**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**TRUNG TÂM THÍ NGHIỆM THỰC HÀNH**

**BÀI THÍ NGHIỆM MÔN HỌC IoT VÀ ỨNG DỤNG**

*(Tài liệu dành cho sinh viên nghành CNTT)*

**Họ tên sinh viên:**....................................

**Mã sinh viên:**…………………………..

**Lớp:**…………………………………….

**BÀI THÍ NGHIỆM SỐ 1**

**CÁC CẢM BIẾN VÀ LẬP TRÌNH THU NHẬN DỮ LIỆU CẢM BIẾN SỬ DỤNG ARDUINO ESP32**

1. **MỤC ĐÍCH, YÊU CẦU**
   1. **Mục đích:**

* Giúp sinh viên hiểu rõ hơn về các thiết bị vật lý trong IoT và nắm được cấu tạo, nguyên tắc hoạt động cơ bản của một số loại cảm biến thông dụng như cảm biến nhiệt độ, độ ẩm, cảm biến âm thanh, ánh sáng, cảm biến siêu âm …
* Nắm được các giao thức truyền tin, ghép nối giữa cảm biến và các bộ xử lý, vi điều khiển như KIT Arduino ESP32 hoặc máy tính Rasberry Pi.
* Nắm được cách thức lập trình cho vi điều khiển và thu nhận dữ liệu thô của các cảm biến dùng ngôn ngữ lập trình C trên môi trường phát triển tích hợp IDE Arduino 2.x
  1. **Yêu cầu:**
* Sinh viên đã được học lý thuyết Chương 1 và 2 của môn học *IoT và ứng dụng*.
* Sinh viên đã có tài liệu thí nghiệm cho môn học *IoT và ứng dụng.*
* Sinh viên đã được học về ngôn ngữ lập trình C/C++. Kỹ thuật vi xử lý và kiến trúc máy tính.

1. **CHUẨN BỊ**

- Sinh viên đọc trước ở nhà nội dung bài thí nghiệm.

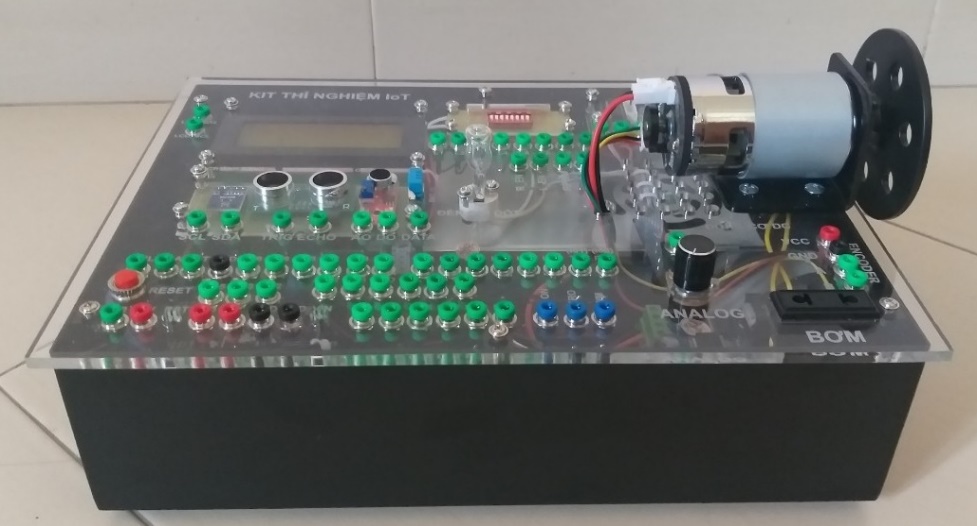
- Xem lại nội dung lý thuyết Chương 1, 2 môn học *IoT và ứng dụng*.

**2.1. Giới thiệu tổng quát về bộ KIT thí nghiệm IoT và phần mềm Arduino IDE**

KIT thí nghiệm IoT là thiết bị thực hành điện tử đa năng giúp người học có khả năng thực hiện việc xây dựng các ứng dụng IoT đơn giản khi kết hợp với các hệ thống mạng Internet và máy tính. KIT thí nghiệm IoT có thành phần chính là là board mạch Arduino ESP32 cùng với các thành phần điện tử ngoại vi phụ trợ tạo nên một hệ vi xử lý đầy đủ. Mỗi một khối mạch điện đều được xây dựng và hoạt động một cách độc lập, các khối mạch điện này có thể được kết nối với nhau thông qua hệ thống các dây cắm để tạo ra các chức năng hoạt động hỗn hợp phức tạp hơn. Hệ thống KIT thí nghiệm IoT bao gồm:

* **Vi xử lý ESP32** (KIT Arduino ESP32 WeMOS D1 R32) được kết nối với máy tính thông qua giao tiếp USB phục vụ việc nạp mã chương trình, gỡ rối…
* **Cơ cấu hiển thị**: bao gồm hệ thống 8 LED đơn đấu Cathode chung và có thể được điều khiển bật/tắt bằng hệ thống chuyển mạch nút gạt. Một màn hình tinh thể lỏng LCD 20x4 có gắn module giao tiếp I2C.
* **Cơ cấu chấp hành**: bao gồm 01 động cơ bước 5Volt-28BYJ-48 được điều khiển thông qua module điều khiển động cơ ULN2003A. 01 động cơ một chiều 24Vdc (CHR775S) có gắn encoder cảm biến từ hiệu ứng Hall. Động cơ một chiều điều khiển bởi mạch điều khiển động cơ BTS-7960. 01 đèn sợi đốt 220Vac được điều khiển bởi mạch công suất với thành phần cơ bản là Triac và mạch điện phát hiện điểm 0. Một lối ra nguồn nuôi điện áp 220V có bảo vệ (thường ngắt) điều khiển được bằng Triac (dùng để điều khiển đóng cắt thiết bị điện tử công suất như máy bơm nước, đèn chiếu sáng, quạt điện…vv).
* **Các cảm biến** nhiệt độ, độ ẩm, cảm biến siêu âm, cảm biến âm thanh và ánh sáng, encoder cảm biến từ hiệu ứng Hall. Hệ thống cũng có thể kết nối trực tiếp với mạng Internet thông qua tính năng thu/phát WiFi của KIT ESP32.
* **Khối Analog** gồm có một nguồn điện một chiều thay đổi được từ 0V÷5Vdc bằng cách vặn biến trở. Khối này có nhiệm vụ cung cấp điện áp cho chức năng chuyển đổi ADC của KIT ESP32.

Hệ thống được nối với máy tính Windows có cài đặt chương trình Arduino IDE.



*Hình vẽ 1.1. KIT thí nghiệm IoT*



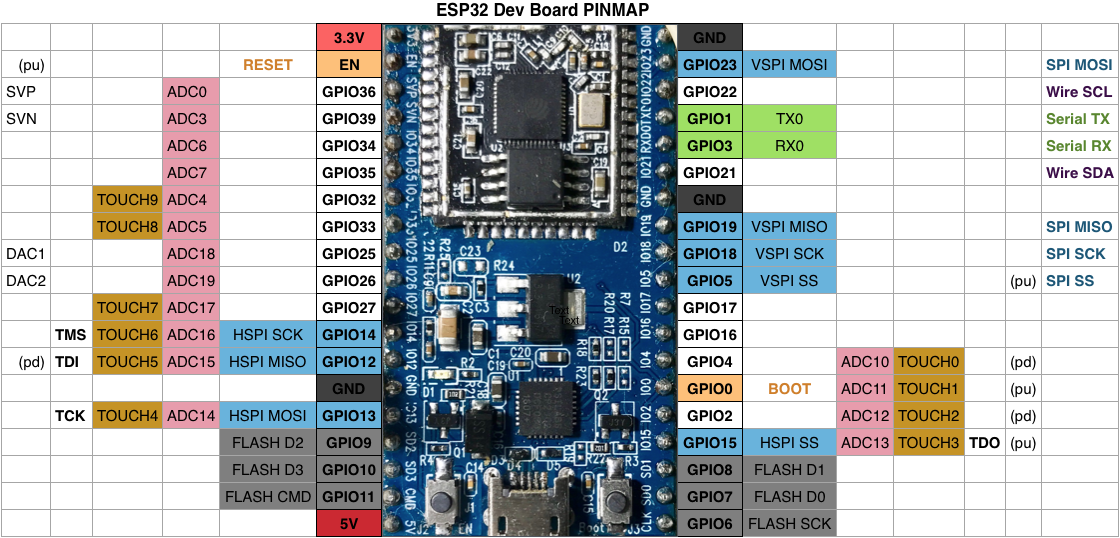
*Hình vẽ 1.2. Sơ đồ khối chức năng của KIT thí nghiệm IoT*

### KIT Arduino ESP32 WeMOS D1 R32

Trung tâm của KIT thí nghiệm IoT được xây dựng trên nền tảng KIT Arduino ESP32 We Mos D1 R32 (có nhân là chip ESP32-WROOM-32). ESP32 được sử dụng trên KIT thí nghiệm IoT tương thích và được IDE Arduino nhận diện dưới tên ESP32 Dev Module (*chú ý: chọn ESP32 Dev module khi kết nối Kit ESP32 với máy tính chạy Arduino IDE*).



*Hình vẽ 1.3. KIT Arduino ESP32 WeMOS D1 R32*



*Hình vẽ 1.4. Tên và chức năng các chân trên KIT ESP32*



*Hình vẽ 1.5. Sơ đồ khối chức năng của chip ESP32*

Arduino WiFi WeMos D1 R32 ESP32 BLE có thiết kế với hình dạng tương tự  Arduino Uno hay WeMos D1 R2. Được thiết kế với mục đích ứng dụng công nghệ WiFi. Module WiFi WeMos D1 R32 được tích hợp công nghệ WiFi, BLE và nhân ARM SoC, KIT có cách sử dụng, lập trình tương tự như KIT RF thu phát WiFi Bluetooth ESP32.

Một số đặc trưng kỹ thuật cơ bản của KIT ESP32 WeMos D1 R32:

* Nguồn nuôi 5Vdc qua cổng USB.
* Nguồn nuôi 5÷12Vdc qua giắc cấp nguồn DC tròn hoặc chân Vin.
* Nhân xử lý trung tâm: Wireless Module – ESP-WROOM-32 dựa trên Espressif ESP32 dual core Tensilica LX6 processor with 802.11 b/g/n WiFi and Bluetooth 4.2 LE.
* Tần số hoạt động lên tới 240Mhz.
* Chip lõi bao gồm 49 chân với 32 chân cắm vào/ra đầy đủ, thiết kế chuẩn chân cắm GPIO và kích thước tương tự KIT Arduino Uno.
* Tích hợp mạch nạp và giao tiếp UART CH340.
* Bộ nhớ trong
* 448 KBytes ROM cho booting và các tính năng của lõi chip.
* 520 KBytes SRAM trên chip dùng cho dữ liệu và các lệnh.
* 8 KBytes SRAM trong RTC (gọi là RTC SLOW Memory) để truy xuất bởi các bộ đồng xử lý co-processor.
* 8 KBytes SRAM trong RTC (gọi là RTC FAST Memory) dùng cho lữu dữ liệu, truy xuất bởi CPU khi RTC đang boot từ chế độ Deep-sleep.
* 1 Kbit EFUSE, với 256 bit cho hệ thống (địa chỉ MAC và cấu hình chip), 768 bit còn lại cho ứng dụng người dùng, gồm cả mã hóa bộ nhớ Flash và định ID cho chip.
* Kết nối WiFi
* Wi-Fi: 802.11 b/g/n/e/i.
* Bluetooth: BR/EDR phiên bản v4.2 và BLE.
* Ethernet MAC hỗ trợ chuẩn: DMA và IEEE 1588.
* Bus hỗ trợ mang CAN 2.0
* Giao tiếp ngoại vi
* Bộ chuyển đổi ADC 12 bit, 16 kênh.
* Bộ chuyển đổi 8-bits DAC: 2 kênh.
* 10 chân để giao tiếp với cảm biến chạm (touch sensor).
* IR (TX/RX)
* Ngõ ra PWM cho điều khiển Motor
* LED PWM: 16 kênh.
* Cảm biến Hall tích hợp sẵn.
* Cảm biến nhiệt độ tích hợp sẵn.
* 4 cổng giao tiếp SPI.
* 2 cổng giao tiếp I²S.
* 2 cổng giao tiếp I²C.
* 3 cổng giao tiếp UART.
* Nhiệt độ hoat động ổn định: -40C đến 85C.
* Điện áp hoạt động: 2.2-3.6V.
* Dòng tiêu thụ ổn định: 80mA.
* Bảo mật
* IEEE 802.11 hỗ trợ các chuẩn bảo mật: WFA, WPA/WPA2 và WAPI.
* Mã hóa Flash.
* 1024-bit OTP, 768-bit cho người dùng.

### Viết và nạp chương trình cho KIT Arduino ESP32

Để viết chương trình cho ESP32 chúng ta sử dụng phần mêm Arduino IDE 2.x. Đây là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino. Đây là một phần mềm Arduino chính thức và nó có các phiên bản cho các hệ điều hành như MAC, Windows, Linux và chạy trên nền tảng Java đi kèm với các chức năng và lệnh có sẵn đóng vai trò quan trọng để gỡ lỗi, chỉnh sửa và biên dịch mã trong môi trường phát triển tích hợp.

Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác. Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin dưới dạng mã.

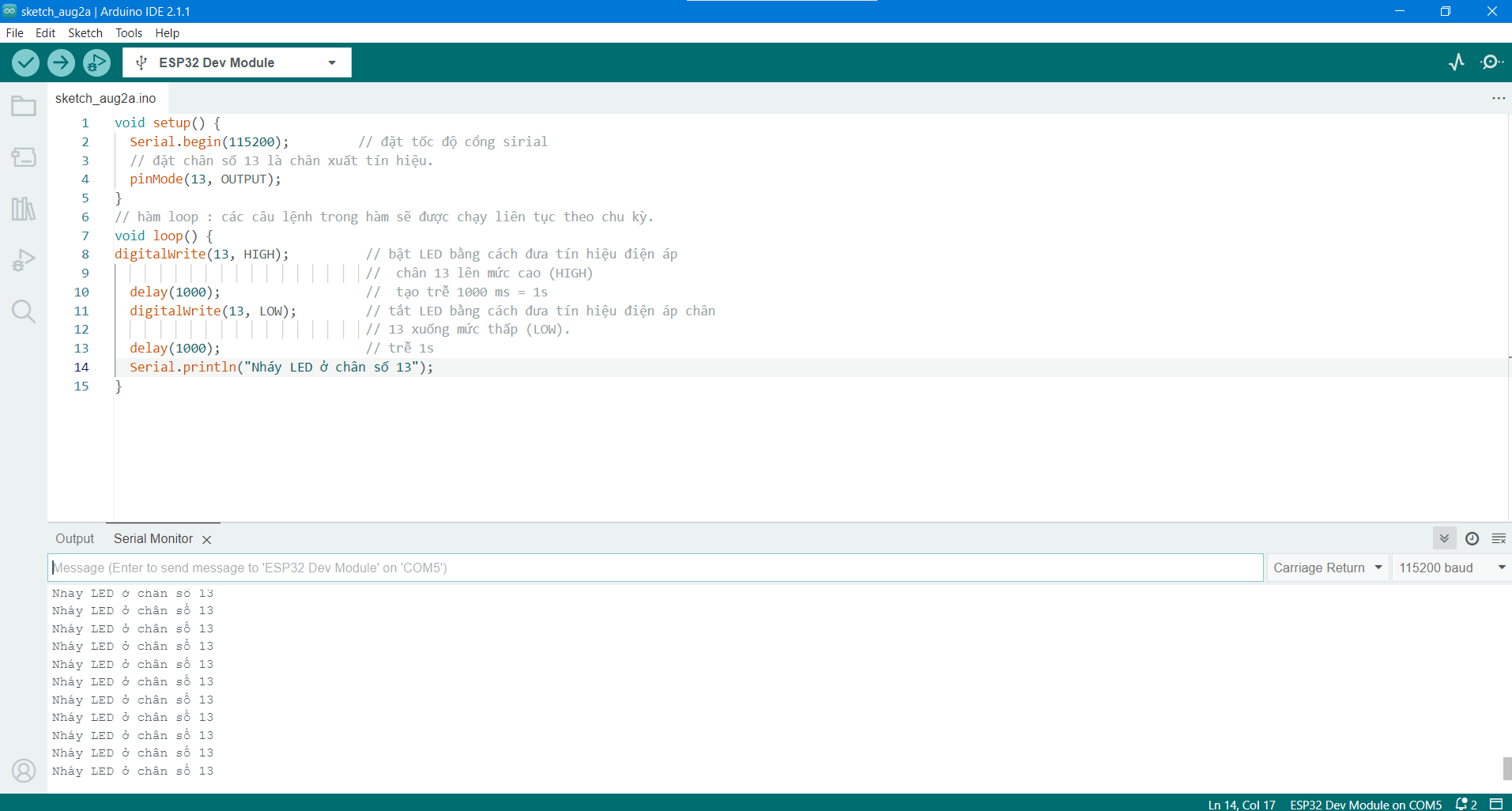
Một chương trình viết cho Arduino, còn được gọi là một sketch, được tạo trên nền tảng Arduino IDE sẽ tạo ra một file Hex, sau đó file này được tải lên (upload) bộ vi điều khiển trên bảng mạch Arduino.

Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình soạn thảo và Trình biên dịch, phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino. Môi trường này hỗ trợ cả ngôn ngữ C và C ++. Khi người dùng viết mã và biên dịch trên máy tính, IDE sẽ tạo file Hex cho mã. File Hex là một tập tin chứa nội dung chương trình cùng các thông tin chỉ dẫn với 6 loại HEX record khác nhau theo định dạng Intel HEX. Mỗi record sẽ bao gồm nhiều ký tự ASCII biểu diễn các giá trị theo mã Hexadecimal và sau đó được gửi đến board mạch bằng cáp USB.

Mỗi board Arduino đều được tích hợp một bộ vi điều khiển, bộ vi điều khiển sẽ nhận file Hex và chạy theo mã được viết.

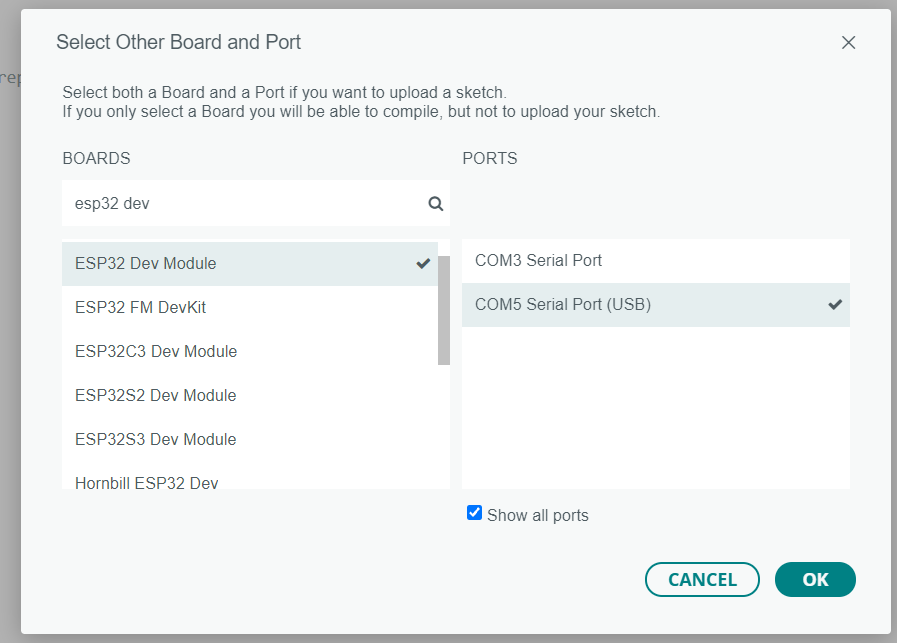
Để viết và nạp chương trình cho KIT Arduino ESP32, thực hiện các bước sau:

* Cắm dây nối nguồn AC220V cho KIT thí nghiệm IoT, bật công tắc nguồn nuôi.
* Nối dây nối USB giữa cổng USB của máy tính và lối vào kết nối USB trên KIT thí nghiệm IoT.
* Khởi động chương trình soạn thảo và biên dịch mã nguồn Arduino IDE (*Chú ý: luôn cập nhật phiên bản mới nhất có thể*).



*Hìn vẽ 1.6. Cửa sổ chương trình Arduino IDE 2.1.1*

* Lựa chọn bảng mạch: ESP32 Dev Module
* Chọn cổng COM ảo cho phù hợp để truyền dữ liệu từ bảng mạch lên terminal của máy tính (tùy từng máy tính khác nhau số hiệu cổng sẽ khác nhau). Trong ví dụ này, cổng COM được chọn là COM5.



*HÌnh vẽ 1.7. Lựa chọn đúng số hiệu cổng COM để kết nối KIT thí nghiệm IoT và máy tính trên phần mềm Arduino IDE*

- **Cấu trúc một chương trình trong Arduino IDE**

Một chương trình hiển thị trên cửa sổ giao diện được gọi là Sketch. Một Sketch được tạo từ hai hàm cơ bản là **setup ()** và **loop ()** :

**Setup()** : Hàm này được gọi khi một sketch khởi động, được sử dụng để khởi tạo biến, đặt các chế độ chân (nhận hay xuất tín hiệu), khởi động một thư viện …vv. Hàm setup() chỉ chạy một lần, sau khi cấp nguồn hoặc reset mạch.

**Loop():** Sau khi khởi tạo hàm setup(), hàm loop() sẽ được khởi tạo và thiết lập các giá trị ban đầu như tên gọi. Hàm loop tạo các vòng lặp liên tục, có cho phép sự thay đổi và đáp ứng. Chức năng tương tự như vòng lặp while() trong C, hàm loop() sẽ điều khiển toàn bộ hoạt động của hệ thống mạch điện.

Sau khi chương trình được viết xong, một sketch sẽ được compile bằng cách nhấn nút



*Hình vẽ 1.8. Nút Compile để biên dịch chương trình*

Tiếp theo nhấn nút Upload để nạp file Hex cho vi điều khiển trên board Arduino



*Hình vẽ 1.9. Nút Upload để nạp file Hex cho vi điều khiển trên board Arduino*

**Ví dụ**: Chương trình điều khiển bật tắt đèn LED đơn trên KIT thí nghiệm IoT

Trong ví dụ này sẽ sử dụng chân số 13 của KIT Arduino để điều khiển bật, tắt một đèn LED đơn trên KIT thí nghiệm Arduino. Nối dây giữa chân số 13 với ký hiệu IO13 trên KIT thí nghiệm IoT với chân L1 tương ứng đèn LED số 1 trong dãy 8 đèn LED được đánh số từ L1 tới L8.

* Viết chương trình trong Arduino IDE như sau:

|  |
| --- |
| void setup() {    Serial.begin(115200);         // đặt tốc độ cổng serial    pinMode(13, OUTPUT);          // đặt chân số 13 là chân xuất tín hiệu.  }  // hàm loop : các câu lệnh trong hàm sẽ được chạy liên tục theo chu kỳ.  void loop() {  digitalWrite(13, HIGH);          // bật LED bằng cách đưa tín hiệu điện áp                                   // chân 13 lên mức cao (HIGH)    delay(1000);                   // tạo trễ 1000 ms = 1s    digitalWrite(13, LOW);         // tắt LED bằng cách đưa tín hiệu điện áp                                   //chân 13 xuống mức thấp (LOW).    delay(1000);                   // tạo trễ 1s    Serial.println("Nháy LED ở chân số 13");                                   // in ra dòng chữ “Nháy LED ở chân số 13                                   // ra cổng serial của Arduino IDE  } |

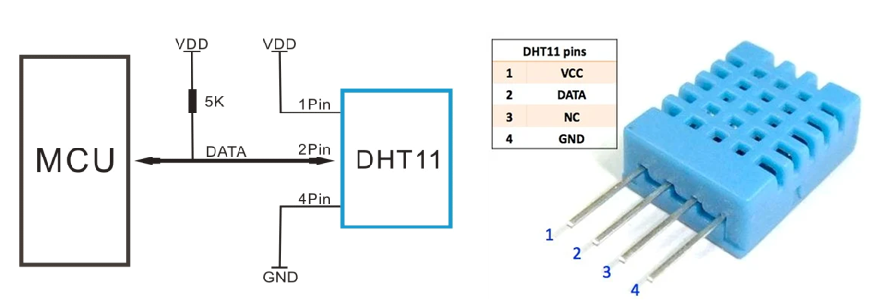
* Biên dịch, nạp chương trình và quan sát kết quả.
* Tần số nháy của đèn LED là bao nhiêu ?............................Giải thích tại sao ? ................................................................................................................................................

**2.2 Cấu tạo và nguyên tắc hoạt động của các cảm biến**

**a. Cảm biến nhiệt độ, độ ẩm DHT11**

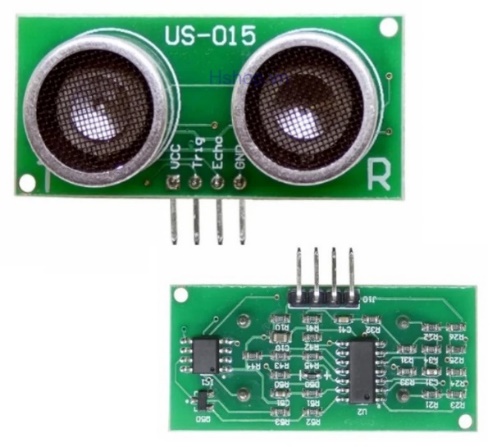
DHT-11 là module cảm biến nhiệt độ, độ ẩm có giao tiếp 1 dây (one wire). Cấu tạo cảm biến gồm 2 phần: một phần tử cảm biến độ ẩm bằng điện dung và một điện trở nhiệt để cảm nhận nhiệt độ. Tụ điện cảm biến độ ẩm có hai điện cực với chất nền giữ ẩm làm chất điện môi giữa chúng. Thay đổi giá trị điện dung xảy ra theo sự thay đổi của các mức độ ẩm. Ngoài ra bên trong module còn có các mạch chuyển đổi tương tự sang số. Cảm biến được tích hợp bộ tiền xử lý giúp cho dữ liệu nhận về được chính xác mà không cần phải qua bất kỳ khâu phân tích hay tính toán nào. Các thông số kỹ thuật của module cảm biến nhiệt độ, độ ẩm DHT-11 được cho như sau:

* Điện áp hoạt động 3,5÷5,5Vdc
* Dòng điện hoạt động cực đại là 2,5mA
* Dải độ ẩm hoạt động 20%÷95% RH, sai số ±5%RH (range humidity)
* Dải nhiệt độ hoạt động 0÷500C, sai số ±20C
* Nhiệt độ và độ ẩm đều có độ phân giải 16 bit.
* Tốc độ lấy mẫu không quá 1Hz (mỗi giây một lần).
* Tốc độ cảm nhận: trung bình 2s
* Khoảng cách truyền tối đa 20m.
* Chân DATA được nối với giắc cắm DATA trên KIT thí nghiệm IoT



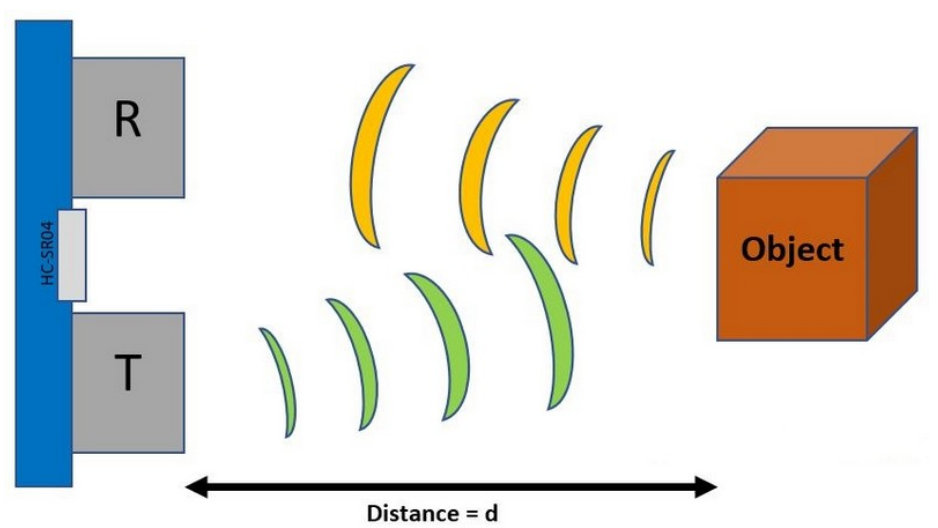
*Hình vẽ 1.10 . Cảm biến nhiệt độ, độ ẩm DHT11 và sơ đồ chân nối với vi điều khiển*

**b. Cảm biến siêu âm**



*Hình vẽ 1.11. Cảm biến siêu âm US-015*

Cảm biến siêu âm là một thiết bị điện tử sử dụng việc phát và thu sóng siêu âm (là sóng lan truyền trong môi trường không khí với vận tốc lớn hơn vận tốc âm thanh, cỡ 340m/s) để cảm nhận sự xuất hiện của vật thể hoặc đo khoảng cách giữa vật thể và cảm biến. Do sóng siêu âm lan truyền trong không khí, khi gặp vật cản sóng siêu âm bị phản xạ trở lại đầu thu sóng siêu âm của cảm biến. Dựa vào tín hiệu sóng siêu âm thu được này sẽ tính được khoảng cách tới vật cản dựa trên khoảng thời gian giữa thời điểm phát tín hiệu sóng siêu âm và thời điểm thu được sóng phản xạ



*Hình vẽ 1.12 Phương pháp đo khoảng cách bằng cảm biến siêu âm*

Cảm biến siêu âm bao gồm một bộ phát ra tín hiệu sóng siêu âm Tx và một đầu thu Rx. Đầu thu của cảm biến hoạt động như một microphone để nhận tín hiệu sóng siêu âm. Chúng được thiết kế với nhiều hình dạng như:

* Đầu dò thẳng (sóng dọc).
* Đầu dò ngang (sóng ngang).
* Đầu dò sóng bề mặt.
* Đầu dò kép (một đầu phát và một đầu nhận tín hiệu).

Mặc dù thiết kế khác nhau nhưng các thiết bị đầu dò cảm biến sóng siêu âm này đều có chung cấu tạo gồm:

* **Bộ phát Tx:** Là bộ phận được cấu tạo từ gốm, với đường kính rộng 15mm, hoạt động nhờ cơ chế chuyển động rung để tạo ra các sóng siêu âm truyền vào không khí.
* **Bộ thu Rx:** Có chức năng hình thành các rung động cơ học tỷ lệ với sóng siêu âm và chuyển đổi sóng siêu âm thành năng lượng điện ở đầu ra của bộ thu.
* **Bộ điều khiển:** Là bộ phận sử dụng mạch điện tích hợp để điều khiển sự hoạt động của bộ phát và bộ thu, đồng thời phát ra tín hiệu điện dạng tương tự hoặc số để có thể truyền thông tin tới các mạch điện ghép nối với cảm biến như vi điều khiển, vi xử lý.

**Nguyên lý hoạt động cơ bản của cảm biến siêu âm US-015**

Sau khi cấp nguồn cho US-015, cấp một xung mức cao vào chân Trigger của US-015, khi đó module sẽ phát ra sóng siêu âm, sóng này lan truyền trong không khí, gặp vật cản, sóng phản hồi lại và được thu bởi cảm biến trên US-015. Khi nhận được sóng siêu âm phản hồi, US-015 sẽ tạo một xung mức cao trên chân Echo, vi điều khiển được ghép nối với cảm biến sẽ đo thời gian tính từ lúc bắt đầu phát sóng siêu âm tới khi nhận được sóng phản hồi, dựa vào tốc độ âm thanh trong không khí để tính ra khoảng cách. Cảm biến US-015 giao tiếp truyền thông tin với vi điều khiển, vi xử lý bằng giao thức truyền tin I2C.

*Chú ý:*

* Cần chia đôi kết quả thu được vì tổng thời gian đo được là thời gian cả lượt đi và lượt về của sóng siêu âm.
* Do cảm biến hoạt động bằng cách thu lại sóng phản hồi nên đối với môi trường quá rộng, bề mặt cần đo hấp thụ hoặc tán xạ sóng siêu âm thì kết quả đo có thể không chính xác.
* Tốc độ sóng siêu âm trong không khí cỡ khoảng 340m/s
* Khi thực hiện phép đo cần tránh không để vật cản nằm trong vùng mù của cảm biến.
* Độ chính xác của phép đo tùy thuộc vào việc hiệu chỉnh vận tốc sóng siêu âm trong môi trường không khí.

**Các đặc trưng kỹ thuật cơ bản của cảm biến siêu âm US-015**

|  |  |
| --- | --- |
| Điện áp hoạt động | 5Vdc |
| Dòng hoạt động | 2,2mA |
| Nhiệt độ làm việc | 0÷700C |
| Các chân lối ra | TRIGGER và ECHO |
| Góc cảm nhận | Nhỏ hơn 150 |
| Khoảng cách phát hiện | Từ 2cm tới 2,5m |
| Độ chính xác phát hiện | 0,1cm ±1% |
| Độ phân giải nhỏ hơn 1mm | Lên tới 0,5mm |

**Bảng 1.3**. Đặc trưng kỹ thuật của cảm biến siêu âm US-015

1. **THÍ NGHIỆM**

**3.1 Viết chương trình đọc dữ liệu nhiệt độ, độ ẩm từ cảm biến DHT11 và hiển thị lên màn hình Serial Monoitor của máy tính**

**Bước 1**: Kết nối dây: Thực hiện việc kết nối dây giữa module cảm biến DHT11 và ESP32 trên KIT thí nghiệm IoT theo bảng sau:

|  |  |
| --- | --- |
| **Chân của module DHT-11** | **Chân của ESP32** |
| DATA | IO 04 |

**Bảng 1.1**. Kết nối dây giữa module DHT-11 và ESP32 trên KIT IoT

**Bước 2**: Viết chương trình cho ESP32

* Sử dụng Arduino IDE viết chương trình đọc và hiển thị dữ liệu nhiệt độ, độ ẩm tại cửa sổ Serial Port của Arduino IDE trên máy tính.
* Biên dịch và nạp mã chương trình cho KIT ESP32

*Chú ý: Thêm vào thư viện cảm biến DHT11 với tên DHT.h và Wire.h cho giao tiếp One-Wire trong Arduino IDE trước khi viết chương trình và biên dịch.*

**Bước 3:** Thay đổi nhiệt độ xung quanh cảm biến và nhận xét, ghi lại kết quả dữ liệu đo nhiệt độ, độ ẩm của phòng thí nghiệm vào dưới đây:

|  |  |  |  |
| --- | --- | --- | --- |
| **Lần đo** | **Nhiệt độ** | | **Độ ẩm**  %RH (range humidity) |
| Độ C (Celsius) | Độ F (Fahrenheit) |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

**Bảng 1.2**. Kết quả đo nhiệt độ, độ ẩm phòng thí nghiệm bằng cảm biến DHT11

**Bước** : Thay đổi chân kết nối giữa ESP32 và DHT11 cũng như thay đổi tốc độ cổng serial. Sửa lại chương trình và thực hiện đọc dữ liệu của DHT11. Ghi nhận xét vào dưới đây:

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Trình bày về nguyên lý cơ bản của giao thức truyền tin One-Wire ?

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**3.2 Viết chương trình đọc dữ liệu khoảng cách gửi lên từ cảm biến siêu âm US-015 và hiển thị kết quả lên màn hình Serial Monoitor của máy tính**

**Bước 1**: Kết nối dây: Thực hiện việc kết nối dây giữa module cảm biến US-015 và ESP32 trên Kit thí nghiệm IoT theo bảng sau:

|  |  |
| --- | --- |
| **Chân của module US-015** | **Chân của ESP32** |
| TRIGGER | IO 13 |
| ECHO | IO 04 |

**Bảng 1.4**. Sơ đồ nối dây giữa cảm biến siêu âm US-015 và ESP32

**Bước 2**: Viết chương trình cho ESP32

* Sử dụng Arduino IDE viết chương trình đọc và hiển thị dữ liệu khoảng cách tại cửa sổ Serial Port của Arduino IDE trên máy tính.
* Biên dịch và nạp mã chương trình cho KIT ESP32

**Bước 3**: Lấy một vật cản (ví dụ tờ giấy hay cuốn sách) che cảm biến siêu âm và thay đổi khoảng cách so với cảm biến. Nhận xét về kết quả thu được trên serial monitor ?

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Bước 4**: Giải thích mã nguồn của chương trình ? Giải thích công thức tính khoảng cách của chương trình ? Tại sao nói “độ chính xác của cảm biến tùy thuộc vào việc hiệu chỉnh tốc độ sóng siêu âm ” ?

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Bước 5**: Thực nghiệm để biết được vùng mù của cảm biến (vùng nằm ngoài khoảng cách nhỏ nhất có thể đo được và lớn nhất có thể đo được). Ghi kết quả vào bảng dưới đây:

|  |  |
| --- | --- |
| **Khoảng cách lớn nhỏ nhất đo được** | **Khoảng cách lớn nhất đo được** |
|  |  |
| **Vùng mù**: | |

**Bảng 1.5**. Bảng kết quả đo xác định vùng mù của cảm biến US-015

**Bước** 6: Thay đổi chân kết nối giữa ESP32 và US-015 cũng như thay đổi tốc độ cổng serial. Sửa lại chương trình và thực hiện đọc dữ liệu của US-015. Ghi nhận xét vào dưới đây:

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

- Trình bày về nguyên lý cơ bản của giao thức truyền tin I2C ?

------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------