Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2024. Vol. 4. No. 1; e-ISSN: 2808-4519

Design and Build Running Text Controls Using IoT

Aji Tyo Syahputra*, Achmad Fauzi[2], Milli Alfhi Syari[3]

[1, 2, 3] STMIK Kaputama

Ajityo767@gmail.com [1], fauzyrivai88@gmail.com [2], milli.fhisya@gmail.com [3]*

Abstract

This research designs and builds an Internet of Things (IoT)-based running text control system using the P10 display module which can be controlled through the Blynk application and displays the temperature in real-time using the DHT22 sensor. The system consists of hardware such as a microcontroller, Wi-Fi module, P10 LED display, and DHT22 sensor to measure temperature and humidity. The Blynk app is used as a user interface to control the messages displayed on the P10 module and monitor the temperature data in real-time. Software development is carried out to integrate the hardware with the Blynk application and enable efficient data transmission. The test results show that this system is able to accurately display text messages and temperature data with low latency. These systems can be applied in a variety of scenarios such as public information systems, environmental surveillance, and dynamic advertising applications.

***Keywords**: IoT, Running Text, Design.*

1. Introduction

In the ever-evolving digital age, technology is increasingly penetrating various aspects of our lives. One area that has undergone a significant transformation is the display of public information, such as running text, which has become an important element in the urban landscape by providing real-time information to the public. However, with the advancement of the Internet of Things (IoT), the concept of running text has undergone a major change. IoT allows running text to become more adaptive, interactive, and responsive to its environment. With connected sensors, running text can retrieve data directly from the surrounding environment, such as weather, traffic, or even human behavior patterns [1].

The main problem faced in this study is the limitations that occur in controlling and monitoring running text remotely. Currently, running text control is often still done manually or using a central controller that requires direct intervention. This obstacle hinders flexibility and ease of access, especially when users want to manage running text from far away [2].

2. Literature Review

2.1. Design

Design is the initial stage of making drawings and sketches that have never been made at all and then managed into images or sketches that have the desired function. System design, or design, is a series of processes for translating the results of a system into a programming language, the purpose of which is to provide a detailed description of how the existing components will be implemented. Meanwhile, the definition of construction or system construction is the activity of creating a new system, replacing all or part of an existing system, or improving it. Design refers to the design of a system and a single unit to design and build a system [3].

2.2. System Control

A control system is a system of regulation or control of one or several quantities (variables, parameters) so that they are at a certain price (range). In the industry, a safe and efficient work process is needed to produce products with good quality and quantity within a certain period of time. Automation is helpful in terms of smooth operation, safety (investment, environment) (production costs), product quality, etc [4].

2.3. Running Text

Running text is a mobile or running display with electronic technology consisting of a series of LEDs (light emitting diodes) connected in a matrix with a combination of LEDs between rows and columns to provide information. The public uses LEDs. Running text is an effective

information and advertising medium because it can attract attention, produce a lot of content (written content), and is simple. In contrast to information and advertising media that use billboards, banners, and posters that are static.

2.4. Internet of Things

The Internet of Things (IoT) is a concept that aims to expand the benefits of continuously connected internet connectivity. Essentially, IoT refers to objects that can be uniquely identified as virtual representatives in an internet-based structure [5].

2.5. Esp8266

ESP8266 is a WiFi module that functions as an additional microcontroller device such as an Arduino, so it can be connected directly to WiFi to establish a TCP/IP connection. Moreover, this module is SOC (Single on Circuit) based which means you can use this device without the help of another microcontroller. The module requires about 3.3V of power and has three WiFi modes: Station, Access point, and both. The module also has a processor, memory, and GPIO, with the number of pins depending on the type of ESP8266 used.

2.6. Blynk

Blynk is an application service to control microcontrollers through the Internet. The application provided by blynk itself must be compiled as needed. The use of the Blynk application in this study is due to the simplicity of the implementation of the Blynk program using a microcontroller, the ease of installation on smartphones, the ability to adjust the appearance of the application according to preferences, and the fact that the Blynk application is free [6].

2.7. Arduino IDE

Arduino IDE is a software used to write programs that contain commands and upload them to a microcontroller for its application. The writing of program code is done to provide commands using the C programming language which aims to run the system so that it can work according to the program code that has been filled into an Arduino. Without program code, the system cannot work because program code is the most important part of building a tool.

2.8. DHT22

The DHT22 sensor is a digital relative humidity and temperature sensor. The DHT22 sensor uses capacitors and fog to measure the ambient air and output signals on the data pins. The DHT22 sensor is very stable, reliable, and the calibration feature provides very accurate results so it is also very easy to implement on Arduino microcontrollers.

3. Analysis and Design

3.1. System Design

The system that will be studied today is a running text control system using the Internet of Things. This control system requires a local internet network and ESP8266 as a general requirement in designing a running text control system. and requires a Blynk that will be connected to the running text as a controller or modifier for the message in the running text.
The process of changing this message will be carried out using the Internet of things where Blynk is the medium in control. Blynk will send data to the microcontroller, then the microcontroller will give a command to the running text that will display the message [7].

3.2. Block Diagram Network

The block diagram of the designed system, as shown in the Figure 1:

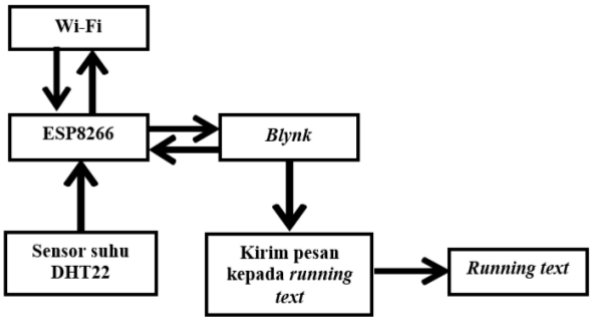

Fig.1: Block Diagram Network

The description of the block diagram above is as follows:
The first process is to connect the hardware used such as a smartphone or computer, and run text to Wi-Fi, send messages or texts through the Blynk application that is already connected through the smartphone or computer used. Then the result data will be displayed in the running text. Then, with a DHT22 temperature sensor that will read the room temperature and display it on the LCD in Real-Time.

3.3. Overall Tool Set

The following is an overview of the running text IoT electronics circuit schematic:

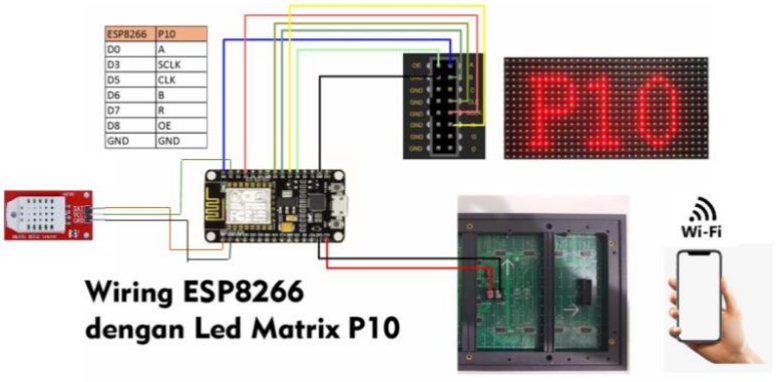

Fig.2: Overall Tool Set

The explanation of the electronic circuit scheme in the design of Running text hardware is as follows:
In the running text design, the pins used to connect the running text and the microcontroller ESP8266 are: pin D_1 is connected to the DHT22 sensor, pins + and – on the DHT sensor are connected to VCC and GND at ESP8266, pin D_0 on ESP8266 is connected to pin A running text, then pin D_3 is connected to the SCLK pin, then pin D_5 is connected to the CLK pin, pin D_6 with pin B, pin D_7 with pin R, pin D_8 with pin OE, and lastly pin GND on ESP8266 connected with pin GND on running text.

3.4. System Flowchart

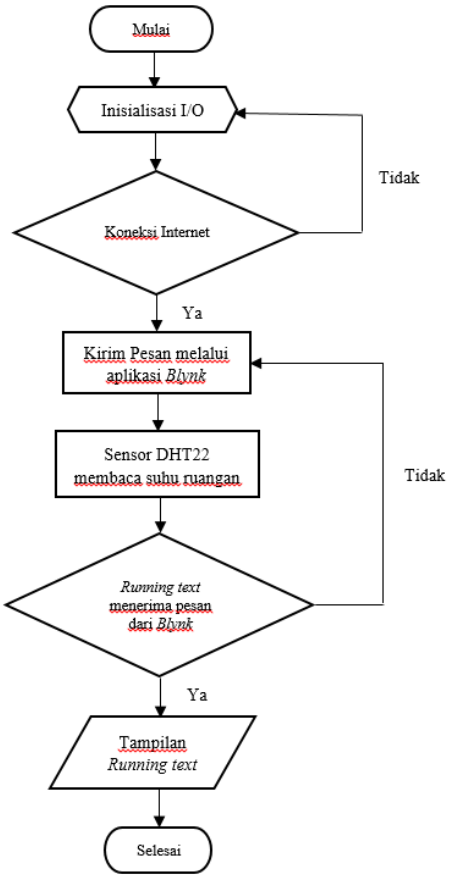

Fig. 3: Flowchart System Control

Description:

1. Get started.
2. Initialization of Inputs and Outputs.
3. Connect Blynk and Running text to the internet.
4. Make sure both are connected to the internet, if they are not connected, they will return to the original connection.
5. Next, if you are connected to the internet. Send a new text using Blynk that is already connected to the internet to the Running text.
6. Then, the DHT22 sensor reads the room temperature that will be displayed on the LCD in real-time.

7. If the running text does not receive a message from blynk, it will refill the text from the blynk application.
8. And finally, the Running text will display the new text that has been sent through Blynk earlier.
9. Done.

4. Discussion and Implementation

4.1. Discussion

In this chapter, the author describes and explains the results of the research by conducting tests. The tests that will be carried out are software and hardware testing.

4.2. Software Testing

To test the program, the initial steps in this experiment are as follows:

1. Open the Arduino IDE software with the unfilled program like this:

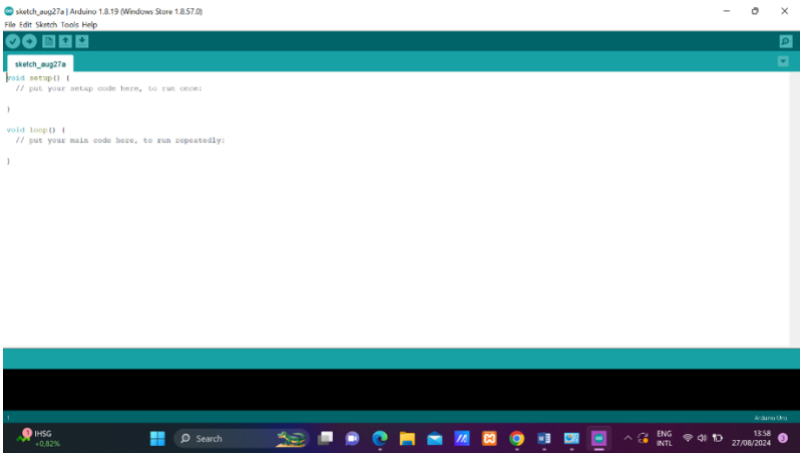

Fig. 4: Arduino IDE Initial Look

2. Enter the program code for the connection below:

```
#include <BlynkSimpleEsp8266.h>
#include <Wire.h>
#include "DHT.h"
#include <DMDESP.h>
#include <EEPROM.h>
#include <fonts/EMSans8x16.h>

#define DISPLAYS_WIDE 2 // Kolom Panel
#define DISPLAYS_HIGH 1 // Baris Panel
#define DHTPIN D1
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
DMDESP Disp(DISPLAYS_WIDE, DISPLAYS_HIGH);

char ssid[] = "Running Text";
char pass[] = "RT123456";
```

Fig. 5: Blynk Connection Program

3. Then resume the previous program, for the timer and set the speed on the running text display:

```
float humid,temp;
unsigned long last;
String suhu;
int timeSuhu = 5;
int timeText = 10;
int addTxt = 0;
int addSpd = 500;
int spd;
int cnt;
static char *teks[] = {""};
int proses;
bool flag;
String txt;
```

Fig.6: Program For Time and Set the Speed on The Running Text Display

4. Next, the program for Blynk, set brightness and text size:

```
void setup() {
  Serial.begin(9600);
  EEPROM.begin(512);
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  dht.begin();
  Disp.start();
  Disp.setBrightness(30);
  Disp.setFont(EMSans8x16);
  spd = EEPROM.read(addSpd);
  txt = readStringFromEEPROM(addTxt);
  teks[0] = const_cast<char*>(txt.c_str());
  Serial.print("Speed : ");Serial.println(spd);
  Serial.print("Text : ");Serial.println(teks[0]);
}
```

Fig.7: Program For Blynk, Set Brightness and Text Size

5. After that program for temperature display:

```
void tampilSuhu(){
  cekSuhu();
  suhu = String(temp,1);
  Disp.drawText(8,0,suhu);
  Disp.drawCircle(43, 1, 1);
  Disp.drawText(47,0,"C");
}

void cekSuhu(){
  humid = dht.readHumidity();
  temp = dht.readTemperature();
  if (isnan(humid) || isnan(temp)) {
    Serial.println(F("Failed to read from DHT sensor!"));
    return;
  }
```

Fig.8: Program For Temperature Display

4.3. Hardware Testing

Once all the programs are typed, design the Hardware as shown in Figure 9:

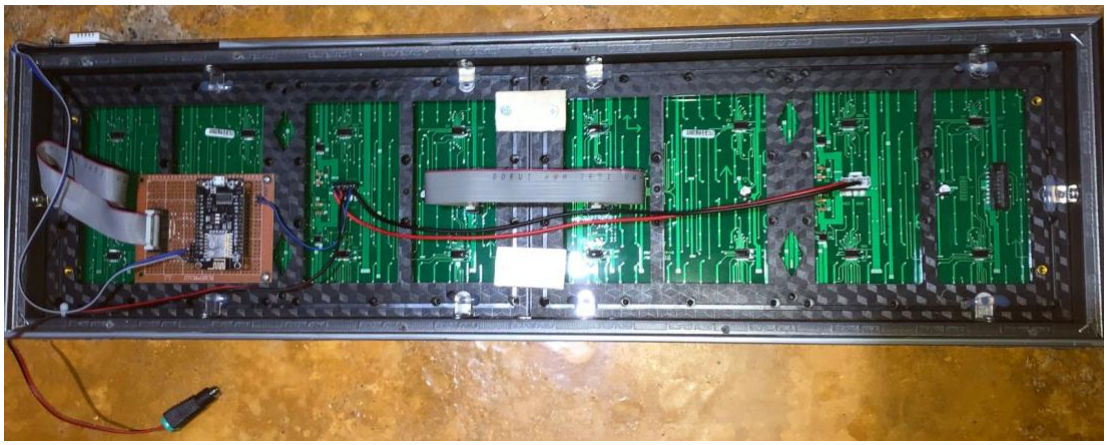

Fig. 9: DHT Sensor Connected to the Running Text Circuit

4.4. Blynk Testing

In this Blynk test, we created a new template with a widget like the image below:

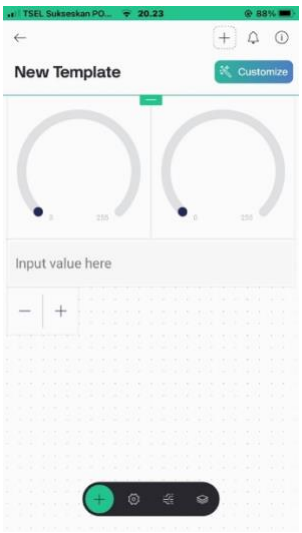

Fig. 10: Blynk Display

After the program has been typed, the next step is to input the program code into the circuit by clicking the Bar menu on the Arduino IDE then click upload with the note Board and Port on the Arduino IDE Bar menu is complete. Next, wait a few moments for the upload process to complete, then the program that has been uploaded will be automatically saved to the microcontroller.

4.5. Implementation

The implementation of this test is carried out to determine the performance of the components that will be used in this thesis with the output in the form of Blynk which will control the Running text by sending a message for the display of the Running text according to the command sent through the Smartphone as well as the room temperature display that is displayed in Real-Time. By being connected to the internet, the microcontroller can receive messages sent by Blynk.
This experiment was carried out by sending messages through Blynk over a long distance and ensuring that the message and display on the Running text had been successfully replaced. And the real-time temperature display experiment was carried out with cold and hot rooms. After all the circuits have been completed in "Design and Build Running Text Control Using IoT", here is a picture of the success of the IoT Running text test in Figures 11 and 12 below:

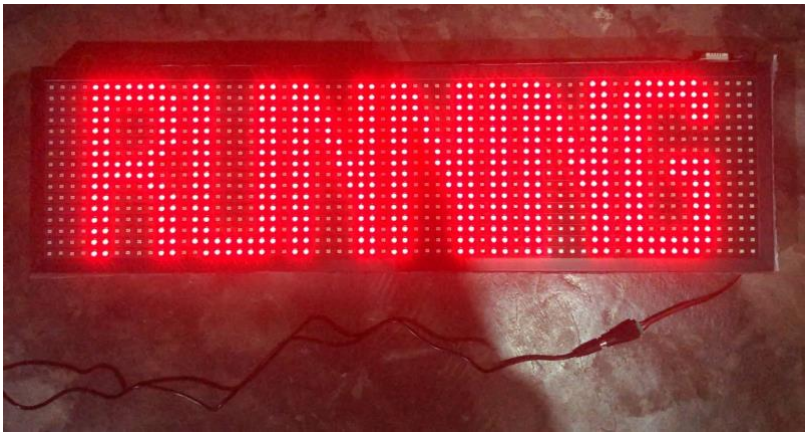

Fig. 11: Display of Running Text After Sending Text via Blynk

Fig. 12: Real-Time Temperature Display

5. Conclusion

After carrying out the design and manufacturing stage of the system which is then continued with the testing and analysis stage, the following conclusions can be drawn:

1. This system is designed to improve efficiency and ease in managing Running text, because users do not need to make manual changes to the hardware.
2. Users can easily access the system remotely via the internet network. Although it succeeded in achieving its main goal, this study still has limitations, such as limited network coverage and response speed that depends on the quality of the internet connection used locally.

Reference

[1] Surahman, Ade, Agung Tri Prastowo, and Lutvi Ashari Aziz. "Design a Honda Beat Motorcycle Safety Device Based on Sim Gsm Using the Design and Build Method." Journal of Embedded Technology and Systems 3.1 (2022).

[2] Faroqi, Adam, Mada Sanjaya WS, and Riyan Nugraha. "Design of lamp automatic control system using arduino-based voice recognition method." TELKA-Journal of Telecommunications, Electronics, Computing and Control 2.2 (2016): 106-117.

[3] Widya, Helma, et al. "Design and Build Running Text Led Display Prayer Time Schedule Based on Arduino Uno as Information Media." JET (Journal of Electrical Technology) 5.2 (2020): 61-67.

[4] NURUL HIDAYATI LUSITA DEWI, NURUL HIDAYATI LUSITA DEWI. Smart home prototype with nodemcu esp8266 module based on the internet of things (iot). Diss. MAJAPAHIT ISLAMIC UNIVERSITY, MOJOKERTO, 2019.

[5] Prayitno, Wahyu Adi, Adharul Muttaqin, and Dahnial Syauqy. "Hydroponic Plant Temperature, Humidity, and Watering Control System using Blynk Android." Journal of Information Technology and Computer Science Development 1.4 (2017): 292-297.

[6] Samsugi, Selamet, Zainabun Mardiyansyah, and Andi Nurkholis. "Automatic irrigation control system using Arduino UNO microcontroller." Journal of Embedded Technology and Systems 1.1 (2020): 17-22.

[7] Roihan, Ahmad, et al. "Simulation of Soil Moisture Detection Using Dht22 Sensor with Proteus." METHODIKA: Journal of Informatics and Information Systems Engineering 7.1 (2021): 25-30.