

ỨNG DỤNG VI ĐIỀU KHIỂN THIẾT KẾ MODULE TỰ ĐỘNG TƯỚI CÂY

Nguyễn Thanh Tùng⁽¹⁾

⁽¹⁾Trường Đại học Thủ Dầu Một

Ngày nhận bài 3/3/2017; Ngày gửi phản biện 20/3/2017; Chấp nhận đăng 30/6/2017

Email: nttung@tdmu.edu.vn

Tóm tắt

Ứng dụng thiết bị phần cứng và phần mềm Arduino là lựa chọn hàng đầu trong lĩnh vực thiết kế các thiết bị tự động hóa với xu hướng gọn nhẹ, kinh phí đầu tư thấp mang lại hiệu quả cao. Trong giới hạn bài viết này, chúng tôi trình bày kết quả nghiên cứu thiết kế một module vi xử lý, tự động điều khiển máy bơm nước tưới cây kiểm thông qua thiết bị Arduino Nano và cảm biến độ ẩm đất. Khi môi trường bên trong vùng đất đang trồng cây xanh có độ ẩm đất thấp, nghĩa là cây xanh bị thiếu nước, độ ẩm giảm vượt ngưỡng đã được ấn định, cảm biến độ ẩm đất cho điện thế đầu ra ở mức cao, hiển thị trên LCD, mạch vi xử lý Arduino kích hoạt role ở trạng thái đóng (mức 1), máy bơm nước được cấp điện nguồn và bắt đầu hoạt động. Khi cây được tưới đủ nước, độ ẩm tăng cao, cảm biến cho ra điện thế thấp, tín hiệu đầu ra từ Arduino ở trạng thái thấp (mức 0) role chuyển sang trạng thái ngắt mạch điện, máy bơm ngừng hoạt động.

Từ khóa: Tưới cây tự động, Cảm biến độ ẩm đất

Abstract

DESIGN MODULE AUTOMOTIVE PUMP WATER FOR PLANTS FROM APPLY ARDUINO EQUIPMENT

Apply hardware and software Arduino Nano is the best choice in design of automotive equipment with a tendency to compact, low investment cost but that brings high effect. Within this article, the author presents the results of the design of a microprocessor module, which automotive controls water pumps for plants through Arduino Nano equipment and land humidity sensors. When the environment inside the soil is planted with low soil moisture, the trees are dehydrated, the humidity decreases beyond the set threshold, the humidity sensor for the high output voltage, the Arduino Nano microprocessor activates the relay in the closed state (level 1), the water pump is powered and starts operating. When the plants are watered enough, the humidity rises, the sensor outputs a low voltage, the output signal from the Arduino Nano microprocessor is in low state (level 0) the relay switches to the cut-off state, the pump stops working.

1. Tổng quan

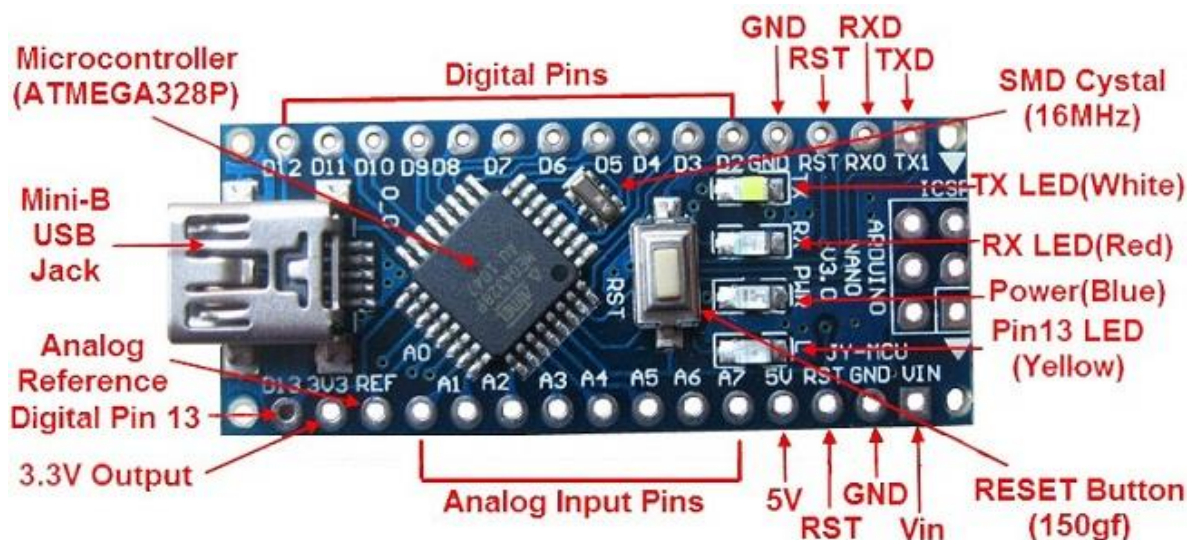
1.1 Giới thiệu

Cuộc cách mạng công nghiệp 4.0 tác động mạnh mẽ đến việc hình thành các dự án nghiên cứu, đầu tư và xây dựng nhằm phát triển nhiều lĩnh vực như: kỹ thuật, công nghiệp, nông nghiệp, xử lý tác động môi trường, chống biến đổi khí hậu, phát triển năng lượng xanh ...

Một trong những thành tố đó là việc phát triển công nghệ vi xử lý, tự động hóa. Vi xử lý là một khái niệm kỹ thuật không thể thiếu cho sự phát triển công nghiệp mang tính hiện đại, với những cái tên “điện thoại thông minh”, “ngôi nhà thông minh”, “thành phố thông minh”... Trong bài viết này, chúng tôi giới thiệu một thiết bị cơ bản và thông dụng với giá thành rẻ, phần mềm miễn phí đã và đang được các nhà nghiên cứu, giảng viên, sinh viên và học sinh trên thế giới sử dụng để chế tạo ra rất nhiều các ứng dụng vi xử lý có hiệu quả cao.

1.2 Thiết bị Arduino Nano

Thiết bị Arduino là phần cứng vi xử lý có khả năng chạy độc lập khi được nạp code. Arduino board có rất nhiều phiên bản với hiệu năng và mục đích sử dụng khác nhau, trong đề tài này tác giả sử dụng Arduino Nano cho thiết kế Module điều khiển tưới cây. Do Arduino Nano có kích thước nhỏ gọn, cách sử dụng và sơ đồ nối chân cũng tương tự như các phiên bản khác của Arduino.



Hình 1. Hình ảnh mạch Arduino Nano

Arduino Nano ra đời nhằm đáp ứng nhu cầu thu gọn kích thước (Hình 1) nhưng vẫn giữ nguyên sức mạnh của Arduino với vi điều khiển ATmega328P – SMD, toàn bộ board mạch có khả năng cắm trực tiếp vào breadboard. Thông số kỹ thuật của Arduino Nano được tóm tắt trong bảng 1.

Bảng 1. Thông số của thiết bị Arduino Nano

Vi điều khiển	ATmega328P
Điện áp hoạt động	5V
Điện áp vào khuyến dùng	7-12V
Điện áp vào giới hạn	6-20V
Digital I/O pin	14 (trong đó 6 pin có khả năng bấm xung)
PWM Digital I/O Pins	6
Analog Input Pins	6
Cường độ dòng điện trên mỗi I/O pin	20 mA
Cường độ dòng điện trên mỗi 3.3V pin	50 mA
Flash Memory	32 KB (ATmega328P); 0.5 KB được sử dụng bởi bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Tốc độ	16 MHz

Chiều dài	68,6 mm
Chiều rộng	53,4 mm
Trọng lượng	25 g

Khác với Arduino Uno sử dụng cổng USB type B, Arduino Nano lại sử dụng một cổng nhỏ hơn có tên là Mini USB. Vì sử dụng cổng này nên kích thước board (về chiều cao) cũng giảm đi khá nhiều.

1.3 Phần mềm Arduino

Môi trường phát triển tích hợp Arduino là một ứng dụng đa nền tảng được viết bằng Java, được dẫn xuất cho ngôn ngữ lập trình xử lý và các dự án lắp ráp. Nó được thiết kế để làm nhập môn lập trình cho các nhà lập trình và những người mới sử dụng khác không quen thuộc với phát triển phần mềm, bao gồm một trình soạn thảo mã với các tính năng như làm nổi bật cú pháp, khớp dấu ngắt khối chương trình, và thực thi dòng tự động, và cũng có khả năng biên dịch và tải lên các chương trình vào bo mạch với một nhấp chuột duy nhất. Một chương trình hoặc mã viết cho Arduino được gọi là "sketch". Chương trình Arduino được viết bằng C hoặc C⁺⁺. Arduino đi kèm với một thư viện phần mềm được gọi là "Wiring" từ dự án lắp ráp ban đầu, cho hoạt động đầu vào/đầu ra phổ biến trở nên dễ dàng hơn nhiều. Người sử dụng chỉ cần định nghĩa hai hàm để thực hiện một chương trình điều hành theo chu kỳ:

Setup(): hàm chạy một lần duy nhất vào lúc bắt đầu của một chương trình dùng để khởi tạo các thiết lập.

Loop(): hàm được gọi lặp lại liên tục cho đến khi bo mạch được tắt đi.

Khi bật điện bảng mạch Arduino, reset hay nạp chương trình mới, hàm setup() sẽ được gọi đến đầu tiên. Sau khi xử lý xong hàm setup(), Arduino sẽ nhảy đến hàm loop() và lặp vô hạn hàm này cho đến khi bạn tắt điện bo mạch Arduino. Arduino IDE sử dụng GNU toolchain và AVR libc để biên dịch chương trình, và sử dụng avrdude để tải lên các chương trình vào bo mạch chủ. Do nền tảng Arduino sử dụng vi điều khiển Atmel, môi trường phát triển của Atmel, AVR Studio hoặc Atmel Studio mới hơn, cũng có thể được sử dụng để phát triển phần mềm cho các Arduino.



Hình 2. Giao diện của Arduino lúc khởi động

```

1 #include <LiquidCrystal.h>
2
3 #define trigger 13
4 #define echo 12
5
6 LiquidCrystal lcd(9, 10, 5, 4, 3, 2);
7
8 float time=0,distanceCm=0, distanceInch=0;
9
10 void setup()
11 {
12   lcd.begin(16,2);
13   pinMode(trigger,OUTPUT);
14   pinMode(echo,INPUT);
15   lcd.print("Ultra sonic");
16   lcd.setCursor(0,1);
17   lcd.print("Distance Meter");
18   delay(2000);
19   lcd.clear();
20   lcd.print(" Circuit Digest");
  
```

Hình 3. Hình ảnh giao diện Arduino khi viết code

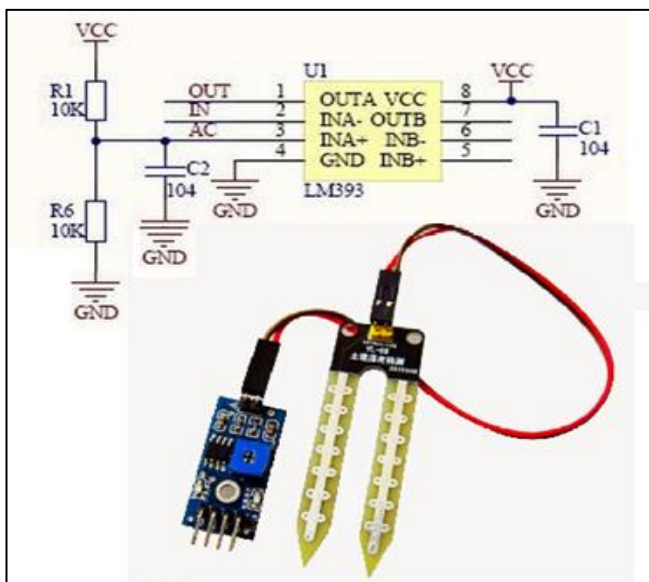
1.4 Cảm biến đo độ ẩm đất

Cảm biến độ ẩm đất, trạng thái đầu ra mức thấp (0V), khi đất thiếu nước đầu ra sẽ là mức cao (5V), độ nhạy cao chúng ta có thể điều chỉnh được bằng biến trở. Phần đầu đo được cắm vào đất để phát hiện độ ẩm của đất, khi độ ẩm của đất đạt ngưỡng thiết lập, đầu ra DO sẽ chuyển trạng thái từ mức thấp lên mức cao. Nhờ thế, chúng ta có thể sử dụng Analog hoặc Digital của Arduino để đọc giá trị từ cảm biến. Khi module cảm biến độ ẩm phát hiện, khi đó sẽ có sự thay đổi điện áp ngay tại đầu vào của IC so sánh LM393. IC này nhận biết có sự thay đổi nó sẽ đưa ra một tín hiệu 0V để báo hiệu và thay đổi như thế nào sẽ được tính toán để đọc độ ẩm đất. Cảm biến độ ẩm đất có 4 chân: Vcc, GND, 2 ngõ ra là D0 (cho giá trị trả về mức logic 0 hoặc 1) và A0, chúng ta có thể dùng 1 trong 2 chân này. Sơ đồ nối chân của Cảm biến đo độ ẩm đất với Arduino Nano như bảng 2.

Bảng 2. Sơ đồ nối chân của Arduino Nano với Cảm biến độ ẩm đất

Cảm biến độ ẩm đất	Arduino Nano
Vcc	3.3V hay 5V
GND	GND
D0	2
A0	A0

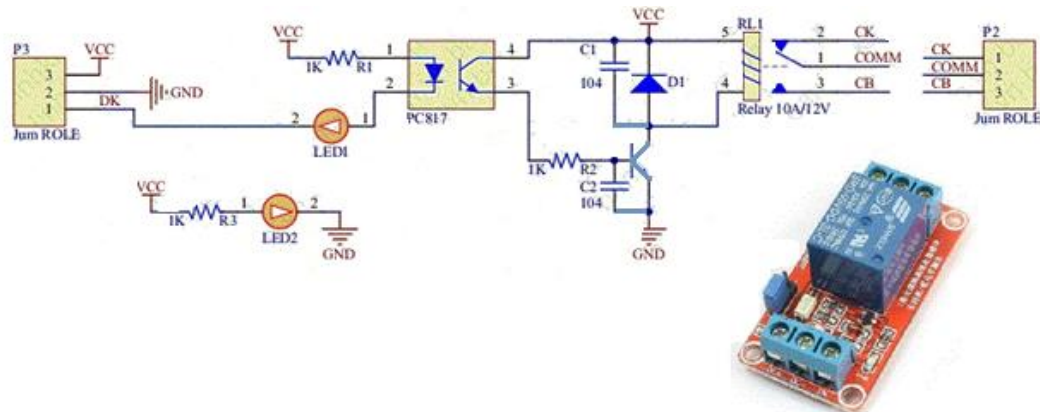
Hình 4. Sơ đồ mạch IC LM393 và cảm biến độ ẩm đất



1.5. Rơ-le (Relay) đóng ngắt mạch 1 kênh 5V

Rơ-le là thiết bị đóng ngắt cơ bản, nó được sử dụng rất nhiều trong các thiết bị điện, điện tử phục vụ cuộc sống hàng ngày. Rơ-le bình thường gồm có 6 chân. Trong đó có 3 chân để kích, 3 chân còn lại nối với đồ dùng điện công suất cao. Ba chân dùng để kích gồm: chân cấp hiệu điện thế dương Vcc (5V), chân cấp điện thế âm GND, chân tín hiệu S, nhận tín hiệu đầu ra từ thiết bị Arduino Nano. Khi chân này có điện thế cao thì rơ-le được kích, ngược lại thì không. Ba chân còn lại nối với thiết bị điện công suất cao (máy bơm nước): COM - chân nối với 1 chân bất kỳ của thiết bị điện, nhưng tốt nhất nên mắc vào dây chân lửa (nóng) nếu dùng hiệu điện thế xoay chiều và cực dương nếu là hiệu điện một chiều; ON hoặc NO (thường mở) - chân này sẽ được nối với chân lửa (nóng) nếu dùng điện xoay chiều và cực dương của nguồn nếu dòng điện một chiều; OFF hoặc NC (thường đóng) - chân này bạn sẽ nối chân lạnh (trung hòa) nếu dùng điện xoay chiều và cực âm của nguồn nếu dùng điện một chiều.

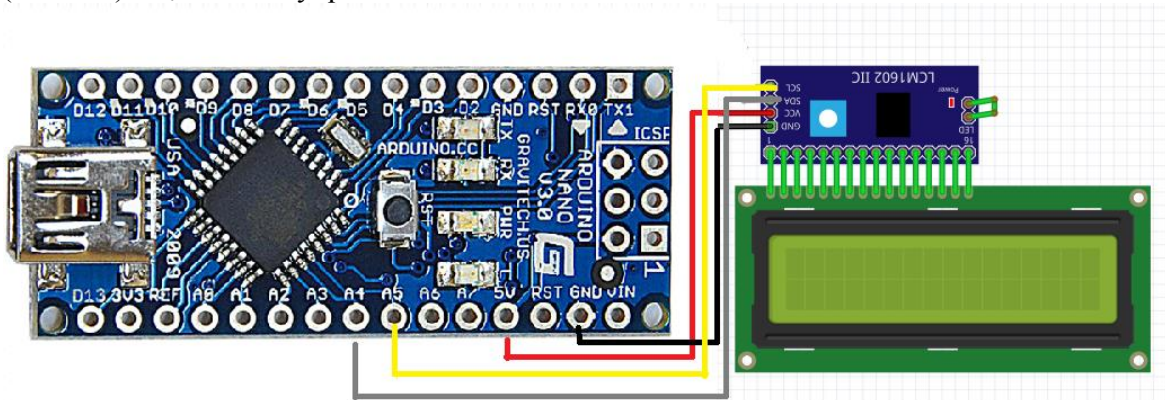
Thiết bị rơ-le trong đề tài này sử dụng có các thông số: điện áp nuôi DC 5V, tiêu thụ dòng khoảng 80mA, điện thế đóng ngắt tối đa: AC 250V-10A hoặc DC 30V-10A, kích thước 53mm (chiều dài), 28,3mm (chiều rộng); 19,3mm (cao).



Hình 5. Sơ đồ mạch Rơ-le 1 kênh 5V

1.6. Màn hình hiển thị LCD

Màn hình tinh thể lỏng hay LCD (Liquid crystal display) là loại thiết bị hiển thị cấu tạo bởi các tế bào (các điểm ảnh) chứa tinh thể lỏng có khả năng thay đổi tính phân cực của ánh sáng và do đó thay đổi cường độ ánh sáng truyền qua khi kết hợp với các kính lọc phân cực. Chúng có ưu điểm là phẳng, cho hình ảnh sáng, chân thật và tiết kiệm năng lượng. Trong đề tài này, tác giả chọn thiết bị LCD cấu hình 16x2 (chiều ngang 16 ký tự, chiều dọc gồm 2 hàng), cấp điện 5VDC tích hợp với mạch I2C còn lại 4 chân ra gồm: Vcc, GND, SDA (chân A4), SCL (chân A5) được trình bày qua sơ đồ Hình 6.



Hình 6. Sơ đồ nối chân giữa LCD-I2C và Arduino Nano

2. Thực hành

2.1. Nguyên lý hoạt động của Module

Khi Cảm biết độ ẩm đất có độ ẩm thấp hơn ngưỡng định sẵn, chỉ số điện thế đầu ra của nó ở mức cao, đèn LED báo hiệu màu xanh sáng và kèm theo dòng tin “EMPTY WATER”, nghĩa là cây trồng đang thiếu nước, cần tưới cây. Khi đó mạch Arduino Nano nhận tín hiệu và cho kết quả sau khi xử lý là đầu ra kích hoạt Rờ le ở mức cao (mức 1), máy bơm nước hoạt động. Sau thời gian bơm tưới, cây nhận lượng nước tăng lên, chỉ số điện thế đầu ra từ cảm biến độ ẩm đất giảm, ngược lại độ ẩm tăng lên vượt ngưỡng mức cao, đèn LED màu đỏ sáng và kèm dòng tin “ENOUGH WATER”, nghĩa là cây đã đủ nước. Khi đó mạch Arduino cho kết quả mức thấp (mức 0) Rờ le ngắt mạch và máy bơm thôi hoạt động. Quy trình bơm tưới của Module sẽ lặp lại nếu như độ ẩm đất lại tiếp tục chuyển sang trạng thái vượt mức ngưỡng thấp.

2.2. Viết code điều khiển Module

Để tiến hành viết code cho Arduino nói chung, chúng ta cần phải khai báo các yếu tố như: Cách nối chân nhập và xuất tín hiệu, các chân cấp điện, thư viện phù hợp cho cảm biến, biến analog hay digital, void setup, vòng lặp void loop, tốc độ truy xuất dữ liệu, cách hiển thị màn hình laptop hay LCD, tần suất lấy đọc dữ liệu, độ trễ (delay)...Dưới đây là nội dung của code được viết để nạp cho Arduino Nano điều khiển các thiết bị:

// Cách nối chân:

// Cảm biến - Arduino nano, AO-A1, DO-D4, GND-âm, V_{CC}-5V.

// LCD-I2C – Arduino nano, SDL-A5, SDA-A4

// Khai báo các biến

#include <LiquidCrystal_I2C.h>

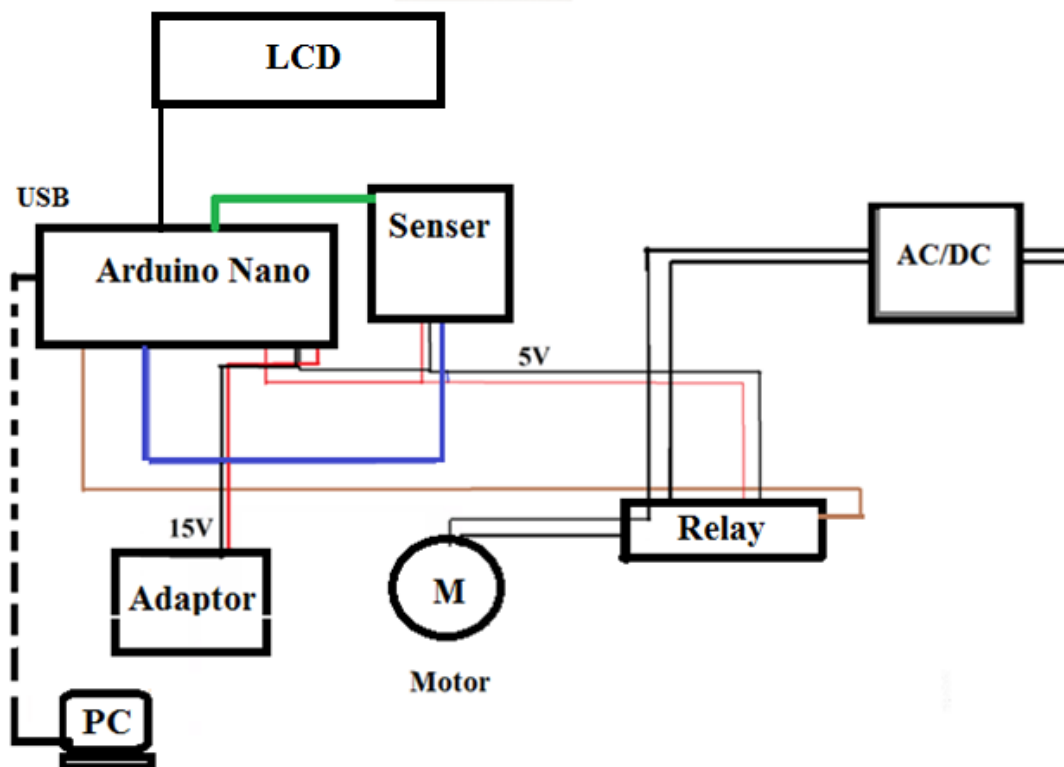
LiquidCrystal_I2C lcd (0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);

```
#define den 11
// Chân đèn LED
int CBDoAm=A0;
//Nối chân ra của cảm biến đo độ ẩm với chân A1
const int D0 = 2;
float Doamdat;
//Biến độ ẩm đất
void setup()
{
    Serial.begin(9600);
//Mở cổng Serial ở mức 9600
    pinMode (D0, INPUT);
    pinMode (A0, INPUT);
    pinMode (12, OUTPUT);
    pinMode(den,OUTPUT);
}
void loop()
{
    Doamdat=analogRead(CBDoAm);
    int value = analogRead(A0);
// Ta sẽ đọc giá trị hiệu điện thế của cảm biến
// Giá trị được số hóa thành 1 số nguyên có giá trị trong khoảng từ 0 đến 1023
    delay(5000);
    lcd.begin(16,2); // bat buoc co
    lcd.print("VOLT INDEX:");
    lcd.setCursor(13,0);
    lcd.print(value);
    if(Doamdat <350)
    {
        digitalWrite(den, HIGH);
        lcd.backlight();
        lcd.setCursor(0,1);
        lcd.print("ENOUGH WATER");
        digitalWrite (12, LOW);
    }
    else
    {
        digitalWrite (12, HIGH);
        digitalWrite (den, LOW);
        lcd.backlight();
        lcd.setCursor(1,1);
        lcd.print ("EMPTY WATER");
    }
}
```

2.3. Sơ đồ khối liên kết các khối trong Module

Theo hình dưới đây chúng ta có thể quan sát sơ đồ khối kết nối chân của thiết bị phần cứng (Arduino Nano) và các thiết bị khác trong Module. Trong khi đó mạch Arduino Nano

được cấp điện ngoài bằng một bộ nguồn chuyển đổi điện từ AC sang DC (Adaptor) có giá trị đầu ra 15VDC được cấp vào chân VIN. Các mạch khác như Cảm biến độ ẩm đất (Sensor), màn hình tinh thể lỏng (LCD) và Rơ le (Relay) đều dùng điện 5V từ mạch này.



Hình 7. Sơ đồ khối nối giữa các thiết bị trong Module

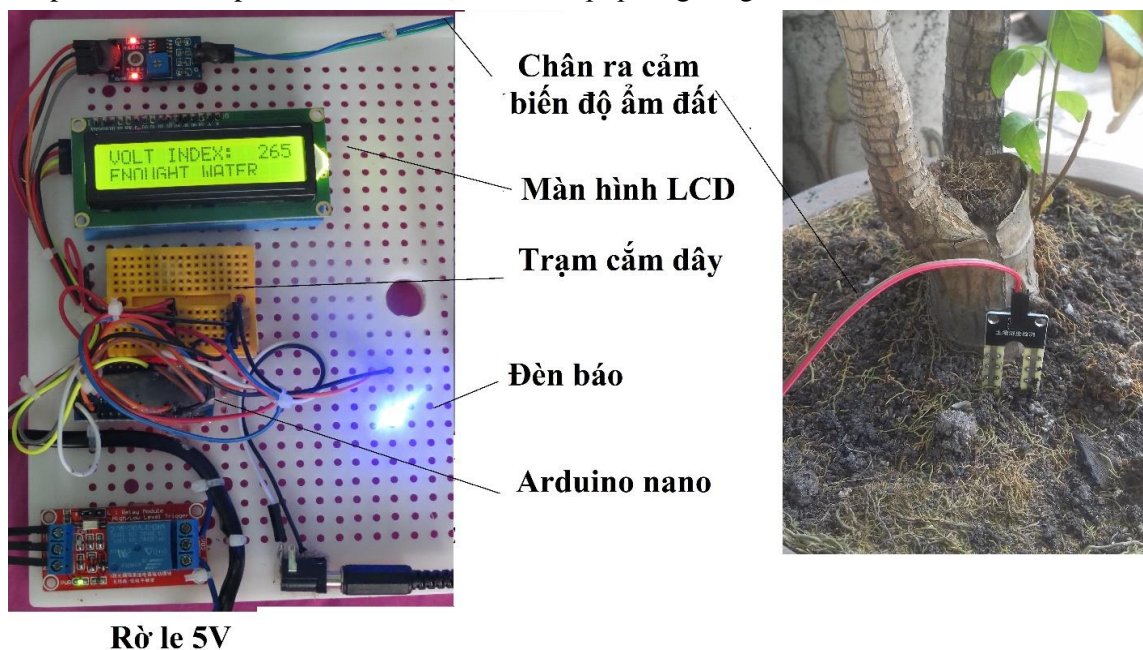
Mô tơ (Motor) bơm nước có thể dùng loại sử dụng dòng điện xoay chiều (AC) công suất lớn hay dòng điện một chiều (DC) công suất nhỏ tùy theo yêu cầu bơm tưới của khu vườn. Code được viết và nạp vào mạch vi xử lý Arduino Nano dựa vào hệ thống dây cáp qua ngõ (USB) nối với máy vi tính (PC).

2.4. Quy trình hoạt động của Module tự động tưới cây

Sau khi áp dụng các kỹ thuật lập trình và công đoạn viết code kết thúc, chúng ta cắm Board mạch Arduino Nano vào máy tính bằng cổng USB Mini. Mở **Menu File**, xem xét các chế độ tùy chọn kiểu board mạch **Tools/Board/Arduino Nano** và **Processor: "ATmega 328"**; chân **Port (COM6)** và upload, biên dịch phần code vừa viết. Nếu không có lỗi về code phần mềm sẽ báo (Done Compiling), nạp code vào mạch bằng lệnh nút **Verify** bên trái màn hình. Sau khi nạp code nếu quy trình được chấp nhận, các đèn LED trên mạch nhấp nháy trong 2 giây, mạch bắt đầu hoạt động độc lập, lúc này chúng ta có thể ngắt cổng USB Mini ra khỏi máy tính. Tuy nhiên, cần nhớ cấp nguồn thường xuyên và phù hợp điện thế cho các mạch Arduino Nano, mạch cảm biến thông qua ngõ V-IN, mạch rơ-le và máy bơm nước được cấp điện riêng. Trong đề tài này chúng tôi sử dụng bơm phun sương công suất 30W, điện thế cung cấp là 24VDC, nguồn cung cấp là Adaptor AC/DC. Hai đầu của cảm biến độ ẩm đất (Hình 8) được cắm sâu trong đất, bên cạnh gốc cây đặt trong vùng trung tâm của khu vườn thực nghiệm. Khi trời nắng gắt, nước bốc hơi nhiều, cây khô nước dần và độ ẩm của đất giảm nhanh xung quanh vị trí đặt cảm biến đến khi hạ thấp hơn ngưỡng đã ấn định (350), màn hình LCD báo dòng

“EMPTY WATER”, đèn LED màu xanh sáng lên, mạch Arduino xử lý tín hiệu và nó phát xung ra cao (mức 1) ở chân 13. Tín hiệu xung kích này được cấp cho chân IN của Rơ-le 5V làm cho nó chuyển sang trạng thái đóng, mạch nguồn máy bơm 24VDC được đóng lại, máy bơm được cấp điện và bắt đầu bơm nước tưới cho hệ thống cây xanh thông qua các béc phun sương trong khu thực nghiệm.

Sau thời gian cây xanh được tưới nước, đất ướt dần, độ ẩm tăng dần xung quanh nơi đặt cảm biến và điện thế đầu ra ở chân D0 của cảm biến cũng dần giảm về mức thấp (mức 0) khi độ ẩm cao vượt ngưỡng cao đã ấn định, đèn LED tắt và màn hình LCD báo tin “ENOUGH WATER”, lúc này Rơ-le mất tín hiệu kích từ chân IN và nó chuyển sang trạng thái ngắt điện, máy bơm mất điện nguồn ngưng hoạt động, công việc tưới cây kết thúc. Cứ như thế quy trình sẽ lặp lại khi đất tiếp tục bị khô nước, độ ẩm thấp quá ngưỡng ấn định.



Hình 8. Hình ảnh các thành phần của Module điều khiển

2.5. Kết quả thực nghiệm

Qua quá trình viết code, lắp ráp phần cứng, nạp code và kiểm tra chạy thử nghiệm thực tế cho thấy thiết bị đã hoạt động độc lập đúng chức năng theo thiết kế (theo 2.1 và 2.4). Thiết bị có độ chính xác, độ nhạy cao khi thay đổi độ ẩm đất, độ ẩm thấp hơn 30% mạch tự động bơm nước, phun sương vào vùng thực nghiệm có chậu kiểng đang được cảm cảm biến. Sau khi bơm nước độ ẩm tăng lên vượt quá 70% máy bơm lại ngừng hoạt động.

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Đức Quý, Nguyễn Văn Dung (2006), *Độ ẩm đất và tưới nước hợp lý cho cây trồng*, NXB Lao động Xã hội.
- [2] Lê Phú Hiếu (2014), *Giáo trình lập trình C++*, Trường Đại học Mở Thành phố Hồ Chí Minh.
- [3] Phạm Văn Ất (2012), *C++ và lập trình hướng đối tượng*, NXB Khoa học và Kỹ thuật.
- [4] Maik Schmidt (2011), *Arduino-A quick Star Guide*, The Pragmatic Bookshelf, USA.
- [5] <http://www.ktphuhung.com/index.php?key=chitietsp&idsp=135>
- [6] <http://www.xahoihoctap.net/magazine/94-arduino/268-gioi-thieu-ve-arduino.html>