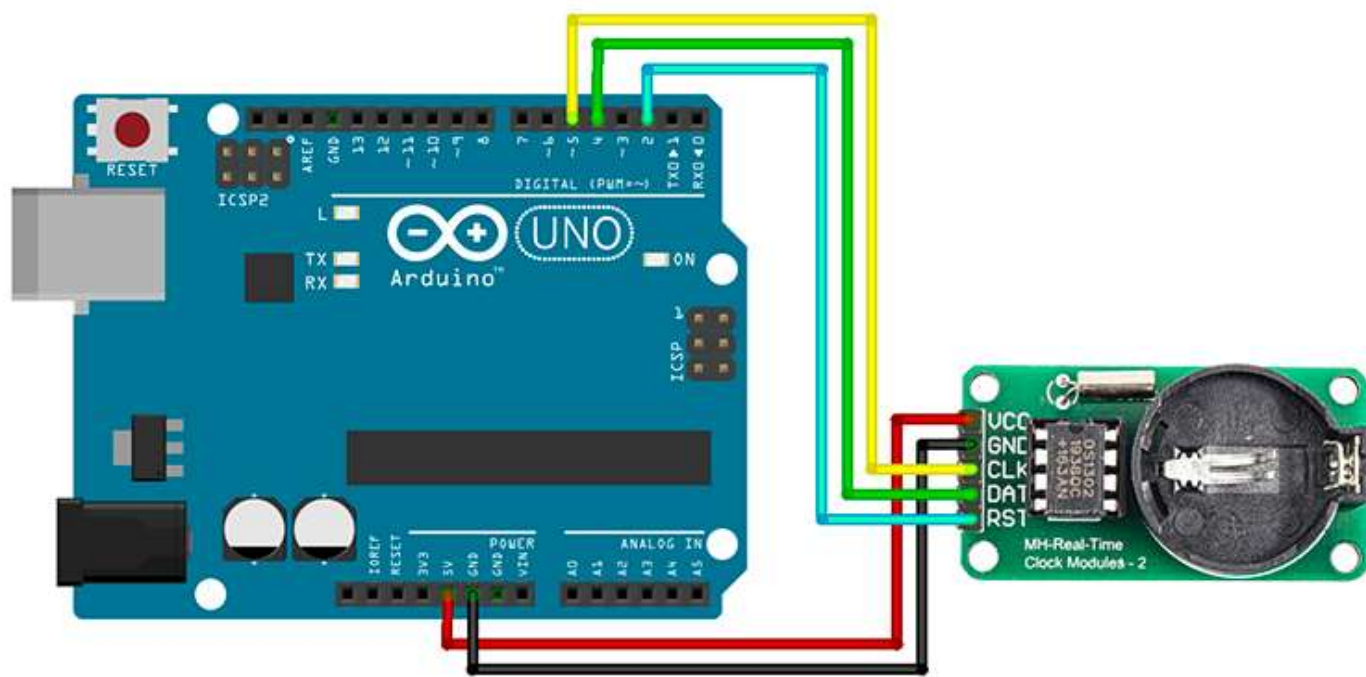


Cảm biến Arduino và module chức năng

Module thời gian thực DS1302

DS1302 là một loại mạch đồng hồ thời gian thời gian thực (Real-Time Clock - RTC). DS1302 có khả năng theo dõi thời gian thực, bao gồm năm, tháng, ngày, giờ, phút và giây. Nó sử dụng một pin ngoại vi để duy trì thông tin về thời gian, cho phép nó tiếp tục hoạt động ngay cả khi nguồn điện chính được tắt



Link thư viện: <https://github.com/Makuna/Rtc/tree/master>

Code gợi ý

```
1 #include <ThreeWire.h>
2 #include <RtcDS1302.h>
3
4 ThreeWire myWire(4, 5, 2);
5 RtcDS1302<ThreeWire> Rtc(myWire);
6
7 void setup ()
8 {
9   Serial.begin(9600);
10
11   Serial.print("compiled: ");
12   Serial.print(__DATE__);
13   Serial.println(__TIME__);
```

```
14
15  Rtc.Begin();
16
17  RtcDateTime compiled = RtcDateTime(__DATE__, __TIME__);
18  printDateTime(compiled);
19  Serial.println();
20
21  if (!Rtc.IsDateTimeValid())
22  {
23      Serial.println("RTC lost confidence in the DateTime!");
24      Rtc.SetDateTime(compiled);
25  }
26
27  if (Rtc.GetIsWriteProtected())
28  {
29      Serial.println("RTC was write protected, enabling writing now");
30      Rtc.SetIsWriteProtected(false);
31  }
32
33  if (!Rtc.GetIsRunning())
34  {
35      Serial.println("RTC was not actively running, starting now");
36      Rtc.SetIsRunning(true);
37  }
38
39  RtcDateTime now = Rtc.GetDateTime();
40  if (now < compiled)
41  {
42      Serial.println("RTC is older than compile time! (Updating DateTime)");
43      Rtc.SetDateTime(compiled);
44  }
45  else if (now > compiled)
46  {
47      Serial.println("RTC is newer than compile time. (this is expected)");
48  }
49  else if (now == compiled)
50  {
51      Serial.println("RTC is the same as compile time! (not expected but all is fine)");
52  }
53 }
54
55 void loop ()
56 {
57     RtcDateTime now = Rtc.GetDateTime();
58     printDateTime(now);
59     Serial.println();
60     if (!now.IsValid())
61     {
62         Serial.println("RTC lost confidence in the DateTime!");
63     }
64
```

```

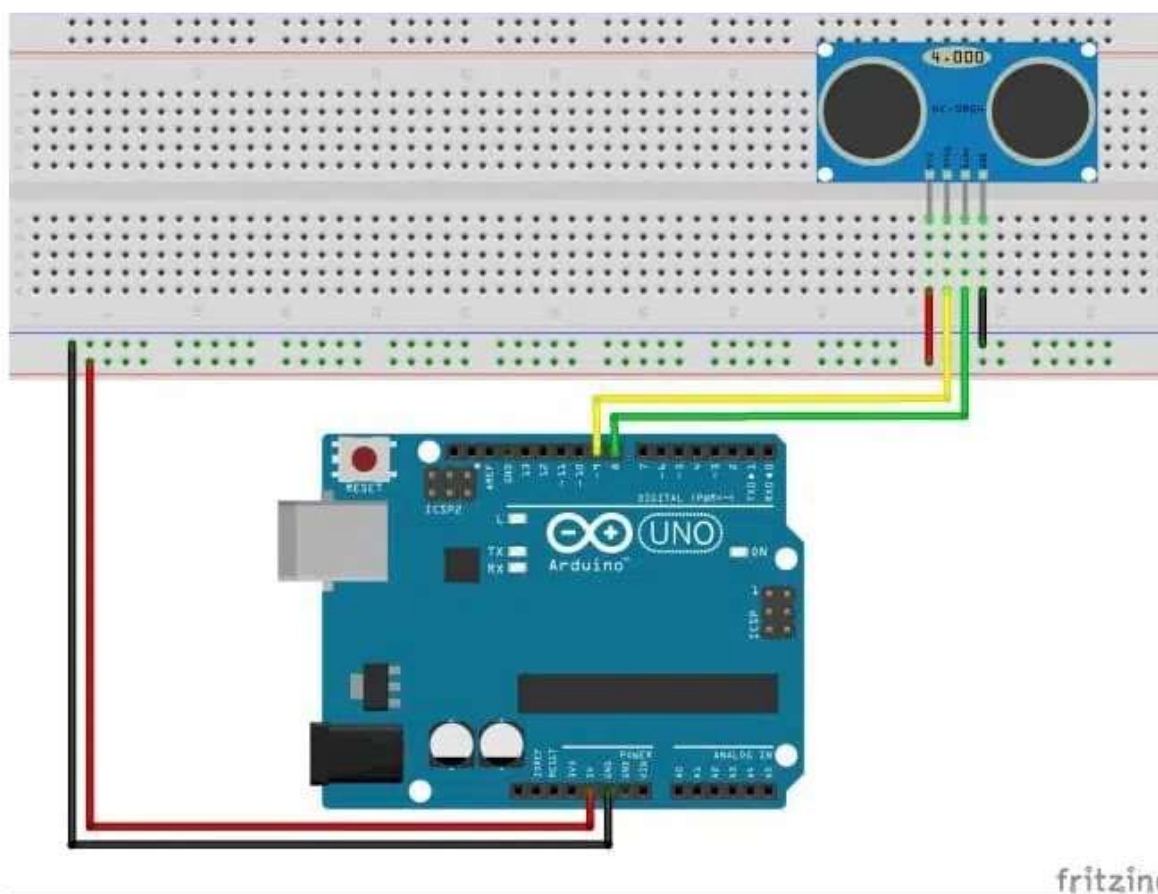
65   delay(5000);
66 }
67
68 #define countof(a) (sizeof(a) / sizeof(a[0]))
69 void printDateTime(const RtcDateTime& dt)
70 {
71   char datestring[20];
72   snprintf_P(datestring,
73             countof(datestring),
74             PSTR("%02u/%02u/%04u %02u:%02u:%02u"),
75             dt.Month(),
76             dt.Day(),
77             dt.Year(),
78             dt.Hour(),
79             dt.Minute(),
80             dt.Second() );
81   Serial.print(datestring);
82 }

```

Bài 1: Sử dụng module **"Real-Time clock DS1307"** để xây dựng đồng hồ thời gian thực và hiển thị lên màn hình LCD? Thực hiện tính năng hẹn giờ qua việc bật 1 đèn LED khi phút ở giá trị 15 và tắt khi giá trị phút lớn hơn 20?

Bài 2: Sử dụng module **"Sensor HC-SR04"** thực hiện tính năng đo khoảng cách đến vật thể? Kết hợp với module L298 Motor Driver để điều khiển động cơ thì có vật cảm?

Link thư viện: <https://github.com/JRodrigoTech/Ultrasonic-HC-SR04>



```
1 byte trig = 12;
2 byte echo = 11;
3 byte buzzer = A5;
4 #include <LiquidCrystal.h>
5 LiquidCrystal lcd(8,7,6,5,4,3);
6 void setup() {
7   Serial.begin(9600);
8   pinMode(trig, OUTPUT);
9   pinMode(echo, INPUT);
10  pinMode(buzzer, OUTPUT);
11  lcd.begin(16,2);
12 }
13
14 void loop() {
15   long thgian;
16   int kcach;
17
18   //phát xung đi
19   digitalWrite(trig, LOW);
20   delayMicroseconds(2);
21   digitalWrite(trig, HIGH);
22   delayMicroseconds(10);
23   digitalWrite(trig, LOW);
24
25   //Tính thời gian
26   thgian = pulseIn(echo, HIGH);
27
28   //tính khoảng cách đơn vị cm
29   kcach = int(thgian/2/29.412);
30
31   Serial.println("Khoang cach: "+ String(kcach) +" cm");
32   Serial.println();
33
34   lcd.setCursor(0, 0);
35   lcd.print("Khoang cach: "+ String(kcach) +" cm");
36   lcd.println();
37
38   // nếu khoảng cách nhỏ hơn 100 thì buzzer sẽ bật.
39   if(kcach < 100){
40     digitalWrite(buzzer,HIGH);
41   }
42   else
43     digitalWrite(buzzer,LOW);
44
45   delay(1000);
46 }
```

Bài 3: Sử dụng cảm biến “**RAIN SENSOR**” kết hợp với bo mạch Arduino để xây dựng hệ thống báo mưa? Kết hợp với “**L298 Motor Driver**” và “**Cảm biến ánh sáng**” để mô phỏng hoạt động của hệ thống giàn phơi thông minh?

+ Nếu giá trị cảm biến mưa =1 và giá trị cảm biến ánh sáng > 35% thì động cơ sẽ kéo giàn phơi ra và hiển thị số liệu, trạng thái lên màn hình LCD 16x2.

+ Nếu giá trị cảm biến mưa =0 hoặc giá trị cảm biến ánh sáng < 35% thì động cơ sẽ thu giàn phơi vào và hiển thị số liệu, trạng thái lên màn hình LCD 16x2.

Bài 4: Hãy sử dụng “**Cảm biến độ ẩm đất**” kết hợp với bo mạch Arduino để xây dựng hệ thống vườn tự động. Kết hợp với “**L298 Motor Driver**” và màn hình LCD để mô phỏng hoạt động của vườn tự động.

+ Nếu độ ẩm nhỏ hơn 70% thì sẽ kích cho motor bơm chạy, đồng thời hiển thị thông báo lên màn hình LCD 16x2.

+ Khi độ ẩm đất từ 70% trở lên, motor bơm sẽ dừng và đèn sẽ tắt.

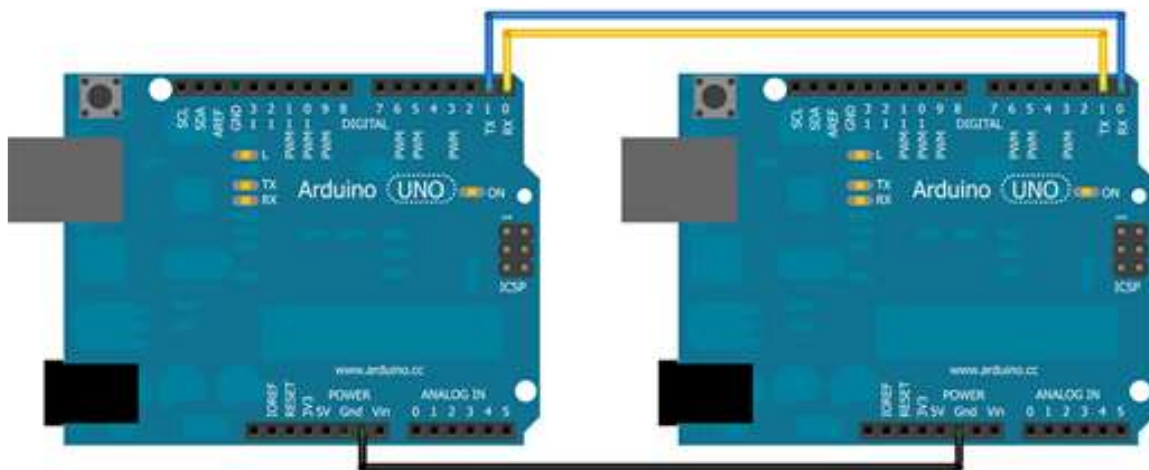
Bài 5: Thực hiện đấu nối và lập trình “**Cảm biến mưa**” kết hợp với “**L298 Motor Driver**” để mô phỏng hệ thống gạt nước mưa trên Ôtô với các kịch bản sau:

- Khi cảm biến mưa nhận biết mưa thì sẽ điều khiển Motor với tốc độ nhanh chậm phụ thuộc vào trời mưa lớn hay mưa nhỏ.
- Khi cảm biến mưa nhận biết mưa thì sẽ điều khiển động cơ sẽ quay với 1 vận tốc cố định.

Bài 6: Thiết kế mạch gồm 1 “**Cảm biến LM35**” kết hợp với 1 đèn **LED** để thực hiện mở đèn khi nhiệt độ môi trường < 30 °C ngược lại thì tắt đèn.

Bài 7: Sử dụng 2 bo Arduino và giao thức UART để truyền và nhận dữ liệu qua lại giữa 2 bo Arduino theo yêu cầu bên dưới?

- Kết nối bo Arduino A với màn hình LCD (I²C), bo Arduino B với màn hình LCD (SPI)
- Kết nối UART giữa 2 bo mạch này, trong đó dữ liệu khi nhận qua lại giữa hai bo Arduino sẽ hiển thị lên màn hình LCD.



Bài 8: Sử dụng 2 bo Arduino và giao thức UART để truyền và nhận dữ liệu qua lại giữa 2 bo Arduino theo yêu cầu bên dưới?

- Kết nối bo Arduino A với cảm biến DHT11 và gửi dữ liệu sang bo Arduino B
- Bo Arduino B trả phản hồi về bo Arduino khi đã nhận được dữ liệu
- Bo Arduino B kết nối với màn hình LCD 16x6 với kết nối I²C để hiển thị lên màn hình LCD.

Bài 9: Kết nối 4 bo mạch Arduino thông qua giao thức I²C để nhận và gửi dữ liệu giữa chúng theo kịch bản bên dưới?

- Trên bo Arduino A, B, C kết nối với 1 LED bất kỳ (A, B, C đóng vai trong Slave)
- Trên bo Arduino D kết nối I²C với 3 bo Arduino còn lại và kết nối 3 nút nhấn
- Trên bo Arduino D khi nhấn 1 nút nhấn sẽ gửi tín hiệu I²C sang 1 bo tương ứng để bật một LED

