

**HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY AND EDUCATION**

**FACULTY OF INFORMATION TECHNOLOGY**

**Course name:** Real Time Operating System

**SUBJECT**

**Smart greenhouse using Freetos operating system**

Lecture: Phan Van Ca

Members:

1. Trần Quang Nhật ID :20119148
2. Huỳnh Minh Nhựt ID :20119149

**HO CHI MINH CITY – 17/12/2022**

**THANK YOU**

I would like to express my sincere thanks and deep gratitude to the teachers of Ho Chi Minh City University of Technology and Education for creating conditions for me to study Realtime Operating System. and I would also like to thank Mr. Phan Van Ca for enthusiastically guiding my team to complete this final project.

In the process of implementing the model, with the enthusiastic help of Mr. Phan Van Ca, I have gained a lot of valuable knowledge that will help me a lot in my learning and working process in the future:

Once again, I would like to thank Mr. Phan Van Ca for the help and concern in the process of implementing this project so that I can complete this model and look forward to the help of teachers in the following projects.

**TÓM TẮT ĐỀ TÀI**

Trên thị trường hiện nay có rất nhiều thiết bị biến một khu vườn bình thường thành một khu vườn thông minh:

Quy mô lớn như: những trang trại trồng hoa, trồng dâu ở đà lạt.

Nhỏ hơn là những kén chăm sóc đặt biệt cho từng cây, từng chậu hoa,..

Trong đề tài lần này chúng em thực hiện làm vườn thông minh, cũng bao gồm các chức năng cơ bản của 1 khu vườn thông minh bình thường nhưng đi kèm với đó là các giải pháp chăm sóc đặt biệt cho từng loại cây. Điều đó sẽ giúp cho cây phát triển tốt hơn, cho năng suất cao hơn.

Đề tài vườn thông minh sẽ giúp cho các bạn hiểu rõ về nguyên lý hoạt động , cách lắp ráp các bộ phận ,cũng như các lợi ích mà vườn thông minh đem lại cho cuộc sống của con người.

**DANH MỤC BẢNG TÍNH VÀ HÌNH**

**\*Danh mục các bảng tính**

Bảng 1.1 Uno R3 Specification

Bảng 1.2

Bảng 1.3 Thông số kỹ thuật của cảm biến dò line thanh 5 led

Bảng 1.4 Thông số kĩ thuật của động cơ

**DANH MỤC BẢNG HÌNH**

**\*Danh mục các hình**

Hình 1.1 Arduino Uno R3

Hình 1.2 Mạch nguồn Arduino Uno R3

Hình 1.3 Mạch dao động Arduino Uno R3

Hình 1.4 Mạch reset

Hình 1.5 Mạch nạp và giao tiếp máy tính

Hình 1.6 Module điều kiển động cơ L298N

Hình 1.7 Sơ đồ nguyên lí của L298N

Hình 1.8 Cảm biến do line thanh 5 led

Hình 1.9 Sơ đồ nguyên lí cảm biến do line

Hình 1.10 Động cơ

Hình 1.11 Bánh xe

Hình 1.12 Khung xe robot

Hình 1.13 Pin có thể sạc lại

Hình 1.14 Hộp để Pin

Hình 1.15 Dây test board

Hình 2.1 Sơ đồ nối mạch

**MỤC LỤC**

LỜI CẢM ƠN………………………………………………………………..…1

**TÓM TẮT ĐỀ TÀI………………………………………………………….…**2

**DANH MỤC BẢNG TÍNH……………………………………………………**3

**DANH MỤC BẢNG HÌNH………………………………………...………….**4

[**NỘI DUNG ĐỀ TÀI**](#_Toc60140682) 7

[**CHƯƠNG 1:TỔNG QUAN VỀ ĐỀ TÀI**](#_Toc60140683) 8

[**1.1 Giới thiệu về ARDUINO 7**](#_Toc60140684)

[1.1.1 Sơ lược về Arduino Uno R3 7](#_Toc60140685)

[1.1.2 Thiết kế nguồn của Arduino Uno R3 8](#_Toc60140686)

[1.1.3 Thiết kế mạch dao động của Arduino Uno R3 9](#_Toc60140687)

[1.1.4 Thiết kế mạch reset 10](#_Toc60140688)

[1.1.5 Thiết kế mạch nạp và giao tiếp máy tính 12](#_Toc60140689)

[**1.2 Module điều khiển động cơ L289N** 13](#_Toc60140690)

[1.2.1 Giới thiệu sơ lược về module L289N 13](#_Toc60140691)

[1.2.2 Tóm tắc qua chức năng các chân của L298N 14](#_Toc60140692)

[1.2.3 Nguyên lí hoạt động của L298N 14](#_Toc60140693)

[**1.3 Cảm biến dò line** 15](#_Toc60140694)

[1.3.1 Giới thiệu cảm biến dò line 15](#_Toc60140695)

[1.3.2 Nguyên lí của cảm biến do line 16](#_Toc60140696)

[**1.4 Các thành phần khác** 16](#_Toc60140697)

[1.4.1 Động cơ, bánh xe, khung xe 17](#_Toc60140698)

[1.4.2 Pin, hộp để pin, dây test board 19](#_Toc60140699)

[**CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI** 20](#_Toc60140700)

[**2.1 Nối mạch** 20](#_Toc60140701)

[**2.2 Nạp chương trình cho arduino** 20](#_Toc60140702)

[2.2.1 Kết nối Arduino Uno R3 vào máy tính 20](#_Toc60140703)

[2.2.2 Tìm cổng kết nối của Arduino Uno R3 với máy tính 20](#_Toc60140704)

[2.2.3 Nạp chương trình cho Arduino Uno R3 22](#_Toc60140705)

[**2.3 Chương trình điều khiển** 24](#_Toc60140706)

[**CHƯƠNG 3: KẾT QUẢ THỰC HIỆN ĐỀ TÀI** 30](#_Toc60140707)

3.1 Sản phẩm của nhóm………………………………………........................30

[**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN** 32](#_Toc60140710)

[**4.1 Kết quả** 32](#_Toc60140711)

[**4.2 Hướng phát triển** 32](#_Toc60140712)

**TÀI LIỆU THAM KHẢO…………………………………………………**33

**NỘI DUNG ĐỀ TÀI**

**Giới thiệu đề tài**

Ngày nay, công nghệ tự động hóa đã đạt được những thành tự rất to lớn trong sản xuất công nghiệp cũng như trong đời sống. Sản xuất các hệ thống tự động là ngành công nghiệp trị giá hàng tỉ USD và ngày càng phát triển mạnh, trong các hệ thống này chúng ta không thể không thể nhắc tới vườn thông minh, các trang trại nuôi trồng thủy hải sản ứng dụng tự động hóa với những đặc thù riêng mà các hệ thống khác không có. Vườn thông minh cũng là một hệ thống rất quan trọng vì nó trực tiếp nâng cao, cải thiện chất lượng các thực phẩm, đồ uống, cái mà chúng ta nạp vào cơ thể hàng ngày.

**Mục đích đề tài**

Ở đây ví dụ điểm hình nhất là cây hoa đồng tiền, với thân hình rất mảnh khảnh, tuy rất cao nhưng thân lại rất nhỏ, cánh hoa mỏng nhẹ, điều này rất bất lợi cho khi gặp mưa lớn rất dễ làm cho cây bị gãy ngã và các cánh hoa sẽ rất dễ rụng, hơn thế nữa ngày nay nước mưa ngày càng trở nên ô nhiễm và điều đó gây ra các loại bệnh cho cây, thậm chí là chết cây, đây là một loại hoa ưa nắng ấm nhưng lại rất ghét mưa. Chính vì vậy chúng em đưa ra một giải pháp để giải quyết vấn đề này, đó chính là thiết kế một mái che, sẽ tự động vở ra khi trời thoáng nắng, và thu lại ngay lập tức khi có mưa để bảo vệ cây, tránh cây tiếp xúc trực tiếp với các cơn mưa nặng hạt gây gãy, rụng hoa, thậm chí là chết cây vì bị nhiễm độc từ nước mưa.

Đối với các hệ thống thông thường được lập trình theo kiểu tuần tự, giả sử khi độ ẩm trong đất đang thấp và hệ thống sẽ tiến hành tưới tiêu để nâng độ ẩm và chức năng tiếp theo sẽ được thực hiện sau khi hệ thống điều đã hoàn thành việc cải thện độ ẩm cho cây, tức là nếu hệ thống có nhiều chức năng thì chức năng thứ hai muốn chạy thì chức năng thứ nhất đã phải hoàn thành công việc. Điều này sẽ gây trì hoãn cho các chức năng khác ví dụ như chức năng thu mái che như đã đề cập ở phía trên. Từ đó sẽ làm cho hệ thống hoạt động không hiệu quả. Để giải quyết vấn đề này, chúng ta sẽ tìm cách để cho các chức năng có thể hoạt động song song với nhau. Một nhân thời gian thực sẽ làm tốt việc này.

**Sơ lược các bước thực hiện**

- Tìm hiểu về các phần cứng cần thiết.

- Thiết kế một sơ đồ nguyên lí thích hợp.

- Tìm hiểu về nhân thời gian thực.

- Lắp ghép và đánh giá.

CHƯƠNG 1:TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu về ARDUINO

1.1.1 Sơ lược về Arduino Uno R3

Arduino board có rất nhiều phiên bản với hiệu năng và mục đích sử dụng khác nhau như: Arduino Mega, Arduino Nano, Arduino LilyPad, ... Trong số đó, Arduino Uno R3 là một trong những phiên bản được sử dụng rộng rãi nhất bởi chi phí và tính linh động của nó. Arduino Uno R3  là một bảng mạch vi điều khiển nguồn mở dựa trên vi điều khiển Microchip ATmega328 được phát triển bởi Arduino.cc. Bảng mạch được trang bị các bộ chân đầu vào/ đầu ra Digital và Analog có thể giao tiếp với các bảng mạch mở rộng khác nhau. Mạch Arduino Uno R3 thích hợp cho những bạn mới tiếp cận và đam mê về điện tử, lập trình...Dựa trên nền tảng mở do Arduino.cc cung cấp các bạn dễ dàng xây dựng cho mình một dự án nhanh nhất ( lập trình Robot, xe tự hành, điều khiển bật tắt led...)



Hình 1.1 Arduino Uno R3

Bảng 1.1 Thông số Arduino Uno R3

|  |  |
| --- | --- |
| Vi điều khiển | Atmega328 họ 8bit |
| Điện áp hoạt động | 5V DC (chỉ được cấp qua cổng USB) |
| Tần số hoạt động | 16 MHz |
| Dòng tiêu thụ | khoảng 30mA |
| Điện áp khuyên dùng | 7-12V DC |
| Điện áp vào giới hạn | 6-20V DC |
| Số chân Digital I/O | 14 (6 chân hardware PWM) |
| Số chân Analog | 6 (độ phân giải 10bit) |
| Dòng tối đa trên mỗi chân I/O | 30 mA |
| Dòng ra tối đa (5V) | 500 mA |
| Dòng ra tối đa ( 3,3V) | 50 mA |
| Bộ nhớ flash | 32 KB (ATmega328) với 0.5KB dùng bởi bootloader |
| SRAM | 2 KB (ATmega328) |
| EEPROM | 1 KB (ATmega328) |

1.1.2 Thiết kế nguồn của Arduino Uno R3

Phần nguồn của Board mạch Arduino được thiết kế để thực hiện các nhiệm vụ sau:

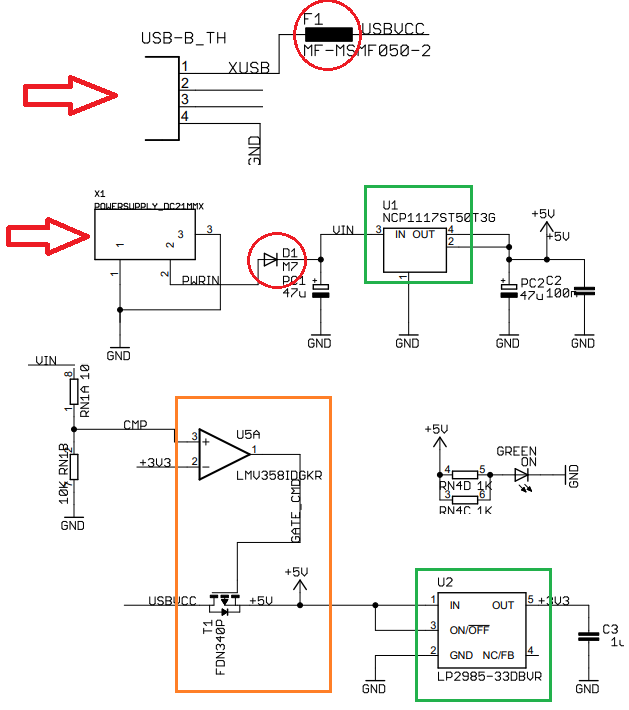
Lựa chọn nguồn cung cấp cho board mạch (khối màu cam trong hình dưới). Board mạch Arduino có thể được cung cấp nguồn bởi Adapter thông qua Jack DC hoặc từ cổng USB (2 mũi tên màu đỏ). Trong trường hợp chỉ có 1 trong 2 nguồn cung cấp thì Board Arduino sẽ sử dụng nguồn cung cấp đó. Trong trường hợp có cả 2 nguồn cung cấp thì Arduino sẽ ưu tiên lựa chọn nguồn cung cấp từ Jack DC thay vì từ cổng USB. Việc ưu tiên này được thực hiện bởi OpAmp trong IC LMV358 và MOSFET FDN340P. Điện áp từ Jack DC sau khi qua Diode bảo vệ D1 thì được gọi là điện áp VIN. Điện áp VIN qua cầu phân áp để tạo thành VIN/2 để so sánh với điện áp 3.3V.  Vì VIN/2 >3.3V nên điện áp đầu ra của OpAmp là 5V, điều này làm cho MOSFET không được kích, nguồn cung cấp cho Board Arduino là từ Jack DC sau khi qua ổn áp.

Tạo ra các điện áp 5v và 3.3v (2 khối màu xanh) để cung cấp cho vi điều khiển và cũng là điểm cấp nguồn cho các thiết bị bên ngoài sử dụng. Mạch Arduino sử dụng IC ổn áp NCP1117 để tạo điện áp 5V từ nguồn cung cấp lớn và IC ổn áp LP2985 để tạo điện áp 3.3V. Đây đều là những IC ổn áp tuyến tính, tuy hiệu suất không cao nhưng ít gợn nhiễu và mạch đơn giản.

Bảo vệ ngược nguồn, quá tải (vòng tròn màu đỏ). F1 là một cầu chì tự phục hồi, trong trường hợp bạn chỉ sử dụng dây cáp USB để cấp nguồn thì tổng dòng tiêu thụ không được quá  500mA. Nếu không cầu chì sẽ ngăn không cho dòng điện

chạy qua. D1 là một Diode, chỉ cho dòng điện 1 chiều chạy qua (từ Jack DC vào mạch), trong trường hợp mạch Arduino của bạn có mắc với các thiết bị khác và có nguồn cung cấp lớn hơn nguồn vào Jack DC, nếu có sai sót chập mạch..vv.. thì sẽ không có trường hợp nguồn các thiết bị bên ngoài chạy ngược vào Adapter.

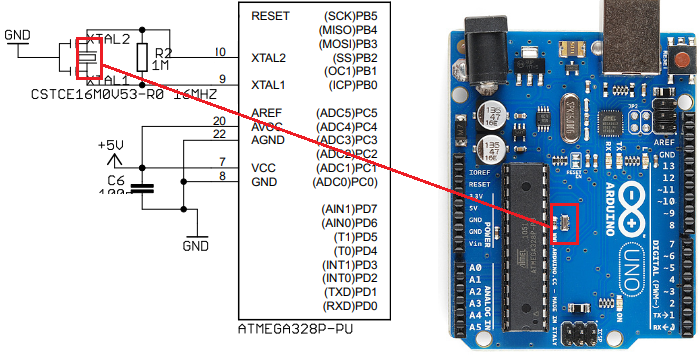
Báo nguồn. Đèn nguồn ON sáng lên báo thiết bị đã được cấp nguồn. Nếu các bạn đã cắm nguồn mà đèn nguồn không sáng thì có thể nguồn cung cấp của bạn đã bị hỏng hoặc jack kết nối lỏng, hoặc mạch Arduino kết nối với các linh kiện bên ngoài bị ngắn mạch.



Hình 1.2 Mạch nguồn Arduino Uno R3

1.1.3 Thiết kế mạch dao động của Arduino Uno R3

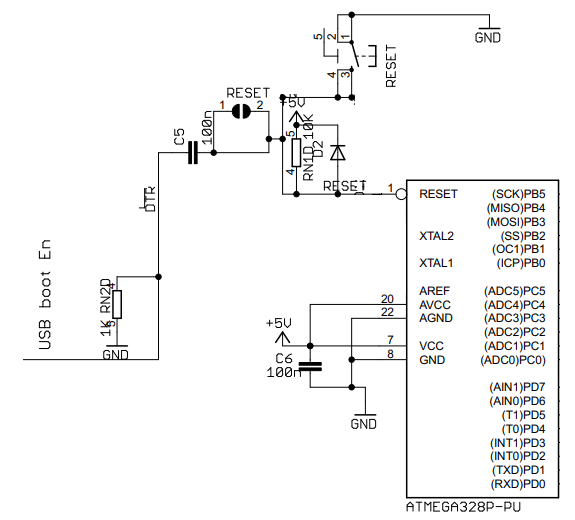
Mạch giao động tạo ra các xung clock giúp cho vi điều khiển hoạt động, thực thi lệnh… Board mạch Arduino Uno R3 sử dụng thạch anh 16Mhz làm nguồn dao động



Hình 1.3 Mạch dao động Arduino Uno R3

1.1.4 Thiết kế mạch reset

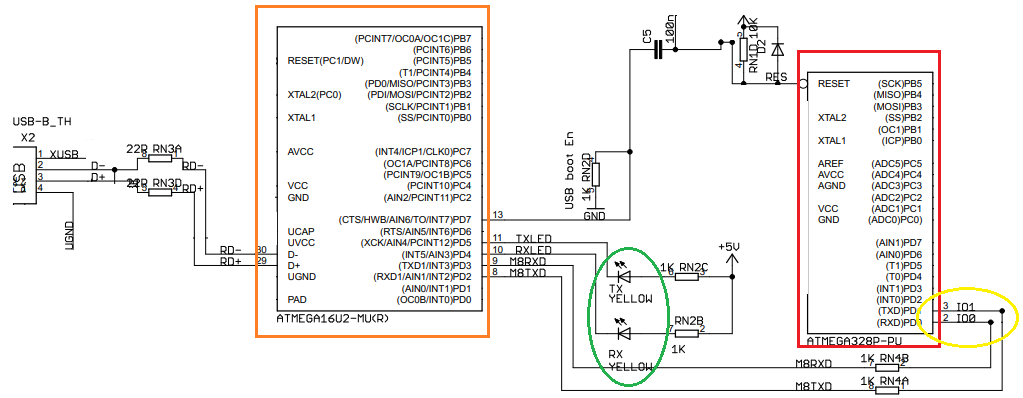
Để vi điều khiển thực hiện khởi động lại thì chân RESET phải ở mức logic LOW (~0V) trong 1 khoản thời gian đủ yêu cầu. Mạch reset của board Arduino UnoR3 phải đảm bảo được 02 việc:  
– Reset bằng tay: Khi nhấn nút, chân RESET nối với GND, làm cho MCU RESET. Khi không nhấn nút chân Reset được kéo 5V.  
– Reset tự động: Reset tự động được thực hiện ngay khi cấp nguồn cho vi điều khiển nhờ sự phối hợp giữa điện trở nối lên nguồn và tụ điện nối đất. Thời gian tụ điện nạp giúp cho chân RESET ở mức LOW trong 1 khoản thời gian đủ để vi điều khiển thực hiện reset.  
– Khởi động vi điều khiển trước khi nạp chương trình mới.



Hình 1.4 Mạch reset

1.1.5 Thiết kế mạch nạp và giao tiếp máy tính

 Vi điều khiển Atmega328P trên Board Arduino UnoR3 đã được nạp sẵn 1 bootloader, cho phép nhận chương trình mới thông qua chuẩn giao tiếp UART (chân 0 và 1) ở những giây đầu sau khi vi điều khiển Reset.  
– Máy tính giao tiếp với Board mạch Arduino qua chuẩn giao tiếp USB (D+/D-), thông qua một vi điều khiển trung gian là ATMEGA16U2 hoặc một IC trung gian là CH340 (thường thấy trong các mạch sử dụng chip dán). Vi điều khiển hoặc IC này có nhiệm vụ chuyển đổi chuẩn giao tiếp USB thành chuẩn giao tiếp UART để nạp chương trình hoặc giao tiếp truyền nhận dữ liệu với máy tính (Serial).  
– Phần thiết kế mạch nạp có tích hợp thêm 02 đèn LED,nên khi nạp chương trình các bạn sẽ thấy 2LED này nhấp nháy. Còn khi giao tiếp, nếu có dữ liệu từ máy tính gửi xuống vi điều khiển thì đèn LED Rx sẽ nháy. Còn nếu có dữ liệu từ vi điều khiển gửi lên máy tính thì đèn Tx sẽ nháy.



Hình 1.5 Mạch nạp và giao tiếp máy tính

1.2 Rain Sensor

1.2.1 Introduce to rain sensor

Rain sensor which detect the sky being rain or hot. The principle of operation is based on the conduction of the electrodes.



Figure 1.6: Rain sensor

Table 1.2 Specification

|  |  |
| --- | --- |
| Power supply | 5VDC |
| Power led | Green |
| Led detect rain | Green |
| Principle of operation | Water drop to the board that create a conductive environment |
| Type of signals | Analog and Digital |
| Used IC | LM358 |

1.2.2 Pin configuration

- Vcc: Power supply for sensor

- Gnd: Ground

- D0: Digital output signal

- A0: Analog output signal

1.2.3 Principle of operation of rain sensor

When water fall into the board that will create a conductive invironment and the IC LM358 will compare with the reference voiltage and generate an analog and a digital output signal.

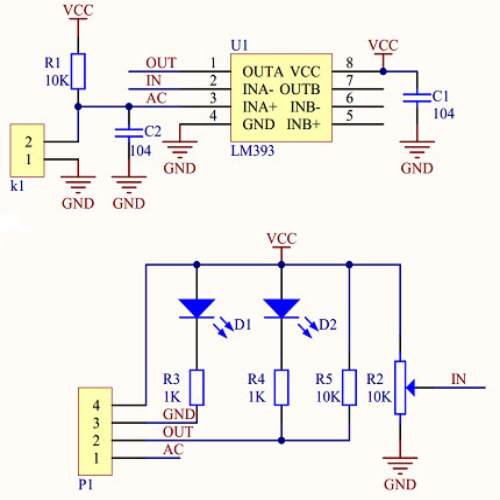


Figure: 1.7 Schematic of rain sensor

1.3 Soil moisture sensor

1.3.1 Introduce to soil moisture sensor

**Humidity sensor used to determin the soil moisture, detect water. When using, please ban the probe into the ground. We will read the output signal on pin A0 and D0.**

Table 1.3 Specification of soil moisture sensor

|  |  |
| --- | --- |
| Power supply | 5VDC |
| Powed led | Green |
| Led detect | Green |
| Type of signals | Analog(A0) and digital(D0) |
| Used IC | LM358 |

Whiteboard

Description automatically generated with medium confidence

Figure: 1.8 Soil moisture sensor

1.3.2 Principle of operationof soil moisture sensor

Principle of operation depend the conduction of environment. If the conduction of the environment is high that mean the moisture is high and vice versa, then the IC358 will compare to the reference voiltage and generate output signal.

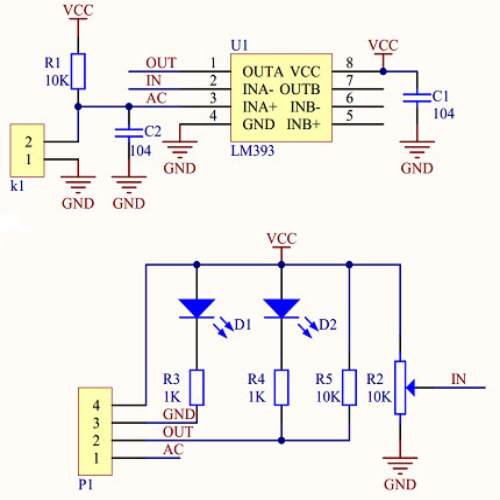


Figure: 1.9 Schematic of soil moisture sensor

1.4 Other components

1.4.1 LCD, Servo, relay, pump,..

- In this project, we use 1602 LCD that integarted the I2C module. It can be controlled by I2C communication (SDA, SCL)

Figure: 1.10 LCD 1602

Table 1.4 Specification of pump

|  |  |
| --- | --- |
| Power supply | 3­-5VDC |
| Consumption current | 300mA |
| Size | 23\*43mm |
| Input diameter | 5mm |
| Output diameter | 7mm |



Figure: 1.10 Pump

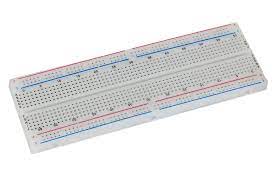


Figure: 1.11 test board

1.4.2 Pin, pin socket, test board wires



Figure: 1.12 Recharge pin

Figure: 1.13 Pin socket Figure 1.14 Test board wires

CHƯƠNG 2: THỰC HIỆN ĐỀ TÀI

2.1 Nối mạch

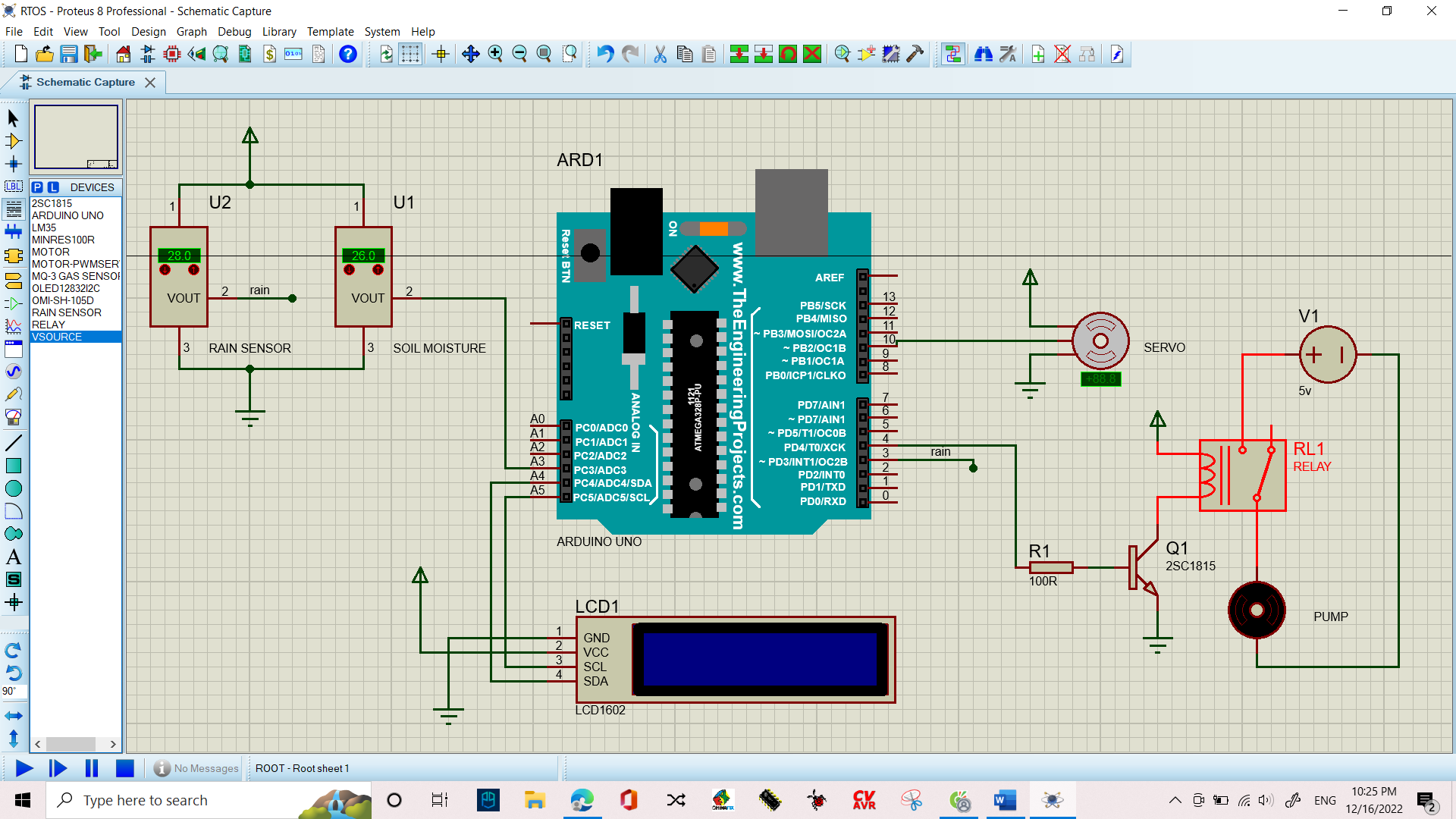


Figure 2.1 Schematic

2.2 Nạp chương trình cho arduino

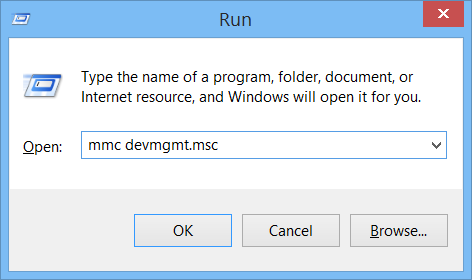
2.2.1 Kết nối Arduino Uno R3 vào máy tính

Arduino Uno R3 được kết nối với máy tính bằng dây cáp USB

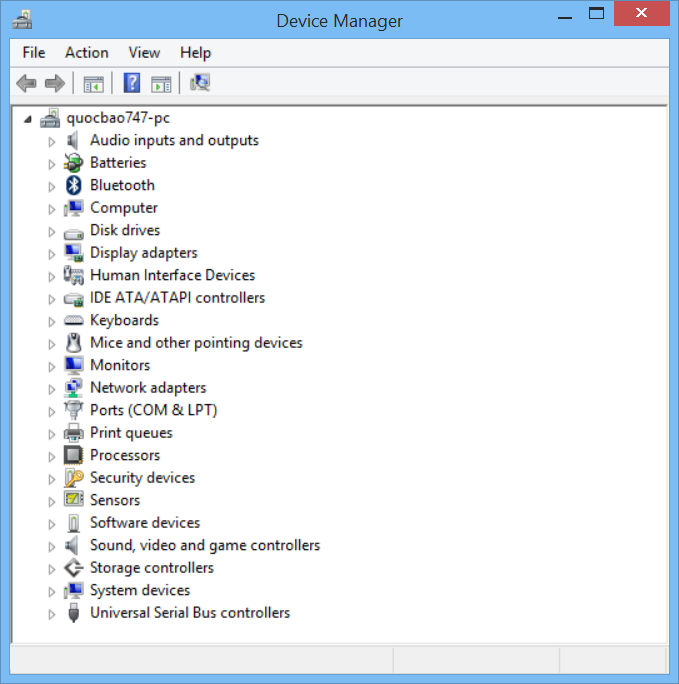
2.2.2 Tìm cổng kết nối của Arduino Uno R3 với máy tính

Khi Arduino Uno R3 kết nối với máy tính, nó sẽ sử dụng một cổng COM (Communication port - cổng dữ liệu ảo) để máy tính và bo mạch có thể truyền tải dữ liệu qua lại thông qua cổng này. Windows có thể quản lí đến 256 cổng COM. Để tìm được cổng COM đang được sử dụng để máy tính và mạch Arduino UNO R3 giao tiếp với nhau, bạn phải mở chức năng Device Manager của Windows.

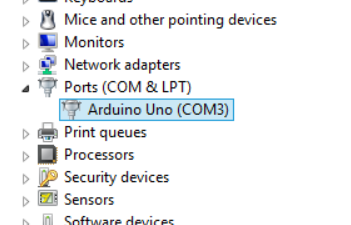
Bạn mở cửa sổ Run và gõ lệnh mmc devmgmt.msc.



Sau đó bấm Enter, cửa sổ **Device Manager** sẽ hiện lên.



Mở mục **Ports (COM & LPT)**, bạn sẽ thấy cổng COM Arduino Uno R3 đang kết nối

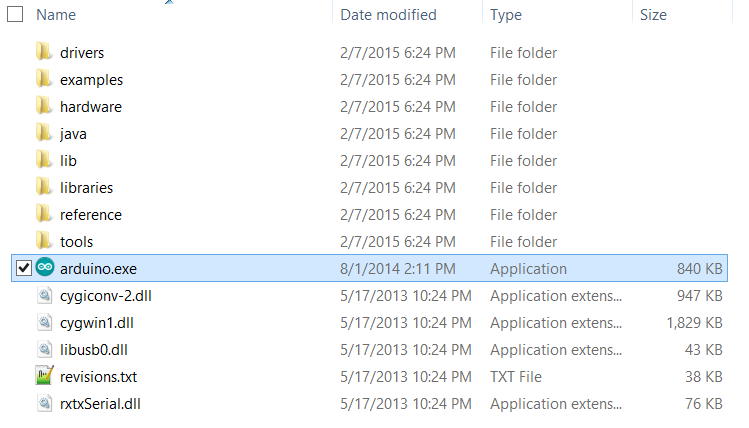


Cổng kết nối ở đây là COM3.

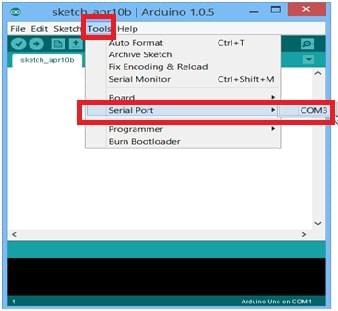
Thông thường, trong những lần kết nối tiếp theo, Windows sẽ sử dụng lại cổng COM3 để kết nối nên bạn không cần thực hiện thêm thao tác tìm cổng COM này nữa.

2.2.3 Nạp chương trình cho Arduino Uno R3

-Đầu tiên khởi động Arduino IDE



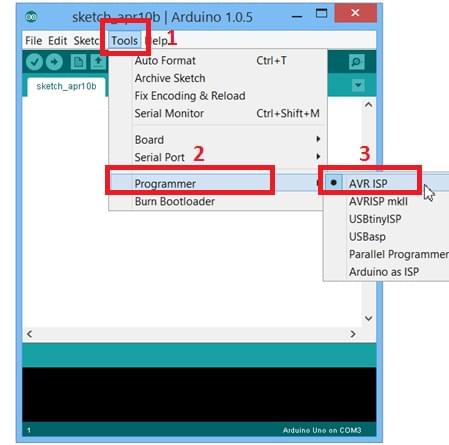
-Tiếp theo vào menu Tools -> Serial Port -> chọn cổng Arduino đang kết nối với máy tính.



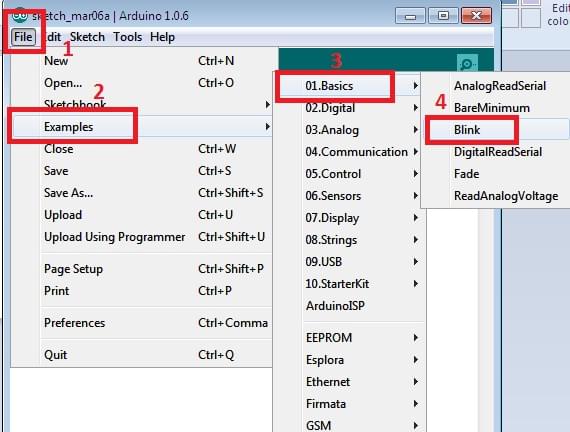
-Xác nhận cổng COM của Arduino IDE ở góc dưới cùng bên phải cửa sổ làm việc.



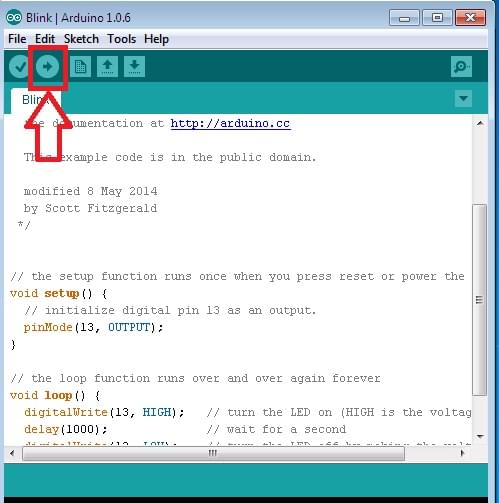
-Vào menu Tools -> Programmer -> chọn AVR ISP



-Tiếp theo ta nạp mã nguồn chương cho arduino uno r3.



-Và ta làm 1 thao tác cuối cùng để nạp chương trình. Đúp chuột vào chổ chỉ của dấu mũi tên để đổ chương trình xuống arduino uno r3



2.3 Chương trình điều khiển

// Import some important libraries

#include <Arduino\_FreeRTOS.h>

#include <Servo.h>

#include <LiquidCrystal\_I2C.h>

// outut and input pin configuration

#define motor 4

#define certain 10

#define rainSensor 3

#define HumSensor A3

// Declare two global variavles in order to tranfer data in other tasks

int globalHumidity;

boolean globalStatus;

// LDC Definition

LiquidCrystal\_I2C lcd(0x27,16,2);

// Servo Definition

Servo myservo;

//Declare Task

void TaskCareObject(void \*pvParameters);

void TaskWeatherDetect(void \*pvParameters);

void TaskPrint(void \*pvParameters);

void setup() {

Serial.begin(9600);

// Declare in, out direction

pinMode(rainSensor, INPUT);

pinMode(certain, OUTPUT);

pinMode(motor, OUTPUT);

// initialize LCD

lcd.init();

// turn on led on screen

lcd.backlight();

// Assign servo pin

myservo.attach(certain);

// Erase screen

lcd.clear();

// Create Task

xTaskCreate(

TaskCareObject

,"Task1"

,128

,NULL

,1

,NULL

);

xTaskCreate(

TaskWeatherDetect

,"Task2"

,128

,NULL

,1

,NULL

);

xTaskCreate(

TaskPrint

,"Task3"

,128

,NULL

,1

,NULL

);

// schdule

vTaskStartScheduler();

}

void loop() {

// put your main code here, to run repeatedly:

}

// check moisture and controll the pump

void TaskCareObject(void \*pvParameters)

{

int temp =0;

while(1)

{

// convert analog signal to moisture percent

temp = analogRead(HumSensor);

temp = abs(temp - 420);

temp = abs(600 - temp);

temp = map(temp, 0, 605, 0, 100);

globalHumidity = temp;

// if the moisture smaller than 70%, turn on the pump

if(temp < 70)

{

digitalWrite(motor, HIGH);

vTaskDelay(10/ portTICK\_PERIOD\_MS);

}

else // when condition is wrong, then turn of the pump

{

digitalWrite(motor, LOW);

vTaskDelay(10/ portTICK\_PERIOD\_MS);

}

}

}

// detect rain and controll the servo

void TaskWeatherDetect(void \*pvParameters)

{

boolean \_status = 0;

while(1)

{

\_status = digitalRead(rainSensor);

globalStatus = \_status;

if(\_status)

{

myservo.write(180);

}

else

{

myservo.write(0);

}

}

}

// wire moisture and weather on screen

void TaskPrint(void \*pvParameters)

{

while(1)

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Humidity:");

lcd.print(globalHumidity);

lcd.print("%");

lcd.setCursor(0,1);

if(!globalStatus)

{

lcd.print("It's rain");

}

else

{

lcd.print("It is not rain");

}

vTaskDelay(500/ portTICK\_PERIOD\_MS);

}

}

CHƯƠNG 3: KẾT QUẢ THỰC HIỆN ĐỀ TÀI

3.1 Sản phẩm của nhóm

A picture containing text

Description automatically generated

Text

Description automatically generated

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết luận

- Kết quả chính xác

- Tốc độ đáp ứng nhanh

- Hoạt động ổn định

4.2 Hướng phát triển

- Quản lí được nhiều khối cùng lúc

- Xây dựng giải thuật hoàn chỉnh hơn.

- Xây dựng thêm các chức năng khác

**TÀI LIỆU THAM KHẢO**

- <https://www.freertos.org/fr-content-src/uploads/2018/07/FreeRTOS_Reference_Manual_V10.0.0.pdf>

- https://docs.arduino.cc/hardware/uno-rev3?queryID=87a7c90bcbfb4e57cfa7f2943f52bda7

- https://www.arduino.cc/reference/en/libraries/freertos/