Literasi Nusantara
vol.5 no. 3, July 2025 – October 2025
https://journal.citradharma.org/index.php/literasinusantara/index

ISSN 2746 8208 (Online)
ISSN 2746 3575 (Print)

Internet of Things-based Mini Conveyor Design using Ubidots platform

Moch Hawin Hamami [1*], Devan Cakra Mudra Wijaya [2], Basuki Rahmat [3], Henni Endah Wahanani [4]

[1,2,3,4] Universitas Pembangunan Nasional Veteran, Jawa Timur, Surabaya, Indonesia
hawin.hamami@gmail.com, mudrawijaya@gmail.com,
basukirahmat.if@upnjatim.ac.id, henniendah.if@upnjatim.ac.id

DOI : *https://doi.org/10.56480/jln.v5i2.1593*

Received: April 28, 2025
Revised: May 25, 2025
Accepted: June 07, 2025

Abstract

The development of digital technology has driven the use of the Internet of Things (IoT) in various fields, including the manufacturing industry. This study aims to provide an overview of the design of an IoT-based monitoring and control system for DC motor speed on conveyors, using the Ubidots platform. The developed system integrates an encoder sensor, a JGA25-370 DC motor, an ESP32 microcontroller, an L298N motor driver, and an I2C LCD. The methodology used in this study is Rapid Application Development (RAD), while the motor speed control technique employs Pulse Width Modulation (PWM). Speed adjustment is performed via a potentiometer, while the On/Off function and motor rotation direction control can be accessed through physical buttons or via the Ubidots interface connected using the MQTT communication protocol. Test results indicate that the system successfully connects to the Wi-Fi network and can communicate in real-time with the Ubidots platform. The system is also capable of effectively monitoring and controlling the speed of the DC motor. However, speed adjustment still relies on manual input via the potentiometer, necessitating further development to enable remote speed control. These findings are expected to provide educational and practical value in IoT applications, while also enhancing efficiency and flexibility in conveyor-based production systems.

Keywords– *ESP32, Motor DC, RPM, Ubidots, MQTT*

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

1. Introduction

The rapid evolution of industrial technology demands automation systems that are not only efficient but also highly adaptable to changing operational requirements (Tamimi & Munawaroh, 2024; Danta et al., 2016; Nurhanudin & Kartimi, 2025). Industries are increasingly shifting toward smart manufacturing, where flexibility, real-time monitoring, and remote control capabilities are essential for maintaining productivity. Traditional systems often rely on manual intervention or localized control mechanisms, which limit scalability and responsiveness. As industries embrace Industry 4.0, there is a growing need for automation solutions that integrate advanced communication technologies, enabling seamless coordination between machines, sensors, and control systems (Parlika et al., 2021; Sinaga & Peniarsih, 2024).

Conveyors play a crucial role in industrial production by facilitating the efficient movement of materials between different stages of manufacturing (Fahlevi et al., 2023; Ushofa et al., 2022). These systems are widely used in sectors such as logistics, automotive, food processing, and pharmaceuticals, where continuous material handling is critical. However, many existing conveyor systems still operate using conventional control mechanisms that require direct human supervision, leading to inefficiencies in dynamic production environments. The lack of remote monitoring and adaptive control features restricts their ability to respond to real-time changes in production demands (Baumann et al., 2021; Kurniawan & Widyatama, 2025).

One of the major limitations of traditional conveyor systems is their dependence on localized control, which restricts flexibility and real-time adjustments (Romadhon & Nawawi, 2024). Operators often need to manually adjust speed, direction, or loading parameters, leading to delays and potential human errors. Additionally, the inability to monitor conveyor performance remotely makes it difficult to optimize operations across multiple production lines. These challenges highlight the need for smarter automation solutions that leverage IoT technology to enable centralized, real-time control and predictive maintenance (Tugino et al., 2019).

The advancement of the Internet of Things (IoT) has opened new possibilities for industrial automation by allowing devices to communicate and exchange data over networks (Saputra et al., 2017). Among various IoT communication protocols, MQTT (Message Queuing Telemetry Transport) stands out due to its lightweight, efficient, and reliable data transmission capabilities, making it ideal for small to medium-sized industrial applications. MQTT's publish-subscribe model ensures seamless communication between sensors, controllers, and cloud-based monitoring systems, enabling real-time adjustments and remote diagnostics. This makes it a strong candidate for integrating smart control features into conveyor systems (Baumann et al., 2021).

In conveyor systems, regulating motor speed is essential for optimizing material flow and energy consumption. Pulse Width Modulation (PWM) has been widely adopted for DC motor speed control due to its efficiency and precision (Petru & Mazen, 2015). By adjusting the duty cycle of PWM signals, operators can fine-tune motor speeds without significant energy loss, making it a cost-effective solution for industrial applications. However, most existing implementations of PWM-based speed control remain localized, lacking integration with IoT platforms for remote management. Bridging this gap could significantly enhance conveyor system flexibility and automation.

Previous studies have explored aspects of IoT-based conveyor systems but have not fully integrated all necessary functionalities. For instance, Putra (2023) developed an IoT conveyor prototype using Blynk but did not incorporate motor speed control. Kusuma & Hendry (2024) implemented MQTT for weight monitoring but lacked active control features. Meanwhile, Abdillah (2023) focused on local PWM-based speed control without remote communication capabilities. These gaps indicate a need for a more comprehensive solution that combines MQTT-based IoT communication with PWM-driven motor control, enabling fully automated, remotely monitored conveyor systems.

To address these limitations, this study proposes an IoT-based conveyor system that integrates MQTT for real-time communication and PWM for precise motor speed control. This system will allow operators to remotely monitor and

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

adjust conveyor operations via a cloud-based dashboard, improving efficiency and reducing manual intervention. By leveraging MQTT's lightweight protocol and PWM's reliable speed regulation, the proposed solution aims to enhance industrial automation, contributing to the advancement of smart manufacturing technologies. Future developments could include AI-driven predictive maintenance and adaptive speed control based on real-time production data, further optimizing industrial workflows.

2. Method

This study uses the RAD (Rapid Application Development) method, which consists of four stages: requirements, prototyping, output, and testing. The requirements stage includes literature review and analysis. Prototyping focuses on hardware and software design. Output describes the final form of the system, while testing includes automated and manual testing.

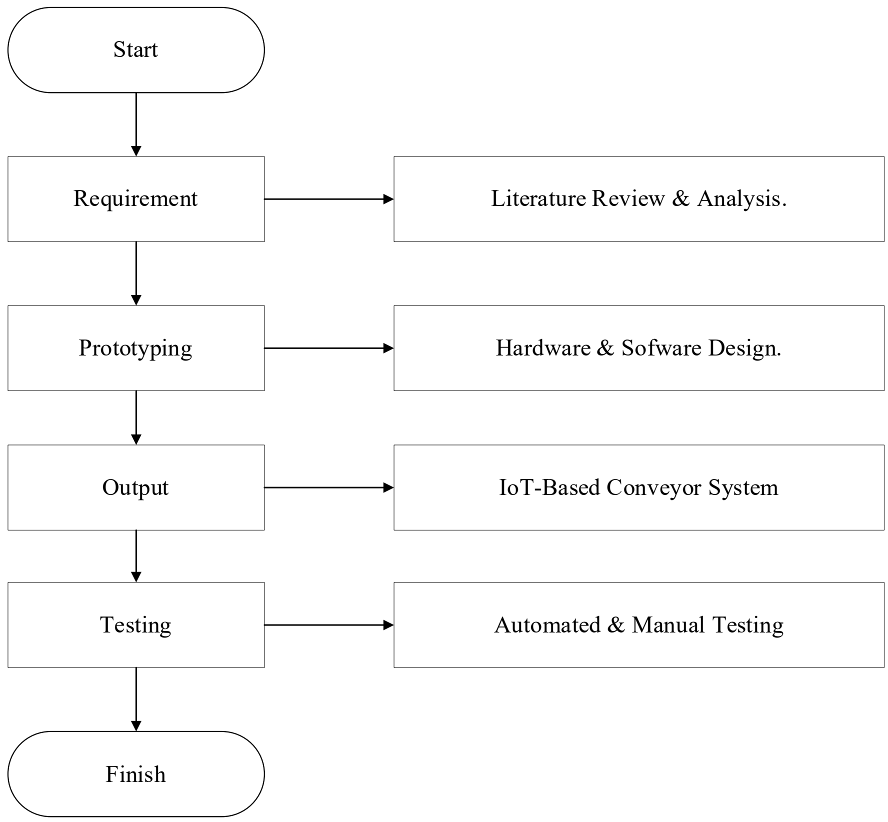

Figure 1. Research Methodology

This system implements Pulse Width Modulation (PWM) as the main method for controlling the speed of a DC motor. PWM regulates the magnitude of the signal by modulating the pulse width. Then, the system output is calculated based on the formula shown below :

$$\text{Duty Cycle (\%)} = \left(\frac{\text{ON Time}}{\text{Total Time Per Period}}\right) \times 100\%$$

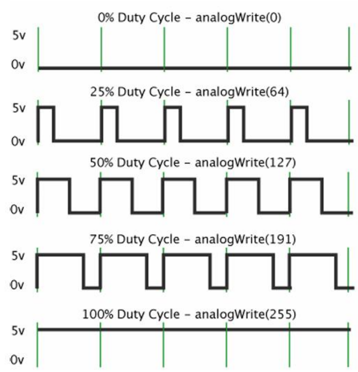

Figure 2. PWM (Pulse Width Modulation)

The development of an IoT-based mini conveyor system requires an integrated design between hardware and software, which is explicitly described below.

Hardware Design

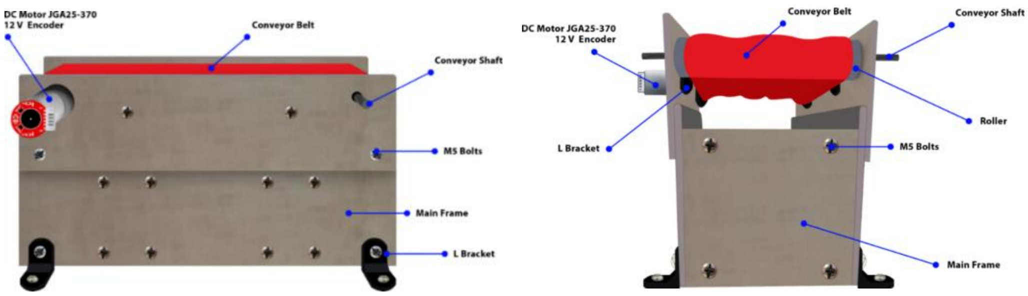

Figure 3. Hardware Design

Figure 3 above shows the main components of the overall system design, which includes the main frame, conveyor belt, rollers, and JGA25-370 DC motor. To maintain structural stability, L-brackets are installed at the bottom of the system.

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

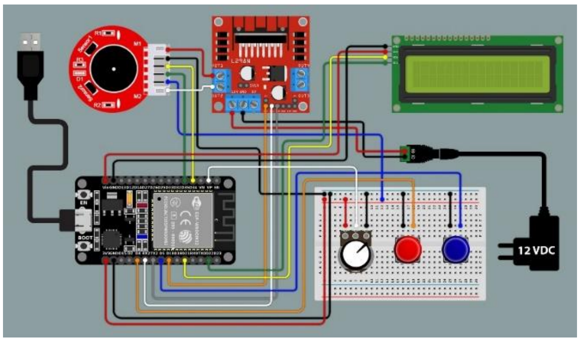

Figure 4. Wiring Diagram

Figure 4 above shows the system cable configuration, which is described in more detail in the following section.

Table 1. Hardware Cabling Details

Hardware Name	Device Pin		Connection Cable Color	Connected Device
	Start Point	End Point		
Breadboard	GND	GND	Black	Encoder -
				ESP32 DOIT DevKit V1
				Potensiometer
				Push button 1
				Push button 2
	VCC	3V3	Red	ESP32 DOIT DevKit V1
		VCC		Potensiometer
		Encoder +	Blue	Motor DC JGA25-370 12V Encoder
ESP32 DOIT DevKit V1	Vin (5V)	VCC	Red	LCD I2C
	GND	GND	Black	
				Breadboard
	3V3	VCC	Red	
	GPIO4		Orange	Push Button 1
	GPIO5		Blue	Push Button 2
	GPIO16	IN1	White	Motor Driver L298N
	GPIO17	IN2	Gray	
	GPIO18	ENA	Orange	
	GPIO21	SDA	Yellow	LCD I2C
	GPIO22	SCL	Green	
	GPIO34	Encoder A	Yellow	Motor DC JGA25-370 12V Encoder
	GPIO35	Encoder B	Green	
	GPIO36	DATA	White	Potensiometer
Motor Driver L298N	OUT1	Motor +	Red	Motor DC JGA25-370 12V Encoder
	OUT2	Motor -	White	
	Vin (Max: 12V)	Adaptor +	Red	Adaptor 12V
	GND	Adaptor -	Black	

Software Design

The workflow of the IoT-based mini conveyor system is shown in Figure 5.

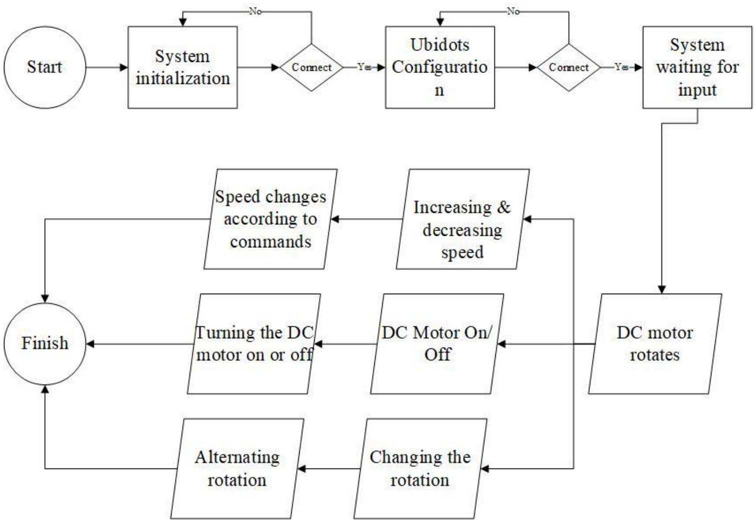

Figure 5. Workflow system

Figure 5 illustrates the workflow of the DC motor speed control system on an IoT-based conveyor. The process begins with device initialization, followed by a network connection check. If the connection fails, the system will repeat the initialization process. If successful, the system configures itself to connect to the Ubidots platform. Once the connection to Ubidots is active, the system enters "waiting for input" mode from the user. Input can include commands to adjust motor speed, turn the motor on/off, or change the direction of rotation (clockwise or counterclockwise). Commands can be sent via physical buttons or online through Ubidots. The system responds to commands in real-time and returns to the waiting state for the next control cycle as long as the device remains active.

3. Result and Discussion

This chapter discusses the implementation of the overall hardware design and the implementation of the program code related to the Ubidots connection, using the MQTT protocol. In addition, the results of a series of tests are also explained, including: WiFi network connectivity tests, IoT communication tests, and tests of the publish-subscribe mechanism in the MQTT protocol. Furthermore, data display verification is performed via an LCD screen, and the

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

rotational speed of the DC motor (RPM) is measured using three methods: data visualization on the Ubidots platform, real-time monitoring through the Arduino IDE, and readings using a Digital Tachometer for result validation.

Hardware Design Implementation

All hardware components are assembled based on the specifications and prototype design described in the previous chapter. Figure 6 below shows the complete hardware assembly, which consists of Display, Actuator, and Sensor components.

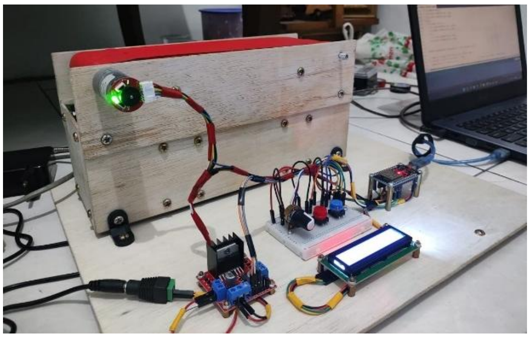

Figure 6. Hardware Design Implementation

3. 2. Ubidots Connection

The following source code is the most important part of the IoT system, which functions to manage device connections with the IoT platform, in this case, Ubidots.

Source Code 1. Ubidots_Connection.ino

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient espClient; PubSubClient client(espClient);
#define WIFI_NAME "YOUR_WIFI_NAME"
#define WIFI_PASSWORD "YOUR_WIFI_PASSWORD"
#define SERVER "industrial.api.ubidots.com"
#define PORT 1883
#define UBD_TOKEN "YOUR_UBIDOTS_TOKEN"
#define UBD_PASSWORD ""
String CLIENT_ID = "esp32-" + String(random(1000));
int retry = 0; const int maxRetry = 20;
bool wasConnected = false; bool isWiFiConnecting = false;
bool isUbidotsConnecting = false; bool hasTriedConnecting = false;
unsigned long lastWiFiCTime = 0, lastWiFiRTime = 0, lastRcnAttempt = 0;
const unsigned long wifiCIntv = 1000, wifiRIntv = 500, rcnIntv = 5000;
```

```
void setup() { Serial.begin(115200);connectWiFi();connectUbidots(); }
void loop() { checkWiFiConnection();checkUbidotsConnection(); }
void connectWiFi() {
  if (!isWiFiConnecting && WiFi.status() != WL_CONNECTED) {
    WiFi.mode(WIFI_STA); WiFi.begin(WIFI_NAME, WIFI_PASSWORD);
    Serial.print("Menghubungkan ke Wi-Fi"); retry = 0;
    lastWiFiRTime = millis(); isWiFiConnecting = true;
    hasTriedConnecting = true;
  }
}
void checkWiFiConnection() {
  unsigned long currentTime = millis();
  if (WiFi.status() == WL_CONNECTED) {
    if (!wasConnected) {
      Serial.print("\n\nBerhasil tersambung ke Wi-Fi: ");
      Serial.println(WIFI_NAME); Serial.print("IP Lokal: ");
      Serial.print(WiFi.localIP().toString());
      if (WiFi.localIP().toString() == "0.0.0.0") {
        Serial.println("IP tidak valid !");
      }
      WiFi.setAutoReconnect(true); WiFi.persistent(true);
      Serial.println();
    }
    wasConnected = true; isWiFiConnecting = false; retry = 0; return;
  }
  if (isWiFiConnecting && currentTime - lastWiFiRTime >= wifiRIntv) {
    lastWiFiRTime = currentTime; retry++; Serial.print(".");
    if (retry >= maxRetry) {
      Serial.print("\n\nGagal terhubung ke WiFi !\n\n\n");
      isWiFiConnecting = false; retry = 0;
    }
  }
  if (!isWiFiConnecting && WiFi.status() != WL_CONNECTED &&
  currentTime - lastWiFiCTime >= 1000) { lastWiFiCTime = currentTime;
    if (wasConnected) { Serial.println("\n\nWi-Fi terputus !\n\n");
    wasConnected = false; } connectWiFi();
  }
}
void connectUbidots() { client.setServer(SERVER, PORT); lastRcnAttempt
= millis(); isUbidotsConnecting = false; }
void checkUbidotsConnection() { if (!client.connected()) { reconnect();
} client.loop(); }
void reconnect() {
  unsigned long currentTime = millis(); if (!client.connected() &&
  (currentTime - lastRcnAttempt >= rcnIntv)) {
    lastRcnAttempt = currentTime; isUbidotsConnecting = true;
    Serial.println("\n\nMenghubungkan ke Ubidots.....");
```

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

```
    if (client.connect(CLIENT_ID.c_str(),UBD_TOKEN,UBD_PASSWORD)) {
      Serial.println("Berhasil tersambung ke Ubidots");
      isUbidotsConnecting = false;
    } else { Serial.print("Gagal, rc = " + String(client.state()));
      isUbidotsConnecting = false;
    }
  }
}
```

Wi-Fi Connectivity Testing

In testing Wi-Fi connectivity, the Arduino IDE Serial Monitor displays the Wi- Fi configuration process on the ESP32 board. The system then displays a message whether Wi-Fi connects successfully or fails.

Figure 7. Wi-Fi Connection

IoT Connectivity Testing

In testing IoT connectivity, the Arduino IDE Serial Monitor displays the IoT configuration process on the ESP32 board. The system then displays an IoT status message whether successfully connected or failed accompanied by an error code.

Figure 8. IoT Connection

Publish-Subscribe Testing on MQTT

The publish and subscribe tests on the MQTT protocol are carried out to ensure that data communication between the ESP32 device and the broker is running properly. In this test, ESP32 successfully received and sent data from variables (topics).

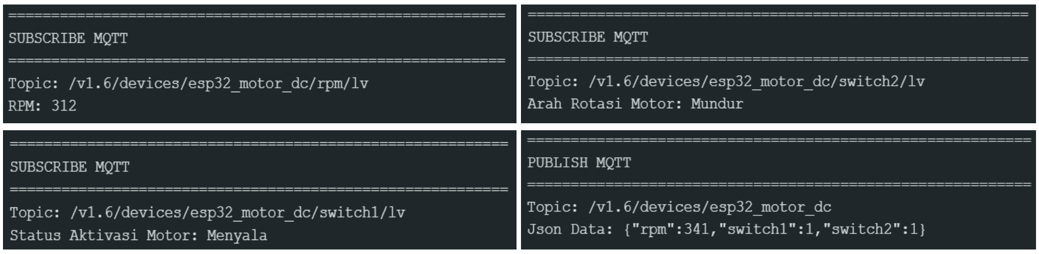

Figure 9. Publish-Subscribe MQTT

I2C LCD Testing

I2C LCD testing is performed to ensure that the screen can display information after being connected to the ESP32. If all information is displayed as expected, then the I2C LCD is declared successfully tested and ready for use in the main application to display motor status and RPM values.

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

Figure 10. I2C LCD Display

DC Motor Rotation Speed (RPM) Testing on the Ubidots Platform

The measurement of the rotational speed (RPM) of a DC motor within a certain time range was carried out using the Ubidots platform. The measurement results are shown in Table 2.

Table 2. Monitoring on the Ubidots Platform

Date	Created At	Timestamp	Value
2025-06-08 13:21:29.350000+07:00	1,749E+12	1,75E+12	210
2025-06-08 13:21:24.352000+07:00	1,749E+12	1,75E+12	210
2025-06-08 13:21:19.350000+07:00	1,749E+12	1,75E+12	261
2025-06-08 13:21:14.404000+07:00	1,749E+12	1,75E+12	382
2025-06-08 13:21:09.366000+07:00	1,749E+12	1,75E+12	312
2025-06-08 13:21:04.308000+07:00	1,749E+12	1,75E+12	329
2025-06-08 13:20:59.305000+07:00	1,749E+12	1,75E+12	300
2025-06-08 13:20:54.288000+07:00	1,749E+12	1,75E+12	319
2025-06-08 13:20:49.126000+07:00	1,749E+12	1,75E+12	326
2025-06-08 13:20:44.071000+07:00	1,749E+12	1,75E+12	287
2025-06-08 13:20:38.990000+07:00	1,749E+12	1,75E+12	316
2025-06-08 13:20:33.945000+07:00	1,749E+12	1,75E+12	333
2025-06-08 13:20:28.861000+07:00	1,749E+12	1,75E+12	324
2025-06-08 13:20:23.842000+07:00	1,749E+12	1,75E+12	382
2025-06-08 13:20:18.844000+07:00	1,749E+12	1,75E+12	316

Figure 11 shows the display of the rotational speed (RPM) graph of a DC motor visualized through the Ubidots platform.

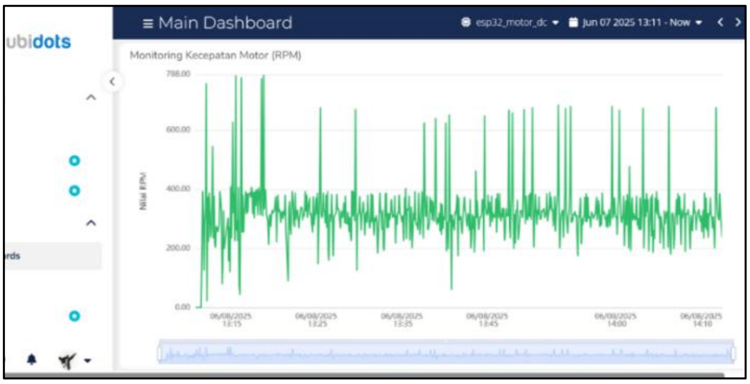

Figure 11. Line Charts Widget

DC Motor Rotation Speed (RPM) Testing on Arduino IDE

The process of measuring the rotational speed (RPM) of a DC motor within a certain time interval is carried out through the Serial Monitor and Serial Plotter on the Arduino IDE, as shown in Figure 12.

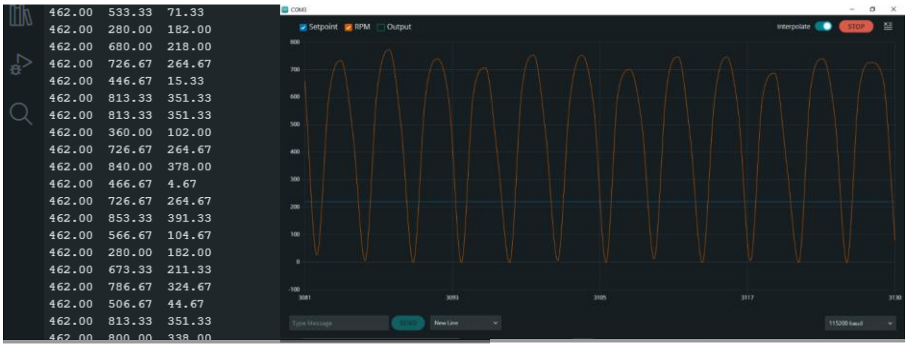

Figure 12. Speed Display on Arduino IDE

Physical and Non-Physical Button Testing

Testing of the DC motor control system on the conveyor was carried out using two types of controls, namely Physical Buttons (Push Buttons) and Non-Physical Buttons (Ubidots). This test aimed to ensure that each type of control functioned properly in activating and deactivating the DC motor (ON/OFF) and regulating the direction of the DC motor rotation (Forward/Reverse). The test results can be seen in Table 3.

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

Table 3. Button Testing Results

No.	Button Type	Function	Action
1	Non-Physical (Ubidots)	ON	Success
2	Non-Physical (Ubidots)	Forward	Success
3	Physical	OFF	Success
4	Non-Physical (Ubidots)	ON	Success
5	Non-Physical (Ubidots)	OFF	Success
6	Non-Physical (Ubidots)	Backward	Success
7	Physical	ON	Success
8	Physical	Forward	Success
9	Non-Physical (Ubidots)	OFF	Success
10	Physical	Backward	Success

Testing the Effect of Item Weight on Conveyor Travel Time

The test aims to determine the travel time of each item when passing through the conveyor line with a certain motor speed. The distance between the starting point and the end point of the conveyor is about ± 30 cm. The test results can be seen in Table 4 below.

Table 4. Results of Goods Transit Time Testing

No.	Object name	Weight	Travel Time at Speed	
			212 (RPM)	492 (RPM)
1	Le Mineral Water 600 ml Bottle	645 grams	17,2 seconds	9,3 seconds
2	Taffware 117 in 1 Toolset Screwdriver	480 grams	6 seconds	1,5 seconds
3	Cleo water 220 ml Bottle	224 grams	1,2 seconds	1 seconds
4	External HDD	166 grams	1,5 seconds	0,8 seconds
5	Kenko Paper Puncher	129 grams	1,4 seconds	0,7 seconds

4. Conclusion

Based on the research that has been carried out by the author, several conclusions can be drawn, including :

1) The system development methodology uses Rapid Application Development (RAD), which has been proven to be efficient in building prototype systems with limited time, without compromising the quality of the final result.
2) On/Off control and rotation control of the DC Motor were successfully implemented, both remotely and closely.

3) Data communication in this system uses the MQTT protocol, which is known to be lightweight and reliable in Internet of Things (IoT) applications. As an IoT platform, Ubidots is used to monitor and control the system in real-time through the internet network.
4) As long as the potentiometer value is in the zero (0) position, the conveyor system will not turn on, even though the activation command has been sent via the Ubidots platform. This condition indicates that speed regulation still requires direct intervention from the operator on site. Therefore, the system needs to be further developed so that the DC motor rotation speed (RPM) setting can be done through the IoT platform, so that operational efficiency and flexibility can be improved.

References

Abdillah, R. (2023). Sistem Kendali Kecepatan Konveyor Dengan Beban Berubah Berbasis Hibrid Fuzzy Logic-Pid. *Jurnal Informatika Dan Teknik Elektro Terapan*, *11*(3), 789–796. https://doi.org/10.23960/jitet.v11i3.3346

Baumann, D., Mager, F., Wetzker, U., Thiele, L., Zimmerling, M., & Trimpe, S. (2021). Wireless Control for Smart Manufacturing: Recent Approaches and Open Challenges. *Proceedings of the IEEE*, *109*(4), 441–467. https://doi.org/10.1109/JPROC.2020.3032633

Danta, B. M., Sentinuwo, S. R., & Putro, M. D. (2016). Implementasi Teknologi Radio Frequency Identification untuk Identifikasi dan Autentikasi pada Gerbang Masuk diUniversitas Sam Ratulangi Manado. *Jurnal Teknik Informatika*, *9*(1), 1–9. https://doi.org/10.35793/jti.9.1.2016.13458

Fahlevi, M. R., Naubnome, V., & Suci, F. C. (2023). Rancang Bangun Mesin Belt Conveyor Dengan Speedcontrol Sebagai Pengatur Kecepatan. *Patria Artha Technological Journal*, *7*(1), 20–26. https://doi.org/10.33857/patj.v7i1.593

Kurniawan, R. Y., & Widyatama, U. (2025). *Penerapan Automasi dan Digitalisasi dalam Industri Strategi Menuju Efisiensi dan Inovasi*. *February*.

Kusuma, D. Y., & Hendry, J. (2024). Purwarupa Sistem Conveyor Scale Berbasis Loadcell, PLC, dan Protokol MQTT dan Analisisnya. *Techno.Com*, *23*(3), 690–703. https://doi.org/10.62411/tc.v23i3.11227

Nurhanudin, & Kartimi. (2025). Memahami Penciptaan, Perkembangan, dan Tantangan Manusia di Era Digital. *Journal on Education*, *7*(2), 9283–9292. https://doi.org/10.31004/joe.v7i2.7868

Parlika, R., Wijaya, D. C. M., & Pratama, A. (2021). BOT PENYIMPAN DATA PENGUMPULAN TUGAS PESERTA ELEARNING BERBASIS

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

TELEGRAM [ER-BOT PDPT]. *SCAN - Jurnal Teknologi Informasi Dan Komunikasi*, *16*(1), 34–41. https://doi.org/10.33005/scan.v16i1.2352

Petru, L., & Mazen, G. (2015). PWM control of a DC motor used to drive a conveyor belt. *Procedia Engineering*, *100*(January), 299–304. https://doi.org/10.1016/j.proeng.2015.01.371

Putra, A. I. (2023). *ANALISIS SISTEM INTERNET OF THINGS PADA PROTOTYPE AUTOMATIC TELESCOPIC CONVEYOR DENGAN PERANGKAT ARDUINO* [Universitas Mercubuana]. http://repository.mercubuana.ac.id/id/eprint/77125.

Romadhon, F. A., & Nawawi, Z. M. (2024). Economic Reviews Journal. *Economic Reviews Journal*, *3*(1), 96–108. https://doi.org/10.56709/mrj.v3i3.525

Saputra, G. Y., Afrizal, A. D., Mahfud, F. K. R., Pribadi, F. A., & Pamungkas, F. J. (2017). Penerapan Protokol MQTT Pada Teknologi Wan (Studi Kasus Sistem Parkir Univeristas Brawijaya). *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *12*(2), 69–75. https://doi.org/10.30872/jim.v12i2.653

Sinaga, D., & Peniarsih. (2024). MENGHADAPI PERUBAHAN DUNIA MELALUI TRANSFORMASI DIGITAL MENUJU KESUKSESAN PADA ERA DIGITALISASI. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, *11*(2), 51–58. https://doi.org/10.31219/osf.io/gdm68

Tamimi, F., & Munawaroh, S. (2024). Teknologi Sebagai Kegiatan Manusia Dalam Era Modern Kehidupan Masyarakat. *Saturnus : Jurnal Teknologi Dan Sistem Informasi*, *2*(3), 66–74. https://doi.org/10.61132/saturnus.v2i3.157

Tugino, U A, E., Basuki, A., & Prasojo, J. (2019). Rancang Bangun Sistem Konveyor Berbasis Internet Of Thing (IOT). *Prosiding Nasional Rekayasa Teknologi Industri Dan Informasi*, *2019*(November), 228–235.

Ushofa, B. D., Anifah, L., Buditjahjanto, I. G. P. A., & Endryansyah. (2022). Sistem Kendali Kecepatan Putaran Motor DC pada Conveyor dengan Metode Kontrol PID. *Jurnal Teknik Elektro*, *11*(2), 332–342. https://doi.org/https://doi.org/10.26740/jte.v11n2.p332-342.