International Journal of Recent Technology and Applied Science
p-ISSN: 2721-2017, e-ISSN: 2721-7280

Original Research Paper

A Design Remote Control System to Feed Bird Using ESP8266

S. Nurmuslimah[1], Naili Saidatin[2], Bagus P.[1]

[1] *Department of Computer System, Faculty of Electrical Engineering and Information Technology, Institut Teknologi Adhi Tama Surabaya.*
[2] *Department of Mechanical Engineering, Faculty of Industrial Technology, Institut Teknologi Adhi Tama Surabaya.*

Article history
Received:
30.08.2020

Revised:
16.09.2020

Accepted:
26.09.2020

***Corresponding Author:**
S. Nurmuslimah
Email:
naili@itats.ac.id

Abstract: Keeping lovebirds at home becomes one of hobbies that is fascinating to some people. Keeping them as pets means educate us to be regular people, particularly related to the process of food provision. Innovation on feeding bird is required such as automatic feeding which can be controlled from far areas. It will be very beneficial when the lovebird keepers are not at home. Therefore, this study develops at designing an automatic system for feeding birds completed with remote control. For this reason, the researcher used ESP8266 connected to NODEMCU as the hardware and BLYNK server as the control bridge of internet. The data would then be transmitted from smartphone to esp8266 system to control the feeding process. In the context of feeding process, the researcher used RTC-clock owned by BLYNK server to substitute RTC as the hardware. Automation system of food provision was carried out by synchronizing the data of hours from the users. The research data were used to design feeder bucket which could give food around 7.2-52.8 grams and feeder bucket for accommodating food about 6.2-48.3 grams. The results of control and connection via BLYNK server demonstrated that 90% data were connected. In terms of data accuracy, this automatic feeder could be accurately 100% successful in a certain time.

Keyword: BLYNK, ESP8266, Feeding Process, Internet of Things, NODEMCU.

1. Introduction

Caring for Lovebird is one of the hobby that is exciting for people. Nurture your beloved Lovebird as you educate yourself to become a regular person. Raising birds such as Lovebird (Agapornis) is now increasingly glance by bird's hobbyists in various regions. Rocketing the selling price of Lovebird is also a trigger for the breeders to expand it to its fullest. Lovebird is also a topic that is currently being talked about in offline forums and online forums on the Internet. This is the popularity factor that causes the bird's price to surge. Another factor that makes the bird hobbyist to nurture this bird is the variety and color that the Lovebird has. The thing to note in maintaining Lovebird colony cage is regular feeding and always available in feed container. But because of the bustle of the hobbyists in the feeding of lovebird regularly become slightly constrained [1].

Research related to the feeding of lovebird birds has been developed by Shoenandia, in its research system designed with an AVR microcontroller. The system is designed only local, when the bird approaches to the container, then the system will open and the bird will be able to eat the contents of the Feed container. In his research there is weakness, which is local or cannot be controlled remotely. The development of remote control technology allows users to monitor the state of cultivated animals with flexibility. Internet network usage plays a role in data transmission and IoT communication.

From exposure by integrating the Internet of things and the world of bird farms, it was designed a bird feeding system. The design of the system is capable of remote control and can be done the RTC timer. System designed using NODEMCU containing chip ESP8266 [2] [3]. The Chip is connected to BLYNK servers. The role of the BLYNK server in this study was for bridge control. The Server provides features that are later used to control, in the opening of the servo, RTC and email notification delivery.

2. Literature Review

2.1. Internet of Things

The Internet of Thing (IoT) is a concept whereby an object has the ability to transfer data over a network without requiring human or human interaction to a computer. IoT has evolved from the convergence of wireless technologies, micro-electromechanical systems (MEMS), and the Internet. "A Things" on the Internet of Things can be defined as the subject of suppose a person with a cardiac implant monitor, a farm animal with a biochip transponder, a car that has equipped a built-in sensor to warn the driver when tire pressure is low. So far, IoT is most closely associated with machine-to-machine (M2M) communication in manufacturing and electricity, demand, and gas. Products built with M2M communication capabilities are often referred to as intelligent or "smart" systems. For example IE smart cable, smart meter, smart grid sensor.

According to Ashton in 2009 the early definition of IoT is the Internet of Things has the potential to change the world as ever done by the Internet, perhaps even better. The statement is taken from the following article: "Today's computers and humans, almost completely dependent on the Internet for all information which all consist of about 50 petabytes (one petabytes is 1,024 terabytes) of data available on the Internet and were first conceived and created by humans. From the magnetic start, to the Record button, take a digital picture or an adequate bar code [4].

The conventional Diagram of the Internet leaves the router the most important part of all. The problem is that people have time, attention and accuracy is limited. They all mean not very good at capturing a variety of data about things in the real world. In terms of physical and so does our environment. Ideas and information are so important, but many more things do not. But the current information technology is very dependent on the data coming from people so that our computer knows more about all the ideas of those things [5].

According to Casagras (Coordinator and Support action for global RFID-related activities and standadisation) defines the IoT as a global network infrastructure, which connects physical and virtual objects through the exploitation of capture data and communication capabilities. Infrastructure consists of existing networks and the Internet follows its network development. All of this will offer identification of objects, sensors and connection capabilities as a basis for the development of independent co-operative services and applications. It is also characterized by high levels of data capture autonomous, event transfer, network connectivity and interoperability [6] [7].

2.2. BLYNK IoT Service

BLYNK is a platform for IOS or ANDROID that is used to control the Arduino modules, Raspberry Pi, Wemos and similar modules via the Internet. This app is very easy to use for people who are still public. This application has many features that facilitate the user in wearing it. How to create a project in this application is very easy, not up to 5 minutes is a drag and drop. BLYNK is not associated with certain modules or boards. From this application, we can control anything remotely wherever we are with Internet-connected records. This is what the IOT (Internet of Things) has to call.

BLYNK is a server service used to support the Internet of Things project. This server service has a mobile user environment both Android and iOS. BLYNK app as an IoT supporter can be abused through Google play. BLYNK supports the different hardware that can be used for the Internet of Things project. BLYNK is a digital Dashboard with graphical interface facilities in the manufacture of the project. Adding components to BLYNK Apps by Drag and Drop makes it easy to add an Input/output component without needing Android or iOS programming capabilities [8] [9].

BLYNK Apps allows to create project interfaces with a variety of input-output components that support for sending or receiving data as well as representing data according to selected components. Data representation can be visual numerals or graphs. There are 4 types of category components that can be used in BLYNK application, namely: controllers are used to transmit data or commands to Hardware; Display is used to display data originating from hardware to smartphone; Notification is used to send messages and notifications; Interface setting display in BLYNK application can be a menu.

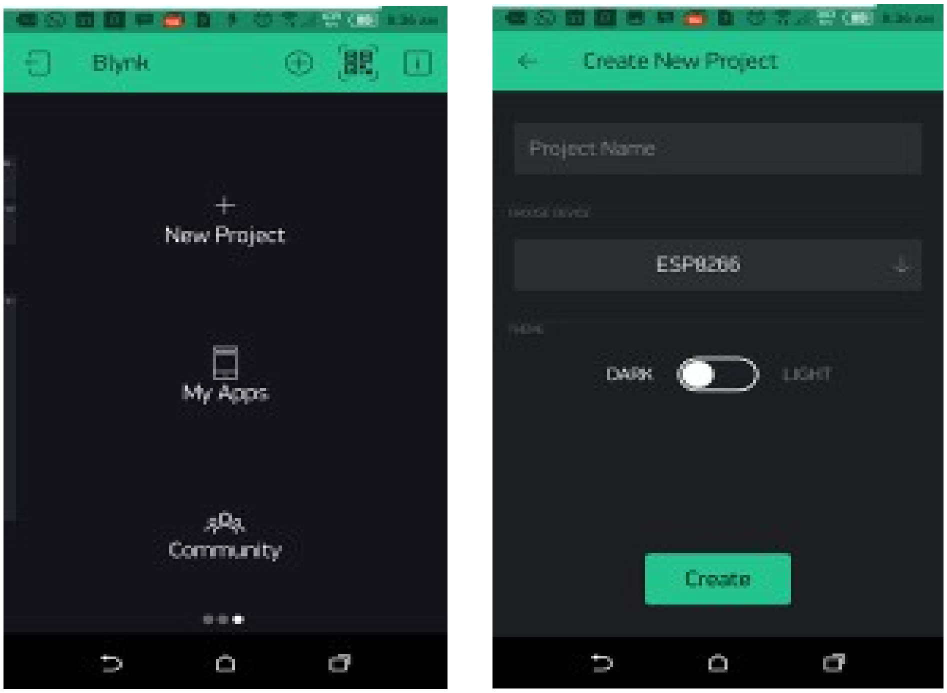


Figure 1. BLYNK and its Fittings [7]

2.3. Node MCU

The MCU ESP8266 Node is a derived module of the development of the family of the ESP-12 type IoT (Internet of Things) platform ESP8266.The ESP8266 module can be learned from previous articles. The function of this module is almost similar to the Arduino module platform, but distinguishing is devoted to "Connected to Internet". In this version (v 0.9) is the first version that has 4 MB flash memory as System on Chip and ESP8266 used is ESP-12.

The downside of this version is in terms of the wide board module size, so if you want to create a porotype using this version of the breadboard, the pin is used only for this module.

This version is a development of version 0.9. And in this version 1.0 ESP8266 used is the type ESP-12E which is considered more stable than the ESP-12 [7]. In addition, the size of the model board is minimized so that compatible used to prototype projects on the Breadboard. And there are pins that are formulated for SPI (Serial Peripheral Interface) Communication and PWM (Pulse Width Modulation) that are not available in version 0.9. It says unofficial board because the product of this module is produced unofficially related to the approval of the Developer Official Node MCU. The difference is not so striking with version 1.0 (official board) is only the addition of USB power output V.

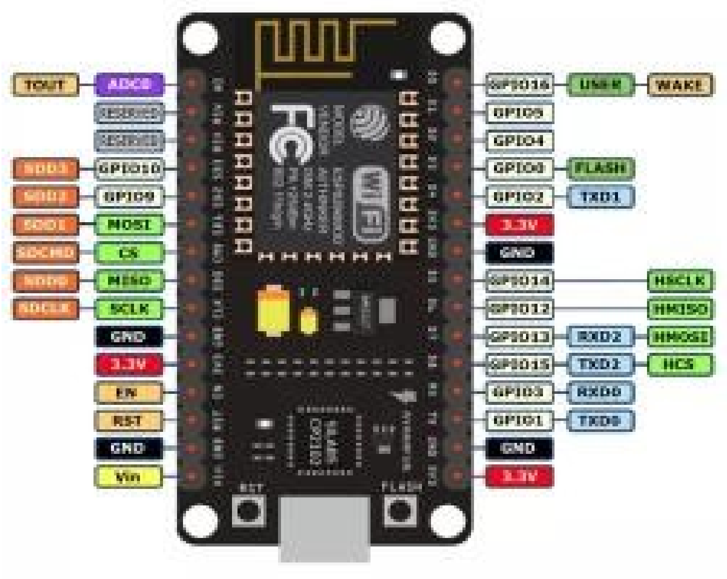


Figure 2. Device Board Wemos D1 Pin

2.4. Method

In the design of the system is done three stages, namely mechanical design, embedded system design and BLYNK component design. Users are able to remotely control the bird's feed control system. When a user travels out of town or not on a home location. The bird cage can still be filled by food. In the design used BLYNK server for ease and the connection path. The mechanical system consists of a feed container connected to the servo motor. The servo Motor serves as the door opening unit of the bird feeding system. The 5 volt voltage servo system is connected from the ESP8266 pin. The data path is inserted from BLYNK to ESP8266. The Data will be processed for feeding the birds. The incoming Data from the ESP8266 chip will be forwarded to the PWM line to soften the servo. When already fed, the user will get a notification on the email that has been fed to the cage. Feeding can be done if there is an Internet connection connected to the system.

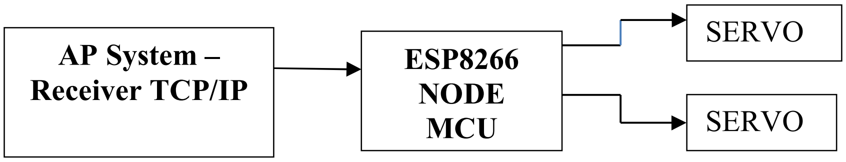


Figure 3. Network Block Diagram

To design a tool, first have to know the system Flow tool how the tool can work. Here will explain the flow diagram of the tool to be created. The tools used are the NODEMCU, servo and mechanical boards. In general, the role of NODEMCU is the connection between digital and internet networks. The use of ESP8266 in the NODEMCU board sets up data transactions that occur in client-server communications. In this research BLYNK act as a server and the unit NODEMCU is client. The Data derived from the user (smartphone) will be sent on the client-server system.

2.5. System Design ESP8266

In research used ESP8266 type NODEMCU LOLIN. The device is a bridge between GPIO communication and WIFI network. The MCU Node is the role of the main controller in the designed system. Fish feeding will interact with users through BLYNK devices. Internet connection and communication path settings are passed on the programmatic range of nodes-MCU.

The programming process is used in the IDEA-Arduino. Programming using the modified C language in a framework. Node MCU has the ESP8266 chipset, as the center recognize the process. ESP8266 handles digital data from embedded encoders in tubes. The Encoder eliminates the digital signal and is read by the NODE-MCU. The NODE-MCU has a connection with BLYNK devices using the API-KEY provided on each device. The data path will be sent and recorded in the webserver provided by BLYNK. The MCU node is shown in Figure 4, which is the block diagram of the one board NODE MCU system [9].

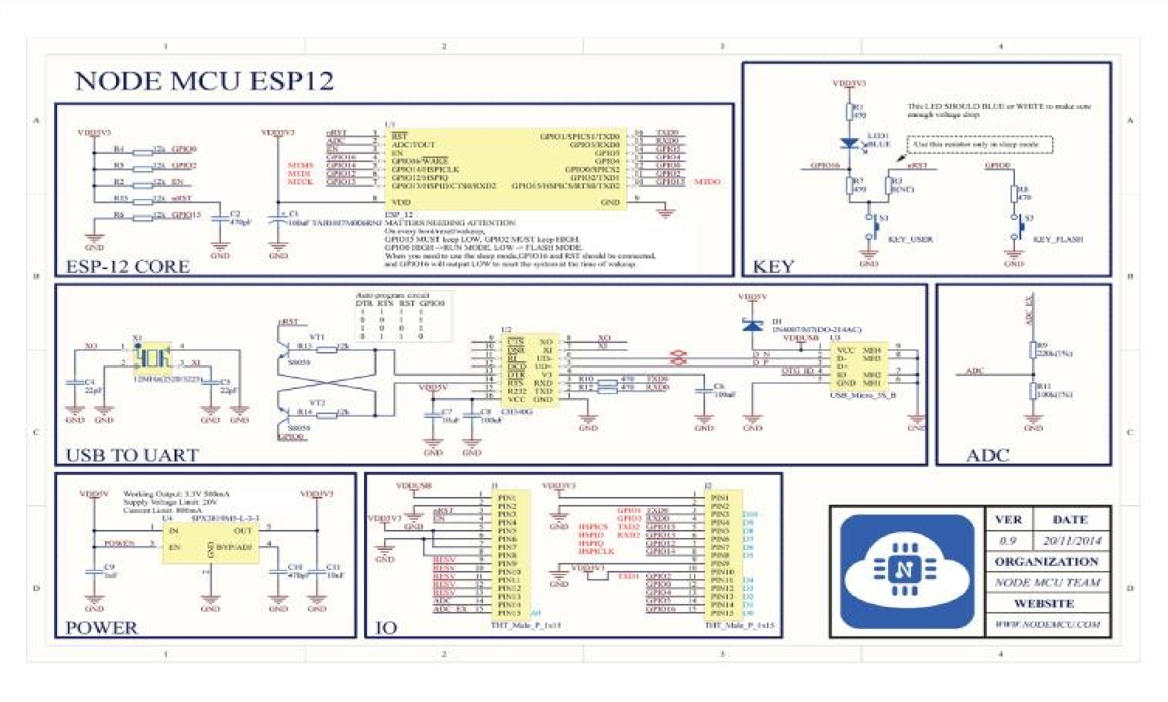


Figure 4. Range of NODE-CMU with ESP8266 Type ESP-12 Chip

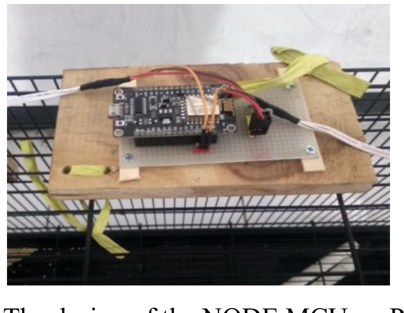


Figure 5. The design of the NODE MCU on PCB Board

The realization of the series of NODEMCU used is shown as in Figure 5, namely the placement of NODEMCU in the bird cage container. In the design of NODEMCU used PCB media dot matrix for easy in laying the legs of the GPIO [10].The GPIO foot role on the ESP series is supporting like the port role in the Arduino microcontroller. The role of the GPIO is more flexible, because in addition to having a 3.3 volt working voltage can also be enabled in the retrieval of sensor data or other. In this study the role of the GPIO is used as a servo curb line for the opening process of the feed container.

2.6. Feed container

In the design of valves designed by using a wooden stick that is in such a way. The stick is attached to the Pro Tower servo and added with foam to the partition so that the bird feed is not out. The partition is used in soft yellow foam material. The stick board is pasted sequentially to form a sheet board. Each board is connected to form a cover unit that can be affixed to the servo. Servo Tower Pro SG90 is a form of servo that has a considerable cap force open.

The role of servo on the research used a set angle of 0-180 range. But there is noise when the set is not precise. At the servo will withstand the load and continue to control according to the given position. The position is determined by the very small servo control inside. The form of control is integrated in the servo gear box. The result of the design and positioning of the servo in bird feed is shown as in Figure 6, which is the installation of SG90 servo for bird feeding.

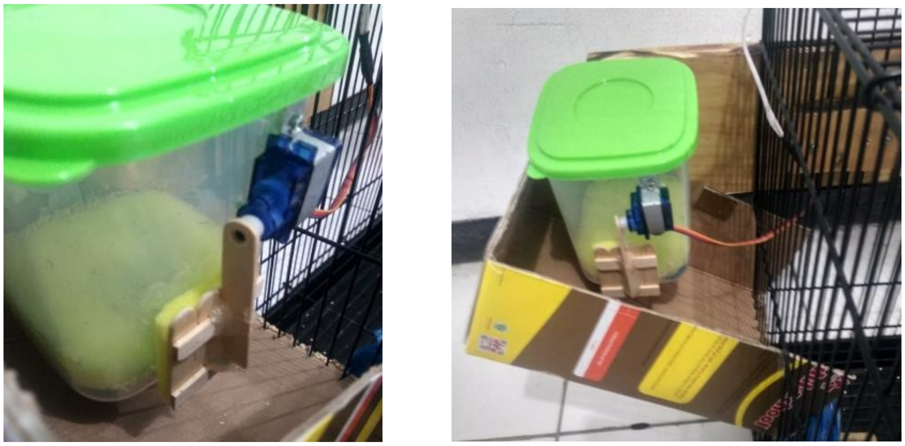


Figure 6. Installation of Bird Feeding Valve

2.7. BLYNK

In addition to mechanical design and electronic devices, in this research also designed BLYNK components. The placement and use of components on BLYNK servers must be designed to optimize energy usage. BLYNK Energy is the allowable use of components that must be no more than 1000.The provision is only for free users. Shown as in Figure 7, namely the registration of BLYNK servers and energy allowance of 1000.From the energy given, the user must compile the form until the target function of the server can work well and able to use the data control process Antra BLYNK and hardware NODEMCU. In the research used several components in BLYNK IE LCD, button, label sensor, email and timer. The use of all these components is mutually integrated and used virtual paths for data settings.

Component preparation needs to be considered when making a project in BLYNK, because it has limitations. The use of energy in this study there are still about 400 left. To start making a display on BLYNK, the new project settings are shown as shown in Figure 7, in the display there are features to add projects. After adding, the setting and laying are done in such a way, so that all components can be used and not overlap between fellow components. When there is overlap from a component it will be difficult for users to operate the tool. The BLYNK server system will hold the main control in the

IoT control process. Components on the BLYNK server require several settings so that data can synchronize with channels programmed in ESP8266. On the ESP8266 hardware path the data virtualization path is used. The choice of data virtualization path is the method used in BLYNK for parsing or shooting data through a specified Virtual ID. The programming pattern is shown in the following list:

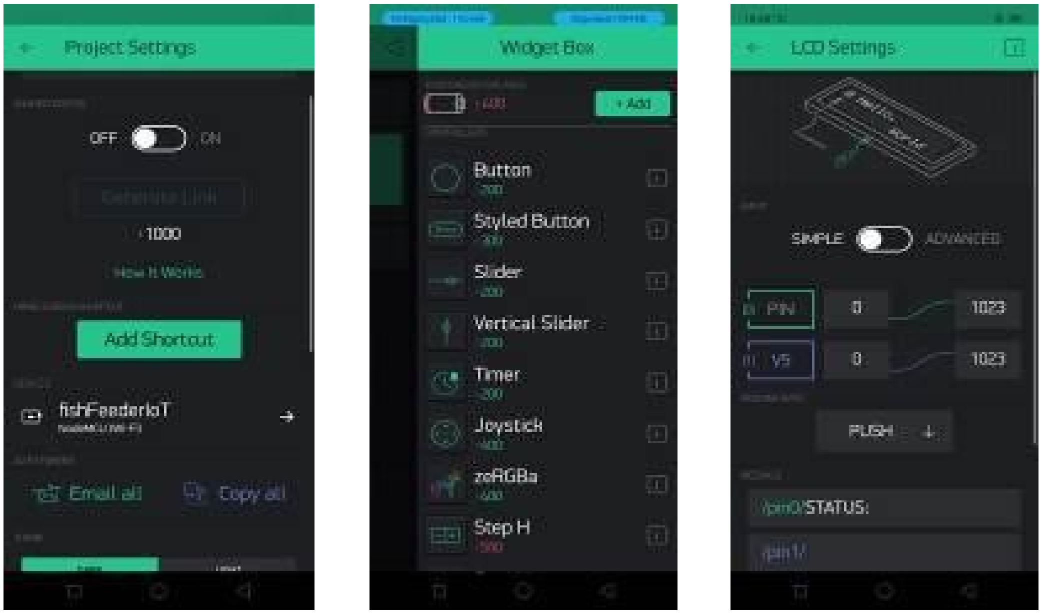


Figure 7. Giving 1000 Energy and Energy Prices in Each Component on the BLYNK Server

3. Results And Discussion

3.1. Servo Test Results

Servo angle testing is done by measuring the servo position command with the measured rotation with a protractor. Measurements were made to find out how true the programming command is with the suitability of the servo opening process. The testing process is shown as in Table 1, namely servo angle testing.

Table 1. Servo Angle Testing A

No	Command	Command Angle	Measuring Angle	Error
1	Servo1.write(0);	0	0	0
2	Servo1.write(10);	10	11	1
3	Servo1.write(20);	20	21	1
4	Servo1.write(30);	30	32	2
5	Servo1.write(40);	40	41	1
6	Servo1.write(50);	50	51	1
7	Servo1.write(60);	60	60	0
8	Servo1.write(70);	70	71	1
9	Servo1.write(80);	80	81	1
10	Servo1.write(90);	90	92	2
			Error (Mean)	**1**

The test starts with an angle of 0-90 ° to see whether the servo can move according to the commands given.

In the second servo test which is located in the two feed containers, shown as in table 2, namely the second servo test. The second servo is placed to open the second feed container. In the second servo test, it is done the same, namely comparing the angle of the command with the measurement of the arc length.

Table 2. Servo Angle Testing B

No	Command	Command Angle	Measuring Angle	Error
1	Servo2.write(0);	0	0	0
2	Servo2.write(10);	10	12	2
3	Servo2.write(20);	20	22	2
4	Servo2.write(30);	30	31	1
5	Servo2.write(40);	40	42	2
6	Servo2.write(50);	50	52	2
7	Servo2.write(60);	60	62	2
8	Servo2.write(70);	70	73	3
9	Servo2.write(80);	80	82	2
10	Servo2.write(90);	90	92	2
			Error (Mean)	**1.**

From the results of the second servo test, the servo motion has an angle difference. Servo one has a smaller average angle difference compared to Servo Two. The difference is caused by the mechanical structure in the servo.

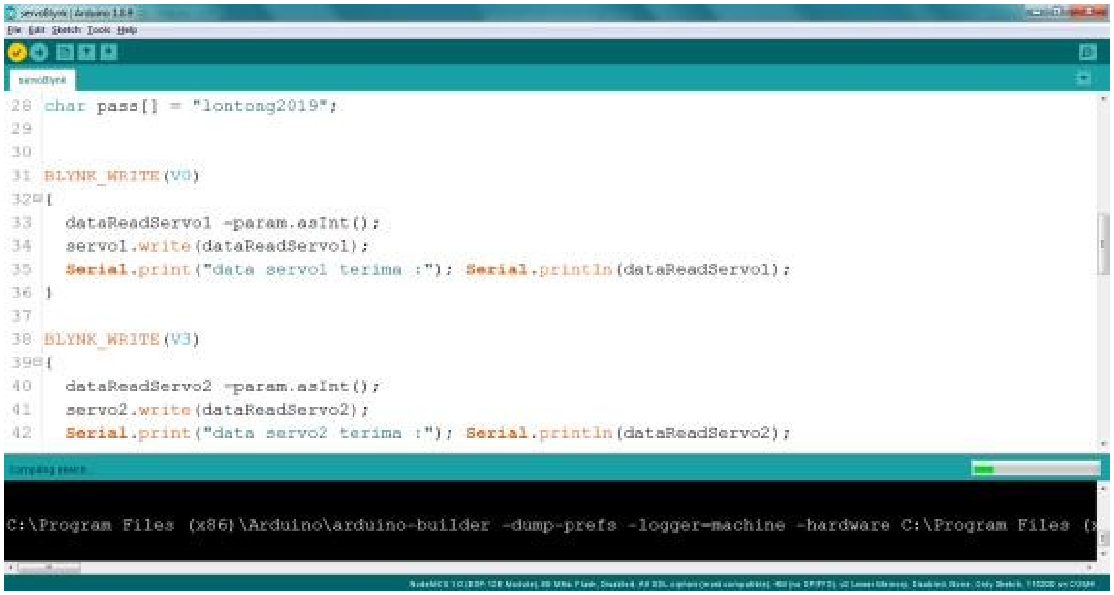
3.2. ESP8266 Filling Testing

The charging test on the ESP8266 board is to find out whether the chip can still receive charging data. If the chip is damaged, then the charging data will not be able to enter. Charging is used IDE-Arduino software.

Table 3. Testing the Charging Program on the ESP8266 Chip

COMMAND	RESULT
1	Instruction – code successfully sent
2	Instruction – code successfully sent
3	Instruction – code successfully sent
4	Instruction – code successfully sent
5	Instruction – code successfully sent
100%	

Charging is used to set the correct board so that data can be entered. Board selection is used ESP8266 NODEMCUV1 which corresponds to the board used. When the compilation process is successful the data will automatically be entered on the ESP8266 chip. The filling process is tested five times the data collection, from the test there is a degree of success is 100%.



```
char pass[] = "lontong2019";


BLYNK_WRITE(V0)
{
  dataReadServo1 =param.asInt();
  servo1.write(dataReadServo1);
  Serial.print("data servo1 terima :"); Serial.println(dataReadServo1);
}

BLYNK_WRITE(V3)
{
  dataReadServo2 =param.asInt();
  servo2.write(dataReadServo2);
  Serial.print("data servo2 terima :"); Serial.println(dataReadServo2);
```

```
C:\Program Files (x86)\Arduino\arduino-builder -dump-prefs -logger=machine -hardware C:\Program Files (x
```

Figure 8. The process of Uploading Data from the Arduino IDE to the ESP8266 Chip

The calculation of the value of percent success follows the Equation 1, where the number of successes is divide by the number of attempts or trials made.

$$\% \text{ Success Rate} = \frac{\text{Number of successes}}{\text{Number of trials}} \text{ x } 100\% \quad (1)$$

4. Conclusion

From a series of studies that have been carried out, the following conclusions are obtained: Design of a bird feed container, able to provide feed with a quantity of feed coverage of around 7.2 - 52.8 grams for container A and about 6.2 - 48.3 grams for container B; Control and BLYNK server connection obtained 90% of data can be connected; Accuracy of data related to feeding automatically, can be done accurately within a certain time with a degree of success of 100%.

References

[1] A. F. Widawan, *IoT. Fakultas Sains dan Matematika.* Universitas Diponegoro Semarang, 2011.

[2] *E-Systems.* Datasheet ESP8266 in ESP8266EX Datasheet. 2016.

[3] M. A. Azizi, and A. Zariman, "Displaying Health Status Based IoT," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 25-35, 2020. [Online]. Available: https://doi.org/10.36079/lamintang.ijortas-0201.57. [Accessed: June 2020].

[3] Agfianto, *Tutorial AT89: RTC DS1307 (64 x 8 Serial Real-Time Clock)*. 2009.

[4] M. U. J. Ibnu, *Aneka Proyek Mikrokontroler.* Jakarta: Elex Media Komputindo. 2009.

[5] G. D. Shivamadhu, A. Venkatesh, A. Alva, D. Nausheer, and K. A, S. Devi, "PortaX Secure Automation System Using IoT – A Survey," *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 66-74, 2020. [Online]. Available: https://doi.org/10.36079/lamintang.ijortas-0201.61. [Accessed: June 2020].

[6] M. P. T. Sulistyanto, D. A. Nugraha, N. Sari, N. Karima, and Asrori, "Implementasi IoT (Internet of Things)," in *pembelajaran di Universitas Kanjuruhan Malang, SMARTICS J.* 2005.

[7] Blynk.io, *Reference Guide.* 2018. [Online]. Available: url:https://Blynk.io. [Accessed: June 2020].

[8] W. Budiharto, *Panduan Praktikum Mikrokontroler AVR ATMega 16*. Jakarta: Elex Media Komputindo, 2008.

[9] H. Hermansyah, K. Kasim, and I. K. Yusri, “Solar Panel Remote Monitoring and Control System on Miniature Weather Stations Based on Web Server and ESP32,” *International Journal of Recent Technology and Applied Science*, vol. 2, no. 1, pp. 1-24, 2020. [Online]. Available: https://doi.org/10.36079/lamintang.ijortas-0201.56. [Accessed: June 2020].

[10] A. Kadir, *Arduino Panduan Mempelajari Aneka Proyek Berbasis Mikrokontroler*. Yogyakarta: Penerbit Andi, 2015.