



BÁO CÁO THỰC HÀNH

Bài thực hành số 01: Arduino và chứng thực

Môn học: Bảo mật Internet of Things

Lớp: NT535.P21.1

THÀNH VIÊN THỰC HIỆN (Nhóm xx):

STT	Họ và tên	MSSV
1	Phạm Thiệu Gia Khang	21520967
2		

Điểm tự đánh
giá

7/10

ĐÁNH GIÁ KHÁC:

Tổng thời gian thực hiện	2 tuần
Phân chia công việc	
Ý kiến (nếu có) + Khó khăn + Đề xuất, kiến nghị	

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

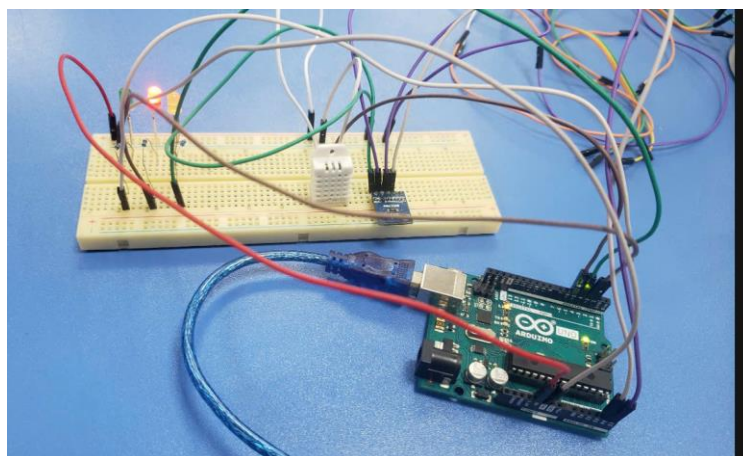
MỤC LỤC

A. BÁO CÁO CHI TIẾT	2
1. Thu dữ liệu từ cảm biến và điều khiển đèn:	2
2. Chức năng chứng thực sử dụng Arduino:.....	5
3. Giải pháp nâng cao bảo mật:	10
B. TÀI LIỆU THAM KHẢO	10

A. BÁO CÁO CHI TIẾT

1. Thu dữ liệu từ cảm biến và điều khiển đèn:

Mô hình:



Kịch bản gồm 1 DHT22, 1 BH1750, 3 đèn LED và 1 Arduino UNO.

- Đèn xanh: sẽ bật sáng khi ánh sáng xung quanh quá thấp
- Đèn đỏ: sẽ bật sáng khi nhiệt độ thay đổi ngoài ngưỡng bình thường (quá nóng, quá lạnh).
- Đèn vàng: Sẽ bật sáng khi độ ẩm thay đổi ngoài ngưỡng bình thường (quá khô, quá ẩm).

Code:

```
#include "DHT.h"
#include <BH1750.h>
#include <Wire.h>
#include "Arduino.h"

#define LED1 2
#define LED2 3
#define LED3 4

#define DHTPIN 5
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);
BH1750 lightMeter;
void setup() {
    Serial.begin(9600);

    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);
    pinMode(LED3, OUTPUT);
    # Khởi tạo các cảm biến
    dht.begin();
    Wire.begin();
    lightMeter.begin();
    Serial.println(F("BH1750 Test begin"));
}

void loop() {
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    digitalWrite(LED3, LOW);

    float h = dht.readHumidity();
    float t = dht.readTemperature();
    float lux = lightMeter.readLightLevel();
    # Khi ánh sáng quá thấp thì bật LED đỏ
    if (lux <= 50) {
        digitalWrite(LED1, HIGH);
    }
    # Khi nhiệt độ nằm ngoài ngưỡng an toàn thì bật LED xanh
    if (t <= 23 || t >= 28) {
        digitalWrite(LED2, HIGH);
    }
    # Khi độ ẩm nằm ngoài mức an toàn thì bật LED vàng
    if (h <= 60 || h >= 70) {
        digitalWrite(LED3, HIGH);
    }

    Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("% Temperature: "));
```

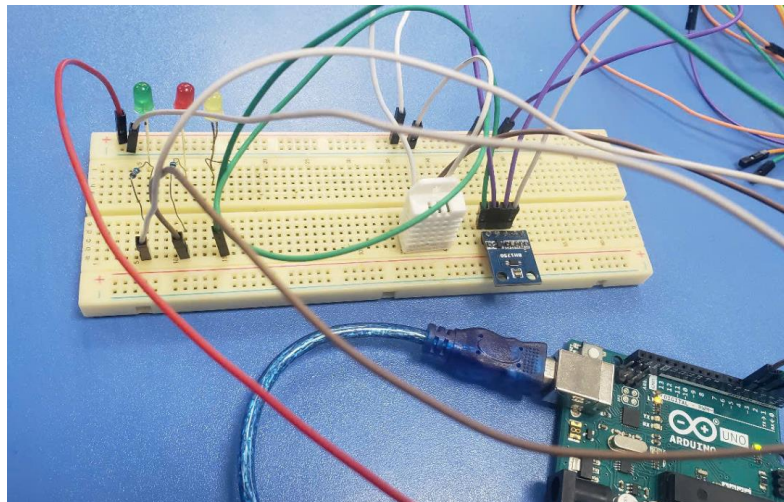
```
Serial.print(t);
Serial.print(F("C Light: "));
Serial.print(lux);
Serial.println(F(" lx"));

delay(5000);
}
```

Khi các giá trị nằm trong ngưỡng an toàn:

```
10:08:58.038 -> Humidity: 60.20% Temperature: 25.10C Light: 277.50 lx
10:09:03.051 -> Humidity: 60.20% Temperature: 25.10C Light: 280.83 lx
10:09:08.072 -> Humidity: 60.30% Temperature: 25.10C Light: 285.83 lx
10:09:13.069 -> Humidity: 60.30% Temperature: 25.10C Light: 285.00 lx
10:09:18.074 -> Humidity: 60.30% Temperature: 25.10C Light: 262.50 lx
10:09:23.098 -> Humidity: 60.30% Temperature: 25.10C Light: 281.67 lx
10:09:28.133 -> Humidity: 60.30% Temperature: 25.10C Light: 283.33 lx
10:09:33.117 -> Humidity: 60.50% Temperature: 25.10C Light: 281.67 lx
10:09:38.139 -> Humidity: 60.60% Temperature: 25.10C Light: 282.50 lx
10:09:43.144 -> Humidity: 60.50% Temperature: 25.10C Light: 280.83 lx
10:09:48.169 -> Humidity: 60.40% Temperature: 25.10C Light: 280.83 lx
```

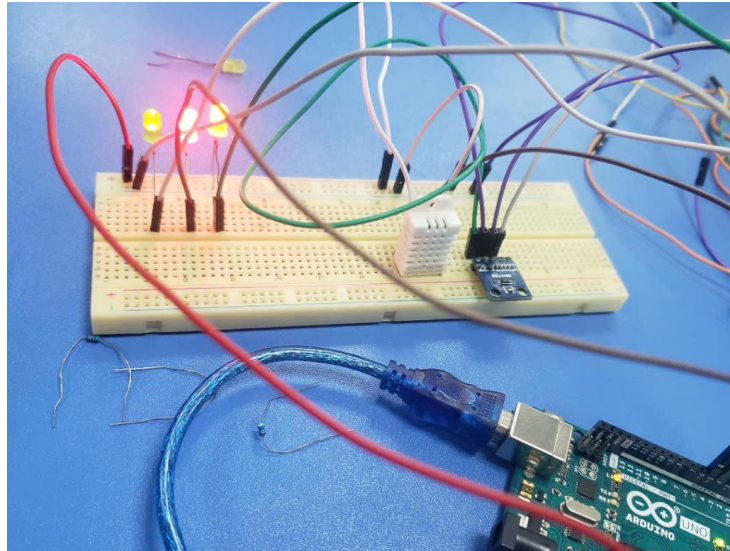
Cả 3 đèn đều tắt:



Chỉ các giá trị test nằm ngoài mức an toàn:

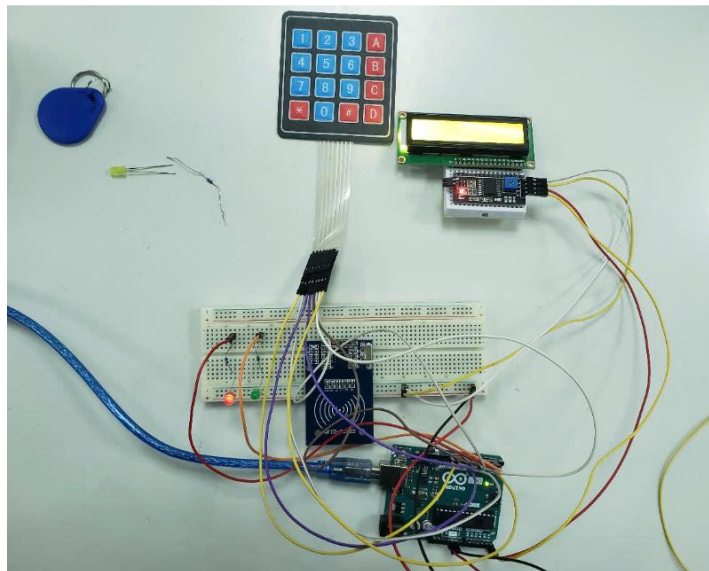
```
10:13:07.138 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:12.175 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:17.159 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:22.189 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:27.177 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:32.174 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:37.207 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:42.209 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:47.213 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:52.222 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:13:57.207 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
10:14:02.216 -> Humidity: 10.00% Temperature: 10.00C Light: 10.00 lx
```

Cả 3 đèn đều sáng:



2. Chức năng chứng thực sử dụng Arduino:

Mô hình:



Các thiết bị bao gồm 1 Arduino UNO, 1 LCD – I2C, 1 Keypad 16x4 còn lại RFID em trienr khai không thành công.

Kịch bản:

- Khi khởi động, LCD hiển thị “SCAN RFID OR PIN” và đèn đỏ luôn sáng.
- Nhập mật khẩu, nếu đúng đèn xanh bật và đèn đỏ tắt trong 5s, hiển thị “Access grants”. Ngược lại đèn đỏ nháy 5s hiển thị “Access denied”.
- Nhập * để reset key.

Code:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <MFRC522.h>
#include <SPI.h>
#include <Keypad.h>
#define RED_LED 3
#define GREEN_LED 2

LiquidCrystal_I2C lcd(0x27, 16, 1);

const byte ROWS = 4;
const byte COLS = 4;

char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'8','9','7','C'},
  {'*','0','#','D'}
};

byte rowPins[ROWS] = {11, 10, 9, 8};
byte colPins[COLS] = {7, 6, 5, 4};

Keypad keypad = Keypad(makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);

String enteredPin = "";
const String correctPin = "2356";

void setup() {
  Serial.begin(9600);

  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  SPI.begin();

  lcd.init();
  lcd.backlight();
  resetToPhase1();
}

void loop() {
  char key = keypad.getKey();
  if (key) {
    Serial.print("Key Pressed: ");
    Serial.println(key);
    processKeyInput(key);
  }

  delay(10);
}
```



```

}
# Hàm kiểm tra input từ keypad reset nếu nhập * và kết thúc nếu đủ 4 ký
tự
void processKeyInput(char key) {
    if (key == '*') {
        enteredPin = "";
        lcd.setCursor(0, 0);
        lcd.print("Enter PIN:      ");
        Serial.println("PIN Cleared");
    }
    else if (enteredPin.length() < 4 && isDigit(key)) {
        enteredPin += key;
        lcd.setCursor(enteredPin.length() - 1, 0);
        lcd.print('*');

        Serial.print("Entered PIN: ");
        Serial.println(enteredPin);

        if (enteredPin.length() == 4) {
            checkPIN();
        }
    }
}
# Kiểm tra mã Pin được nhập
void checkPIN() {
    if (enteredPin == correctPin) {
        authenticateSuccess();
    } else {
        authenticateFail();
    }
    enteredPin = "";
}
# Hàm khi xác nhận thành công đèn đỏ tắt, đèn xanh bật
void authenticateSuccess() {
    digitalWrite(RED_LED, LOW);
    digitalWrite(GREEN_LED, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Authenticated");
    Serial.println("Access Granted");
    delay(5000);
    resetToPhase1();
}
# Khi xác nhận thất bại đèn đỏ nhấp 5 lần
void authenticateFail() {
    lcd.clear();

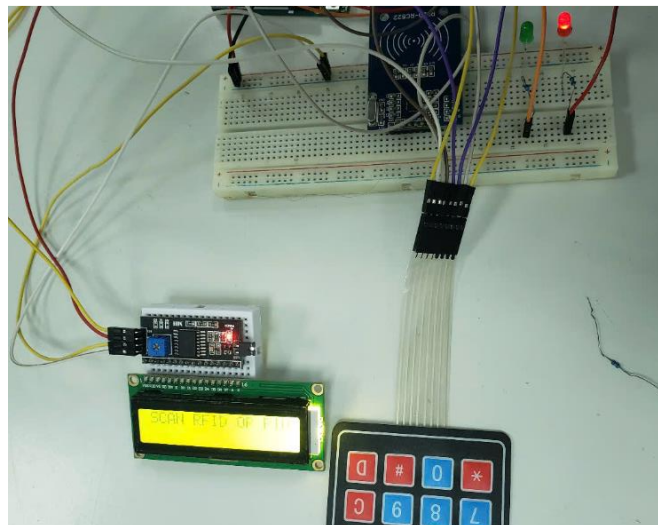
```

```

lcd.setCursor(0, 0);
lcd.print("Access Denied");
Serial.println("Access Denied");
for (int i = 0; i < 5; i++) {
    digitalWrite(RED_LED, LOW);
    delay(300);
    digitalWrite(RED_LED, HIGH);
    delay(300);
}
delay(2000);
resetToPhase1();
}
# Khi hoàn thành 1 vòng lặp qua trở lại bước 1
void resetToPhase1() {
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("SCAN RFID OR PIN:");
    Serial.println("Phase reseat");
    enteredPin = "";
}

```

Demo:



Khi bắt đầu, LCD hiển thị SCAN RFID OR PIN và đèn đỏ luôn bật.

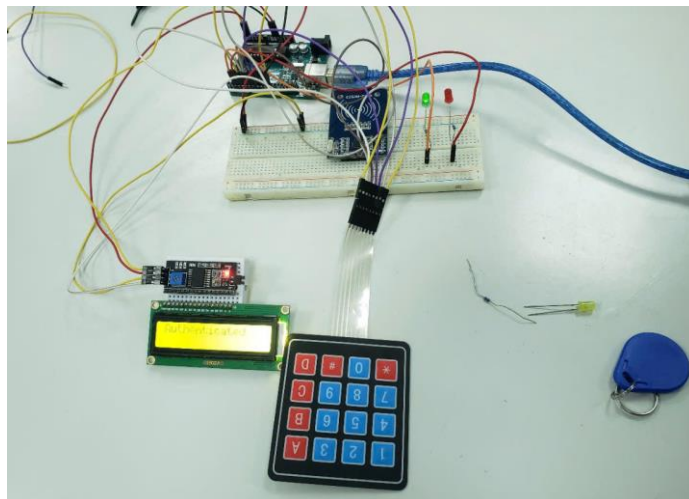
Trường hợp nhập đúng mã pin 2356:

```

15:49:49.725 -> Key Pressed: 2
15:49:49.725 -> Entered PIN: 2
15:49:51.146 -> Key Pressed: 3
15:49:51.146 -> Entered PIN: 23
15:49:51.995 -> Key Pressed: 5
15:49:52.027 -> Entered PIN: 235
15:49:52.797 -> Key Pressed: 6
15:49:52.830 -> Entered PIN: 2356
15:49:52.830 -> Access Granted
15:49:57.880 -> Phase reseat

```

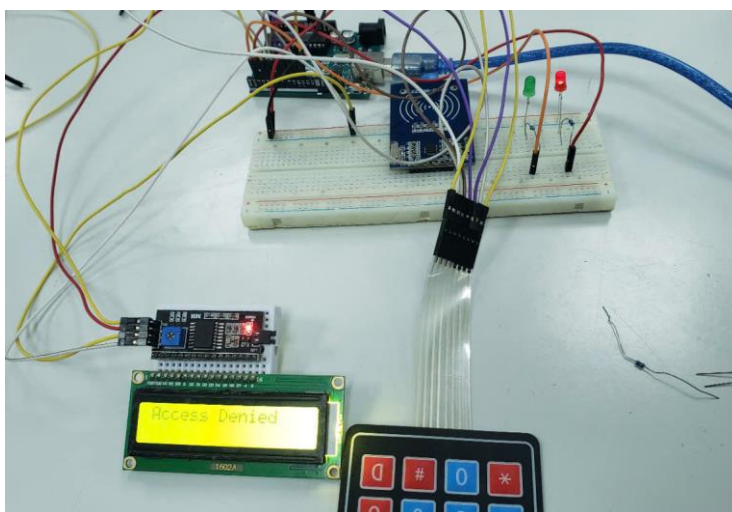
Đèn đỏ tắt, đèn xanh bật, lcd hiển thị "Authenticated", sau 5s sẽ reseat trở lại phase 1:



Nếu nhập sai:

```
15:49:57.880 -> Phase reseat
15:53:38.233 -> Key Pressed: 2
15:53:38.233 -> Entered PIN: 2
15:53:38.934 -> Key Pressed: 2
15:53:38.934 -> Entered PIN: 22
15:53:39.663 -> Key Pressed: 2
15:53:39.663 -> Entered PIN: 222
15:53:40.441 -> Key Pressed: 2
15:53:40.473 -> Entered PIN: 2222
15:53:40.518 -> Access Denied
15:53:45.534 -> Phase reseat
```

Hiện thị Access Denied và đèn đỏ nháy 5 lần:



Nhấn * để Reset key:

```
15:53:45.534 -> Phase reseat
15:55:42.288 -> Key Pressed: 2
15:55:42.333 -> Entered PIN: 2
15:55:42.698 -> Key Pressed: 2
15:55:42.730 -> Entered PIN: 22
15:55:43.447 -> Key Pressed: 2
15:55:43.480 -> Entered PIN: 222
15:55:44.639 -> Key Pressed: *
15:55:44.683 -> PIN Cleared
```

Video demo tại đây [Video demo.](#)

3. Giải pháp nâng cao bảo mật:

Đối với bài 1 đo giá trị các cảm biến, các dữ liệu này có thể dễ dàng bị thay đổi thông qua các yếu tố ngoại cảnh tới các thiết bị vậy nên cần bao bọc các cảm biến trong không gian an toàn hơn.

Đối với bài toán sử dụng keypad và RFID, mật khẩu 4 ký tự có thể dễ dàng bị phá giải thủ công nếu không giới hạn số lần nhập cũng như thẻ ID có thể bị sao chép.

Nên giới hạn số lần nhập mã PIN và sử dụng mã PIN dài hơn cũng như có độ phức tạp cao hơn.

B. TÀI LIỆU THAM KHẢO