# SU 2024 IOT102 Digital Clock

1st Nguyen Hoang Tue Sang, 2nd Mach Gia Hao, 3rd Tran Chi Thuan, 4th Luong Hong My and Duc Ngoc Minh Dang
FPT University, Ho Chi Minh Campus, Vietnam
{sangnhtse184333, haomgse184349 thuantcse184519, mylhse184354}@fpt.edu.vn, and ducdnm2@fe.edu.vn

***Abstract*—Put the abstract here.**

## I. Introduction

In the era of smart devices and the Internet of Things (IoT), the development of multifunctional digital clocks has emerged as a significant research problem. This project aims to address this problem by designing and implementing a multifunctional digital clock that not only displays time but also integrates several additional features.

The challenge lies in creating a user-friendly device. The integration of these features requires a deep understanding of sensor technology and programming techniques. The clock uses an LCD1602 display for its good visibility, readability, and energy efficiency.

The ESP8266 microcontroller is used for data processing and internet connectivity, enabling the clock to retrieve and display real-time weather data. This report will detail the design considerations, implementation process, and performance evaluation of the multifunctional digital clock.

## II. Main Proposal

**Necessity and Topicality**: In the era of IoT and smart devices, the integration of various functionalities into a single device has become a necessity. The digital clock project addresses this need by combining time display, environmental monitoring, and health tracking into a single device. The use of ESP8266 microcontroller and Arduino platform makes this project highly topical, as these are widely used in IoT applications.

**Scientific Significance**: The project demonstrates the application of several scientific concepts and technologies. It uses the ESP8266 microcontroller for data processing and internet connectivity, enabling real-time weather data retrieval from the OpenWeatherMap API. It also uses the Arduino platform for sensor data processing and display. The project thus contributes to the field of IoT and embedded systems by demonstrating how various technologies can be integrated to create a multifunctional device.

**Practical Significance**: The digital clock has significant practical implications. It serves as a personal health monitoring device, an environmental sensor, and a weather station, in addition to displaying time. This makes it a valuable tool in both personal and professional settings. Furthermore, the potential for future enhancements such as integration with smart home systems and personal health tracking adds to its practical significance.

This proposal provides a high-level overview of the project. The detailed design and implementation process, performance evaluation, and potential future enhancements will be discussed in the subsequent sections of the report.

### A. System models and block diagram

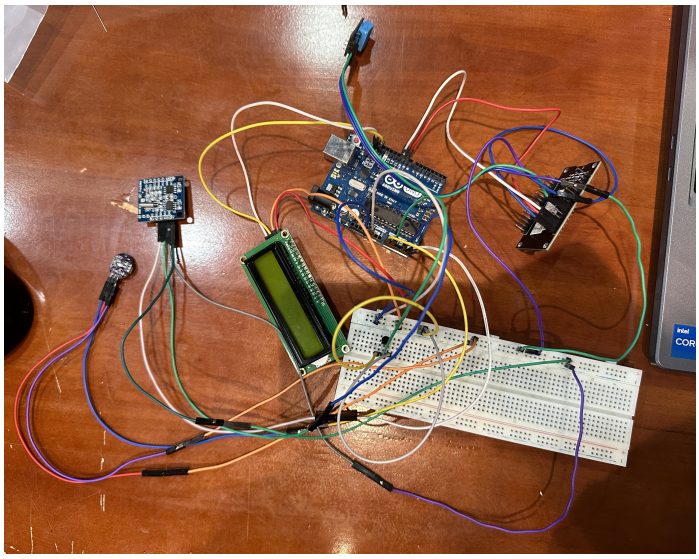

Fig. 1. Model Picture

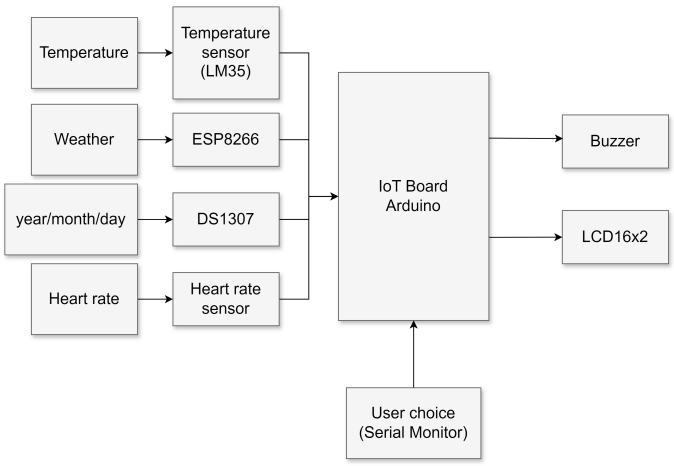


Fig. 2. Block diagram of the developed system.

### B. Components and peripheral devices

LM35, DS1307, ESP8266, LCD16x2, Heart Rate sensor, Buzzer, Arduino Uno R3

### C. Software programming

File unoCode.ino:

```
    #include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <DHT.h>
#include <DHT_U.h>
#define USE_ARDUINO_INTERRUPTS true
LiquidCrystal_I2C lcd(0x27, 16, 2); // I2C address 0x27, 16 columns, 2 rows
#include <PulseSensorPlayground.h>
#include "Adafruit_Sensor.h"
#include "RTClib.h"
//received user
int receivedInt = 0;
int option = 0;
//Temp room
int val;
int tempPin = 1;
const int PulseWire = 0;
#define RX_PIN 9 // RX pin of SoftwareSerial
#define TX_PIN 10 // TX pin of SoftwareSerial
DHT dht(7, DHT11);
PulseSensorPlayground pulse;
SoftwareSerial esp(RX_PIN, TX_PIN); // Create a SoftwareSerial object
int alarmHour = 0; // Initialize alarm hour
int alarmMinute = 0; // Initialize alarm minute
RTC_DS1307 rtc;

void showMenu() {
  Serial.println("GROUP 6");
  Serial.println("------------------------");
  Serial.println("1. Real Time");
  Serial.println("2. Temperature in room");
  Serial.println("3. Heart rate");
  Serial.println("4. Weather From Internet");
  Serial.println("5. Exit");
  Serial.println("------------------------");
  Serial.println("Please enter number: ");
}


void setup() {
  Serial.begin(9600);
  esp.begin(2400);
  pulse.analogInput(PulseWire);
  pulse.blinkOnPulse(13);
  pulse.setThreshold(515);
  dht.begin();

  if (! rtc.begin()) {
    Serial.println("Couldnt find RTC");
  }
  else
  {
    Serial.println("Found RTC");

  }
  if (pulse.begin())
  {
    Serial.println("PulseSensor object created successfully!");
  }
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running, lets set the time!");
  }
  rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));


 lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  showMenu();
  receivedInt = 0;
}


void loop() {
  if (receivedInt == 0) {
    receivedInt = getSerialInt();
    Serial.print("Your choice: ");
    Serial.println(receivedInt);
  }

  // Now you can perform actions based on the receivedInt value
  if (receivedInt == 1) {
    // Your code for option 1
    while (receivedInt == 1) {
      showRealTime(); // Chay lien tuc hom cho tay chon 1
      checkForNewOption();
    }
  } else if (receivedInt == 2) {
```

```
    // Your code for option 2
    while (receivedInt == 2) {
      showTemperature(); // Chạy liên tục hàm cho tùy chọn 2
      checkForNewOption();
    }
  } else if (receivedInt == 3) {
    while (receivedInt == 3) {
      showBPM(); // Chạy liên tục hàm cho tùy chọn 3
      checkForNewOption();
    }
  } else if (receivedInt == 4) {
    while (receivedInt == 4) {
      showTemperatureFromInternet(); // Chạy liên tục hàm cho tùy chọn 4
      checkForNewOption();
    }
  } else {
    // Handle other cases if needed
    Serial.println("!INPUT SAI!");
  }

  showMenu();
  receivedInt = 0;
}

void checkForNewOption() {
  if (Serial.available() > 0) {
    int newInt = getSerialInt(); // Lấy giá trị mới từ Serial
    if (newInt == 5) {
      receivedInt = newInt; // Nếu giá trị mới là 5, cập nhật receivedInt
      lcd.clear();
    }
  }
}


int getSerialInt() {
  int retInt;
  Serial.flush();

  while (!Serial.available()) {} // Waits for an input on the serial device
  retInt = Serial.parseInt(); // Takes the Serial input and looks for an integer
  Serial.parseInt(); // throw away carriage return
  Serial.flush();

  return retInt;
}

void clearRXBuffer() {
  int x;
  while (x = esp.available() > 0) {
    while (x--) esp.read();
  }
}
//REAL TIME
void showRealTime() {
  lcd.clear();
  lcd.setCursor(0, 0);

  DateTime now = rtc.now();
  lcd.print("Date:");
  lcd.print(now.year());

  lcd.print("/");
  lcd.print(now.month());

  lcd.print("/");
  lcd.print(now.day());

  lcd.setCursor(0, 1);
  lcd.print("Time: ");
  lcd.print(now.hour());
  lcd.print(":");
  lcd.print(now.minute());
  lcd.print(":");
  lcd.print(now.second());


  delay(1000);
}
//SHOW TEMP INTERNET

String getValue(String data, char separator, int index) {
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length() - 1;

  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0], strIndex[1]) : "";
}
```

```
185 void showTemperatureFromInternet() {
186   if (esp.available() > 0) {
187     String data_recived = esp.
          readStringUntil('\n');
188     Serial.print("Data received:");
189     Serial.println(data_recived);
190     Serial.print("Data length:");
191     Serial.println(data_recived.length())
          ;
192     Serial.flush();
193     esp.flush();
194     String temp = getValue(data_recived,
          ';', 0);
195     float t = temp.toFloat();
196     t -= 273.15; //convert to Celcius
197     String humid = getValue(data_recived,
          ';', 1);
198     String pressure = getValue(
          data_recived, ';', 2);
199     clearRXBuffer();
200     lcd.clear();
201     lcd.setCursor(0, 0);
202     lcd.print("Temp: ");
203     lcd.print(t);
204     lcd.print("C-");
205     lcd.setCursor(0, 1);
206     lcd.print("Humid: ");
207     lcd.print(humid);
208     lcd.print("%");
209     delay(5000);
210   }
211 }
212
213 // SHOW TEMP IN ROOM
214 void showTemperature()
215 {
216   val = analogRead(tempPin);
217   float mv = (val / 1023.0) * 5000;
218   float cel = mv / 10;
219
220   lcd.clear();
221   lcd.setCursor(0, 0);
222   lcd.print("TEMP = ");
223   lcd.print(cel);
224   lcd.print("*C");
225   Serial.print("TEMPRATURE = ");
226   Serial.print(cel);
227
228   Serial.print("*C");
229   Serial.println();
230   delay(1000);
231
232 }
233
234 //HEART
235 void showBPM() {
236   lcd.clear();
237   lcd.setCursor(0, 0);
238   if (pulse.sawStartOfBeat())
239   {
240     lcd.print("Nhip tim:");
241     int currentBPM = pulse.
          getBeatsPerMinute();
242     lcd.print(currentBPM);
243     Serial.println(currentBPM);
244     lcd.print(" BPM");
245   }
246   else
247   {
248     lcd.print("BPM stop...");
249   }
250   delay(1000);
251 }
```

File esp8266Humidity.ino:

```
1     #include <ESP8266WiFi.h>
2 #include <ESP8266HTTPClient.h>
3 #include <WiFiClient.h>
4 #include <Arduino_JSON.h>
5 #include <SoftwareSerial.h>
6 #define RX_PIN 4
7 #define TX_PIN 5
8
9 SoftwareSerial esp826(RX_PIN,TX_PIN);
10 const char* ssid = "FPTU_Student";
11 const char* password = "12345678";
12 String openWeatherMapApiKey = "8
     ffd9d8ce7d7fb8ac5777e1206dfcb35";
13 String city = "Ho Chi Minh";
14 String countryCode = "VN";
15 unsigned long lastTime = 0;
16 unsigned long timerDelay = 20000;
17
18 String jsonBuffer;
19 void setup() {
20   Serial.begin(9600);
21   esp826.begin(2400);
22   WiFi.begin(ssid, password);
23   Serial.println("Connecting");
24   while (WiFi.status() != WL_CONNECTED) {
25     delay(500);
26     Serial.print(".");
27   }
28   Serial.println("");
29   Serial.print("Connected to WiFi network
       with IP Address: ");
30   Serial.println(WiFi.localIP());
31
32   Serial.println("Timer set to 20 seconds
       (timerDelay variable), it will take
       20 seconds before publishing the
       first reading.");
```

```
}

void loop() {
  // Send an HTTP GET request
  if ((millis() - lastTime) > timerDelay) {
    // Check WiFi connection status
    if (WiFi.status() == WL_CONNECTED) {
      String serverPath = "http://api.openweathermap.org/data/2.5/weather?q=" + city + "," + countryCode + "&APPID=" + openWeatherMapApiKey;

      jsonBuffer = httpGETRequest(serverPath.c_str());
      Serial.println(jsonBuffer);
      JSONVar myObject = JSON.parse(jsonBuffer);

      // JSON.typeof(jsonVar) can be used to get the type of the var
      if (JSON.typeof(myObject) == "undefined") {
        Serial.println("Parsing input failed!");
        return;
      }

      Serial.print("JSON object = ");
      Serial.println(myObject);
      Serial.print("Temperature: ");

      String temp = JSON.stringify(myObject["main"]["temp"]);

      String humid = JSON.stringify(myObject["main"]["humidity"]);

      String pressure = JSON.stringify(myObject["main"]["pressure"]);
      Serial.println(temp);

      Serial.print("Humidity: ");
      Serial.println(humid);

      Serial.print("Pressure: ");
      Serial.println(pressure);

      String Data = temp + ";" + humid + ";" + pressure;

      Serial.print("Data sent to Arduino is : " );

      Serial.println(Data);
      esp826.println(Data);
      esp826.flush();

    }
    else {
      Serial.println("WiFi Disconnected");
    }
    lastTime = millis();
  }
}

String httpGETRequest(const char* serverName) {
  WiFiClient client;
  HTTPClient http;

  // Your IP address with path or Domain name with URL path
  http.begin(client, serverName);

  // Send HTTP POST request
  int httpResponseCode = http.GET();

  String payload = "{}";

  if (httpResponseCode > 0) {
    Serial.print("HTTP Response code: ");
    Serial.println(httpResponseCode);
    payload = http.getString();
  }
  else {
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
  }
  // Free resources
  http.end();

  return payload;
}
```

### D. Programming Flowchart

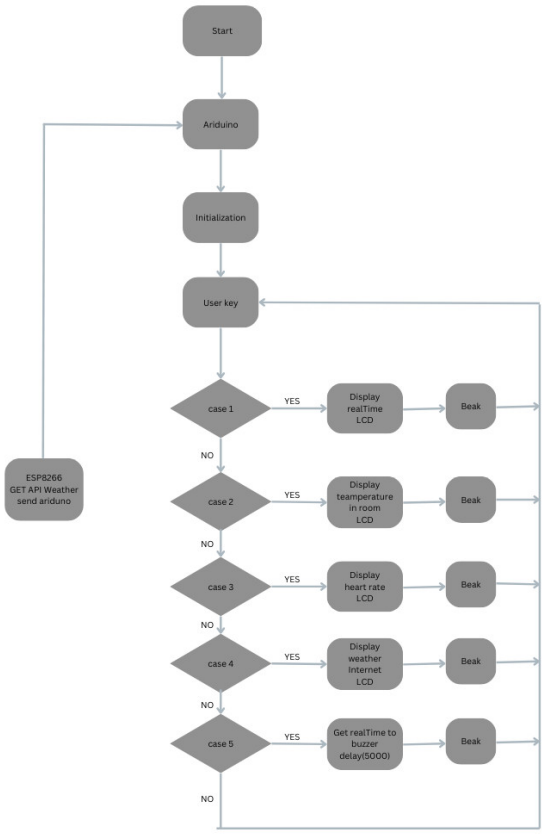


Fig. 3. Model Picture

## III. Results and discussion

A clear concise description of the research methods used.

### A. Prototype Implementation

After verifying the code, open the Serial Monitor screen. Press 1 to display real time by LCD. Press 2 to display room temperature and humidity on LCD. Press 3 to display heart rate on LCD. Press 4 to display the weather on the LCD. Press 5 to set the timer. When the time is right, the buzzer will sound

### B. Experimental Results

The outcomes of the experiment are notably encouraging when contrasted with the date, time, and the present readings of heart rate and temperature.

### C. Discussion

Presenting real-time data such as temperature/humidity and heart rate on an I2C LCD via Arduino is an excellent utilization of the LCD, enabling local, tangible interaction. The ability to update data over the network is a valuable addition. Some factors to consider would include the use of a wifi or ethernet shield for network connectivity, deciding on a remote data source, and appropriately parsing/formatting the data for transmission to the Arduino.

TABLE I
Research Plan

| No | Task | Student Name |
|---|---|---|
| 1 | Make models, Code | Mach Gia Hao |
| 2 | Make models, Code | Tran Chi Thuan |
| 3 | Slide, Block Diagram | Luong Hong My |
| 4 | Overleaf, flowchart | Nguyen Hoang Tue Sang |

## IV. Conclusion

Give a clear concise description of the project's outputs [1]

*This is an example of a citation.* For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [2]–[5] [6].

New here [7]

## REFERENCES


[1] D. Li, D. Zhao, Y. Chen, and Q. Zhang, "Deepsign: Deep learning based traffic sign recognition," in *2018 international joint conference on neural networks (IJCNN)*. IEEE, 2018, pp. 1–6.

[2] D. N. M. Dang, D. H. Ngoc, N. Vandung, H. Zaw, and C. S. Hong, "Hermac: A hybrid efficient and reliable mac for vehicular ad hoc networks," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*. IEEE, 2014, pp. 186–193.

[3] T. Anh Khoa, C. H. Phuc, P. D. Lam, L. M. B. Nhu, N. M. Trong, N. T. H. Phuong, N. V. Dung, N. Tan-Y, H. N. Nguyen, and D. N. M. Duc, "Waste management system using iot-based machine learning in university," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–13, 2020.

[4] E. Khorov, A. Lyakhov, and R. Yusupov, "Two-slot based model of the ieee 802.11ah restricted access window with enabled transmissions crossing slot boundaries," in *IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2018, pp. 1–9.

[5] D. N. M. Dang, C. S. Hong, and S. Lee, "A hybrid multi-channel mac protocol for wireless ad hoc networks," *Wireless Networks*, vol. 21, pp. 387–404, 2015.

[6] N. T. Pham, S. D. Nguyen, V. S. T. Nguyen, B. N. H. Pham, and D. N. M. Dang, "Speech emotion recognition using overlapping sliding window and shapley additive explainable deep neural network," *Journal of Information and Telecommunication*, vol. 0, no. 0, pp. 1–19, 2023.

[7] T. Anagnostopoulos, A. Zaslavsky, K. Kolomvatsos, A. Medvedev, P. Amirian, J. Morley, and S. Hadjieftymiades, "Challenges and opportunities of waste management in iot-enabled smart cities: a survey," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 3, pp. 275–289, 2017.