

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Implementation

This project uses the Arduino IDE app. This app is used for compilation and upload the program. This program can be seen as follows:

1. Libraries used in this project.\

```
1. #include <SPI.h>
2. #include <Ethernet.h>
3. #include <PubSubClient.h>
4. #include <Servo.h>
```

2. Setting network.

```
5. byte mac[] = { 0xDE, 0xED, 0xBA, 0xFE, 0xFE, 0xED };
6. IPAddress ip(192, 168, 1, 177);
7. IPAddress server(192, 168, 1, 10);
```

3. Function callback is used to process data commands from Domoticz and Node-RED frameworks. This code controls the servo.

```
8. void callback(char* topic, byte* payload, unsigned int
length) {
9.   Serial.print(topic);
10.  Serial.print(" ");
11.  for (int i=0;i<length;i++) {
12.    Serial.print((char)payload[i]);
13.  }
14.  Serial.println();
15.  if (payload[0] == '1' && payload[1] == '1') {
16.    myservo.write(0);
17.  }
18.  if (payload[0] == '1' && payload[1] == '0') {
19.    myservo.write(90);
20.  }
21. }
```

4. The function below connects Arduino UNO to MQTT.

```

22.     void reconnect() {
23.         // Loop until we're reconnected
24.         while (!client.connected()) {
25.             Serial.print("Attempting MQTT
connection...");
26.             // Attempt to connect
27.             if (client.connect("arduinoClient")) {
28.                 Serial.println("connected");
29.                 // Once connected, publish an
announcement...
30.                 client.publish("domoticz/out","hello
world");
31.                 // ... and resubscribe
32.                 client.subscribe("pintu");
33.             } else {
34.                 Serial.print("failed, rc=");
35.                 Serial.print(client.state());
36.                 Serial.println(" try again in 5 seconds");
37.                 // Wait 5 seconds before retrying
38.                 delay(5000);
39.             }
40.         }
41.     }

```

5. The following program is used to read the upper and lower limits of water in the water reservoir, if the upper limit of the sensor reads ≥ 250 then the water pump machine will die and if the lower limit of the sensor reads < 250 then the water pump machine will light to fill the water at tub shelter.

```

42.     dataAir1 = analogRead(pinAir1);
43.     dataAir2 = analogRead(pinAir2);
44.     if (dataAir1  $\geq$  250)
45.     {
46.         digitalWrite(pinPump, LOW);
47.     }
48.     if (dataAir2  $<$  250)
49.     {
50.         digitalWrite(pinPump, HIGH);
51.     }

```

6. The following program is used to send notifications to the Domoticz framework via MQTT by publishing it when the rain sensor reads the analog value. If the sensor reads ≤ 500 then it will send a notification that the water pump machine is alive and if not the water pump machine is dead.

```

52.     sensorValue = analogRead(sensorPin);
53.     if (sensorValue<=500) {
54.         client.publish("domoticz/in","{\"idx\":4
,\"nvalue\":0,\"svalue\":\"Mesin Pump Hidup\" }");
55.     }
56.     else {
57.         client.publish("domoticz/in","{\"idx\":4
,\"nvalue\":0,\"svalue\":\"Mesin Pump Mati\" }");
58.     }

```

7. This program is used to send notifications to the Domoticz framework via MQTT by publishing it when the obstacle sensor reads the value. If the reading sensor detects an obstacle it will send a notification that the door is open, otherwise it will send a notification that the door is being closed.

```

59.     obstacle1 = digitalRead(obstaclePin1);
60.     if (obstacle1 == LOW){
61.         client.publish("domoticz/in","{\"idx\":5,\"n
value\":0,\"svalue\":\"Pintu Terbuka\" }");
62.     }
63.     else{
64.         client.publish("domoticz/in","{\"idx\":5,\"n
value\":0,\"svalue\":\"Pintu Tertutup\" }");
65.     }
66.     delay(1000);

```

8. The following program is used to count water debit using waterflow sensor, in this program I use constant variable which serves to set the water speed. After that the value of the discharge of water in units of milliliter will be published to MQTT and will be displayed data on Domoticz framework.

```

67.     if((millis() - oldTime) > 1000)
68.     {
69.         detachInterrupt(sensorInt);
70.         debit = ((1000.0 / (millis() - oldTime)) *
pulseCount) / konstanta;
71.         oldTime = millis();
72.         flowmlt = ((debit / 60) * 1000)*2;
73.         totalmlt += flowmlt;
74.
75.         unsigned int frac;
76.         char data[200];
77.         float temp_coop = totalmlt;
78.         Serial.println(temp_coop);
79.
payload="{"idx":9,"nvalue":0,"svalue":" "+String
(temp_coop)+"\} ";
80.
payload="{"Suhu":" "+String(temp_coop)+"} "; //String
81.     payload.toCharArray(data, (payload.length()
+ 1));
82.     client.publish("domoticz/in", data);
83.
84.     pulseCount = 0;
85.
86.     attachInterrupt(sensorInt, pulseCounter,
FALLING);
87.     }
88.     delay(1000);

```

5.2 Testing

The first microcontroller will run the program to connect to MQTT and Node-RED. Once connected, if the upper waterlevel sensor value has a value <250 then it will turn on the air pump machine and if the top waterlevel sensor has value $> = 250$ then the water is full and will turn off the water pump machine. When the water pump engine turns on the rain sensor it will read if any water flow will send a message to Domoticz that the water machine is on, otherwise it will send a message to Domoticz that the water machine is dead.



Illustration 5.1: The rain sensor reads the flow of water from the water pump machine

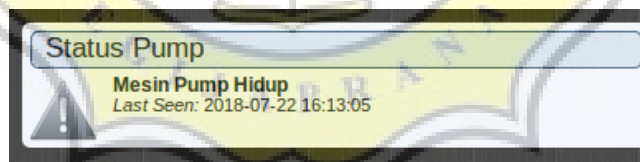


Illustration 5.2: Status Pump ON

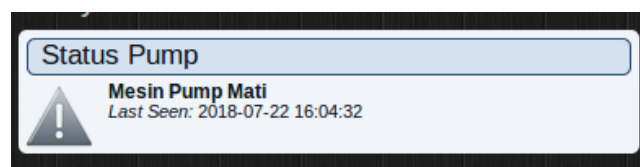


Illustration 5.3: Status Pump OFF

The water flow function of this project is to calculate the water discharge in milliliter size. To accurate the flow of water, then experiment.

Table 5.1: Water Debit Counting Experiments

Volume & height water	Konstanta (flowmeter)	200ml	300ml	500ml
1500ml & 9cm	4.5	205ml	322ml	496ml
	4.0	222ml	334ml	562ml
1000ml & 6cm	4.5	196ml	305ml	490ml
	4.0	218ml	332ml	561ml

From the above experimental results, water pressure in the shelter affects the process of calculating the flow of water.

On home gate security system, if ordered by domoticz to lock, the servo will work to lock the gate. And users can monitor from Domoticz if the servo has worked properly. And if the door is open or closed the user can see the status in Domoticz.

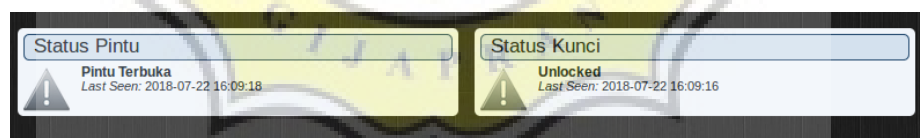


Illustration 5.4: Status Open Door and Unlocked

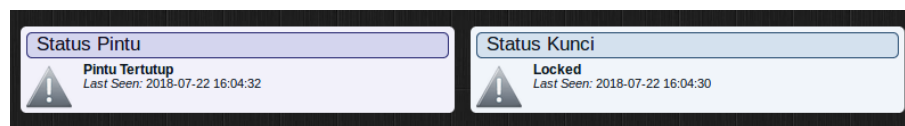


Illustration 5.5: Status Close Door And Locked

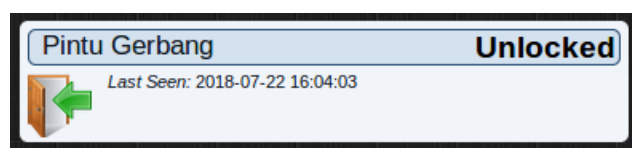


Illustration 5.6: Switch Servo