labor such as in term of manually count the product by human workers. This machine is ready to be applied in the industry and it would provide many benefits to local

Small-Medium Enterprise to move forward in the future.

V. ACKNOWLEDGMENT

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah for His blessing and for the strength he has granted in completing this research paper for ICE2T 2019. To those who directly and indirectly contributed to this research paper, your benevolence means a lot to me.

REFERENCES

- [1]. H. Merchant, and D. D. Ahire, "Industrial Automation using IoT with Raspberry Pi," *Int. J. Computer. Appliance*, vol. 168, pp. 44–48, 2017.
- [2]. H. Kopetz, "Internet of Things in Real-Time Systems," *Real-Time Systems Series*, vol. 1, pp.43-57, 2011.
- [3]. P. N. Margarret, N. Madhukar, A. Ashwini, J. Muddsar, and M. Saish, "IOT Based Industrial Automation," pp. 36–40, 2017.
- [4]. K. Gawli, P. Karande, P. Belose, T. Bhadirke, and A. Bhargava, "Internet of Things (IOT) Based Robotic Arm," pp. 757–759, 2017.
- [5]. U. Berger, and C. Lehmann (2015, September). Development of a Mobile Robot System for Assembly Task on Continuous Conference, pp. 7–11.
- [6]. P. Simoens, M. Dragone, and A. Saffiotti, "The Internet of Robotic Things: A review of concept, added value and applications," *Int. J. Advanced Robot System*, vol. 15, pp. 1–11, 2018.
- [7]. T. P. B. Daisy, and R. S. Albert, "Design and Development of Integrated Secured and Intelligent Architecture for Internet of Things and Cloud Computing," *Future Internet of Things and Cloud (FiCloud)*, pp. 817-822, 2015.
- [8]. L. D. Hung, N. Narendra, and H. L. Truong, "HINC - Harmonizing Diverse Resource Information across IoT Network Functions and

- Clouds,” *Future Internet of Things*, pp. 317-324, 2016.
- [9]. K. D. B. July, D. G. R. Christian, and I. R. M. José, “Learning Technology for Education in Cloud – The Changing Face of Education,” vol. 620, pp. 77, 2016.
- [10]. H. L. Truong, C. Narendra, and J. L. Kwei, “Ensembles of IoT Network Functions and Clouds for Service-Oriented Computing and Applications,” *Service Oriented Computing and Applications*, pp. 65, 2018.