

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

\*\*\*\*\*



**HCMUTE**

**BÁO CÁO ĐỒ ÁN MÔN**

**BẾ CÁ THÔNG MINH**

SVTH: HOÀNG VIẾT HƯNG

MSSV: 18119082

Khóa: 18

Ngành: CÔNG NGHỆ KĨ THUẬT MÁY TÍNH

GVHD: THS. NGUYỄN NGÔ LÂM

Tp Hồ Chí Minh --- tháng 7, năm 2021



----\*\*\*----

Tp. Hồ Chí Minh, ngày 7 tháng 7 năm 2021

## NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên sinh viên: Hoàng Việt Hưng

MSSV: 18119082

Ngành: Công nghệ kỹ thuật máy tính

Lớp: 18119CL3B

Giảng viên hướng dẫn: Nguyễn Ngô Lâm

ĐT: 0866507210

Ngày nhận đề tài: 20/03/2021

Ngày nộp đề tài: 07/07/2021

1. Tên đề tài : Bể cá thông minh

2. Các số liệu, tài liệu ban đầu: Các kiến thức liên quan đến môn Mạch điện, điện tử cơ bản, kỹ thuật số, kiến trúc và tổ chức máy tính.

3. Nội dung thực hiện đề tài:

- Tóm tắt lý thuyết có liên quan và đặc tính của từng linh kiện
- Hoàn thiện sơ đồ mạch nguyên lý của hệ thống
- Hoàn thiện bản thiết kế
- Mô phỏng hệ thống
- Test từng khối giao tiếp với khối Xử lý trung tâm
- Test toàn hệ thống
- Viết báo cáo trình bày kết quả

4. Sản phẩm:

Hệ thống bể cá thông minh

GIẢNG VIÊN HƯỚNG DẪN



---\*\*\*---

## **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên: Hoàng Việt Hưng.....MSSV: 18119082.....

Ngành: Công nghệ kỹ thuật Máy tính.....

Tên đề tài: Bé cá thông minh .....

Họ và tên Giáo viên hướng dẫn: Nguyễn Ngô Lâm.....

### **NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.....  
.....  
.....  
.....

2. Ưu điểm:

.....  
.....  
.....  
.....

3. Khuyết điểm:

.....  
.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

.....

6. Điểm: .....(Bằng chữ: ..... )

.....  
Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên hướng dẫn  
(Ký & ghi rõ họ tên)

-----\*\*\*-----

## PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên Sinh viên: Hoàng Việt Hưng.....MSSV: 18119082.....

Ngành: Công nghệ kỹ thuật Máy tính.....

Tên đề tài: Bể cá thông minh.....

.....

Họ và tên Giáo viên phản biện:.....

### NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....  
.....  
.....  
.....  
.....

2. Ưu điểm:

.....  
.....  
.....  
.....  
.....

3. Khuyết điểm:

.....  
.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

.....

6. Điểm: .....(Bằng chữ:.....)

.....

Tp. Hồ Chí Minh, ngày tháng năm 2021

Giáo viên phản biện

(Ký & ghi rõ họ tên)

## MỤC LỤC

<b>NHIỆM VỤ ĐỒ ÁN MÔN HỌC.....</b>	<b>2</b>
<b>PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....</b>	<b>3</b>
<b>PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN.....</b>	<b>4</b>
<b>DANH MỤC CÁC TỪ VIẾT TẮT.....</b>	<b>7</b>
<b>DANH MỤC CÁC BẢNG BIỂU.....</b>	<b>8</b>
<b>DANH SÁCH CÁC HÌNH ẢNH, BIỂU ĐỒ.....</b>	<b>9</b>
<b>CHƯƠNG 1 .....</b>	<b>11</b>
<b>TỔNG QUAN.....</b>	<b>11</b>
1.1. GIỚI THIỆU.....	11
1.2. LÍ DO CHỌN ĐỀ TÀI.....	11
1.3. CÔNG NĂNG HỆ THỐNG.....	11
1.4. LINH KIỆN CẦN DÙNG.....	12
1.5. BỐ CỤC.....	12
<b>CHƯƠNG 2.....</b>	<b>13</b>
<b>CƠ SỞ LÝ THUYẾT.....</b>	<b>13</b>
2.1. CƠ BẢN VỀ ARDUINO.....	13
2.1.1. Lý thuyết về Arduino.....	13
2.1.2. Nguyên lý và cơ chế hoạt động của Arduino.....	13
2.1.3. Arduino UNO R3.....	14
2.2. CƠ BẢN VỀ LCD 20x4.....	17
2.2.1. Lý thuyết về LCD 20x4.....	17
2.2.2. Nguyên lý và cơ chế hoạt động của LCD.....	17
2.2.3. LCD 20x4.....	18
2.3. CƠ BẢN VỀ SERVO MG90S.....	21
2.3.1. Lý thuyết về Servo.....	21
2.3.2. Nguyên lý và cơ cấu hoạt động của Servo.....	21
2.3.3. Servo MG90S.....	22
2.4. CƠ BẢN VỀ WATER SENSOR.....	24
2.4.1. Lý thuyết về cảm biến mực nước.....	24
2.4.2. Nguyên lý và cơ chế hoạt động.....	24
2.4.3. Cảm biến mực nước (Water Sensor).....	24
2.5. CƠ BẢN VỀ CẢM BIẾN ĐỘ ĐÚC.....	26
2.5.1. Lý thuyết về cảm biến độ đục.....	26
2.5.2. Nguyên lý và cơ chế hoạt động.....	26
2.5.3. Cảm biến đo độ đục (SKU SEN0189).....	26
2.6. CƠ BẢN VỀ DS3231.....	28
2.6.1. Lý thuyết về hệ thống thời gian thực.....	28
2.6.2. Nguyên lý và cơ chế hoạt động.....	28
2.6.3. Module thời gian thực DS3231.....	29
<b>CHƯƠNG 3.....</b>	<b>32</b>
<b>THIẾT KẾ HỆ THỐNG.....</b>	<b>32</b>
3.1. YÊU CẦU VÀ SƠ ĐỒ KHỐI CỦA HỆ THỐNG.....	32
3.1.1. YÊU CẦU CỦA HỆ THỐNG.....	32

3.1.2. SƠ ĐỒ KHỐI, CHỨC NĂNG CỦA TỪNG KHỐI TRONG HỆ THỐNG.....	32
3.1.3. HOẠT ĐỘNG CỦA HỆ THỐNG.....	33
<b>3.2. THIẾT KẾ HỆ THỐNG PHẦN CỨNG.....</b>	<b>34</b>
3.2.1. KHỐI THỜI GIAN THỰC.....	34
3.2.2. KHỐI CẢM BIẾN MỤC CHẤT LỎNG.....	35
3.2.3. KHỐI CẢM BIẾN ĐỘ ĐỨC.....	36
3.2.4. KHỐI SERVO.....	36
3.2.5. KHỐI LED.....	37
3.2.6. KHỐI HIỂN THỊ.....	38
3.2.7. KHỐI XỬ LÝ TRUNG TÂM.....	39
3.2.8. KHỐI NGUỒN.....	41
<b>3.3. THIẾT KẾ HỆ THỐNG PHẦN MỀM.....</b>	<b>41</b>
3.3.1. LUU ĐỒ GIẢI THUẬT.....	41
3.3.2. PHÂN TÍCH THUẬT TOÁN.....	42
<b>CHƯƠNG 4.....</b>	<b>44</b>
KẾT QUẢ, NHẬN XÉT, ĐÁNH GIÁ.....	44
4.1. KẾT QUẢ.....	44
4.2. NHẬN XÉT VÀ ĐÁNH GIÁ.....	49
<b>CHƯƠNG 5.....</b>	<b>50</b>
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	50
5.1. KẾT LUẬN.....	50
5.2. HƯỚNG PHÁT TRIỂN.....	50
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>51</b>
<b>PHỤ LỤC.....</b>	<b>52</b>

## **DANH MỤC CÁC TỪ VIẾT TẮT**

LED: Light Emitting Diode

LCD: Liquid Crystal Display

AI: Artificial Intelligence

SPI: Serial Peripheral Interface

SDA: Serial Data

SCL: Serial Clock

MOSI: Master Out Slave In

MISO: Master In Slave Out

SCK: Serial Clock

UART: Universal Asynchronous Receive Transmitter

I2C: Inter-Integrated Circuit

I/O: Input/Output

EEPROM: Electrically Erasable Programmable Read-Only Memory

PWM: Pulse Width Modulation

TSS: Total Suspended Solids

NTU: Nephelometric Turbidity Units

IOTs: Internet Of Things

## **DANH MỤC CÁC BẢNG BIỂU**

BẢNG 1.1 BẢNG THỐNG KÊ CÁC LINH KIỆN DÙNG TRONG HỆ THỐNG...	12
BẢNG 2.1 BẢNG THỐNG KÊ MỘT SỐ ĐẶC TÍNH CƠ BẢN CỦA ARDUINO UNO R3.....	15
BẢNG 2.2 BẢNG ĐỊA CHỈ CÁC Ô NHỚ TRONG LCD 20X4.....	19
BẢNG 2.3 BẢNG CHỨC NĂNG TỪNG CHÂN CỦA LCD 20X4.....	20
BẢNG 2.4 BẢNG CHI TIẾT SƠ ĐỒ CHÂN CỦA CẢM BIẾN ĐỘ ĐỨC.....	27
BẢNG 2.5 BẢNG SƠ ĐỒ CHÂN CỦA DS3231.....	31
BẢNG 3.1 BẢNG CẤP NGUỒN CHO CÁC LINH KIỆN.....	41

## DANH SÁCH CÁC HÌNH ẢNH, BIỂU ĐỒ

HÌNH 2.1 GIAO DIỆN ARDUINO IDE.....	13
HÌNH 2.2 SƠ ĐỒ CHÂN ARDUINO UNO R3 (CHIP SMD).....	14
HÌNH 2.3 KÝ HIỆU CỦA ARDUINO UNO R3 (CHIP SMD).....	14
HÌNH 2.4. STRUCTURE OF A LCD.....	17
HÌNH 2.5 LCD 20X4.....	18
HÌNH 2.6 SƠ ĐỒ CHÂN CỦA LCD 20X4.....	18
HÌNH 2.7 KÍ HIỆU LCD.....	18
HÌNH 2.8 SƠ ĐỒ NỐI CHÂN CỦA LCD VỚI MODULE I2C.....	19
HÌNH 2.9 CẤU TRÚC CỦA MỘT SERVO.....	21
HÌNH 2.10 SERVO MG90S.....	22
HÌNH 2.11 SƠ ĐỒ CHÂN SERVO MG90S.....	22
HÌNH 2.12 KÍ HIỆU SERVO MG90.....	22
HÌNH 2.13. SƠ ĐỒ NỐI CHÂN GIỮA ARDUINO VỚI SERVO MG90S.....	22
HÌNH 2.14 NGUYÊN LÝ HOẠT ĐỘNG CỦA CẢM BIẾN MỤC NƯỚC.....	24
HÌNH 2.15 WATER SENSOR.....	24
HÌNH 2.16 SƠ ĐỒ CHÂN WATER SENSOR.....	24
HÌNH 2.17 KÍ HIỆU WATER SENSOR.....	24
HÌNH 2.18 SƠ ĐỒ NỐI CHÂN GIỮA ARDUINO VỚI WATER SENSOR.....	25
HÌNH 2.19 CẢM BIẾN ĐỘ ĐỨC.....	26
HÌNH 2.20 SƠ ĐỒ CHÂN CẢM BIẾN ĐỘ ĐỨC.....	26
HÌNH 2.21 KÝ HIỆU CẢM BIẾN ĐỘ ĐỨC.....	26
HÌNH 2.22 SƠ ĐỒ NỐI CHÂN GIỮA CẢM BIẾN ĐO ĐỘ ĐỨC VỚI ARDUINO.....	27
HÌNH 2.23 GIAO TIẾP I2C CHO DS3231.....	29
HÌNH 2.24 DS3231.....	29
HÌNH 2.25 SƠ ĐỒ CHÂN DS3231.....	29
HÌNH 2.26 KÝ HIỆU DS3231.....	29
HÌNH 2.27 SƠ ĐỒ NỐI CHÂN GIỮA DS3231 VỚI ARDUINO.....	30
HÌNH 3.1 SƠ ĐỒ KHỐI CỦA HỆ THỐNG.....	33
HÌNH 3.2 KẾT NỐI ARDUINO VỚI DS3231.....	34
HÌNH 3.3 KẾT NỐI WATER SENSOR VỚI ARDUINO.....	35
HÌNH 3.4 KẾT NỐI ĐỘ ĐỨC VỚI ARDUINO.....	36
HÌNH 3.5 KẾT NỐI SERVO VỚI ARDUINO.....	36
HÌNH 3.6 KẾT NỐI LED VỚI ARDUINO.....	37
HÌNH 3.7 KẾT NỐI LCD VỚI ARDUINO.....	38
HÌNH 3.8 SƠ ĐỒ NGUYÊN LÝ CỦA HỆ THỐNG.....	40

HÌNH 3.9 GIẢI THUẬT CỦA TOÀN HỆ THỐNG.....	41
HÌNH 3.10 GIẢI THUẬT CHO NÚT NHẤN THÚ 2 TRONG HỆ THỐNG.....	41
HÌNH 4.1 MẠCH IN CỦA HỆ THỐNG.....	43
HÌNH 4.2 MẠCH IN SAU KHI LÀM RA.....	43
HÌNH 4.3 MẠCH HOÀN CHỈNH.....	44
HÌNH 4.4 LCD KHI NUỚC TRONG, MỤC NUỚC ÔN ĐỊNH.....	44
HÌNH 4.5 LCD KHI NUỚC ĐỤC, MỤC NUỚC ÔN ĐỊNH.....	45
HÌNH 4.6 LCD KHI NUỚC TRONG, MỤC NUỚC THẤP.....	46
HÌNH 4.7 LCD KHI NUỚC ĐỤC, MỤC NUỚC THẤP.....	46
HÌNH 4.8 LED TRANG TRÍ KHI KHÔNG SÁNG VÀ SÁNG.....	47

# CHƯƠNG 1

## TỔNG QUAN

### 1.1. GIỚI THIỆU.

Đồ án 1 là sự tổng hợp kiến thức về các môn học quan trọng như mạch điện, điện tử cơ bản, kỹ thuật số,... Từ những kiến thức và kinh nghiệm đã được trau dồi qua các bài học, em đã có đủ điều kiện để thực hiện đồ án 1 của mình.

### 1.2. LÍ DO CHỌN ĐỀ TÀI.

Hiện nay với việc nuôi thú cưng đã trở thành thú vui của mọi người nói chung và những nhà có hoàn cảnh khá giả nói riêng. Và trong đó “cá kiểng” không thể nghi ngờ là một trong những thú nuôi phổ biến và khá được nhiều người yêu thích. Nhưng bên cạnh đó là việc nuôi cá cũng có chút không dễ dàng, nếu chúng không chăm sóc đúng mực thì chúng rất dễ bị chết đi, thậm chí có thể gây ra mùi cá tanh lan ra trong gia đình. Điều đó khẳng định rất dễ xảy ra với các gia đình thường hay không ở nhà do đi làm, đi công tác quá nhiều. Từ đó, tôi có một giải pháp giúp các hộ gia đình yên tâm hơn trong việc nuôi cá. Đó là tạo ra một hồ cá thông minh nơi có thể hỗ trợ người dùng chăm sóc những thú cưng của mình.

### 1.3. CÔNG NĂNG HỆ THỐNG.

Hệ thống bể cá thông minh được tạo ra với một số công năng chính sau:

- Hệ thống có thể tự động cho cá ăn hoặc dùng nút bấm điều khiển cho cá ăn.
- Xác định được độ đục của nước, từ đó cảnh báo người nuôi thay nước cho cá khi độ đục vượt ngưỡng.
- Xác định được mực nước trong bể cá. Đồng thời cũng đưa ra cảnh báo nếu thông số vượt ngưỡng.
- Hiển thị những thông số đo được từ cảm biến lên màn hình LCD.

## 1.4. PHẠM VI NGHIÊN CỨU

Đề tài nghiên cứu chỉ dùng ở mức độ điều khiển các thiết bị thông qua việc tự động đã được thiết lập sẵn hoặc thông qua việc sử dụng các nút nhấn điều khiển theo ý của người dùng. Và mọi hoạt động của hệ thống sẽ được lập trình thông qua phần mềm Arduino IDE.

## 1.5. LINH KIỆN CẦN DÙNG

Thông kê một số linh kiện sẽ được sử dụng để thiết kế hệ thống:

STT	Linh Kiện	Số Lượng
1.	Vi điều khiển Arduino UNO R3	1
2.	Servo MG90S	1
3.	LCD 20x4	1
4.	Module I2C LCD	1
5.	Cảm biến mực nước, độ đục	1
6.	DS3231	1
7.	Led (5mm)	2
8.	Trở R (100Ohm, 220Ohm, 4.7kOhm)	10
9.	Button	2
10.	Dây Bus	Tùy trường hợp

Bảng 1.1. Bảng thống kê các linh kiện dùng trong hệ thống

## 1.6. BỐ CỤC.

Báo cáo đồ án 1 sẽ được trình bày theo bố cục như sau:

- Chương 1: Tổng quan của đồ án
- Chương 2: Cơ sở lý thuyết
- Chương 3: Thiết kế hệ thống
- Chương 4: Kết quả, nhận xét và đánh giá
- Chương 5: Kết luận và hướng phát triển

# CHƯƠNG 2

## CƠ SỞ LÝ THUYẾT

### 2.1. CƠ BẢN VỀ ARDUINO.

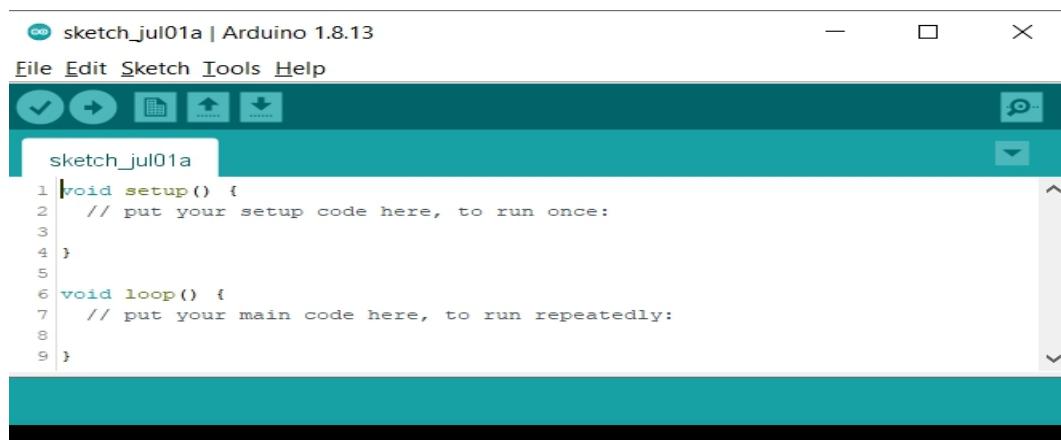
#### 2.1.1. Lý thuyết về Arduino.

Arduino là công cụ phù hợp với hầu hết các sinh viên chuyên ngành kỹ thuật hiện nay. Nhất là trong các lĩnh vực về điều khiển tự động, IoTs, hay thậm chí là AI. Arduino được biết đến như là một dự án mã nguồn mở (open source), kết hợp giữa phần mềm và phần cứng để mọi người có thể dễ dàng tiếp cận và làm quen với lập trình nhúng. Điều đặc biệt ở đây chính là Arduino được thiết kế để có thể lập trình bằng ngôn ngữ bậc cao như C hay C++, các board mạch đều được chuẩn hóa.

#### 2.1.2. Nguyên lý và cơ chế hoạt động của Arduino.

Để có thể dùng được Arduino ngoài việc ta cần có một board Arduino thích hợp. Mà bên cạnh đó còn cần tải phần mềm Arduino IDE.

Đây là link tải của Arduino: <https://www.arduino.cc/en/software>



**Hình 2.1** Giao diện Arduino IDE.

Đương nhiên để code cho Arduino hoạt động điều mà mỗi người cần biết là phải quên với việc lập trình qua C hoặc là C++.

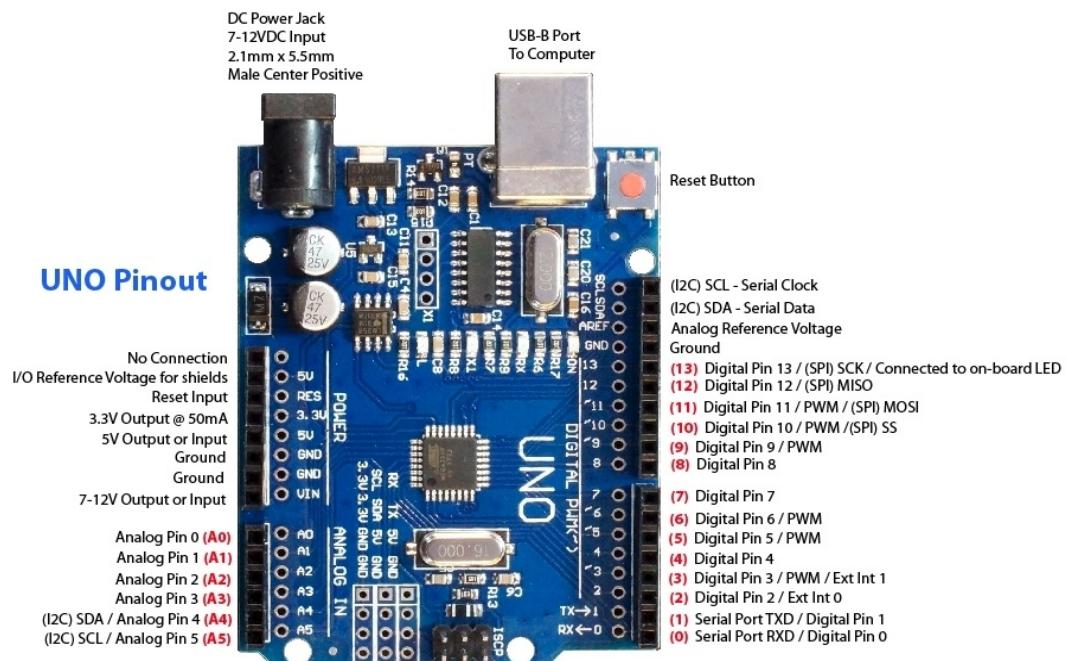
Tùy thuộc vào ứng dụng ta muốn tìm hiểu, những module cảm biến mà ta muốn dùng mà sẽ có những thư viện phù hợp để người mới vào nghề cũng có

thể dễ dàng lập trình được. Với một cộng đồng rộng lớn như Arduino thì thư viện luôn là thứ không thể thiếu. Và việc chọn thư viện vào để hỗ trợ bản thân trong quá trình tìm hiểu ứng dụng là tùy thuộc vào mỗi người lựa chọn.

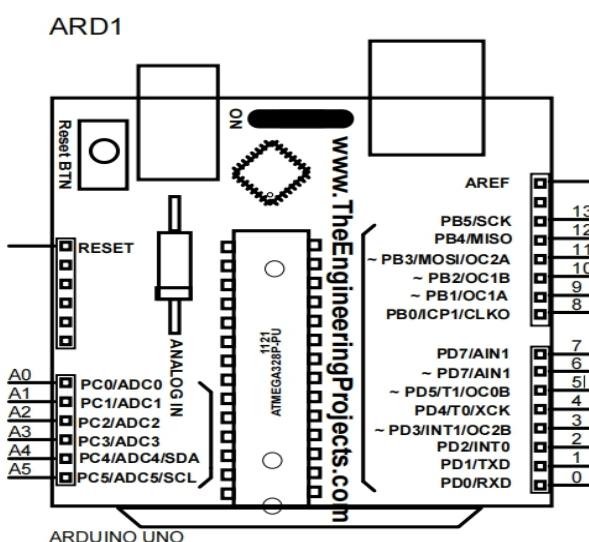
Arduino được cấu tạo để giao tiếp với nhiều loại chuẩn truyền thông khác nhau như SPI, UART, I2C. Và còn có nhiều chân ngoại vi I/O để thực hiện kết nối với các cảm biến các module được kèm theo tùy vào loại ứng dụng là người lập trình mong muốn tìm hiểu.

### 2.1.3. Arduino UNO R3. [5]

#### 1) Hình ảnh.



**Hình 2.2 Sơ đồ chân Arduino Uno R3 (chip SMD)**



**Hình 2.3 Ký hiệu của Arduino Uno R3 (chip SMD)**

## 2) Các đặc điểm về Arduino Uno R3.

a) Một số đặc điểm, đặc tính của Arduino UNO R3.

<b>STT</b>	Microcontroller (loại vi điều khiển):	ATmega328
<b>1.</b>	Operating Voltage (điện áp hoạt động):	5 V
<b>2.</b>	Input Voltage (recommended) (mức điện áp hoạt động bình thường):	Từ 7 V đến 12 V
<b>3.</b>	Input Voltage (limits) (mức điện áp hoạt động giới hạn):	Từ 6 V đến 20 V
<b>4.</b>	Digital I/O Pins (số lượng chân I/O số):	Tổng có 14 chân ngoại vi số trong đó 6 chân PWM
<b>5.</b>	Analog Input Pins (số lượng chân I/O tương tự):	Tổng có 6 chân ngoại vi tương tự
<b>6.</b>	DC Current per I/O Pin (dòng điện định mức trên mỗi chân I/O):	40 mA
<b>7.</b>	DC Current for 3.3V Pin (dòng điện trên chân nguồn 3.3V):	50 mA
<b>8.</b>	Flash memory (bộ nhớ flash):	32 KB
<b>9.</b>	SRAM (bộ nhớ RAM tĩnh):	2 KB
<b>10.</b>	EEPROM (bộ nhớ EEPROM):	1 KB
<b>11.</b>	Clock Speed (tốc độ xung nhịp):	16 MHz

**Bảng 2.1** Bảng thống kê một số đặc tính cơ bản của Arduino UNO R3

b) Công suất và năng lượng.

Ta có thể cấp nguồn để Arduino Uno R3 hoạt động bằng 2 cách:

- Cách 1: ta có thể cấp nguồn qua cổng USB interface nơi giao tiếp với máy tính để nạp code để chạy chương trình
- Cách 2: ta có thể cấp nguồn qua cổng “External Power Supply”, để cấp nguồn tại cổng này ta cần một adapter với một jack cáp 2.1mm

Bên cạnh đó ta yêu cầu được cấp nguồn từ 7V đến 12V nếu cấp ít hơn 7V thì các chân I/O sẽ đưa tín hiệu ra có điện áp có thể nhỏ hơn 5V dẫn đến mạch chạy không ổn định, ngược lại nếu điện áp lớn hơn 12V sẽ có khả năng dẫn đến mạch bị quá tải hoặc gây ra hư hại cho mạch.

c) Bộ nhớ.

Ta có thể đọc hoặc viết dữ liệu trên bộ nhớ EEPROM với dung lượng khoảng 1KB.

d) Các chân I/O của Arduino UNO R3.

Digital Pin:

- **Serial: 0 (RX) and 1 (TX).** Dùng để giao tiếp UART (chỉ có 1 cổng)
- **External Interrupts: 2 and 3.** Dùng để thực hiện ngắt ngoài, xử lý ngắt.
- **PWM: 3, 5, 6, 9, 10, and 11.** Độ phân giải của các tín hiệu xung PWM là 8-bit.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** Dùng để giao tiếp SPI giúp cho Arduino giao tiếp với các thiết bị như LCD, bộ điều khiển video game, bộ điều khiển cảm biến các loại, đọc thẻ nhớ SD và MMC...

Analog Pin (ngõ vào Analog được trang bị với độ phân giải 10-bit):

- **I2C: 4 (SDA) and 5 (SCL).** Dùng để giao tiếp I2C, kết nối được với các loại cảm biến như thermostat của CPU, tốc độ quạt, một số màn hình OLED/LCD, đọc real-time clock, chỉnh âm lượng cho một số loại loa...

e) Cài đặt Board Arduino cho Arduino IDE.

Đầu tiên vào phần mềm Arduino IDE, sau đó ở thanh tùy chọn, chọn File -> Preference. Tại mục Addition Boards Manager URLs: nhập vào đường dẫn “[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)”.

Bên cạnh đó trước khi kết nối Arduino với máy tính thông qua cổng Usb ta cần phải tải driver cho Arduino trước. Thông thường thì driver để Arduino có thể giao tiếp được với máy tính là CH340. Đây là đường link để tải driver: “<https://sparks.gogo.co.nz/ch340.html>”.

## 2.2. CƠ BẢN VỀ LCD 20x4.

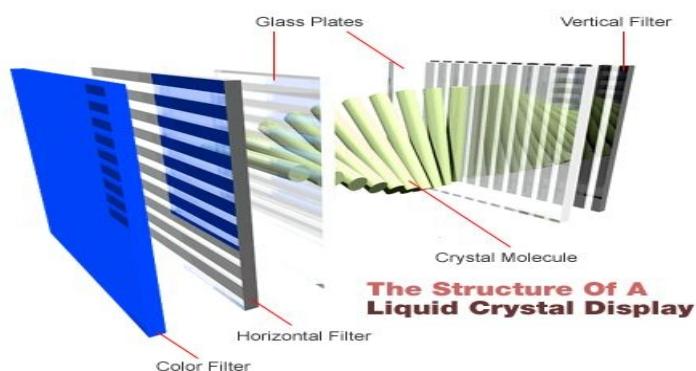
### 2.2.1. Lý thuyết về LCD 20x4.

Màn hình LCD hay còn được gọi dưới cái tên Liquid Crystal Display, đang được ứng dụng khá rộng rãi vì nó tương thích với nhiều loại vi điều khiển khác nhau. Led LCD 20\*4 được kết hợp từ nhiều led đơn tạo ra nhiều khôi ô vuông có kích thước ngang như mỗi ô vuông đó sẽ có thể hiện ra 1 ký tự hoặc một chữ số.

Có nhiều loại LCD khác nhau như LCD 16x2, LCD 20x4, LCD tích hợp sẵn Arduino. Điều làm nên sự thu hút của LCD là nó có thể dễ dàng đưa vào mạch ứng dụng với nhiều giao thức giao tiếp khác nhau có thể kể đến như I2C, trực tiếp kết nối qua bus,... Bên cạnh đó giá thành của một LCD lại không quá cao phù hợp với nhiều người mới bắt đầu học điện tử.

### 2.2.2. Nguyên lý và cơ chế hoạt động của LCD.

LCD được ra đời vào năm 1970, theo đúng như cái tên của nó LCD là một loại vật chất phản xạ ánh sáng khi điện thế thay đổi, Nó hoạt động dựa trên nguyên tắc ánh sáng nền (Back Light) . Nó bao gồm một lớp chất lỏng nằm giữa 2 lớp kính phân cực ánh sáng. Bình thường, khi không có điện áp các tinh thể này được xếp thẳng hàng giữa hai lớp cho phép ánh sáng truyền theo hình xoáy ốc và khi đó ta sẽ không nhìn thấy được ánh sáng từ LCD. Hai bộ lọc phân cực, 2 bộ lọc màu và 2 bộ cân chỉnh sẽ xác định cường độ ánh sáng đi qua và màu nào được tạo ra trên mỗi pixel. Khi có điện áp cấp vào LCD thì lớp cân chỉnh sẽ tạo ra một xung điện tích, để cân chỉnh lại các tinh thể lỏng. Từ đó các tinh thể sẽ ngăn cản không cho ánh sáng đi qua, tạo nên một điểm sáng hiển thị trên LCD. Thật ra các điểm sáng trên màn hình LCD là một transistor cực nhỏ điều khiển bật tắt của bóng led tại vị trí điểm sáng đó.



Hình 2.4. Structure of A LCD.

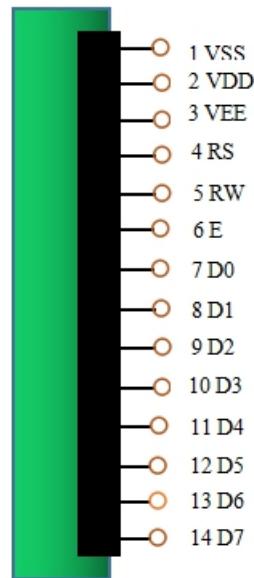
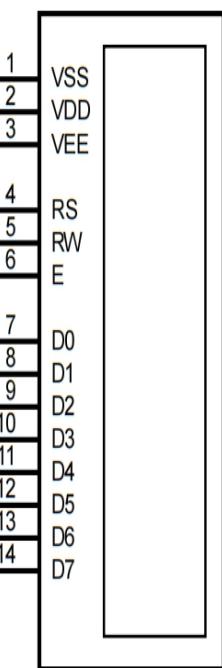
Qua đó ta biết được ánh sáng được phát ra từ đèn nền có tác dụng chỉnh độ tương phản của ánh sáng. Khi chúng ra ở ngoài trời hay những nơi có ánh sáng cao thì ta nên tăng mức sáng lên mới có thể thấy được điểm sáng hiển thị trên LCD, ngược lại nếu ở trong điều kiện hoàn cảnh tối tăm thì nên giảm mức sáng xuống.

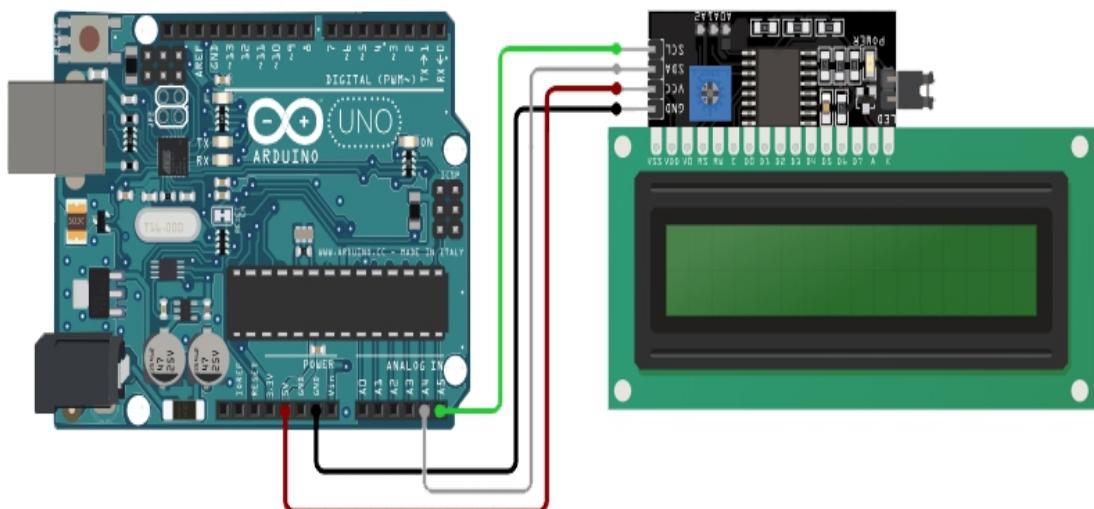
Mặt khác để có thể sử dụng được LCD ta có thể sẽ tốn không ít chân vi điều khiển, điều này hoàn toàn là không xứng đáng. Vậy nên việc kết hợp giữa màn hình LCD với module I2C LCD luôn là lựa chọn ưu dùng của rất nhiều người. Việc dùng giao thức truyền thông I2C không những giảm bớt việc tiêu tốn nhiều chân vi điều khiển còn giúp cho việc kết nối bus giữa chúng dễ dàng hơn. Thậm chí việc lập trình cũng trở nên dễ dàng hơn với thư viện hỗ trợ giao tiếp giữa màn hình LCD với module I2C, và thư viện đó là “*LiquidCrystal\_I2C*”.

Ta có thể thực hiện download thư viện trên thông qua đường link dưới đây “<https://www.arduinolibraries.info/libraries/liquid-crystal-i2-c>”.

### 2.2.3. LCD 20x4. [6]

1) Hình ảnh.

		
<b>Hình 2.5 LCD 20x4</b>	<b>Hình 2.6 Sơ đồ chân</b>	<b>Hình 2.7 Kí hiệu LCD của LCD 20x4</b>



**Hình 2.8** Sơ đồ nối chân của LCD với module I2C

## 2) Các đặc điểm của LCD 20x4.

### a) Một số đặc trưng của LCD 20x4.

- Gồm 4 dòng, mỗi dòng có 20 ô kí tự, mỗi ô kí tự được tạo thành bởi ma trận 5x8 ô nhỏ.
- Dùng vi xử lý HD44780 hoặc vi điều khiển khác tương đương để tạo thành.
- Hoạt động ở điện áp 5V (cũng có loại có thể hoạt động ở điện áp 3V).

### b) Địa chỉ lưu trữ của các ô nhớ.

DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53
DD RAM Address	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
DD RAM Address	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67

**Bảng 2.2** Bảng địa chỉ các ô nhớ trong LCD 20x4

c) Chân I/O của LCD 20x4.

STT	Pin	Symbol (kí hiệu)	Function (chức năng)
1.	1	VSS	Ground (đất)
2.	2	VDD	+ 3 V or + 5 V (cấp nguồn)
3.	3	V0	Contrast adjustment (điều chỉnh mức sáng nền)
4.	4	RS H/L	Thanh ghi lựa chọn tín hiệu
5.	5	R/W	H/L read/write signal (tín hiệu đọc/ghi)
6.	6	E	H -> L enable signal (ngõ vào điều kiện)
7.	7	DB0	H/L data bus line (đường bus dữ liệu đưa vào)
8.	8	DB1	H/L data bus line (đường bus dữ liệu đưa vào)
9.	9	DB2	H/L data bus line (đường bus dữ liệu đưa vào)
10.	10	DB3	H/L data bus line (đường bus dữ liệu đưa vào)
11.	11	DB4	H/L data bus line (đường bus dữ liệu đưa vào)
12.	12	DB5	H/L data bus line (đường bus dữ liệu đưa vào)
13.	13	DB6	H/L data bus line (đường bus dữ liệu đưa vào)
14.	14	DB7	H/L data bus line (đường bus dữ liệu đưa vào)
15.	15	A	Power supply for LED (4.2 V)
16.	16	K	Power supply for B/L (0 V)
17.	17	NC/VEE	NC or negative voltage output
18.	18	NC	NC connection

**Bảng 2.3** Bảng chức năng từng chân của LCD 20x4.

## 2.3. CƠ BẢN VỀ SERVO MG90S.

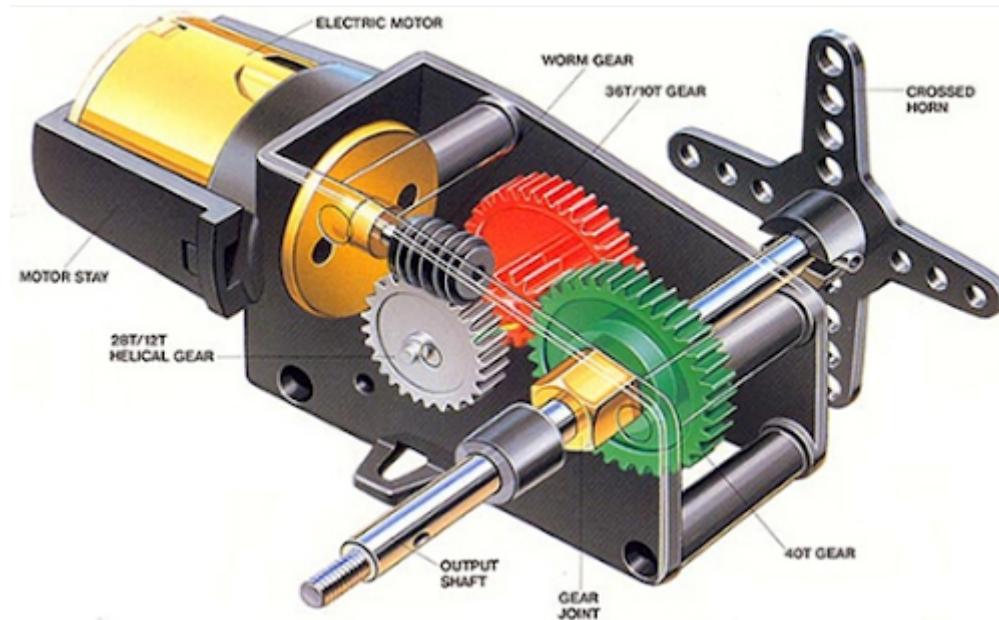
### 2.3.1. Lý thuyết về Servo.

Động cơ Servo là một động cơ được kết hợp chung với bộ mã hóa quay (encoder) hoạt động chiết áp để tạo thành một cơ cấu quay Servo. Bộ chiết áp này được dùng để cung cấp vị trí dựa trên tín hiệu analog, tuy nhiên do ảnh hưởng từ nhiệt độ thay đổi trong môi trường sẽ có thể làm cho bộ chiết áp đổi khi xác định vị trí không chính xác tốc độ phản hồi lại không ổn định. Mặt khác bộ mã hóa sẽ sử dụng bộ điều khiển PID để điều khiển chính xác vị trí với một tốc độ phản hồi luôn được duy trì ổn định.

### 2.3.2. Nguyên lý và cơ cấu hoạt động của Servo.

Động cơ Servo được thiết kế là những hệ thống hồi tiếp vòng kín. Tín hiệu ra của động cơ sẽ được nối với một mạch điều khiển. Khu động cơ quay, vận tốc và vị trí sẽ được hồi tiếp về mạch điều khiển này. Nếu có bất cứ lý do nào ngăn cản chuyển động quay của động cơ, cơ cấu hồi tiếp sẽ nhận thấy tín hiệu ra chưa đạt được vị trí mong muốn. Mạch điều khiển tiếp tục chỉnh sai lệch cho động cơ đạt được điểm chính xác.

Phương thức liên lạc được dùng phổ biến trong các động cơ Servo là liên lạc vô tuyến (Servo RC - Radio Controlled) và những Servo dạng như vậy thường dùng trong các máy bay, xe hơi.

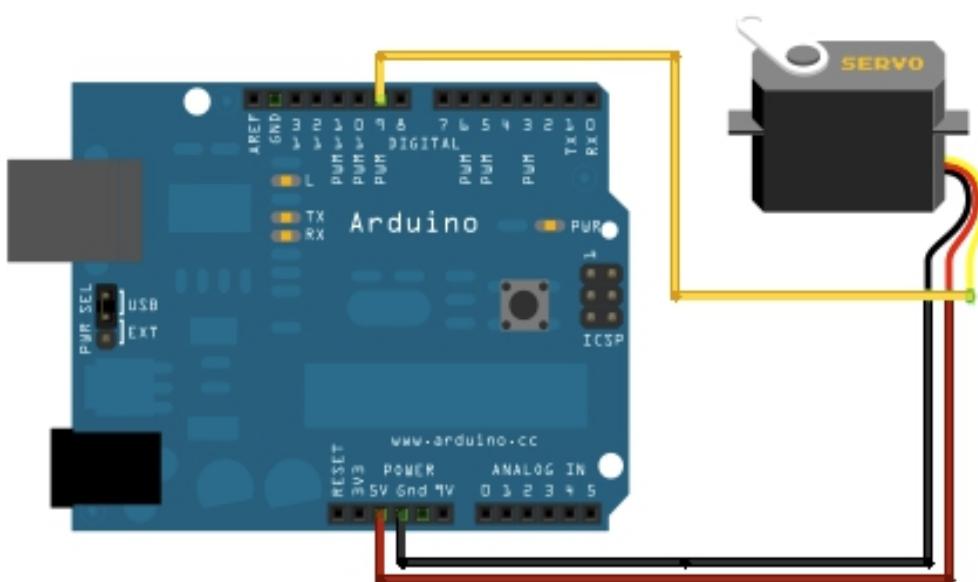
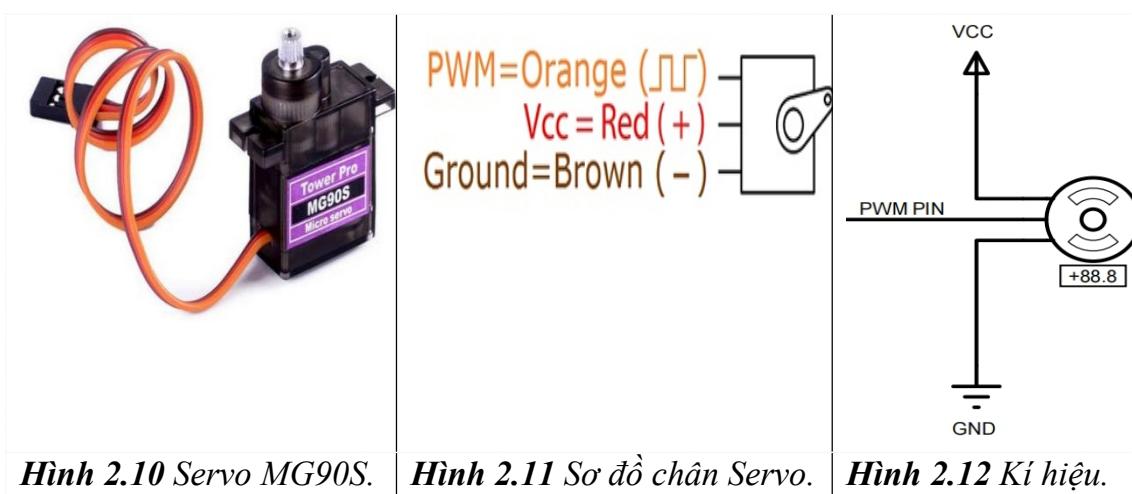


Hình 2.9 Cấu trúc của một Servo.

Đối với Servo dùng trong điện tử như MG90S, SG90, Motor Servo thì hoạt động như thế này: Rotor của động cơ điện (phản quay quanh trục) được làm bằng nam châm vĩnh cửu có từ trường mạnh. Phần vỏ của động cơ có các lá thép hoạt nhựa được ghép lại với nhau (nhằm khắc phục dòng điện Foucault) được quấn thành các cuộn dây riêng biệt. Và được cấp nguồn dựa trên hệ thống điều khiển, trong một chu kỳ có thể quay theo lập trình hoặc tự động.

### 2.3.3. Servo MG90S. [7]

1) Hình ảnh.



**Hình 2.13.** Sơ đồ nối chân giữa Arduino với Servo MG90S.

## 1) Các đặc điểm của Servo MG90S.

a) Một số đặc trưng của Servo MG90S.

- Weight (trọng lượng): 13.4 g.
- Dimension (kích thước): 22.5 x 12 x 35.5 mm.
- Stall torque: 1.8 kgf·cm (4.8V ), 2.2 kgf·cm (6 V).
- Operating speed (tốc độ): 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V).
- Operating voltage (điện áp hoạt động): 4.8 V - 6V.
- Dead band width (băng thông chệch): 5  $\mu$ s.

b) Chân I/O của Servo.

- Pin 1: chân ngõ vào điều khiển PWM (bus color is orange).
- Pin 2: chân cấp nguồn cho servo hoạt động (bus color is red)
- Pin 3: chân mass cho servo (bus color is brown).

c) Thư viện hỗ trợ.

Để dễ dàng sử dụng Servo cộng đồng người dùng đã cung cấp ra một thư viện hỗ trợ để có thể lập trình cho Servo đó là “Servo.h”.

Ta có thể thực hiện download thư viện trên thông qua đường link dưới đây “<https://www.arduino.cc/reference/en/libraries/servo/>”.

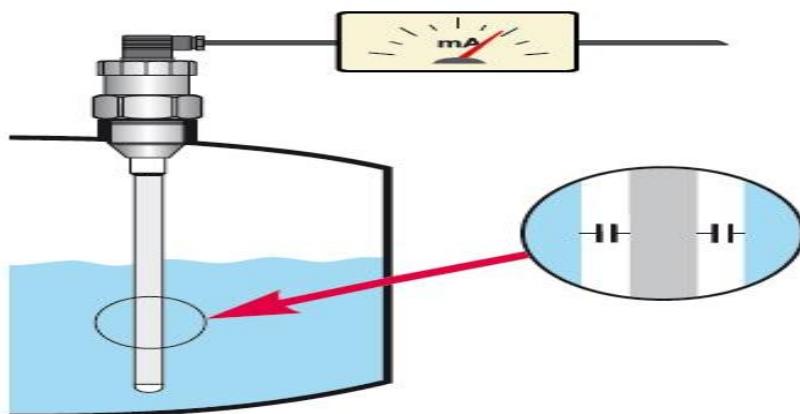
## 2.4. CƠ BẢN VỀ WATER SENSOR.

### 2.4.1. Lý thuyết về cảm biến mực nước.

Cảm biến mực nước là một trong những cảm biến khá phổ biến hiện nay.

### 2.4.2. Nguyên lý và cơ chế hoạt động.

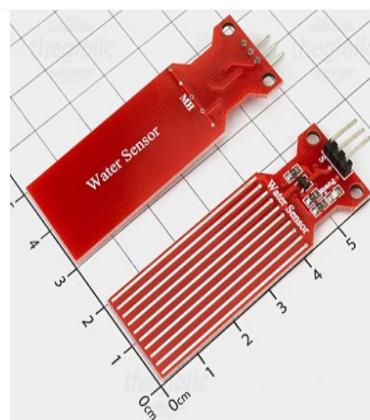
Cảm biến sẽ dựa vào sự thay đổi của điện dung khi mực nước trong bồn chứa nước thay đổi. Từ đó tín hiệu sẽ được chuyển đổi và truyền về vi điều khiển để xử lý.



**Hình 2.14** Nguyên lý hoạt động của cảm biến mực nước

### 2.4.3. Cảm biến mực nước (Water Sensor).

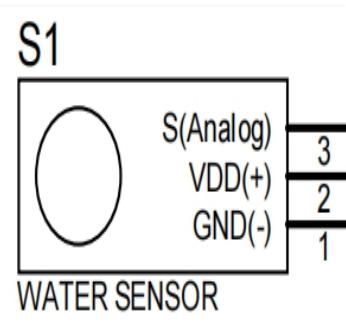
#### 1) Hình ảnh.



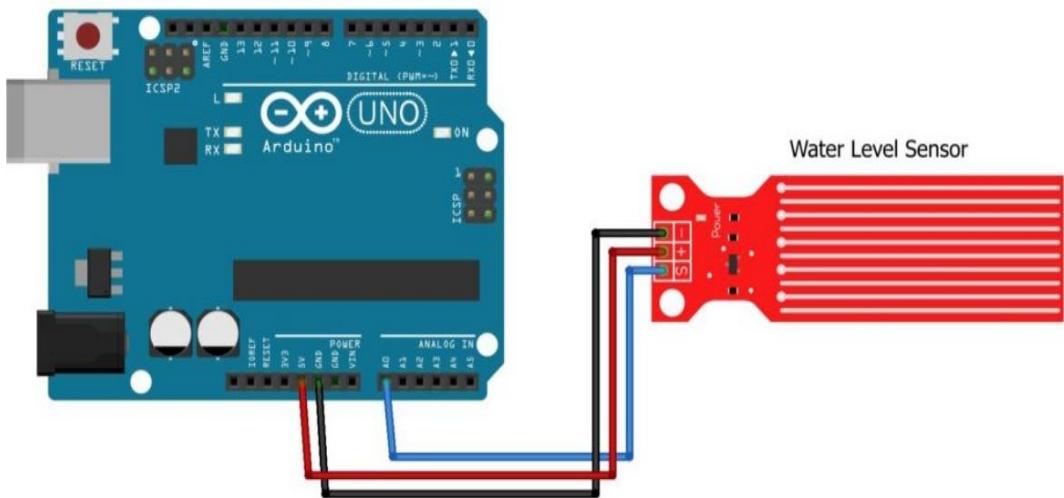
**Hình 2.15** Water sensor.



**Hình 2.16** Sơ đồ chân.



**Hình 2.17** Kí hiệu



**Hình 2.18** Sơ đồ nối chân giữa Arduino với Water Sensor.

2) Các đặc điểm cơ bản của cảm biến mức nước (Water Sensor).

a) Một số đặc điểm của cảm biến mức nước (Water Sensor).

- Tên: Cảm biến mức nước
- Điện áp hoạt động: DC3-5V
- Dòng định mức: ít hơn 20mA
- Loại cảm biến: Analog
- Diện tích phát hiện: 40mm x 16mm
- Quy trình sản xuất: phun FR4 hai mặt
- Nhiệt độ làm việc: 10°C -30°C
- Độ ẩm hoạt động: 10% -90% không ngưng tụ
- Trọng lượng: 3,5g
- Kích thước: 62mm x 20mm x 8mm

b) Chân I/O của cảm biến mức nước (Water Sensor).

- Pin 1: Ground (đất).
- Pin 2: VCC (Nguồn 5v).
- Pin 3: Chân ngõ ra Analog S.

## 2.5. CƠ BẢN VỀ CẢM BIẾN ĐỘ ĐỤC.

### 2.5.1. Lý thuyết về cảm biến độ đục.

Như ta đã biết độ đục của nước có thể xem như là thước đo để đánh giá mức độ an toàn của nước. Độ đục là đặc tính quang học của nước và là biểu hiện của lượng ánh sáng được phân tán bởi vật thể trong nước khi ánh sáng được chiếu xuyên qua mảnh nước đó. Cường độ ánh sáng tán xạ càng chứng tỏ độ đục càng cao.

Nước đục nguyên nhân là do các hạt lơ lửng tồn lại bên trong như bùn, đá sét, vật liệu vô cơ, các chất hữu cơ như tảo, sinh vật phù du, vật liệu phân rã. Nói chung lại các yếu tố chính ảnh hưởng tới độ đục của nước là chất rắn lơ lửng trong nước, chay hữu cơ hòa tan màu, chất hữu cơ hòa tan huỳnh quang,...

Độ đục sẽ càng cao khi các hạt vật chất trong nước càng nhiều ảnh hưởng đến việc thâm nhập của ánh sáng.

### 2.5.2. Nguyên lý và cơ chế hoạt động.

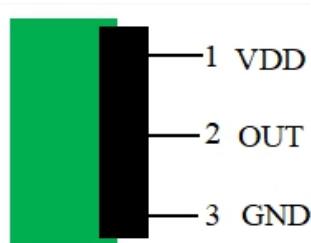
Cảm biến đo độ đục sẽ sử dụng ánh sáng để phát hiện các hạt lơ lửng trong nước bằng cách đo truyền ánh sáng và tỷ lệ tán xạ ánh sáng đó, mà thay đổi với có lượng tổng chất rắn lơ lửng (TSS) trong nước. Khi TSS tăng, mức độ đục của chất lỏng tăng khi đó ánh sáng nhận được thông qua cảm biến sẽ giảm đi.

### 2.5.3. Cảm biến đo độ đục (SKU SEN0189). [8]

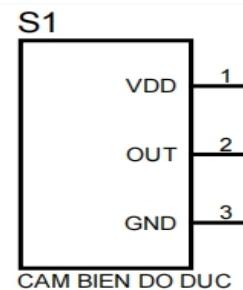
1) Hình ảnh.



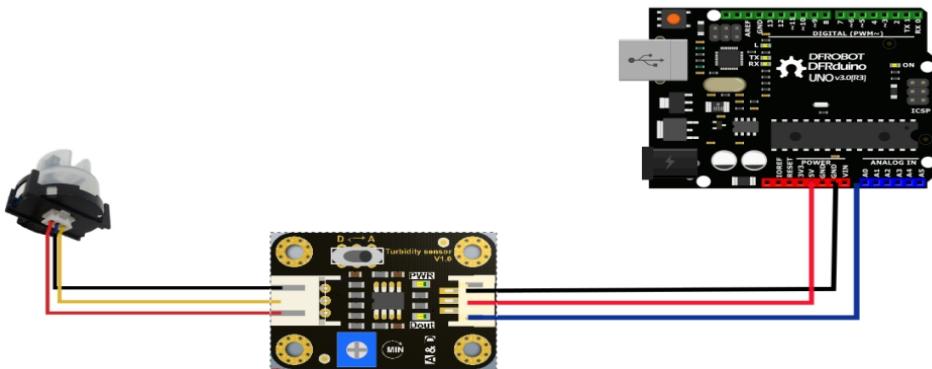
**Hình 2.19** Cảm biến độ đục.



**Hình 2.20** Sơ đồ chân.



**Hình 2.21** Ký hiệu.



**Hình 2.22** Sơ đồ nối chân giữa cảm biến đo độ đục với Arduino

## 2) Các đặc điểm cơ bản của cảm biến đo độ đục.

a) Một số đặc trưng của cảm biến đo độ đục.

- Điện áp hoạt động: 5V
- Dòng điện làm việc: 40mA (Max)
- Thời gian đáp ứng: < 500ms
- Điện trở cách điện: 100M(Min)
- Đầu ra Analog 0 ~ 4.5V
- Đầu ra Digital: High/Low (Có thẻ điều chỉnh giá trị ngưỡng bằng biến trờ)
- Nhiệt độ hoạt động: 5°C ~ 90°C
- Nhiệt độ dự trữ: -10°C ~ 90°C
- Kích thước: 38mm\*28mm\*10mm
- Trọng lượng: 30g
- Dải đo độ đục là 0 - 1000 NTU.

b) Chân I/O của cảm biến đo độ đục.

STT	Pin	Symbol	Function
1.	1	D	Chân dữ liệu đầu ra
2.	2	+	VCC: 4-5V
3.	3	-	Ground

**Bảng 2.4** Bảng chi tiết sơ đồ chân của cảm biến độ đục

Chân dữ liệu đầu ra “D” có thẻ chọn là chân ra kiểu Digital hoặc Analog thông qua công tắc trên mạch chuyển đổi của cảm biến.

Khi ta gạt công tắc sang D thì dữ liệu đầu ra là dạng số 0 và 1, ngưỡng so sánh được điều chỉnh thông qua biến trờ trên mạch chuyển đổi của cảm biến

Như ở lại khi ta gạt công tắc sang A thì dữ liệu đầu ra là dạng Analog, có thể nội suy ra chỉ số độ đục của nước,

Một số ví dụ cho ngõ ra Analog

NTU 500: 3.27V

NTU 50: 4.1V

NTU 0.5 4.21V

## 2.6. CƠ BẢN VỀ DS3231.

### 2.6.1. Lý thuyết về hệ thống thời gian thực.

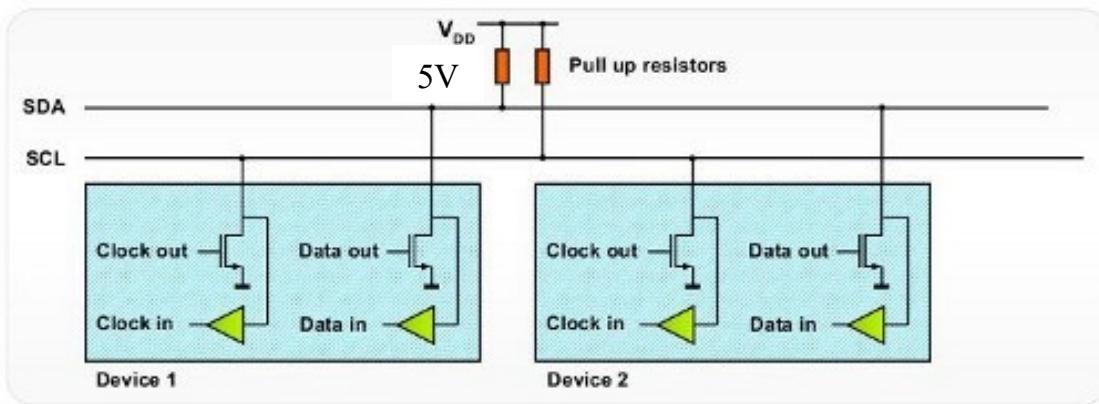
Hệ thống thời gian thực là một hệ thống có khả năng thực hiện thời gian thực nghĩa là hệ thống phải thực hiện được chức năng của mình trong một khoảng thời gian nhất định và nhỏ nhất có thể. Khi đã đáp ứng được yêu cầu như thế này thì hệ thống đó được gọi là hệ thống thời gian thực.

Qua đó thời gian thực luôn có vai trò trong việc hoạt động các mạch điện, các project IoT,.. Có thời gian thực vừa giúp hệ thống quản lý chính xác được thời gian vừa có thể thực hiện đáp ứng các sự kiện hoặc ngõ ra một cách nhanh chóng và chính xác.

### 2.6.2. Nguyên lý và cơ chế hoạt động.

DS3231 là module thời gian thực hay còn được gọi dưới cái tên RTC (Real time clock). Module DS3231 bên trong có tồn tại 7 thanh ghi dữ liệu mỗi thanh ghi có 8-bit đại diện cho “thứ, ngày tháng, năm, giờ, phút, giây). Ngoài ra DS3231 còn có một thanh ghi dùng để điều khiển ngõ ra phụ và 56 thanh ghi trống được dùng như là RAM cho Module DS3231.

DS3231 được đọc thông qua chuẩn truyền thông I2C nên do đó để đọc được và ghi từ DS3231 thông qua chuẩn truyền thông này



- 2 wire bus:
  - SDA: Serial Data Line
  - SCL: Serial Clock Line

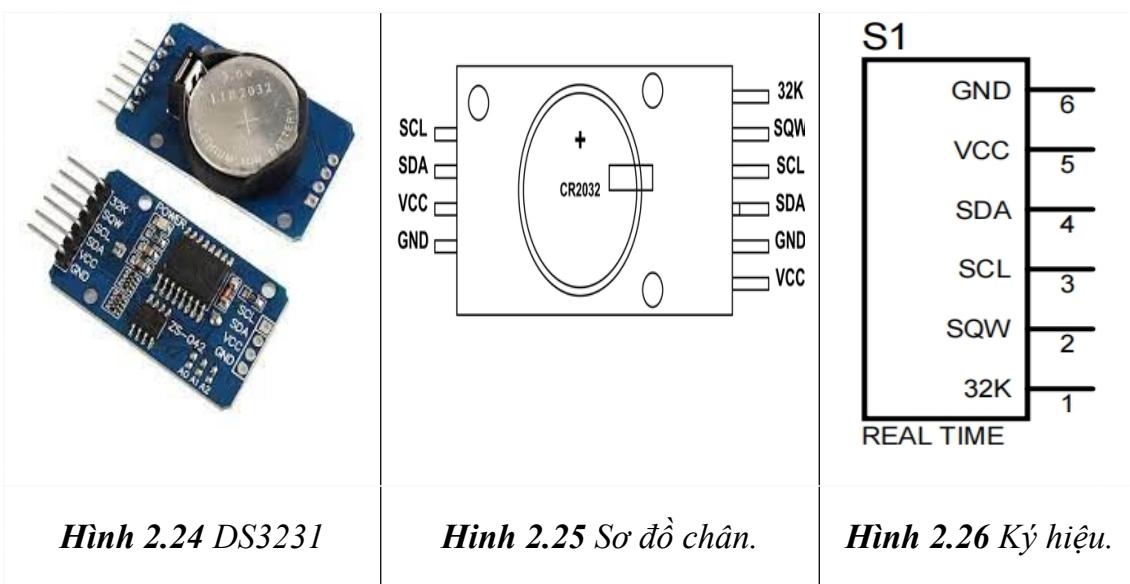
**Hình 2.23 Giao tiếp I2C cho DS3231.**

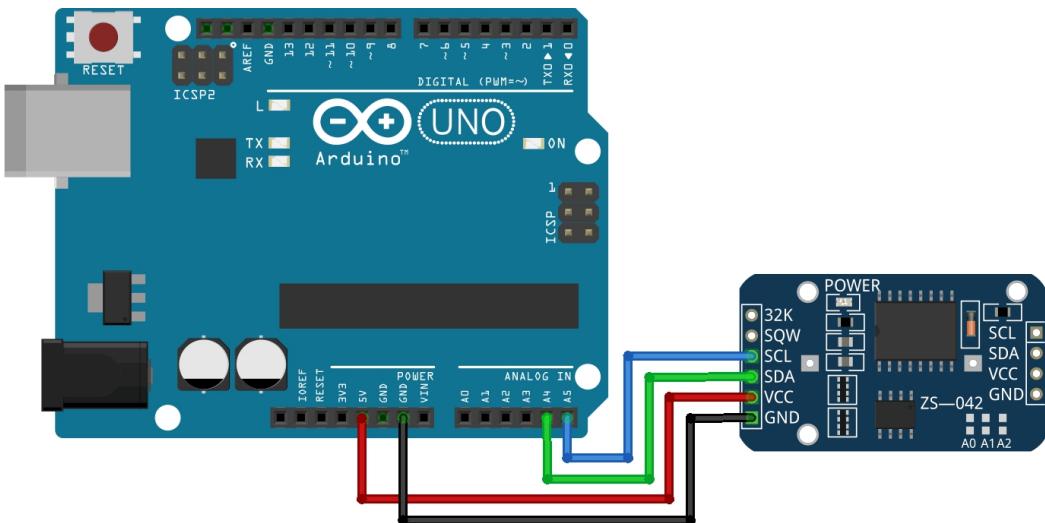
Cũng như nhiều cảm biến khác DS3231 cũng như các loại RTC thì đều có chung một thư viện để hỗ trợ cho việc lập trình thời gian thực. Thư viện đó chính là “RTC.h”, “RTCLib-master.h” hoạt thư viện dành chính cho module DS3231 là “DS3231.h”.

Ta có thể download các thư viện trên thông qua đường link dưới đây “<https://www.arduinolibraries.info/libraries/ds3231>”

### 2.6.3. Module thời gian thực DS3231. [9]

1) Hình ảnh.





**Hình 2.27** Sơ đồ nối chân giữa DS3231 với Arduino.

## 2) Các đặc điểm cơ bản của module DS3231.

### a) Các đặc trưng của DS3231.

- Kích thước: 38 x 22 x 14 mm
- Trọng lượng: 8g
- Điện áp làm việc: 3.3 - 5.5V DC
- IC đồng hồ: độ chính xác cao sử dụng IC DS3231.
- Độ chính xác: 0 - 40°C phạm vi, độ chính xác 2ppm, lối khoảng 1 phút.
- Thời gian thực phát điện đồng hồ giây, phút, giờ, ngày, tháng, năm và thời gian cung cấp giá trị cho đến năm 2100 (năm nhuận).
- IC bộ nhớ: AT24C32 (Dung lượng lưu trữ 32K)
- Kiểu truyền bus I2C, tốc độ truyền tối đa 400kHz (Điện áp làm việc 5V)
- Có thể kết nối với các thiết bị khác I2C, 24C32 địa chỉ có thể thiếu A0 / A1 / A2 thay đổi địa chỉ mặc định là 0x57

b) Chân I/O của DS3231.

STT	Pin	Symbol	Function
1.	1	32K	Một cái nguồn cung cấp dự phòng
2.	2	VCC	Cấp nguồn 3.3-5V
3.	3	INT/SQW	Dùng để cài đặt báo thời gian
4.	4	RST	Reset
5.	5-12	NC	Không kết nối, nên nối đất
6.	13	GND	Nối đất, 0V
7.	14	VBat	Ngõ vào nguồn gắn thêm tụ để có thể sử dụng
8.	15	SDA	Giao tiếp I2C
9.	16	SCL	Giao tiếp I2C

*Bảng 2.5 Bảng sơ đồ chân của DS3231*

## CHƯƠNG 3

### THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

#### 3.1. YÊU CẦU VÀ SƠ ĐỒ KHỐI CỦA HỆ THỐNG.

##### 3.1.1. Yêu cầu của hệ thống.

Hệ thống có những chức năng như sau:

- Giám sát mực nước trong bể cá là có đủ tiêu chuẩn không.
- Giám sát độ đục của nước trong bể cá.
- Đưa ra cảnh báo nếu có dấu hiệu vượt ngưỡng chỉ số của các cảm biến.
- Nhận biết và hiển thị thời gian thực.
- Tự động cho cá ăn theo thời gian quy định hoặc trực tiếp cho cá ăn thông qua điều khiển.
- Thông tin về thời gian mức độ đục hay mực nước trong bể cá được hiển thị trên màn hình led LCD 20x4.
- Các đèn led được lắp đặt xung quanh bể cá trang trí và đồng thời cũng là cảnh báo khi có vấn đề xảy ra.

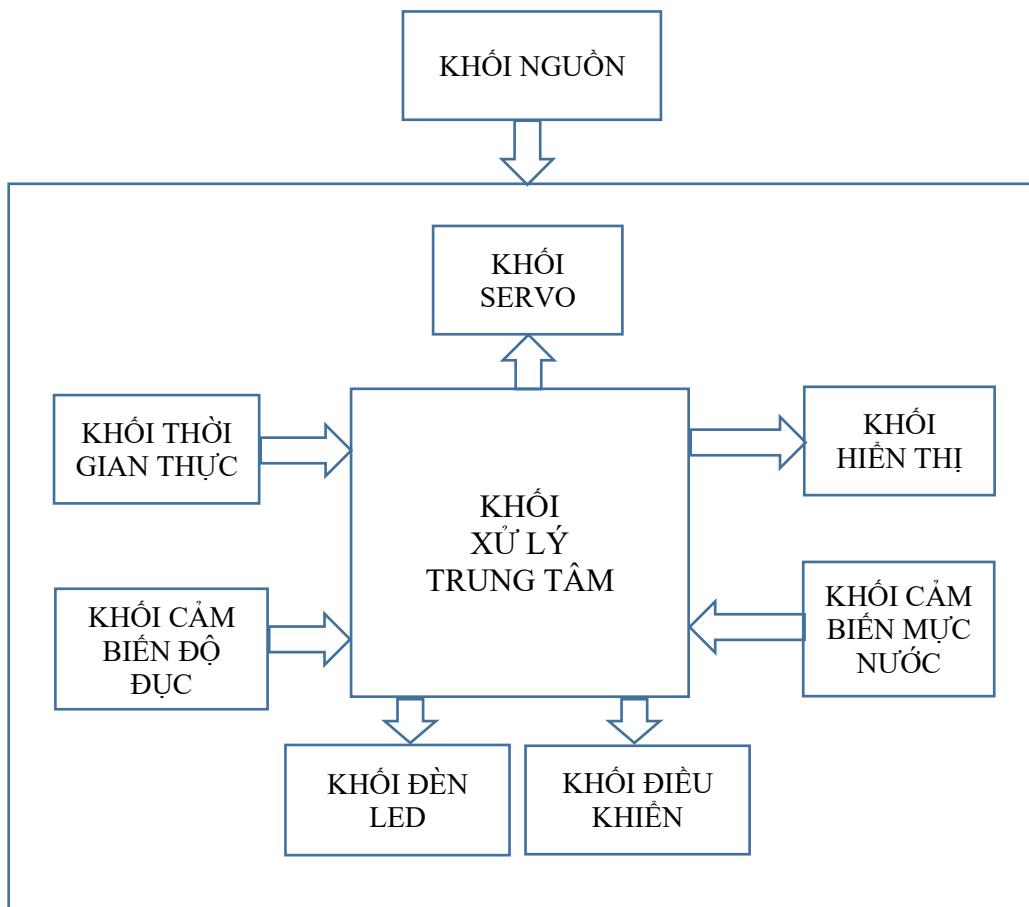
##### 3.1.2. Sơ đồ khối và chức năng của từng khối trong hệ thống.

###### 1) Chức năng của từng khối.

- **Khối xử lý trung tâm:** Nhận dữ liệu, thông tin của bể cá từ các cảm biến. Qua đó đưa xử lý và đưa ra tín hiệu điều khiển hợp lý để các khối thực thi như Servo hay khối hiển thị như màn hình led LCD.
- **Khối thời gian thực:** Cập nhật thời gian hiện tại, rồi đưa về bộ xử lý trung tâm để xử lý và đưa ra khối hiển thị
- **Khối cảm biến độ đục:** Đọc dữ liệu từ cảm biến độ đục trong bể cá và gửi về bộ xử lý trung tâm để xử lý.
- **Khối cảm biến mực nước:** Đọc dữ liệu từ cảm biến mực nước trong bể cá và gửi về bộ xử lý trung tâm để xử lý.
- **Khối Servo:** Bộ xử lý trong tâm đưa ra tín hiệu để điều khiển động cơ Servo để cho cá ăn.

- **Khối hiển thị:** Nhận tín hiệu điều khiển từ bộ xử lý trung tâm, hiển thị mực nước, độ đục, thời gian,..
- **Khối điều khiển:** Khối các nút bấm để điều khiển on/off đèn hoặc điều khiển hoạt động của khói Servo.
- **Khối LED:** Gồm các led được trang trí xung quanh bể cá, nhận tín hiệu điều khiển từ bộ xử lý trung tâm.
- **Khối Nguồn:** Cung cấp nguồn cho toàn hệ thống hoạt động.

## 2) Sơ đồ khối của hệ thống.



**Hình 3.1** Sơ đồ khối của hệ thống.

### 3.1.3. Hoạt động của hệ thống.

Ban đầu khi cấp nguồn cho hệ thống hoạt động, thì các khói cảm biến mực chất lỏng, cảm biến độ đục, cảm biến thời gian thực sẽ thu thập dữ liệu từ bể cá, sau đó đưa về cho khói xử lý trung tâm để xử lý. Khói xử lý trung tâm sẽ so sánh các thông số được lấy về từ cảm biến với các thông số người lập trình cài

đặt sẵn sau đó đưa ra tín hiệu điều khiển phù hợp đến các ngoại vi như servo, và hiển thị các thông số cần thiết trên màn hình led LCD.

Hệ thống có khói đèn vừa dùng để trang trí vừa dùng để cảnh báo đèn người dùng nếu có thông số nào từ cảm biến vượt qua ngưỡng mà người dùng đã quy định từ đó đưa ra các người dùng có thể đưa ra cách biện pháp giải quyết kịp thời.

Bên cạnh việc điều khiển tự động thông qua khối xử lý trung tâm thì người dùng có tự điều khiển qua khói điều khiển với các nút nhấn được cài đặt sẵn sẽ nhận tín hiệu nếu có và đưa về bộ xử lý trung tâm để xử lý và điều khiển các ngoại vi như đèn led trang trí hay cho cá ăn.

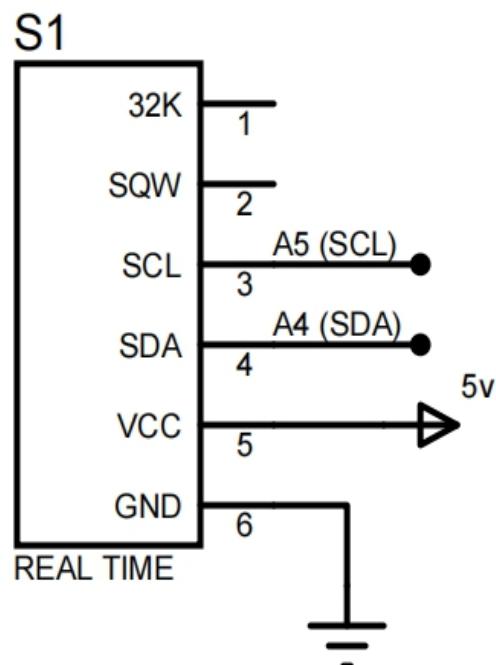
### 3.2. THIẾT KẾ HỆ THỐNG PHẦN CỨNG.

#### 3.2.1. Khối thời gian thực.

DS3231 là module thời gian thực hay còn được gọi dưới cái tên RTC (Real time clock). Module DS3231 bên trong có tồn tại 7 thanh ghi dữ liệu mỗi thanh ghi có 8-bit đại diện cho “thứ, ngày tháng, năm, giờ, phút, giây”. Ngoài ra DS3231 còn có một thanh ghi dùng điều khiển ngoài ra phụ và 56 thanh ghi trống được dùng như là RAM cho Module DS3231.

DS3231 giao tiếp được với vi điều khiển thông qua chuẩn truyền thông I2C do đó để đọc và ghi từ DS3231 phải thông qua chuẩn truyền thông này. DS3231.

Bên cạnh DS3231 vẫn còn khá nhiều module thời gian thực khác có thể kể như DS1307, DS1302, DS3231,... nhưng em chọn DS3231 vì đây là module được sử dụng khá rộng rãi trên thị trường, có nhiều công dụng và chức năng hơn các module thế hệ trước, giá cả lại phù hợp với túi tiền. Với DS3231 việc đọc thời gian thực là chính xác và ổn định.



Hình 3.2 Kết nối Arduino với DS3231.

Dòng điện định mức dùng cho DS3231 là 1.5mA.

Tần số dao động thạch anh dùng trong DS3231 là 32kHz.

Kết nối với Arduino:

- Chân SCL của DS3231 nối với chân A5 của Arduino Uno.
- Chân SDA của DS3231 nối với chân A4 của Arduino Uno.
- Chân GND và VDD của DS3231 nối với nguồn 5V và GND của Arduino Uno.

*Thu vien dung để điều khiển module thời gian thực DS3231:*

- *Thu vien DS3231.h.*
- *Thu vien RTClib.h.*
- *Thu vien RTC.h.*

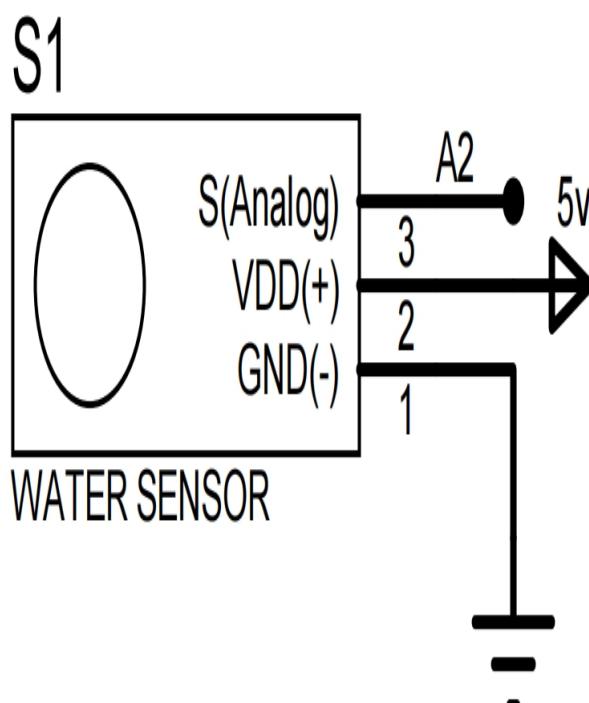
### 3.2.2. Khối cảm biến mực chất lỏng.

Cảm biến sẽ dựa vào sự thay đổi của điện dung khi mực nước trong bồn chứa nước thay đổi. Từ đó tín hiệu sẽ được chuyển đổi và truyền về vi điều khiển để xử lý.

Dòng điện: 20mA.

Kết nối với Arduino:

- Chân S của cảm biến mực chất lỏng sẽ nối với chân A2 của Arduino Uno.
- Chân ‘-’ của cảm biến mực chất lỏng sẽ nối với GND của Arduino Uno.
- Chân ‘+’ của cảm biến mực chất lỏng sẽ nối với 5V
- của Arduino Uno.



**Hình 3.3** Kết nối Water Sensor với Arduino.

Bên cạnh việc sử dụng Water Sensor để đo mực nước, ta còn có thể sử dụng cảm biến siêu âm, một cảm biến khá tiện dụng cho việc đo khoảng cách và được sử dụng khá phổ biến trên thị trường. Ta có thể dùng cảm biến siêu âm để đo khoảng cách từ nó đến mặt nước và từ đó tính ra mực nước trong bể cá. Nhưng do mức độ của đồ án, việc dùng cảm biến siêu âm là không cần thiết vì nó sẽ làm cho hệ thống khá cồng kềnh, phức tạp hóa hệ thống.

### 3.2.3. Khối cảm biến độ đục.

Cảm biến đo độ đục sẽ sử dụng ánh sáng để phát hiện các hạt lỏng lửng trong nước bằng cách đo truyền ánh sáng và tỷ lệ tán xạ ánh sáng đó, mà thay đổi với có lượng tổng chất rắn lỏng (TSS) trong nước. Khi TSS tăng, mức độ đục của chất lỏng tăng khi đó ánh sáng nhận được thông qua cảm biến sẽ giảm đi.

Thay vì sử dụng cảm biến siêu âm HC-SR04 để đo mức nước thường thấy. Nhưng do tính chất là 1 đồ án không lớn bể cá chỉ mang tính chất mô phỏng nên không cần thiết phải dùng cảm biến siêu âm này

Dòng điện: 40mA.

Kết nối với Arduino:

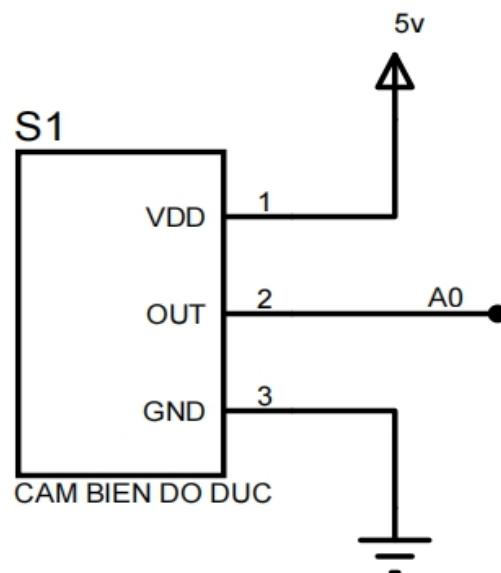
- Chân Out của cảm biến độ đục sẽ nối với A0 của Arduino Uno.
- Chân VCC của cảm biến độ đục sẽ nối với 5V của Arduino Uno.
- Chân GND của cảm biến độ đục sẽ nối với GND của Arduino Uno.

### 3.2.4. Khối Servo (động cơ MG90S).

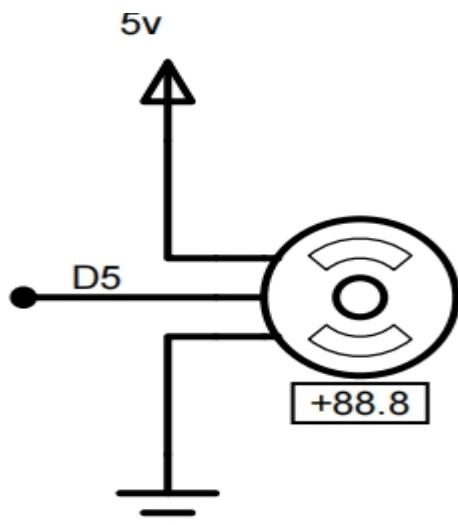
Đối với Servo dùng trong điện tự như MG90S, SG90, Motor Servo thì hoạt động như thế này: Rotor của động cơ điện (phản quay quanh trục) được làm bằng nam châm vĩnh cửu có từ trường mạnh. Phản vỏ của động cơ có các lá thép hoạt nhựa được ghép lại với nhau (nhằm khắc phục dòng điện Foucault) được quấn thành các cuộn dây riêng biệt. Và được cấp nguồn dựa trên hệ thống điều khiển, trong một chu kỳ có thể quay theo lập trình hoặc tuyến tính.

Dòng điện: 1.5mA.

Kết nối với Arduino:



**Hình 3.4** Kết nối Độ Đục với Arduino.



**Hình 3.5** Kết nối Servo với Arduino.

- Chân VCC (màu cam) của Servo MG90S được nối với 5v của Arduino Uno.
- Chân điều khiển (màu đỏ) của Servo MG90S được nối với chân D5 (PWM) của Arduino Uno.
- Chân GND (màu nâu) của Servo MG90S được nối với chân GND của Arduino Uno.

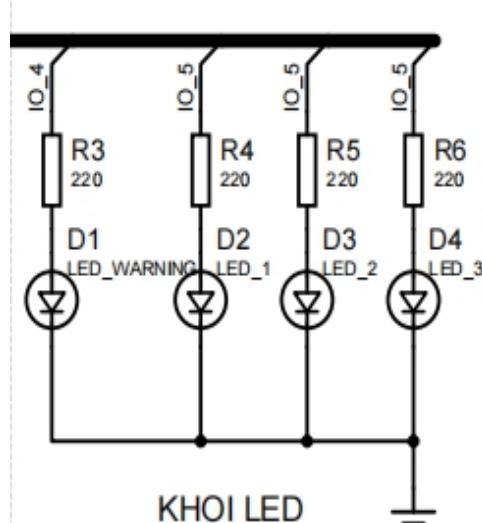
\*Thu viện thường dùng cho Servo MG90S là Servo.h.

### 3.2.5. Khối LED.

Bao gồm các led được mắc song song dùng để trang trí và cảnh báo đèn người dùng nếu có vấn đề về các thông số vượt ngưỡng mà cảm biến đo được. Từ đó người dùng có thể dễ dàng phát hiện và đưa ra biện pháp giải quyết phù hợp.

Kết nối với Arduino:

- Các chân Anode của LED(trang trí) được nối chung và nối với chân D10 của Arduino Uno.
- Các chân Anode của LED(cảnh báo) được nối chung và nối với chân D12 của Arduino Uno.
- Các chân Cathode của LED được nối chung và nối với GND của Arduino Uno.



Hình 3.6 Kết nối LED với Arduino

Với nguồn cấp cho các led là 5V, dòng điện định mức cho các led 5mm được dùng là khoảng 20mA - 30mA

$$\Rightarrow R = \frac{V}{I} = \frac{5}{20 \cdot 10^{-3}} = 250 \text{ Ohm} \text{ hoặc } R = \frac{V}{I} = \frac{5}{30 \cdot 10^{-3}} = 166.67 \text{ Ohm}$$

=>Kết luận điện trở nên dùng trong khoảng từ 200 Ohm đến 300Ohm  
Nhưng do ta thực hiện mắc song song các led trang trí nên dòng của các led sẽ bị chia ra. Ở đây  $I_1 = I_2 = I_3 = \frac{30 \cdot 10^{-3}}{3} = 10 \cdot 10^{-3} = 10\text{mA}$ , nên led sẽ sáng yếu.

Việc led sáng yếu sẽ làm việc quan sát trở nên khó khăn nhưng bên cạnh đó lại giúp làm giảm hiệu năng của hệ thống.

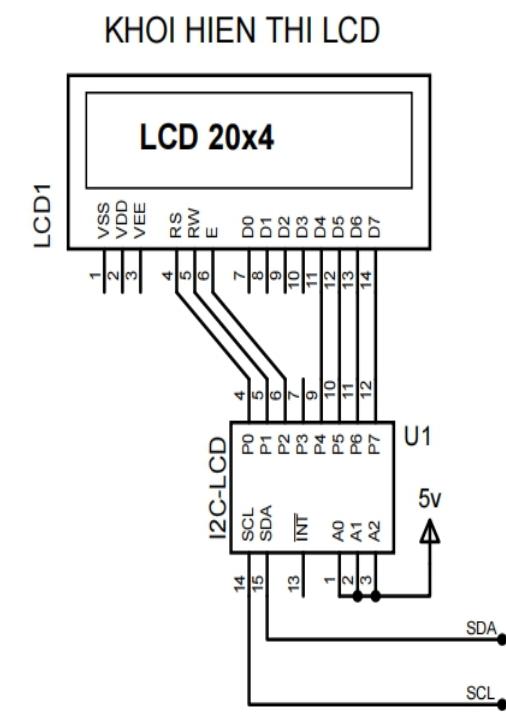
Trên thị trường ta có thể chọn các điện trở thông dụng như 220 (Ohm) hoặc 470 (Ohm)

### 3.2.6. Khối hiển thị.

Nhằm mục đích hiển thị các thông tin như mực nước, độ đục, thời gian thực cảnh báo,... thì màn hình LCD là lựa chọn khả dĩ nhất và được nhiều người thích dùng. Ở đây để phù hợp với thiết kế của bản thân nên em đã chọn màn hình LCD 20x4 tuy nó hơi có chút to nhưng có thể hiển thị một lần được nhiều thông tin nhất.

Cách kết nối giữa màn hình LCD với vi điều khiển thì có không chỉ một, nhưng để tránh lãng phí chân bằng cách kết nối trực tiếp. Vậy nên để tiết kiệm chân trách lãng phí thì nhiều người thiết kế vẫn tin dùng cách kết nối bằng giao thức giao tiếp I2C , điều này vừa giúp tiết kiệm I/O cho những ngoại vi khác vừa giúp hệ thống được gọn gàng hơn.

Dòng điện: 0.35-0.6mA.



**Hình 3.7** Kết nối LCD với Arduino.

Kết nối với Arduino:

- Chân SCL của I2C LCD nối với SCL của Arduino Uno
- Chân SDA của I2C LCD nối với SDA của Arduino Uno.
- Chân VCC của I2C LCD nối với 5v của Arduino Uno.
- Chân GND của I2C LCD nối với GND của Arduino Uno.

*Thu viện để điều khiển hoạt động của LCD là:*

- *Thu viện LiquidCrystal.*
- *Thu viện LiquidCrystal\_I2C.*

### 3.2.7. Khối xử lý trung tâm.

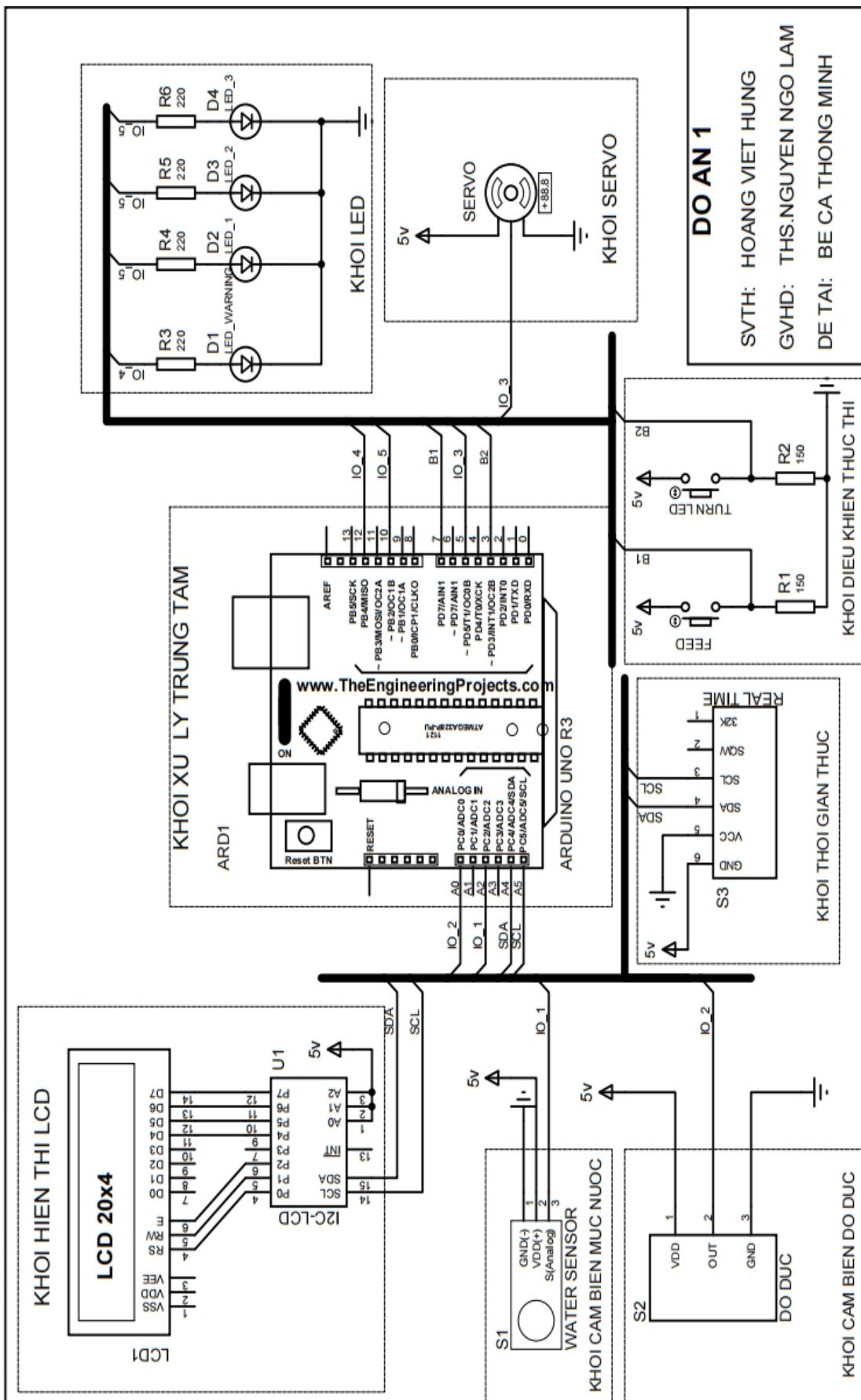
Hiện nay trên thị trường có rất nhiều board vi điều khiển khác nhau với các đặc điểm khác nhau như PIC, Arduino, Raspberry Pi, ESP,... Nhưng nếu nói về lập trình điều khiển thì người ta vẫn luôn nhớ đến là Arduino Uno, Arduino luôn là món ăn tinh thần phổ biến đối với nhiều người đặc biệt là những ai mới đặt chân tìm tòi về lĩnh vực điện-điện tử.

Vậy nên để dễ dàng nghiên cứu và làm việc thì em chọn Broad Arduino Uno R3. Vì nó khá phù hợp với chủ đề và quy mô nghiên cứu lần này. Với Arduino Uno R3, hỗ trợ được 2 cổng ngoại vi giao tiếp bằng chuẩn truyền thống I2C phù hợp để sử dụng vừa I2C LCD và DS3231. Bên cạnh đó Arduino Uno hiện đang được dùng phổ biến nên nó có nhiều thư viện được viết sẵn, hỗ trợ việc lập trình trở nên dễ dàng và nhanh chóng hơn.

Từ đó, ta có khái niệm trung tâm được kết nối như sau:

- Chân SCL của Arduino Uno nối với chân SCL của I2C LCD.
- Chân SDA của Arduino Uno nối với chân SDA của I2C LCD.
- Chân A5 của Arduino Uno nối với chân SCL của DS3231.
- Chân A4 của Arduino Uno nối với chân SDA của DS3231.
- Chân A0 của Arduino Uno nối với chân Out của Cảm biến độ đục.
- Chân A2 của Arduino Uno nối với chân Out của Cảm biến mực chất lỏng.
- Chân D5 của Arduino Uno nối với chân Out của Servo MG90S.
- Chân D12 của Arduino Uno nối với chân Anode các LED cảnh báo.
- Chân D10 của Arduino Uno nối với chân Anode các LED trạng trí.
- Chân D7, D3 của Arduino Uno nối với lần lược 2 nút ấn.

Từ các mô tả ở trên, ta có được một sơ đồ nguyên lý sau:



*Hình 3.8 Sơ đồ nguyên lý của hệ thống.*

### 3.2.8. Khối nguồn.

STT	Thiết bị	Dòng(mA)	Áp(V)
1.	Arduino Uno R3	30	7.0-12.0
2.	Cảm biến độ đục	40	5
3.	Cảm biến đo mực nước	20	3-5V
4.	LCD I2C 20x4	0.35-0.6	5
5.	Realtime clock DS3231	1.5	5
6.	Động cơ Servo MG90S	1.5	4.8 -6.0
7.	Tổng	93.6	3.0-12.0

Bảng 3.1 Bảng cấp nguồn cho các linh kiện

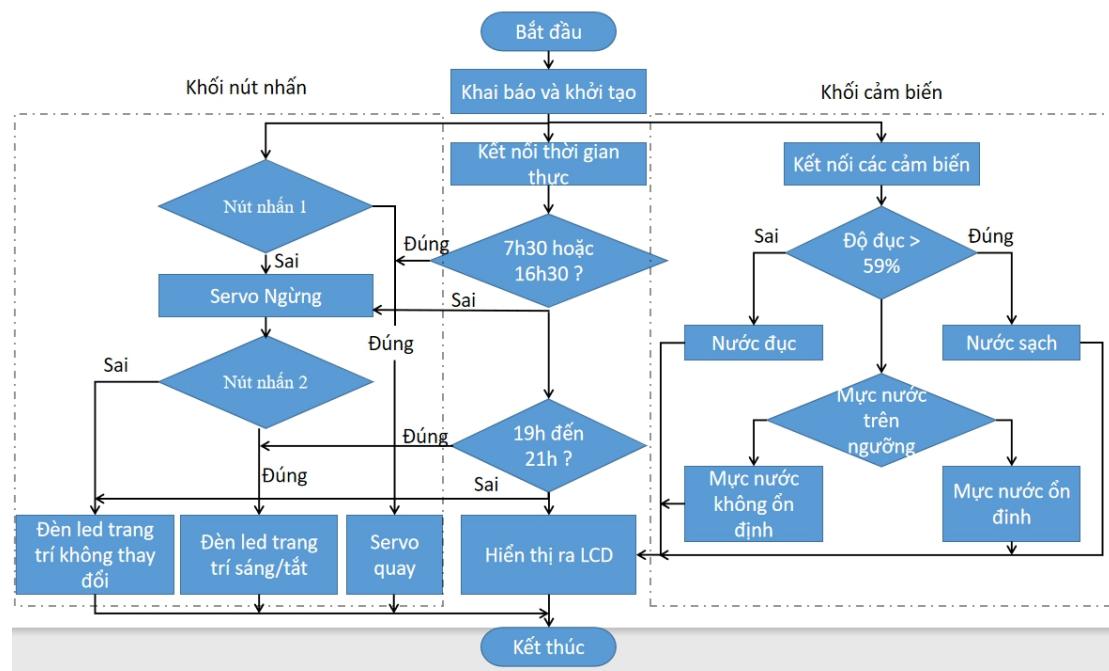
Vậy mạch tiêu thụ tối thiểu gần 0.1Ampe nguồn dao động từ 3.0 đến 12.0V.

Vậy nên em quyết định dùng Adaptor 9-12V để cấp nguồn cho Arduino Uno R3 được hoạt động. Adaptor 5V để chạy các cảm biến khác hoặc lấy nguồn từ Arduino cấp ra.

## 3.3. THIẾT KẾ HỆ THỐNG PHẦN MỀM.

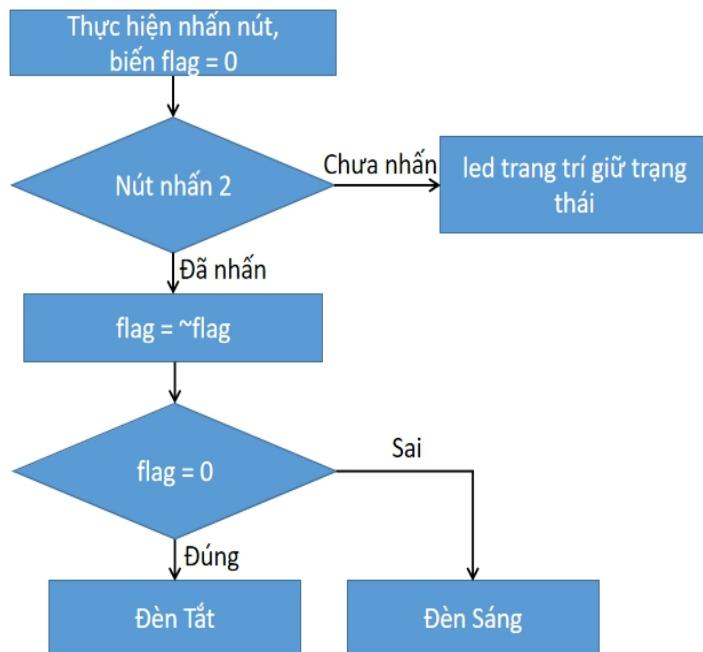
### 3.3.1. Lưu đồ giải thuật.

Hệ thống được lập trình dựa trên sơ đồ giải thuật sau:



Hình 3.9 Giải thuật của toàn hệ thống.

Giải thuật cho nút nhấn thứ hai (nút nhấn dùng để trang trí đèn):



**Hình 3.10** Giải thuật cho nút nhấn thứ 2 trong hệ thống

### 3.3.2. Phân tích thuật toán.

- Đối với lưu đồ giải thuật chính:

Đầu tiên ta tiến hành khai báo các thư viện cũng như các biến toàn cục cần thiết cho chương trình. Sau đó, tiến hành khởi động kết nối với module thời gian thực DS3231, và cho kết quả là hiển thị thời gian trên màn hình LCD. Sau đó, ta tiến hành xem xét các khoảng thời gian. Nếu khoảng thời gian là 7h30 sáng hay 16h30 chiều thì servo của mạch sẽ hoạt động, servo sẽ quay một góc  $150^\circ$  để cho cá ăn. Trong thời gian servo quay mạch sẽ bị delay, các chức năng khác sẽ không hoạt động (thời gian delay khoảng 3s). Còn nếu không phải thời gian đó, Servo sẽ không quay. Tiếp tục xét khoảng thời gian có nằm trong từ 19h đến 21h nếu có thì đèn led trang trí sẽ tự động sáng và ngược lại nếu không. Bên cạnh hoạt động của module thời gian thực thì các cảm biến như cảm biến mực nước hay độ đục cũng được khởi động và sẽ in kết quả đo được về màn hình LCD. Đối với khối nút nhấn thì nút nhấn thứ nhất là nút nhấn điều khiển servo, khi nhấn vào servo sẽ quay một góc  $150^\circ$ , delay 3s rồi quay về vị trí ban đầu.

- Đối với lưu đồ giải thuật cho nút nhấn thứ hai:

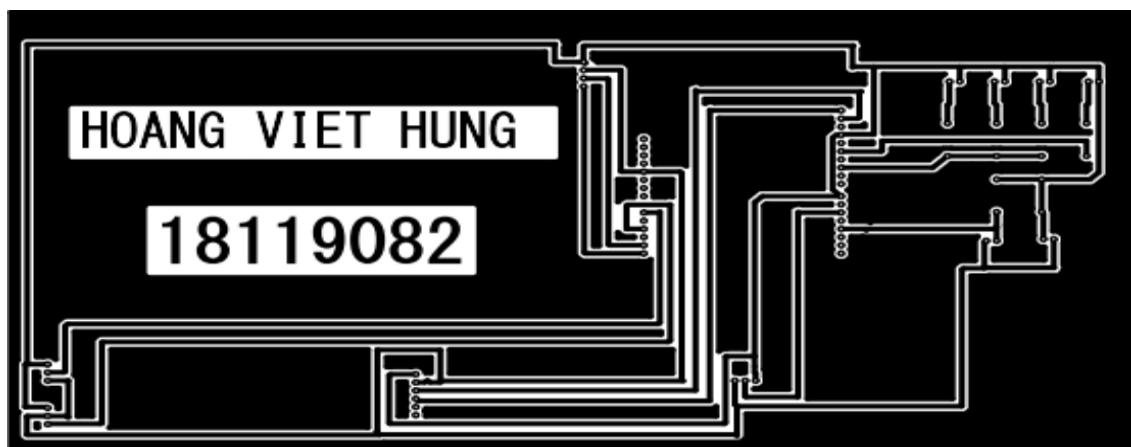
Nút nhấn thứ hai dùng để điều khiển các led trang trí. Khi nhấn nút, ta sẽ tạo và cho một biến cờ có giá trị đảo với giá trị của nó trước khi nhấn nút. Nếu biến cờ bằng 0 thì led trang trí sẽ tắt và ngược lại nếu biến cờ bằng 1. Khi không nhấn nút thì led giữ trạng thái trước đó.

## CHƯƠNG 4

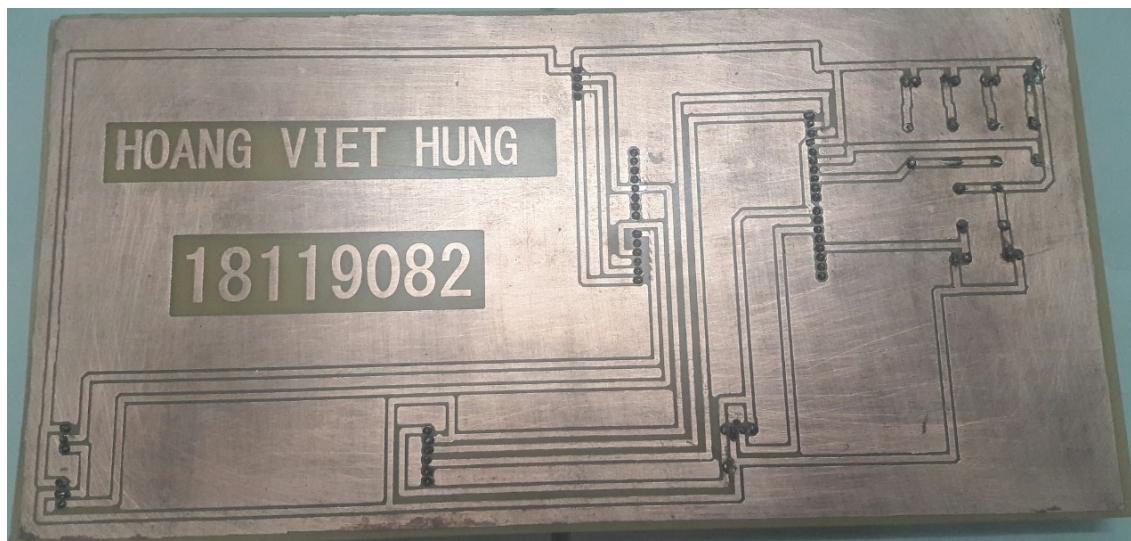
### KẾT QUẢ, NHẬN XÉT, ĐÁNH GIÁ

#### 4.1. KẾT QUẢ

Phần cứng của hệ thống sẽ được làm trên mạch in:



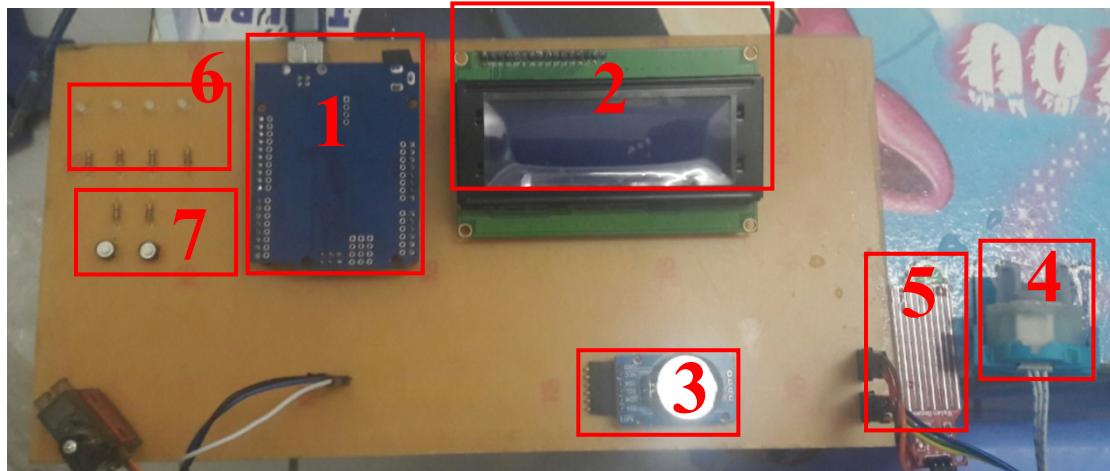
*Hình 4.1 Mạch in của hệ thống.*



*Hình 4.2 Mạch in sau khi làm ra*

Mạch in được thiết kế khá to, nhưng các đường dây thể hiện rõ ràng, to, dễ thực hiện hàn chân linh kiện. Mạch in lớn có thể bao quát được tất cả các linh kiện đều nằm trên nó. Các chân hàng được nối với các hàng rào chân, từ đó có thể gán được linh kiện vào cũng như gỡ linh kiện ra khi có vấn đề trực tiếp với linh kiện nào đó.

Sau khi đã hoàn thành mạch in, ta tiến hành lắp mạch và chạy thử để xem kết quả, sử dụng các linh kiện đã nói ở phần đầu ta tiến hành lắp ghép mạch lại như sau:

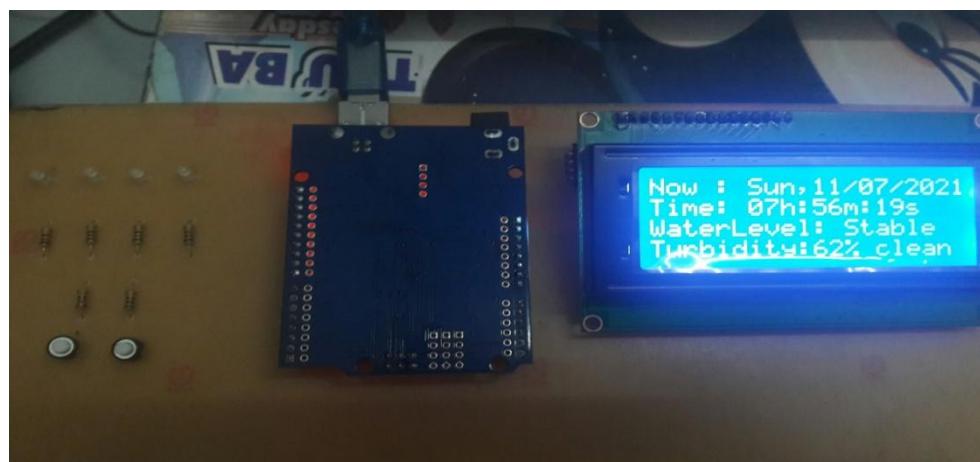


**Hình 4.3** Mạch hoàn chỉnh

Chú thích:

- 1\_ Khối xử lý trung tâm (Arduino Uno R3).
- 2\_ Khối hiển thị (LCD 20x4).
- 3\_ Khối thời gian thực (DS3231).
- 4\_ Khối cảm biến (Cảm biến độ đục, cảm biến mực nước).
- 5\_ Khối Servo (Servo MG30S).
- 6\_ Khối led cảnh báo và trang trí.
- 7\_ Khối điều khiển bằng nút nhấn.

Ban đầu ta thực hiện test mạch với mực nước đầy đủ và nước có độ trong nhất định thì mạch sẽ cho ra kết quả trên LCD như sau:



**Hình 4.4** LCD khi nước trong và mực nước ổn định.

Giải thích kết quả được hiện trên LCD:

Dòng đầu tiên: LCD hiện kí tự “Now : ” đồng thời lấy giá trị ngày tháng năm hiện tại đọc được từ module thời gian thực DS3231 để hiển thị ra.

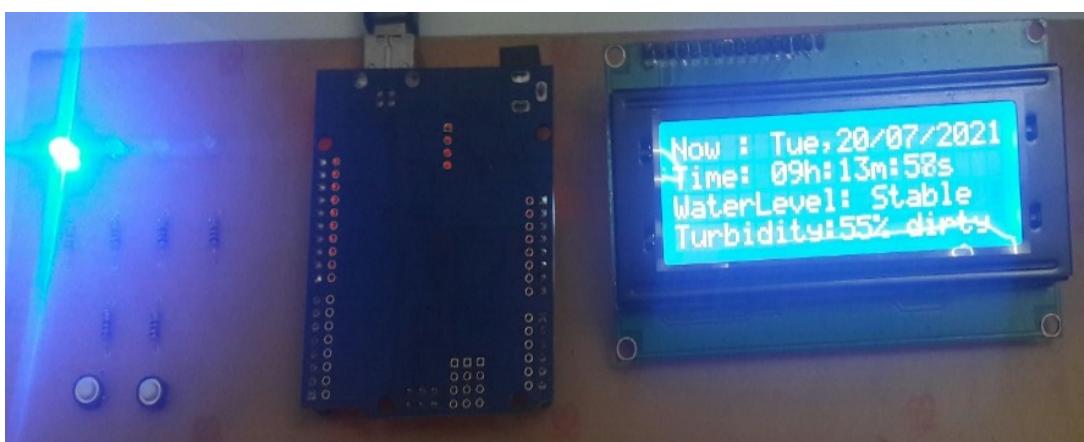
Dòng thứ hai: LCD hiện kí tự “Time : ” đồng thời lấy giá trị giờ phút giây hiện tại đọc được từ module thời gian thực DS3231 để hiển thị ra.

Dòng thứ ba: LCD hiện kí tự “Water level: ” đồng thời hiển thị kết quả đo được từ cảm biến mực nước, nếu mực nước trong bể đã đủ với yêu cầu đề ra thì LCD sẽ hiển thị kí tự “Stable”, và ngược lại, nếu mực nước trong bể không đủ với yêu cầu đề ra thì LCD sẽ hiển thị kí tự “Unstable”. Như trên hình thì mực nước đã ổn định.

Dòng thứ tư: LCD hiện kí tự “Turbidity:” đồng thời hiện kết quả đo được từ cảm biến độ đục của hệ thống. Kết quả đo được sẽ chuyển thành phần trăm độ đục và được hiển thị ra trên LCD đồng thời nếu kết quả lớp hơn 59% LCD sẽ hiện thêm từ “clean” nghĩa là độ đục của nước lúc này đã ổn, nước không quá dơ, còn ngược lại nếu phần trăm độ đục nhỏ hơn 59% thì LCD sẽ hiển thị kí tự “dirty” nghĩa là nước trong bể đã bẩn cần được thay nước. Như trên hình thì độ đục trong nước là 62%, nước sạch.

Và do cả cảm biến mực nước và cảm biến độ đục đều đưa ra kết quả tốt nên đèn cảnh báo sẽ không phát sáng.

Nhưng khi ta cho thử cảm biến độ đục vào nước đục thì kết quả trả về trên LCD sẽ là:

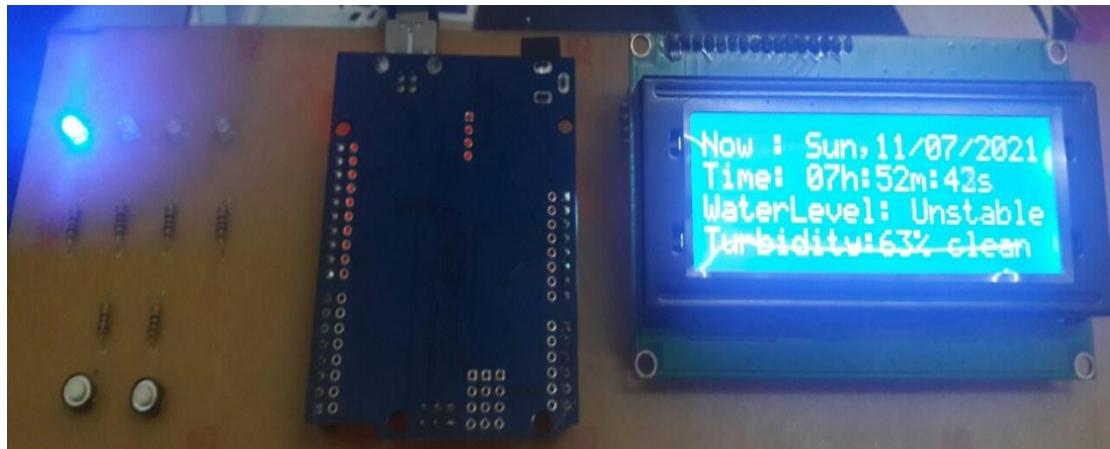


*Hình 4.5 LCD khi nước đục, mực nước ổn định.*

Khi này độ đục của nước đo được là 54% hoặc thấp hơn, và lúc này led cảnh báo sẽ được bật lên như trên hình. Tuy nhiên lúc này led cảnh báo sẽ sáng theo dạng nhấp nháy. Điều này là do hệ thống bị delay, lập trình chưa được tối ưu. Bên cạnh cảnh báo từ cảm biến độ đục thì cảm biến mực nước vẫn cho ra

kết quả là mức nước ổn định. Thời gian hiển thị trên LCD thì luôn được cập nhập sau mỗi giây

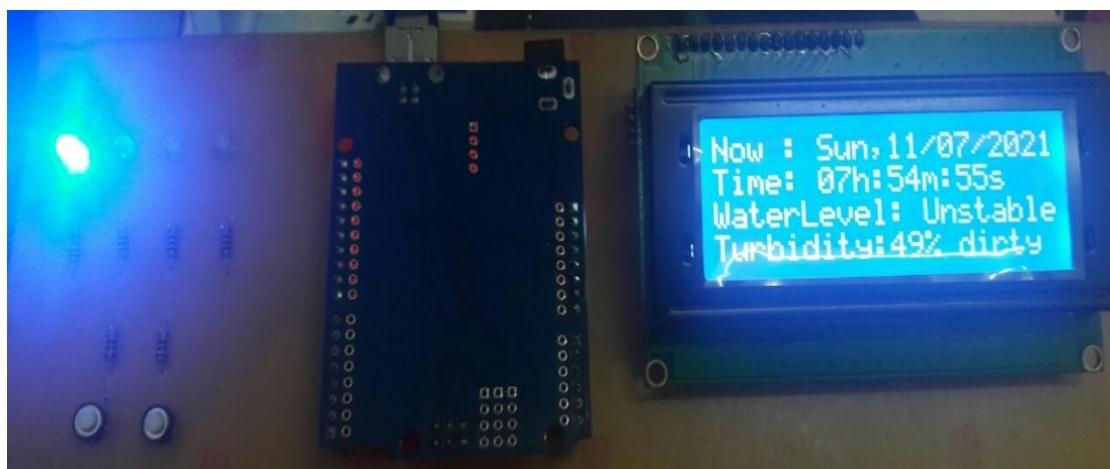
Tương tự khi độ đục của nước trở lại là nước trong nhưng cảm biến mức nước sẽ được lấy ra:



**Hình 4.6 LCD khi nước trong, mức nước thấp**

Khi này mức nước hiện ra trên LCD là không ổn định. Đèn cảnh báo sẽ sáng nhấp nháy. Bên cạnh cảnh báo đèn từ mức nước thì độ đục của nước trong tình huống này tốt, nước trong bể sạch. Thời gian vẫn cập nhật lên LCD đúng với thời gian thực tế.

Và bây giờ, khi test thử mức nước trong bể thấp đồng thời độ đục của nước là bẩn thì kết quả trên LCD sẽ là:



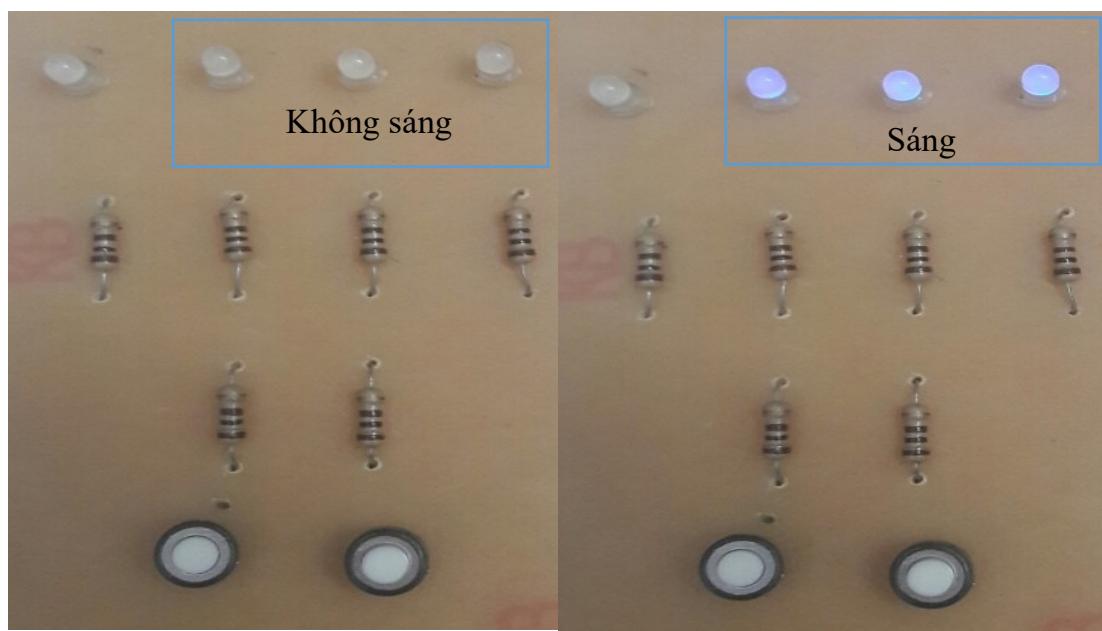
**Hình 4.7 LCD khi nước đục, mức nước thấp**

Kết quả lần này cho ra mức nước là không ổn định (Unstable), đồng thời độ đục của nước là bẩn 58%. Vậy nên đèn cảnh báo sẽ sáng nhấp nháy. Bên

cạnh đó thì module thời gian thực vẫn hoạt động bình thường với thời gian luôn được cập nhật trên LCD.

Qua các trường hợp vừa rồi, có thể rút ra được kết luận là, mạch hoạt động khá ổn định trong điều kiện nhiệt độ từ 25°C đến 35°C. Luôn đưa ra cảnh báo ngay khi có cập nhật mới từ cảm biến gửi về, cho thấy tốc độ đáp ứng cao. Nhưng đèn led còn sáng nhấp nháy do delay của hệ thống.

Đối với nút nhấn đèn trang trí (3 bóng led ngoài cùng bên phải) trong mạch:



**Hình 4.8** Led trang trí khi không sáng và sáng

Qua đây ta thấy được, nút nhấn hoạt động ổn khi chạy theo đúng chương trình đã viết ra nhưng đèn led trang trí lại không hoạt động hiệu quả vì led sáng quá yếu. Điều này như đã trình bày ở phần 3.2.5, do mắc các led song song, dòng sẽ bị chia ra nên led không thể sáng được tốt.

Các bước để sử dụng hệ thống:

- Bước 1: Cấp nguồn 5V vào hệ thống bằng adapter.
- Bước 2: Đợi khoảng sau 1s, thông số về thời gian, mục nước và độ đục sẽ được cập nhật từ cảm biến và sẽ hiển thị trên màn hình led LCD.
- Bước 3: Khi ta nhấn nút nhấn bên phải, servo sẽ quay khi đó ta có thể thực hiện việc tự cho cá ăn theo ý chí của bản thân. Hoặc ta có thể để cho cá ăn

tự động thông qua 2 mốc thời gian là 7h30 sáng hoặc 16h30 chiều servo sẽ tự động quay và cho cá ăn.

- Bước 4: Khi ta nhấn nút nhấn bên trái một lần, đèn trang trí sẽ sáng nhấp nháy. Nhưng khi ta nhấn nút bên trái thêm 1 lần nữa thì đèn trang trí sẽ tắt (đèn trang trí sẽ là 3 bóng đèn ngoài cùng bên tay phải). Hoặc có thể để đèn trang trí tự động sáng tắt trong mốc thời gian từ 19h đến 21h.
- Bước 5: Hệ thống sẽ cập nhật liên tục kết quả đo được từ cảm biến nên nếu có vấn đề xảy ra với mức nước hay độ đục trong nước thì đèn cảnh báo sẽ tự động bật lên.

Link quay video mô phỏng sẽ được gửi lên google drive:  
<https://drive.google.com/file/d/1wKfgjtH49DoqYgicDYK66s1qSCkIytnT/view?usp=sharing>

## 4.2. NHẬN XÉT VÀ ĐÁNH GIÁ.

### \*Những điều làm được:

- Hệ thống hoạt động với mức điện áp ổn định 5V
- Mạch chạy được đúng chức năng, đánh giá được độ đục của mức và hiển thị chính xác độ đục trong bể cá ra màn hình LCD, cũng như cho biết mức nước trong bể là ổn định hay không.
- Tự động hỗ trợ cho cá ăn theo giờ hoặc người dùng có thể điều khiển qua nút nhấn.
- Có hệ thống cảnh báo bằng đèn led.
- Công suất tiêu thụ của mạch là không quá lớn, phù hợp với các gia đình thích nuôi cá làm kiểng.

### \*Những điều cần cải thiện:

- Nút nhấn điều khiển đèn led trang trí đôi lúc còn bị dội.
- Đèn led trang trí có độ sáng còn yếu.
- Mực nước đo được nhưng chưa cấp thông số cụ thể là bao nhiêu.
- Mạch in có độ thẩm mĩ còn kém.
- Hệ thống chưa tối ưu, còn bị delay nhiều chỗ.
- Thiết kế mạch còn khá to, cần thiết kế nhỏ gọn hơn

## CHƯƠNG 5

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. KẾT LUẬN

Sau quá trình nghiên cứu, thiết kế và thi công đề tài “Bể cá thông minh mini” em đã hoàn thành cơ bản được các nội dung chủ yếu sau:

- Hoàn thiện được mạch in của hệ thống.
- Lập trình Arduino điều khiển được các chức năng của hệ thống.
- Khối cảm biến độ đục và mực nước hoạt động ổn định, luôn cho ra kết quả hiển thị trên LCD.
- Hệ thống cảnh báo hoạt động tốt trong thời gian dài, và sẽ được ra cảnh báo khi có tín hiệu không ổn định từ các cảm biến.
- Có chức năng tự động cho cá ăn là chủ yếu thì luôn hoạt động tốt.
- Hệ thống nút nhấn để điều khiển, giúp người dùng dễ dàng sử dụng hơn.

### 5.2. HƯỚNG PHÁT TRIỂN

- Cần phát triển thêm về hệ thống, tăng cường giám sát chi tiết hơn về tình hình nước trong bể cá như việc dùng các cảm biến như cảm biến nhiệt độ, cảm biến đo độ pH, độ mặn của nước để việc chăm sóc cá trở nên tốt hơn.
- Thiết kế mạch cần nhỏ gọn hơn.
- Cần một hệ thống lọc nước tự động trong bể cá để có thể khi không có người thì bể cá có thể tự động thay nước khi cần thiết.
- Và sẽ tốt hơn nếu có một hệ thống điều chỉnh nhiệt độ cho bể cá bằng sò nóng lạnh.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt (Sách, tạp chí,... Bảng in)

- [1] <https://www.slideshare.net/trongthuy2/luan-van-thiet-ke-he-thong-nuoi-ca-tu-dong-hot-9d>
- [2] Phạm Minh Tuấn 2010 “Arduino cho người mới bắt đầu” trang 105-109  
Link:<https://drive.google.com/file/d/1dPdaxe86k7mE1dXXtBEHaHW7BO4B9RKF/view>

### Tiếng Anh (Sách, tạp chí,... Bảng in)

- [3] <http://fusol.vn/kho-do-an-ve-arduino-iot.html>
- [4] Michael Margolis 2011 “Arduino Cookbook” page 217 -222  
Link:<https://drive.google.com/file/d/1L6ffMNtaXj2Ha9mFspD2BwPTouHb3QBz/view>

### Đường link web tham khảo

- [5] **Datasheet Arduino**  
LINK 1: <https://www.farnell.com/datasheets/1682209.pdf>  
LINK 2: [https://store.arduino.cc/usa/arduino-uno\\_rev3?queryID=undefined](https://store.arduino.cc/usa/arduino-uno_rev3?queryID=undefined)
- [6] **Datasheet LCD 20x4**  
LINK 1: <https://www.vishay.com/docs/37314/lcd020n004l.pdf>  
LINK 2: <https://www.futurlec.com/LED/LCD16X2BLa.shtml>
- [7] **Datasheet Servo**  
LINK 1: <https://www.alldatasheet.com/datasheet-pdf/pdf/1132104/ETC2/MG90S.html>  
LINK 2: <https://datasheetspdf.com/pdf-file/1106582/ETC/MG90S/1>
- [8] **Datasheet Cảm biến độ đục**  
LINK 1: [https://wiki.dfrobot.com/Turbidity\\_sensor\\_SKU\\_SEN0189](https://wiki.dfrobot.com/Turbidity_sensor_SKU_SEN0189)  
LINK 2: <http://www.farnell.com/datasheets/2700162.pdf>
- [9] **Datasheet DS3231**  
LINK 1: <https://www.datasheetq.com/DS3231-doc-Dallas>  
LINK 2:<https://datasheetspdf.com/pdffile/1081920/MaximIntegrated/DS3231/1>

# PHỤ LỤC

Code chương trình của hệ thống.

## 1) Khai báo và khởi tạo.

Bước đầu, ta tiến hành gọi thư viện và khai báo các biến cục bộ cần thiết:

Code	Chú thích
#include "RTClib.h" #include <LiquidCrystal_I2C.h> #include <Wire.h> #include <Servo.h>	//khai báo thư viện cho led LCD I2C và module thời gian thực DS3231
RTC_DS3231 rtc; LiquidCrystal_I2C lcd(0x27, 40, 2);	//tạo biến rtc có kiểu dữ liệu RTC_DS3231 //thiết lập biến lcd có kiểu dữ liệu là LiquidCrystal_I2C. thiết lập địa chỉ LCD là 0x27, thiết lập loại LCD sử dụng là LCD 40x2
Servo myServo;	//tạo một biến myServo có kiểu dữ liệu là Servo
bool temp2 = false; int BTN1 = 7;	//biến cờ cho nút nhấn 2 //tạo nút nhấn cho ăn tại chân D7 của Arduino
int mucNuoc = A2;	//tạo biến đọc dữ liệu từ cảm biến mực chất lỏng
int doDuc = A0; int WarningLed = 12; int BTN2 = 3;	//tạo biến đọc dữ liệu từ cảm biến độ đục //led cảnh báo cho chân D4
int LedDecorate = 10; char daysOfTheWeek[7][12] ={"Sun","Mon","Tue","Wed", "Thu","Fri","Sat"};	//tạo biến btn vào led để trang trí //thiết lập mảng nhằm tạo biến cho các thứ trong tuần
int Days; int Months; int Years; int Seconds;	//khai báo các biến chưa ngày giờ

int Minutes;	
int Hours;	
String DayOfWeek;	//biến chứa chuỗi ngày trong tuần
String NowDate;	//biến chứa chuỗi ngày tháng năm
String NowTime;	//biến chứa chuỗi giờ phút giây

## 2) Hàm thiết lập và đọc thời gian thực từ DS3231.

Tiếp theo ta sẽ tạo một hàm để thiết lập kết nối cho Arduino với DS3231:

Code	Chú thích
<pre>void SetRTC_Time() {     rtc.begin();     Serial.println("\n* * * * * RTC * * * * * *");      Serial.print    ("Is      RTC      Operating (Waiting!!!) : ");     if(!rtc.begin())     {         Serial.println("Could't fine RTC");         while(1);     }     else     {         Serial.println("Connected Success :)");         Serial.println("");     }     if(rtc.lostPower())     {         Serial.print("RTC lost power or pin is out of date, lets set the time!!!");         rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));     } }</pre>	<p>//hàm khởi tạo RTC cho DS3231 //khởi động RTC in DS3231</p> <p>//hàm check xem DS3231 đã kết nối với Arduino chưa</p> <p>//check xem pin từ DS3231 có bị mất hay hết pin, nếu bị thì cần cập nhật lại thời gian</p> <p>// for example to set January 27 2017 at 12:56 you would call: //</p>

```

void readTimeRTC_PrintLCD()
{
    DateTime now = rtc.now();
    Days = now.day();
    Months = now.month();
    Years = now.year();
    Seconds = now.second();
    Hours = now.hour();
    Minutes = now.minute();
    DayOfWeek =
        daysOfTheWeek[now.dayOfTheWeek()];

    NowDate = NowDate + DayOfWeek +
    "," + Days + "/" + Months + "/" + Years ;
    NowTime = NowTime + Hours + "h:" +
    Minutes + "m:" + Seconds + "s";

    Serial.print("Today is: ");
    Serial.println(NowDate); // send to serial monitor
    Serial.print("Time is : ");
    Serial.println(NowTime);
    Serial.println("");

    lcd.setCursor(0,0);
    lcd.print("Now : "); //Print on lcd

    lcd.print(daysOfTheWeek[now.dayOfThe
    Week()]);
    lcd.print(",");
    if(now.day()<=9)
    {
        lcd.print("0");
        lcd.print(now.day());
    }
    else {
        lcd.print(now.day());
    }
}

```

```
lcd.print('/');
if(now.month()<=9)
{
    lcd.print("0");
    lcd.print(now.month());
}
else {
    lcd.print(now.month());
    lcd.print('/');
    if(now.year()<=9)
    {
        lcd.print("0");
        lcd.print(now.year());
    }
    else {
        lcd.print(now.year());
    }
    lcd.setCursor(0,1);
    lcd.print("Time: ");
    if(now.hour()<=9)
    {
        lcd.print("0");
        lcd.print(now.hour());
        lcd.print("h");
    }
    else {
        lcd.print(now.hour());
        lcd.print("h");
    }
    lcd.print(':');
    if(now.minute()<=9)
    {
        lcd.print("0");
        lcd.print(now.minute());
        lcd.print("m");
    }
    else {
```

```

lcd.print(now.minute());
lcd.print("m");
}
lcd.print(":");
if(now.second()<=9)
{
    lcd.print("0");
    lcd.print(now.second());
    lcd.print("s");
}
else {
    lcd.print(now.second());
    lcd.print("s");
}
}

```

### 3) Hàm khởi tạo và quay Servo.

Để chạy được động cơ Servo không thể thiếu hàm khởi tạo:

Code	Chú thích
<pre> void Init_Servo() {     myServo.attach(5); } </pre>	//cấu hình chân D5 là chân OUT cho Servo
<pre> void CheckTimes() { int btnStatus = digitalRead(BTN1);  if(btnStatus == HIGH) {     myServo.write(0);     delay(3000);     myServo.write(150);     delay(1000); } </pre>	//Sau khi đã có hàm khởi tạo ta sẽ thực hiện quay Servo. Servo sẽ quay khi có sự kiện nút nhấn 1, hoặc trong khoảng thời gian 7h30 và 16h30: //đọc trạng thái Button cho cá ăn /*nên cho cá ăn 1 ngày 2 lần đó là

<pre> DateTime now = rtc.now(); Hours = now.hour(); Minutes = now.minute(); Seconds = now.second();  if( ((Hours == 7) &amp;&amp; (Minutes == 30) &amp;&amp; (Seconds == 00))    ((Hours == 16) &amp;&amp; (Minutes == 30) &amp;&amp; (Seconds == 00)) ) {     myServo.write(0);     delay(3000);     myServo.write(150);     delay(1000); } </pre>	<p>7h30 sáng và 16h30 chiều*/ //gọi thời gian</p> <p>//kiểm tra thời gian</p>
---	---

#### 4) Hàm led cảnh báo sáng/tắt.

<pre> void LedWarning_OFF() {     digitalWrite(WarningLed, LOW); }  void LedWarning_ON() {     digitalWrite(WarningLed,HIGH); } </pre>	<p>//Hàm thực hiện việc tắt đèn cảnh báo: //hàm led tắt</p> <p>//Hàm thực hiện việc bật đèn cảnh báo: //hàm led sáng</p>
--	--

#### 5) Hàm đọc cảm biến đo đục.

Hàm cảm biến độ đục sẽ được đọc và chuyển về % độ đục để hiển thị trên LCD, nếu phần trăm độ đục dưới 59%, nghĩa là nước đang dơ led cảnh báo sẽ được bật lên, còn ngược lại là nước sạch, led cảnh báo tắt:

Code	Chú thích
<pre> void Read_CamBienDoDuc() {     int Turbidity = analogRead(doDuc);      float voltages = Turbidity * (5.0/1023.0);     int turbidity = ((voltages * 100) / 5);     lcd.setCursor(20,1);     lcd.print("Turbidity: ");      Serial.print("\nTurbidity: ");     Serial.println(turbidity);      //send to lcd     if(turbidity &lt; 59)     {         lcd.setCursor(30,1);         lcd.print(turbidity);         lcd.print("% ");         lcd.print("dirty");         LedWarning_ON();     }     else     {         lcd.setCursor(30,1);         lcd.print(turbidity);         lcd.print("% ");         lcd.print("clean");         LedWarning_OFF();     } } </pre>	<p>//đọc cảm biến</p> <p>//chuyển thành độ đục thành điện áp</p> <p>//tính ra phần trăm độ đục</p> <p>//send to monitor serial</p>

## 6) Hàm đọc cảm biến mực nước.

Hàm cảm biến mực nước sẽ đọc giá trị ghi được từ cảm biến mực nước về, nếu giá trị đọc về dưới ngưỡng (<100) thì sẽ báo hiển thị trên LCD là lượng

nước trong bể cá đang không ổn, led cảnh báo sẽ được bật lên. Ngược lại thì lượng nước trong bể vẫn bình thường, led cảnh báo tắt:

Code	Chú thích
<pre>void Read_CamBienMucNuoc() {     int WaterLevel = analogRead(mucNuoc);     lcd.setCursor(20,0);     lcd.print("WaterLevel: ");      Serial.print("\nWater Level is: ");     Serial.println(WaterLevel);      if(WaterLevel &lt;= 100)     {         lcd.setCursor(32,0);         lcd.print("Unstable");         LedWarning_ON();         Serial.print("Water Level: Unstable");     }     else     {         lcd.setCursor(32,0);         lcd.print("Stable");         LedWarning_OFF();         Serial.print("Water Level: Stable");     } }</pre>	<p>//đọc dữ liệu</p> <p>// send to monitor serial</p> <p>//nếu mức nước dưới 100 thì sẽ hiển thị lên lcd</p> <p>//thông báo mức nước không ổn định ra LCD</p> <p>//thông báo mức nước ổn định ra LCD</p>

### 7) Hàm bật/tắt led trang trí.

Led trang trí sẽ được bật/tắt theo nút nhấn thứ hai và theo thời gian, nếu trong khoảng thời gian từ 19h đến 21h thì led sẽ được bật để soi sáng bể cá:

Code	Chú thích
<pre>void Decorate() {     DateTime now = rtc.now();</pre>	//đọc trạng thái chân button

<pre> int buttonStatus2 = digitalRead(BTN2); if((digitalRead(BTN2)==HIGH)){     temp2 = !temp2;      delay(200); } if(temp2)     digitalWrite(LedDecorate, HIGH); else     digitalWrite(LedDecorate, LOW);  if((now.hour() &gt; 19) &amp;&amp; (now.hour() &lt; 21)) {     digitalWrite(LedDecorate, HIGH); } else     digitalWrite(LedDecorate, LOW); } </pre>	<p>//kiểm tra nút bấm //đảo trạng thái biến temp khi nút nhấn được nhấn</p> <p>//nếu biến temp có mức logic 1 thì đèn sáng và ngược lại</p> <p>//hiển thị led nếu trong khoảng thời gian từ 19h đến 21h, đèn led trang trí sẽ sáng</p>
---	--

### 8) Chương trình chính của hệ thống.

Code	Chú thích
<pre> void setup() {     Serial.begin(9600);      pinMode(BTN1,INPUT);      pinMode(mucNuoc,INPUT);      pinMode(doDuc, INPUT);      pinMode(WarningLed, OUTPUT);      pinMode(LedDecorate,OUTPUT);     pinMode(BTN2,INPUT);      lcd.init(); } </pre>	<p>//khởi tạo màn hình monitor Serial để in kết quả muốn hiển thị</p> <p>//cấu hình chân cho BTN1</p> <p>//cấu hình chân cho mucNuoc</p> <p>//cấu hình chân cho doDuc</p> <p>//cấu hình chân cho led cảnh báo</p> <p>//cấu hình cho led và btn trang trí</p> <p>//hàm khởi tạo hoạt động LCD</p>

Wire.begin();	//khởi tạo hoạt động I2C
lcd.backlight();	//cho LCD đèn nền sáng lên
SetRTC_Time();	//cài đặt thời gian của bạn
Init_Servo();	//khởi động servo
}	
void loop() {	//hàm loop sẽ chạy lặp đi lặp lại các hàm con trong chương trình
readTimeRTC_PrintLCD();	
CheckTimes();	
Read_CamBienMucNuoc();	
Read_CamBienDoDuc();	
Decorate();	
}	