

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ**



VÕ THỊ BẢO TRÂN

ĐO VÀ ỔN ĐỊNH ĐỘ ẨM ĐẤT

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

NGÀNH VẬT LÝ KỸ THUẬT

Cần Thơ – 2018

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CẦN THƠ**

VÕ THỊ BẢO TRÂN

ĐO VÀ ỔN ĐỊNH ĐỘ ẨM ĐẤT

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

CÁN BỘ HƯỚNG DẪN

PGS. TS. NGUYỄN TRÍ TUẤN

NGÀNH VẬT LÝ KỸ THUẬT

Cần Thơ – 2018

LỜI CẢM ƠN

Để đề tài hoàn thành và đạt được kết quả như mong muốn, đó không chỉ là sự nỗ lực của bản thân mà còn có sự chỉ dạy tận tình của quý thầy cô cùng với sự giúp đỡ của bạn bè.

Em xin được gửi lời cảm ơn chân thành đến thầy PGS.TS. Nguyễn Trí Tuấn và thầy Vương Tấn Sĩ, đã giúp đỡ em tận tình trong những lúc khó khăn để hoàn thành đề tài nghiên cứu của mình, giúp em định hướng được mục tiêu của đề tài, biết nghiên cứu khoa học và cách trình bày luận văn. Qua đó, em đã đúc kết được nhiều kinh nghiệm và nó sẽ trở thành hành trang quan trọng cho tương lai của em.

Một lần nữa, em xin chân thành cảm ơn quý thầy cô và bạn bè đã luôn đồng hành cùng em trong suốt thời gian qua. Em xin gửi lời chúc sức khỏe và thành công đến tất cả mọi người.

Sinh viên thực hiện

Võ Thị Bảo Trân

DANH MỤC VIẾT TẮT

Viết tắt	Thuật ngữ
API	Application Programming Interface
EEPROM	Electrically Erasable Programmable Read-Only Memory
I/O	Input/ Output
I ² C	Inter-Integrated Circuit
IC	Integrated Circuit
IoT	Internet of Things
LCD	Liquid Crystal Display
LED	Light Emitting Diode
PIC	Programmable Intelligent Computer
RAM	Random Access Memory
SCL	Serial Clock
SDA	Serial Data
TWI	Two Wire Serial Interface

DANH MỤC BẢNG

Bảng 2.1: Các chân của LCD 20x4.....	3
Bảng 2.2: Cách nối các chân giữa LCD và Arduino UNO	5
Bảng 2.3: Cách nối các chân giữa LCD I ² C Arduino UNO	6
Bảng 2.4: Thông số kỹ thuật của Arduino UNO.....	12
Bảng 2.5: Nhóm các phần cơ bản của Arduino UNO.....	14
Bảng 2.6: Nhóm chân Power của Arduino UNO.....	15
Bảng 2.7: Nhóm chân I/O của Arduino UNO.....	16
Bảng 2.8: Thông số của ATmega328.....	17
Bảng 2.9: Các chân ATmega328	18
Bảng 2.10: Thông số kỹ thuật của cảm biến độ ẩm đất	24
Bảng 2.11: Cách nối chân giữa cảm biến và Arduino	24
Bảng 4.1: Giai đoạn vườn ươm.....	39

DANH MỤC HÌNH VÀ SƠ ĐỒ

Hình 1.1: Quy trình tổng quát	2
Hình 1.2: Quy trình mạch đo độ ẩm đất.....	2
Hình 2.1: Màn hình LCD 20x4	3
Hình 2.2: Cách nối dây giữa LCD 20x4 với Arduino.....	5
Hình 2.3: Cách nối dây giữa LCD I ² C và Arduino	6
Hình 2.4: Xung giữ nhịp trên SCL.....	8
Hình 2.5: Phương thức hoạt động của I ² C	9
Hình 2.6: Kết nối các thiết bị ngoại vi với SDA, SCL.....	9
Hình 2.7: Quá trình truyền nhận dữ liệu từ Master đến Slave	10
Hình 2.8: Board Arduino UNO	11
Hình 2.9: Vi điều khiển ATmega328.....	17
Hình 2.10: Các chân của ATmega328	20
Hình 2.11: Sơ đồ bên trong ATmega328	21
Hình 2.12: Cảm biến độ ẩm đất	22
Hình 2.13: Sơ đồ nguyên lý của bộ cảm biến đo độ ẩm đất	23
Hình 3.1: Trang chủ ThingSpeak	26
Hình 3.2: Tạo tài khoản trên ThingSpeak	26
Hình 3.3: Đăng nhập vào ThingSpeak	31
Hình 3.4: Tạo kênh dữ liệu trên ThingSpeak.....	31
Hình 3.5: Kênh dữ liệu trên ThingSpeak	32
Hình 3.6: Kiểm tra kênh dữ liệu ThingSpeak	33
Hình 3.7: Sửa đổi thông tin dữ liệu trên ThingSpeak	33

Hình 4.1: Mạch đo độ ẩm đất.....	34
Hình 4.2: Mạch Relay	35
Hình 4.3: Mạch in mạch đo độ ẩm đất.....	36
Hình 4.4: Mạch 3D mạch đo độ ẩm đất	36
Hình 4.5: Mạch đo độ ẩm đất hiển thị lên LCD <i>I2C</i>	37
Hình 4.6: Mặt trước của thiết bị.....	37
Hình 4.7: Mặt sau của thiết bị	38
Hình 4.8: Mặt bên của thiết bị.....	38

MỤC LỤC

PHẦN MỞ ĐẦU	1
PHẦN NỘI DUNG	2
CHƯƠNG 1: GIỚI THIỆU CHUNG VỀ QUÁ TRÌNH ĐO ĐỘ ẨM	2
CHƯƠNG 2: GIỚI THIỆU LINH KIỆN.....	3
2.1 MÀN HÌNH HIỂN THỊ	3
2.1.1 Màn hình LCD 20x4	3
2.1.2 Màn hình LCD I ² C	6
2.2 GIAO TIẾP I ² C – TWI	7
2.2.1 Định nghĩa thuật ngữ.....	7
2.2.1.1 Master.....	7
2.2.1.2 Slave	7
2.2.1.3 Serial Data (SDA)	7
2.2.1.4 Serial Clock (SCL).....	7
2.2.1.5 START Condition	8
2.2.1.6 STOP Condition	8
2.2.2 Đặc điểm giao tiếp I ² C	9
2.2.3 Cơ cấu hoạt động.....	10
2.3 ARDUINO UNO.....	11
2.3.1 Giới thiệu chung.....	11
2.3.2 Thông số kỹ thuật.....	12
2.3.3 Cấu tạo	14
2.3.3.1 Nhóm các phần cơ bản.....	14
2.3.3.2 Nhóm chân Power	14
2.3.3.3 Nhóm chân I/O.....	16
2.4 VI ĐIỀU KHIỂN ATMEGA328PU	17
2.4.1 Vi điều khiển Atmega328P-PU.....	17

2.4.2 Một số chân của Atmega328.....	18
2.5 CẢM BIẾN ĐỘ ẨM ĐẤT (SOIL MOISTURE SENSORS)	22
2.5.1 Giới thiệu chung.....	22
2.5.2 Đặc điểm	22
2.5.3 Sơ đồ nguyên lý.....	23
2.5.4 Thông số kỹ thuật.....	24
2.5.5 Kết nối.....	24
CHƯƠNG 3: THINGSPEAK INTERNET OF THINGS.....	25
3.1 GIỚI THIỆU	25
3.2 TẠO TÀI KHOẢN ĐĂNG NHẬP.....	26
3.3 THIẾT KẾ CHƯƠNG TRÌNH ĐƯA DỮ LIỆU LÊN TRANG THINGSPEAK.....	27
3.4 TẠO KÊNH DỮ LIỆU	31
3.5 THAY ĐỔI THÔNG TIN HIỂN THỊ TRÊN KÊNH DỮ LIỆU	33
CHƯƠNG 4: THIẾT KẾ MẠCH VÀ THI CÔNG ĐỀ TÀI	34
4.1 MẠCH ĐO ĐỘ ẨM ĐẤT.....	34
4.1.1 Sơ đồ khối của mạch	34
4.1.2 Nguyên lý mạch phát hiện sự thay đổi độ ẩm đất.....	34
4.1.3 Nguyên lý mạch chuyển đổi và điều khiển	34
4.1.4 Nguyên lý mạch ngoại vi, thiết bị theo dõi và ổn định độ ẩm đất	35
4.2 THIẾT KẾ MẠCH VÀ THI CÔNG MẠCH	36
4.2.1 Mạch in.....	36
4.2.2 Ảnh thực tế	37
4.3 ỨNG DỤNG	39
PHẦN KẾT LUẬN	41
TÀI LIỆU THAM KHẢO.....	42

PHẦN MỞ ĐẦU

1. LÝ DO CHỌN ĐỀ TÀI

Người lao động đa số làm việc theo cảm hứng, họ không thể kiểm soát lượng nước cần tưới cho cây trồng, khi họ tưới quá ít, khi họ tưới quá nhiều, đôi khi vì khoảng cách từ điểm tưới nước đến điểm tắt nước lại quá xa, hay khi họ nghỉ ngơi, nghe điện thoại. Họ sẽ để nước chảy tự do. Như vậy nguồn nước của chúng ta sẽ rất lãng phí.

Vậy có cách nào để tưới tiêu một cách hợp lý, vừa tiết kiệm nước, tiết kiệm thời gian, tiết kiệm sức người, tiết kiệm chi phí bỏ ra hay không?

Xuất phát từ các vấn đề trên, tôi đã chọn “ĐO VÀ ỔN ĐỊNH ĐỘ ẨM ĐẤT” để làm đề tài luận văn tốt nghiệp

2. MỤC ĐÍCH NGHIÊN CỨU

- Tạo ra sản phẩm ứng dụng giúp tưới tiêu hiệu quả.
- Quản lý và nhận biết được xu hướng thay đổi của độ ẩm đất.
- Ứng dụng thành tựu của Công nghệ 4.0 (IoT) vào nông nghiệp nước ta. Giúp nền nông nghiệp nước ta phát triển, giúp cải tiến quy trình quản lý trong nông nghiệp.

3. ĐỐI TƯỢNG NGHIÊN CỨU

- Cảm biến độ ẩm đất và trang web Thingspeak.

4. PHẠM VI NGHIÊN CỨU

- Nghiên cứu tổng quát về lý thuyết của cảm biến độ ẩm đất.
- Thiết kế mạch ứng dụng sử dụng cảm biến đo độ ẩm đất.
- Tạo ra sản phẩm.
- Quản lý kết quả thực nghiệm trên trang ThingSpeak.

5. PHƯƠNG PHÁP NGHIÊN CỨU

- Sử dụng phương pháp nghiên cứu lý thuyết áp dụng vào thực nghiệm.

6. GIỚI HẠN ĐỀ TÀI

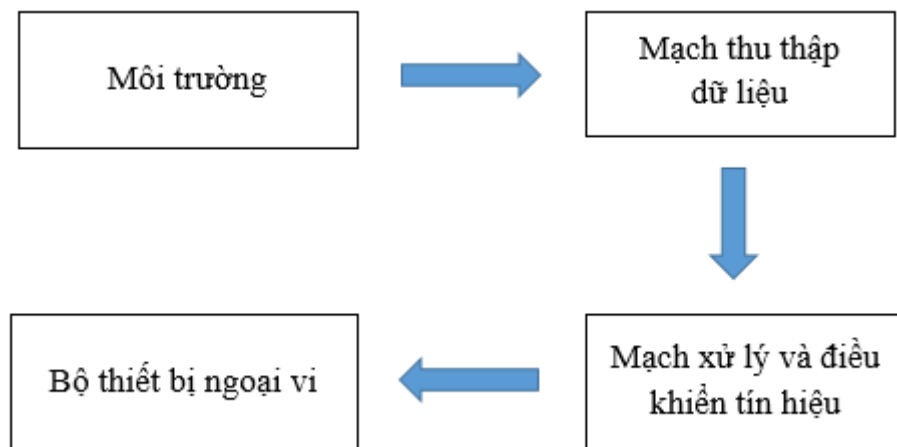
Trong bài luận văn này em chỉ nghiên cứu về: cảm biến đo độ ẩm đất, board Arduino Uno, lập trình Arduino, I²C, trang web ThingSpeak.

PHẦN NỘI DUNG

CHƯƠNG 1

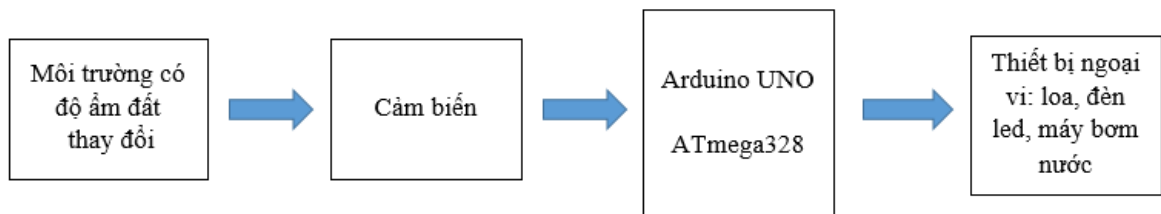
GIỚI THIỆU CHUNG VỀ QUÁ TRÌNH ĐO ĐỘ ẨM

Cảm biến là một thuật ngữ chuyên dùng trong ngành Vật lý. Cảm biến dùng để xác định sự thay đổi vật lý của môi trường, từ không điện (môi trường) sang có điện (qua bộ thu thập dữ liệu). Tín hiệu được xử lý và điều khiển tùy thuộc vào mục đích sử dụng mà đưa ra các lệnh điều khiển cho các thiết bị ngoại vi.



Hình 1.1: Quy trình tổng quát. Ảnh: Internet

Khi môi trường có độ ẩm thay đổi, cảm biến sẽ thu nhận dữ liệu từ sự thay đổi của môi trường, tín hiệu được truyền đến board Arduino, nhờ vào ngôn ngữ lập trình, cùng với ATmega328 được xem là bộ não của mạch và có chức năng xử lý tín hiệu, đưa tín hiệu lên máy tính hoặc LCD và điều khiển các thiết bị ngoại vi như loa, đèn led, máy bơm nước.



Hình 1.2: Quy trình mạch đo độ ẩm đất. Ảnh: Internet

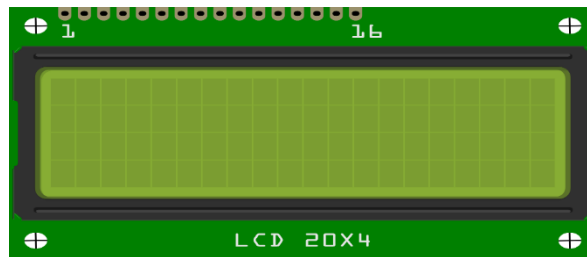
CHƯƠNG 2

GIỚI THIỆU LINH KIỆN

2.1 MÀN HÌNH HIỂN THỊ

2.1.1 Màn hình LCD 20x4

Màn hình LCD 20x4 gồm 4 hàng (hàng 0, 1, 2, 3), và 20 ô. (ô 0, 1, 2,..., 19); được dùng để hiển thị dữ liệu, gồm 16 chân kết nối với Arduino UNO.



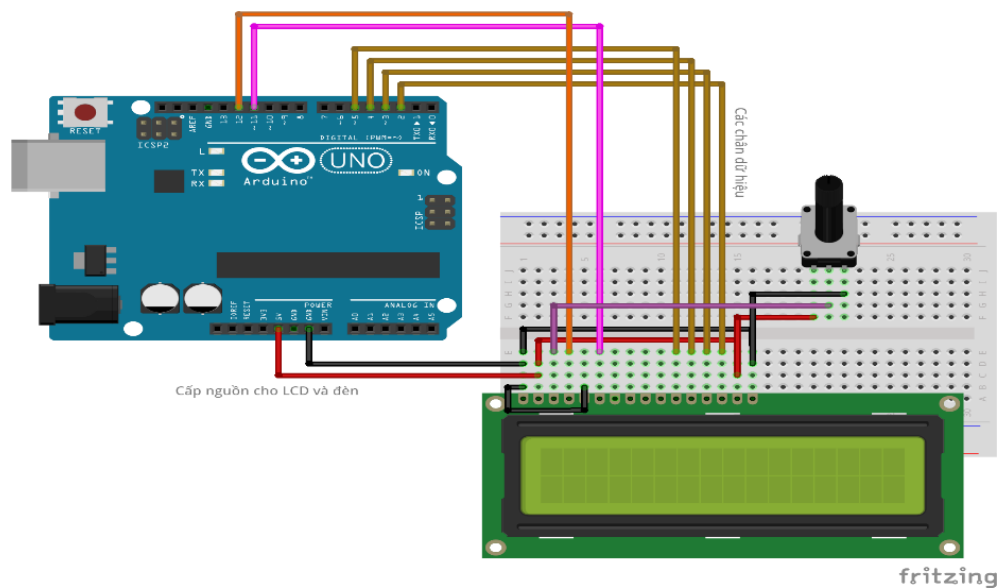
Hình 2.1: Màn hình LCD 20x4. Ảnh: Internet

Bảng 2.1: Các chân của LCD 20x4

Chân	Ký hiệu	Mức Logic	I/O	Chức năng
1	V_{SS}	-	-	GND
2	V_{CC}	-	-	Nguồn +5V
3	V_{ee}	-	-	Độ tương phản
4	RS	0/1	I	0 = nhập lệnh 1 = nhập dữ liệu
5	R/W	0/1	I	0 = ghi dữ liệu 1 = đọc dữ liệu
6	E	1,1 → 0	I	Tín hiệu cho phép

7	DB0	0/1	I/O	Bus dữ liệu 0
8	DB1	0/1	I/O	Bus dữ liệu 1
9	DB2	0/1	I/O	Bus dữ liệu 2
10	DB3	0/1	I/O	Bus dữ liệu 3
11	DB4	0/1	I/O	Bus dữ liệu 4
12	DB5	0/1	I/O	Bus dữ liệu 5
13	DB6	0/1	I/O	Bus dữ liệu 6
14	DB7	0/1	I/O	Bus dữ liệu 7
15	LAMP +	-	-	Đèn nền cực anode
16	LAMP -	-	-	Đèn nền cực cathode

Sơ đồ nối dây

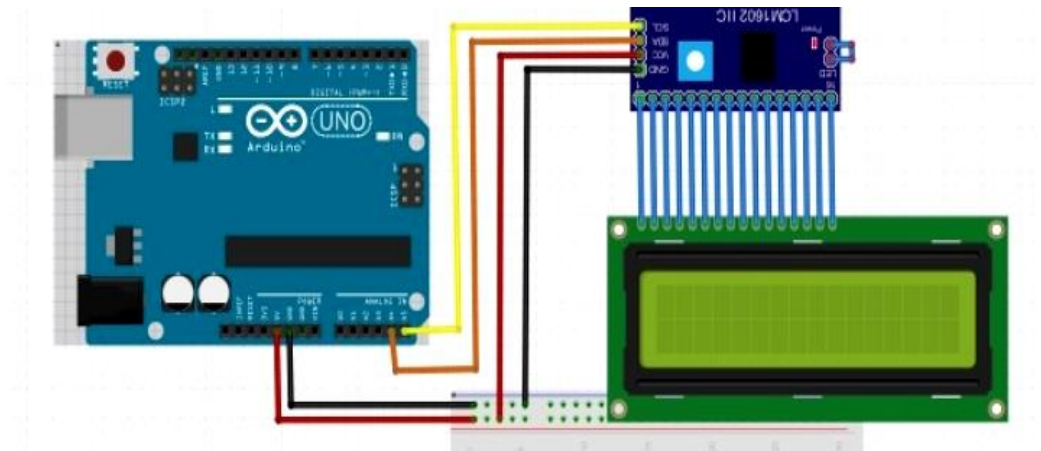


Hình 2.2: Cách nối dây giữa LCD 20x4 với Arduino. Ảnh: Internet

Bảng 2.2: Cách nối các chân giữa LCD và Arduino UNO

BIẾN TRỞ		LCD		ARDUINO
Chân		Chân		Chân
1	–	1	–	+5V
3	–	2	–	GND
2	–	3		
		4	–	12
		5	–	+5V
		6	–	11
		11	–	5
		12	–	4
		13	–	3
		14	–	2
		15	–	+5V
		16	–	GND

2.1.2 Màn hình LCD I²C



Hình 2.3: Cách nối dây giữa LCD I²C và Arduino. Ảnh: Internet

Thông thường, sử dụng màn hình LCD sẽ phải nối rất nhiều dây. Để đơn giản hóa công việc, một loại vi điều khiển LCD sử dụng giao tiếp I²C được tạo ra. Với cách nối dây đơn giản hơn.

Bảng 2.3: Cách nối các chân giữa LCD I²C Arduino UNO

LCD I ² C	ARDUINO
V_{CC}	+5V
GND	GND
SDA	A_4
SCL	A_5

2.2 GIAO TIẾP I²C – TWI

2.2.1 Định nghĩa thuật ngữ

TWI là một module truyền thông nối tiếp đồng bộ trên các chip AVR dựa trên chuẩn truyền thông I²C. I²C là một chuẩn truyền thông do hãng điện tử Philips Semiconductor sáng lập.

Giao tiếp I²C là chuẩn giao tiếp nối tiếp 2 dây.

2.2.1.1 Master

Là chip khởi động quá trình truyền nhận, phát đi địa chỉ của thiết bị cần giao tiếp và tạo xung giữ nhịp trên đường SCL.

2.2.1.2 Slave

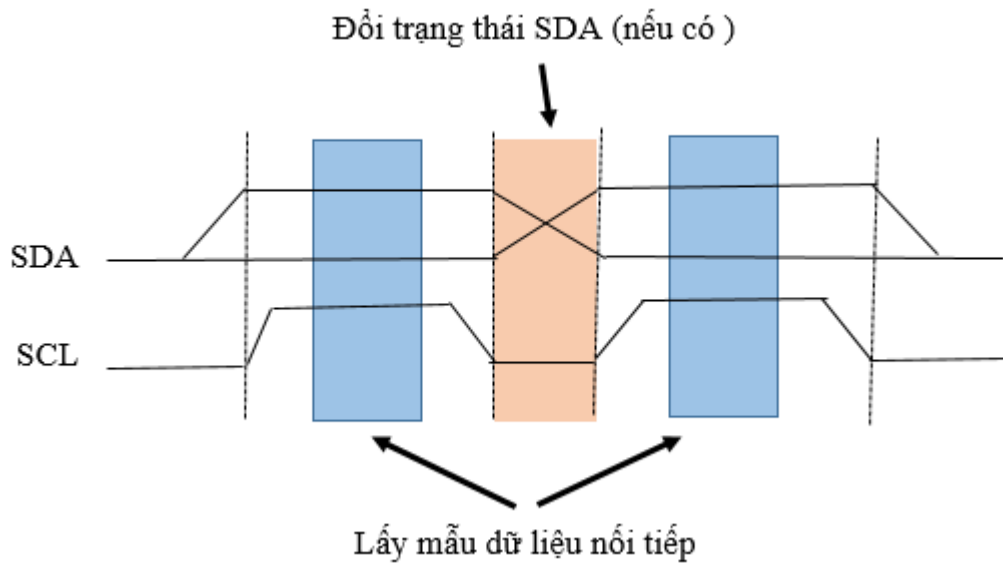
Là chip có một địa chỉ cố định, được gọi bởi Master và phục vụ yêu cầu từ Master.

2.2.1.3 Serial Data (SDA)

Là đường dữ liệu nối tiếp, tất cả các thông tin về địa chỉ hay dữ liệu đều được truyền trên đường này theo thứ tự từng bit một.

2.2.1.4 Serial Clock (SCL)

Là đường giữ nhịp nối tiếp. TWI (I²C) là chuẩn truyền thông nối tiếp đồng bộ, cần có 1 đường tạo xung giữ nhịp cho quá trình truyền/nhận, cứ mỗi xung trên đường giữ nhịp SCL, một bit dữ liệu trên đường SDA sẽ được lấy mẫu. Dữ liệu nối tiếp trên đường SDA được lấy mẫu khi đường SCL ở mức cao trong một chu kỳ giữ nhịp, vì thế đường SDA không được đổi trạng thái khi SCL ở mức cao (trừ START và STOP condition). Chân SDA có thể được đổi trạng thái khi SCL ở mức thấp.



Hình 2.4: Xung giữ nhịp trên SCL. Ảnh: Internet

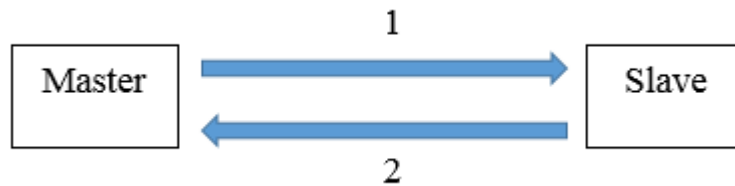
2.2.1.5 START Condition

Điều kiện bắt đầu từ trạng thái nghỉ, khi cả SDA và SCL ở mức cao nếu Master muốn thực hiện một “cuộc gọi”, Master sẽ kéo chân SDA xuống thấp trong khi SCL vẫn cao. Trạng thái này gọi là START Condition (chúng ta gọi tắt là S).

2.2.1.6 STOP Condition

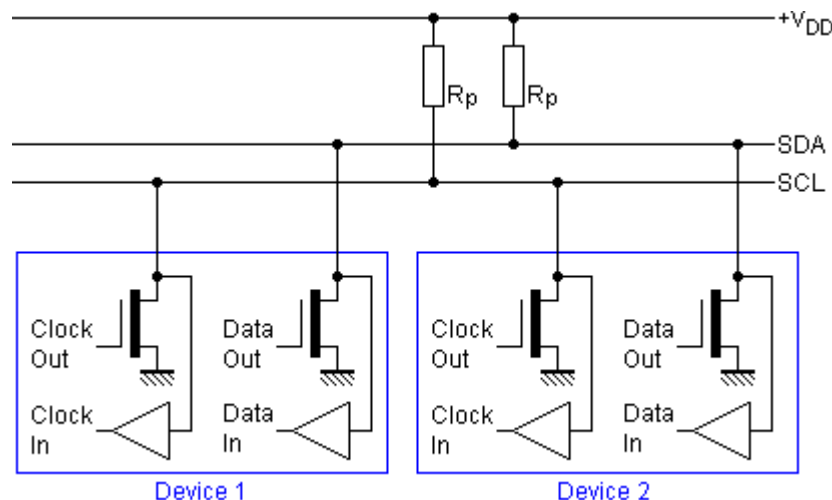
Điều kiện kết thúc sau khi thực hiện truyền/nhận dữ liệu, nếu Master muốn kết thúc quá trình nó sẽ tạo ra một STOP condition. STOP condition được Master thực hiện bằng cách kéo chân SDA lên cao khi đường SCL đang ở mức cao. STOP condition chỉ được tạo ra sau khi địa chỉ hoặc dữ liệu đã được truyền/nhận.

2.2.2 Đặc điểm giao tiếp I²C



Hình 2.5: Phương thức hoạt động của I²C. Ảnh: Internet

- Quá trình 1: Master “gọi” Slave gửi dữ liệu về cho Master
- Quá trình 2: Master gửi dữ liệu về cho Slave.
- Slave chỉ làm việc khi được Master yêu cầu.
- Giữa Slave và Slave không thể trao đổi dữ liệu cho nhau.
- Khi một thiết bị ngoại vi kết nối với đường Bus I²C thì chân SDA của nó sẽ nối với SDA của bus, chân SCL sẽ nối với chân SCL.



Hình 2.6: Kết nối các thiết bị ngoại vi với SDA, SCL. Ảnh: Internet

Mỗi dây SDA và SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên. Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I²C của các thiết bị ngoại vi đều ở dạng cực máng hở. Giá trị của các điện trở này khác nhau tùy thuộc vào từng thiết bị và chuẩn giao tiếp.

- Một thiết bị hay một IC kết nối với Bus I²C đều có 1 địa chỉ riêng.

2.2.3 Cơ cấu hoạt động

Quá trình truyền nhận dữ liệu từ Master đến 1 Slave



Hình 2.7: Quá trình truyền nhận dữ liệu từ Master đến Slave. Ảnh: Internet

Để truyền một dữ liệu, điều kiện cần phải có là: tín hiệu Start, địa chỉ cần truyền đến và Stop.

❖ Chế độ hoạt động (tốc độ truyền)

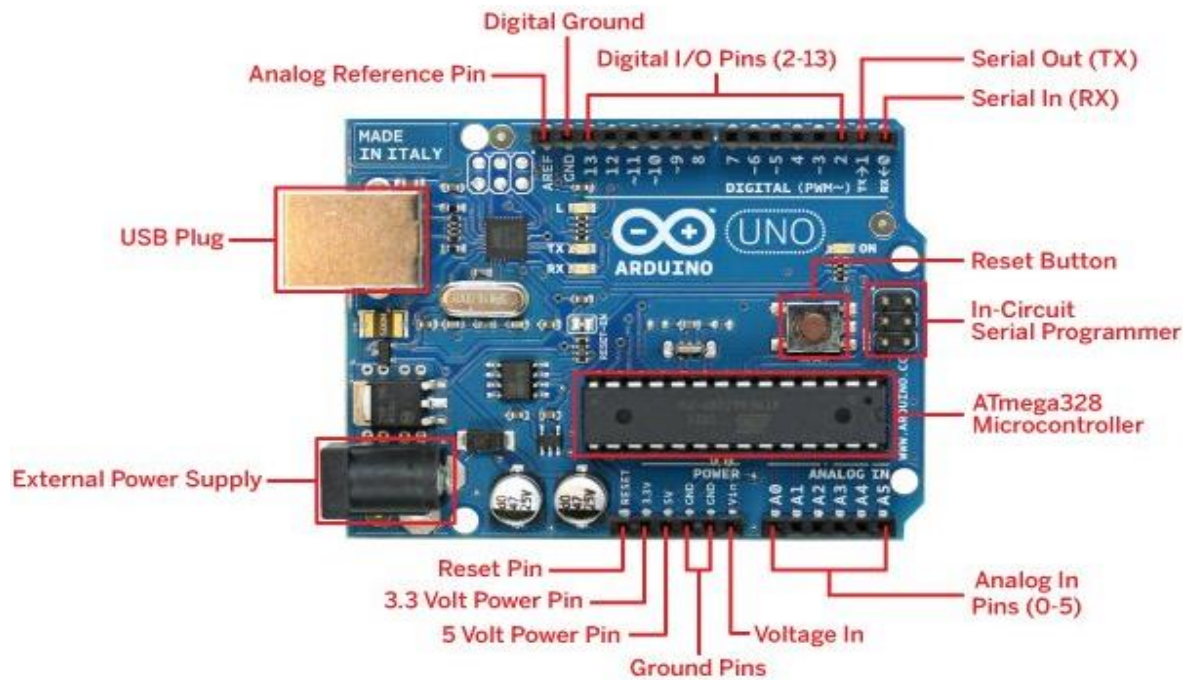
Các Bus I²C có thể hoạt động ở 3 chế độ hay nói cách khác, các dữ liệu trên Bus I²C được truyền trong 3 chế độ khác nhau:

- Chế độ tiêu chuẩn:
 - Chế độ tiêu chuẩn ban đầu được phát hành vào những năm 80.
 - Tốc độ dữ liệu tối đa 100kbps.
 - Bit địa chỉ: 7 bit, cho phép tối đa 112 thiết bị kết nối (Slave)
- Chế độ tăng cường hoặc chế độ nhanh:
 - Tốc độ dữ liệu tối đa 400kbps.
 - Chân SDA và SCL của thiết bị I²C ở trạng thái trở kháng cao khi không cấp nguồn.
- Chế độ cao tốc:
 - Tốc độ dữ liệu cao gấp 35 lần so với chế độ tiêu chuẩn.
 - Cung cấp 1,7Mbps.

❖ Một Bus I²C có thể hoạt động ở nhiều chế độ:

- Một chủ - một tớ.
- Một chủ - nhiều tớ.
- Nhiều chủ - nhiều tớ.

2.3 ARDUINO UNO



Hình 2.8: Board Arduino UNO. Ảnh: Internet

2.3.1 Giới thiệu chung

- Board Arduino là thiết bị phần cứng dùng để “phiên dịch” ngôn ngữ giữa máy tính và tín hiệu đầu vào; đồng thời cũng là thiết bị dùng để nạp chương trình cho Vi điều khiển.
- Cấu tạo board Arduino có thể chia thành 3 phần:
 - + Nhóm phần cơ bản.
 - + Nhóm chân nguồn.
 - + Nhóm chân I/O.

2.3.2 Thông số kỹ thuật.

Các thông số kỹ thuật chính của Arduino UNO

Bảng 2.4: Thông số kỹ thuật của Arduino UNO

Tên	Thông số
Điện áp hoạt động	5V
Điện áp đầu vào (được đề nghị)	7-12V
Điện áp đầu vào (giới hạn)	6-20V
Số chân I/O kỹ thuật số	14 (trong đó có 6 chân cung cấp đầu ra PWM)
PWM Digital I/O Pins	6
Analog Input Pins	6
Dòng điện DC mỗi I/O	20mA
Dòng điện DC với chân 3.3V	50mA
Bộ nhớ flash	32KB (Atmega328P) Trong đó 0.5KB sử dụng bởi bộ nạp khởi động
SRAM	2KB (Atmega328P)
EEPROM	1KB (Atmega328P)
Tốc độ đồng hồ	16MHz

Chiều dài	68.6mm
Bề rộng	53.4mm
Cân nặng	25g

Lưu ý:

- Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó, phải hết sức cẩn thận, kiểm tra các cực tính của nguồn trước khi cấp cho Arduino UNO.
- Các chân 3.3V và 5V trên Arduino là các chân dùng để cấp nguồn ra cho các thiết bị khác, không phải là các chân cấp nguồn vào. Việc cấp nguồn sai vị trí có thể làm hỏng board.
- Cấp nguồn ngoài không qua cổng USB cho Arduino UNO với điện áp dưới 6V có thể làm hỏng board.
- Cấp điện áp trên 13V vào chân RESET trên board có thể làm hỏng vi điều khiển Atmega328.
- Cường độ dòng điện I/O ở tất cả các chân Digital và Analog của Arduino UNO nếu vượt quá 200mA sẽ làm hỏng vi điều khiển.
- Cấp điện áp trên 5.5V vào các chân Digital hoặc Analog của Arduino UNO sẽ làm hỏng vi điều khiển.
- Cường độ dòng điện qua một chân Digital hoặc Analog bất kì của Arduino UNO vượt quá 40mA sẽ làm hỏng vi điều khiển. Do đó nếu không dùng để truyền nhận dữ liệu, nên mắc một điện trở hạn dòng.

2.3.3 Cấu tạo

2.3.3.1 Nhóm các phần cơ bản

Arduino Uno có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Mức điện áp logic là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở pull-up được thiết kế có sẵn trong vi điều khiển Atmega328. Khi muốn dùng các điện trở pull up này, thì cần đến lập trình.

Bảng 2.5: Nhóm các phần cơ bản của Arduino UNO

Tên	Chức năng
USB plug	Cổng liên kết với USB hoặc cáp để truyền dữ liệu
External Power Supply	Cổng cấp nguồn 5V
Atmega328 Microcontroller	Đề liên kết với Atmega328
Reset Button	Nút khởi động lại Arduino UNO
In-Circuit Serial Programmer	
Power pin	Các chân nguồn
Analog and Digital ground	Các chân tín hiệu vào/ra của Arduino UNO

2.3.3.2 Nhóm chân Power

Có thể cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyến dùng là 7-9V DC cho kit Arduino UNO. Khi không có sẵn nguồn từ cổng USB, sẽ có cấp nguồn bằng pin vuông 9V. Nếu cấp nguồn vượt quá ngưỡng giới hạn 20V, kit sẽ cháy ngay.

Bảng 2.6: Nhóm chân Power của Arduino UNO

Chân	Công dụng	Mô tả
GND (Ground)	Đầu nối đất	Các chân nối đất của các linh kiện được nối với nhau.
5V	Đầu ra điện áp 5V ()	Dòng tối đa cho phép cấp ở pin này là 0.5A.
3.3V	Đầu ra điện áp 3.3V	Dòng tối đa cho phép cấp ở pin này là 0.05A.
Vin	Cấp nguồn ngoài cho kit	Nối cực dương của nguồn với pin Vin và cực âm của nguồn với pin GND.
IOREF	Điện áp hoạt động của vi điều khiển	Điện áp luôn là 5V. Tuy nhiên, không được lấy nguồn từ pin này cấp đi chỗ khác.
RESET	Reset lại vi điều khiển	Chân reset được nối với nút bấm. nguyên lý là chân Reset sẽ được nối với GND qua 1 điện trở 10KΩ.

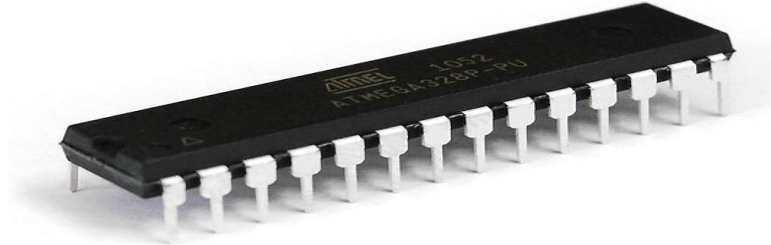
2.3.3.3 Nhóm chân I/O

Arduino Uno có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Mức điện áp logic là 0V và 5V với dòng vào/ra tối đa trên mỗi chân là 40mA. Ở mỗi chân đều có các điện trở được thiết kế có sẵn trong vi điều khiển Atmega328.

Bảng 2.7: Nhóm chân I/O của Arduino UNO

Nhóm	Chân	Chức năng
Analog In	A_0 , A_1 , A_2 , A_3 , A_4 , A_5	Đầu vào Analog. Dùng để ghi nhận các tín hiệu thô được Sensor truyền về.
Digital I/O	0	Pin RX: Gửi dữ liệu đi
	1	Pin TX: nhận dữ liệu đến
	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13	Là đầu ra Digital. Dùng để truyền tín hiệu ra được điều khiển thông qua quá trình lập trình, với code: digitalWrite().
	GND	Digital Ground

2.4 VI ĐIỀU KHIỂN ATMEGA328PU



Hình 2.9: Vi điều khiển ATmega328. Ảnh: Internet

Mô tả

ATmega328 có tên đầy đủ là Atmega328P-PU. Atmega328 là linh hồn của board mạch Arduino, sức mạnh phần cứng mà Arduino có được là từ đây.

Là “bộ não” của mạch điều khiển, bộ phận để Arduino nạp chương trình và dữ liệu. Có thể sử dụng tách biệt khi không có Arduino.

2.4.1 Vi điều khiển Atmega328P-PU.

Các thông số của vi điều khiển Atmega328P-PU được mô tả bởi

Bảng 2.8: Thông số của ATmega328

Nội dung	Thông số	Ghi chú
Kiến trúc	8 bit	AVR
Xung nhịp lớn nhất	20MHz	
Bộ nhớ chương trình (Flash)	32KB	Những đoạn lệnh lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển.

Bộ nhớ EEPROM	1KB	Giống như một chiếc ổ cứng mini – nơi có thể đọc và ghi dữ liệu của mình mà khi mất điện dữ liệu sẽ không bị mất đi.
Bộ nhớ RAM	2KB	Lưu trữ các biến khai báo. Khi mất điện RAM sẽ bị mất.
Điện áp hoạt động	1.8-5V	
Số timer	3	Gồm 2 timer 8 bit và 1 timer 16 bit.
Số kênh xung PWM	6	1 timer gồm 2 kênh.

2.4.2 Một số chân của Atmega328

Với 23 chân có thể sử dụng cho các kết nối I/O. Chia làm 3 cổng:

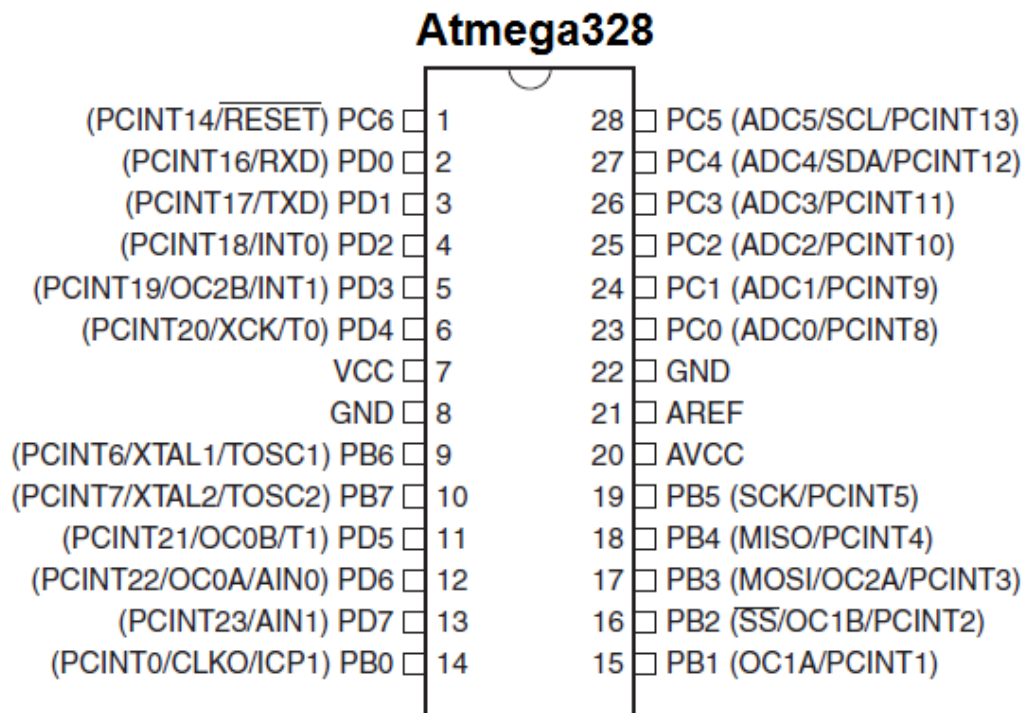
- Cổng B: 8 chân (PB0 → PB7)
- Cổng C: 7 chân (PC0 → PC6)
- Cổng D: 8 chân (PD0 → PD7)

Bảng 2.9: Các chân ATmega328

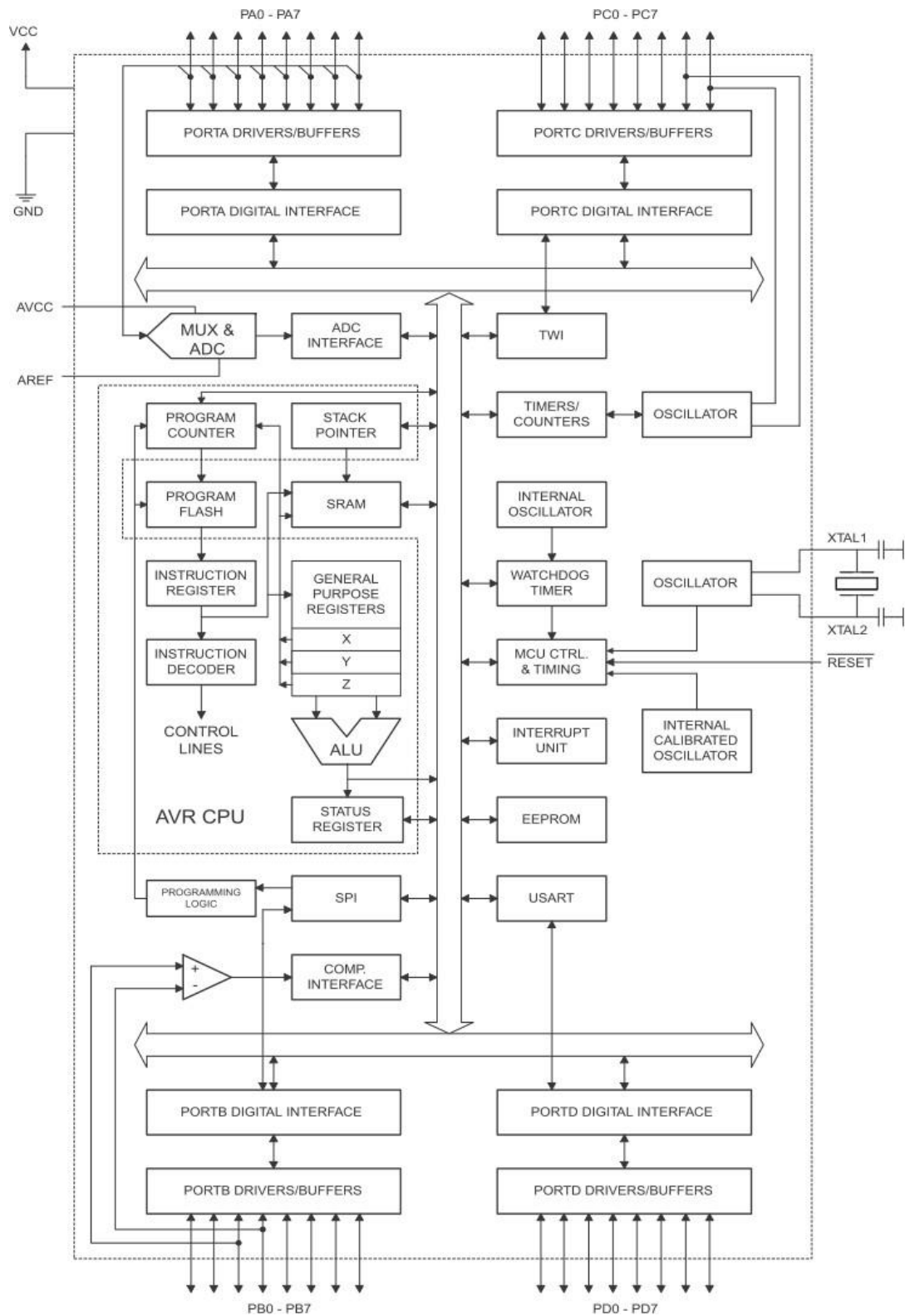
Port	Số chân	Chân	Tên	Chức năng	Ghi chú
B	14	PB0	ICP1	Là cổng vào ra hai hướng 8 bit, có điện trở nối lên nguồn	
	15	PB1	OC1A		

	16	PB2	SS/OC1B	dương bên trong. Cung cấp các chức năng ứng với các chức năng đặc biệt của ATmega32.	
	17	PB3	MOSI/OC2		
	18	PB4	MISO		
	19	PB5	SCK		
	9	PB6	XTAL1/TOSC1		
	10	PB7	XTAL2/TOSC2		
C	23	PC0	ADC0	Là cổng vào tương tự cho sự chuyển đổi sang số. Nó cũng là cổng vào ra hai hướng 8 bit trong trường hợp không sử dụng làm cổng chuyển đổi tương tự, có điện trở nối lên nguồn dương bên trong. Cung cấp đường dữ liệu địa chỉ vào ra theo kiểu hợp kều khi dùng bộ nhớ bên ngoài.	
	24	PC1	ADC1		
	25	PC2	ADC2		
	26	PC3	ADC3		
	27	PC4	ADC4		SDA
	28	PC5	ADC5		SCL
D	2	PD0	RXD	Là cổng vào ra hai hướng 8 bit, có điện trở nối lên nguồn dương bên trong. Cung cấp các chức năng ứng với các chức năng đặc biệt của Atmega32.	
	3	PD1	TXD		
	4	PD2	INT0		
	5	PD3	INT1		
	6	PD4	XCK/T0		

	11	PD5	T1		
	12	PD6	AIN0		
	13	PD7	AIN1		
	1	PC6	RESET	Lỗi vào đặt lại.	
	7		V_{CC}	Điện áp nguồn của Atmega328.	
	20		AVCC		
	8,22		GND	Nối đất.	
	21		AREF		



Hình 2.10: Các chân của ATmega328. Ảnh: Internet



Hình 2.11: Sơ đồ bên trong ATmega328. Ảnh: Internet

2.5 CẢM BIẾN ĐỘ ẨM ĐẤT (SOIL MOISTURE SENSORS)

2.5.1 Giới thiệu chung

Cảm biến phát hiện độ ẩm đất, bình thường đầu ra mức thấp, khi đất thiếu nước đầu ra sẽ mức cao, bộ cảm biến có thể sử dụng để tưới nước tự động khi không có người quản lý. Khi độ ẩm đất thấp, kích relay bật hệ thống tưới nước tự động. Cảm biến phát hiện độ ẩm đất có thể tùy chỉnh được.

Độ nhạy của cảm biến độ ẩm đất có thể điều chỉnh được. Phần đầu dò được cắm vào đất để phát hiện độ ẩm đất, khi độ ẩm đạt ngưỡng thiết lập, đầu ra sẽ chuyển từ trạng thái thấp lên cao.



Hình 2.12: Cảm biến độ ẩm đất. Ảnh: Internet

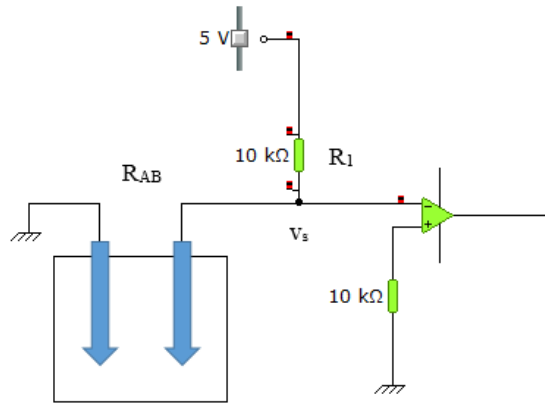
2.5.2 Đặc điểm

Bộ cảm biến gồm 2 phần:

- Đầu dò kim loại:
 - Bản chất: 2 que kim loại tách biệt nhau, chống rỉ sét.
 - Một que được nối và nguồn 5V, một que
 - Khi môi trường thay đổi, tức độ ẩm thay đổi, sẽ có sự chênh lệch điện trở của 2 que.
- Module:
 - Là mạch khuếch đại tín hiệu, khi nhận được tín hiệu từ sự thay đổi điện trở của đầu dò do độ ẩm đất thay đổi, sự thay đổi này không quá lớn. Nên tín hiệu qua module sẽ được khuếch đại và tiếp tục truyền đến thiết bị xử lý kế tiếp.
 - Module có 2 mức truyền:
 - + Tín hiệu Analog: luôn trả về giá trị là một số nguyên từ 0 đến 1023 tương ứng với thang điện áp từ 0 – 5V.

- + Tín hiệu Digital: trả về giá trị ở 2 mức trạng thái là Low (0) và High (1).
- Trên module có 1 biến trở dùng để điều chỉnh độ khuếch đại của mạch.

2.5.3 Sơ đồ nguyên lý



Hình 2.13: Sơ đồ nguyên lý của bộ cảm biến đo độ ẩm đất. Ảnh: Internet

Dựa vào sơ đồ nguyên lý của bộ cảm biến đo độ ẩm đất (Hình 2.13), ta có:

Điện thế vào Analog V_s (V) của tín hiệu:

$$V_s = \frac{R_{AB}}{R_1 + R_{AB}} \times V_{CC}$$

$$V_s = \frac{V_{CC}}{1 + \frac{R_1}{R_{AB}}}$$

Thay $V_{CC} = 5V$, $R_1 = 10K$ vào công thức, ta được:

$$V_s = \frac{5V}{1 + \frac{10k}{R_{AB}}}$$

- Khi môi trường đất khô (khô nhất):

$$R_{AB} \approx \infty \quad V_s \approx 5V$$

- Khi môi trường đất ướt (ướt nhất):

$$R_{AB} \approx 0 \quad V_s \approx 0V$$

Tín hiệu của cảm biến được diễn tả qua thang đo điện thế từ 5-0V (khô – ướt). Do giới hạn của thang đo điện thế không đủ rộng để người sử dụng có thể nhận biết được sự thay đổi độ ẩm trong đất (các thông số hiển thị không có sự

thay đổi đáng kể). Nên một phép ánh xạ từ thang đo điện thế 5-0V sang thang đo 0- 1023; thang đo mới này giúp người dùng có thể nhận biết được sự thay đổi độ ẩm trong đất tốt hơn.

2.5.4 Thông số kỹ thuật

Bảng 2.10: Thông số kỹ thuật của cảm biến độ ẩm đất

Nội dung	Thông số
Điện áp hoạt động	3.3-5V
Kích thước PCB	3x1.6cm
Led đỏ	Báo vào nguồn
Led xanh	Báo độ ẩm
DO	Đầu ra tín hiệu số
AO	Đầu ra tín hiệu Analog

2.5.5 Kết nối

Bảng 2.11: Cách nối chân giữa cảm biến và Arduino

Cảm biến	Arduino
V_{cc}	5V
GND	GND
D_0	D_1
A_0	A_0

CHƯƠNG 3

THINGSPEAK INTERNET OF THINGS

3.1 GIỚI THIỆU

ThingSpeak là một trang web cung cấp các dịch vụ nhằm xây dựng các ứng dụng IoT. Nó có khả năng thu thập dữ liệu tại thời gian thực, biểu diễn các dữ liệu thu thập được dưới dạng biểu đồ, có khả năng tạo ra các ứng dụng cho các dịch vụ web, mạng xã hội và các API khác.

ThingSpeak bao gồm:

- 8 trường để lưu trữ dữ liệu của bất kỳ dữ liệu nào. Đây là nơi có thể sử dụng để lưu trữ các dữ liệu từ một cảm biến hoặc từ một thiết bị nhúng.
- 3 vị trí dùng để lưu trữ các vĩ độ, kinh độ và độ cao; nó rất hữu ích cho việc theo dõi thiết bị di chuyển.
- 1 tình trạng dùng để mô tả dữ liệu được lưu trữ trong kênh.

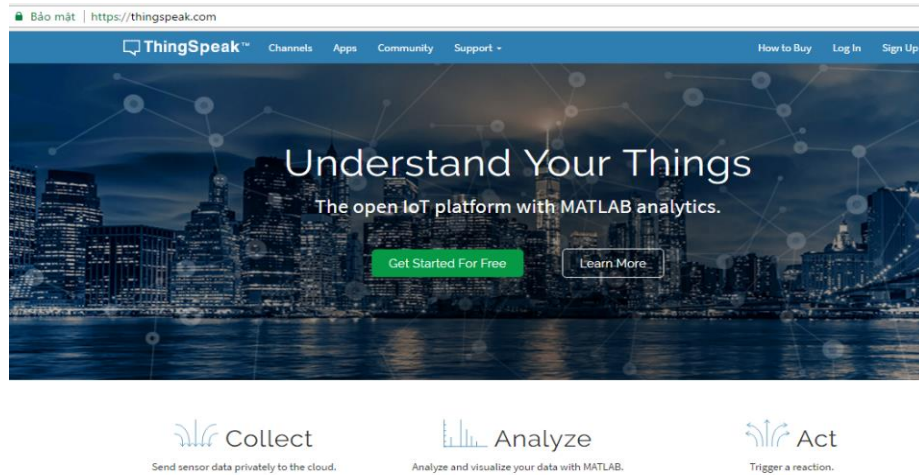
Để sử dụng ThingSpeak ta phải đăng ký và tạo ra một kênh. Một khi chúng ta có kênh, ta có thể gửi dữ liệu.

Internet of Thing (IoT) là một hệ thống kết nối vạn vật, bao gồm: hệ điều hành nhúng, khả năng giao tiếp với Internet. Một trong những điểm chung của hệ thống IoT là kết nối vạn vật bởi dịch vụ IoT. Cái thú vị trong IoT là nó có khả năng kết nối với mọi thứ mà chúng ta không làm được. Khai thác IoT ở việc kết nối với những dịch vụ trực tuyến. Dịch vụ đóng vai trò như một người quản lý vô hình bằng cách cung cấp các khả năng khác, từ các dữ liệu đơn giản và giám sát để phân tích dữ liệu phức tạp

Link dẫn đến trang ThingSpeak: <https://www.thingspeak.com/login>

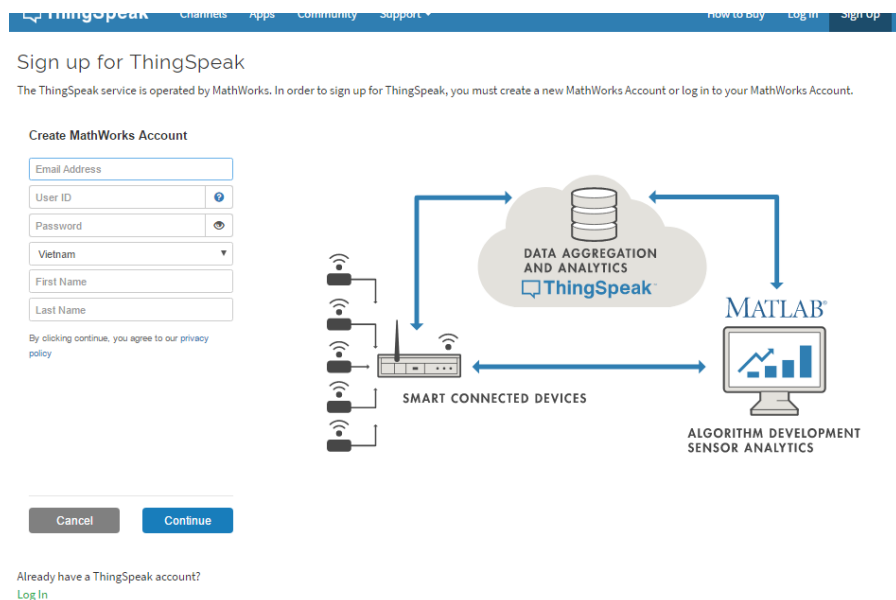
3.2 TẠO TÀI KHOẢN ĐĂNG NHẬP

Bước 1: Vào trang web Thingspeak bởi: <https://www.thingspeak.com/login>



Hình 3.1: Trang chủ ThingSpeak. Ảnh thực nghiệm

Bước 2: Sign up → Create Mathworks Account → điền đầy đủ thông tin → Continue → làm theo các hướng dẫn trên trang web.



Hình 3.2: Tạo tài khoản trên ThingSpeak. Ảnh thực nghiệm

3.3 THIẾT KẾ CHƯƠNG TRÌNH ĐƯA DỮ LIỆU LÊN TRANG THINGSPEAK

Sử dụng lập trình Arduino để đưa dữ liệu, thông tin lên trang ThingSpeak.

```
#include <Dhcp.h>
#include <LCD.h>
// #include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
#include <Wire.h>
#define BLYNK_PRINT Serial
#include <Ethernet.h>
#include <EthernetClient.h>
#include <SPI.h>
#include <SimpleTimer.h>
#include <BlynkSimpleEthernet.h>
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xED, 0xFE, 0xEE };
char thingSpeakAddress[] = "api.thingspeak.com"; // địa chỉ tìm đến
String apiKey = "P8XVIWIRKAN19BOB"; // ghi dữ liệu vào kênh mình đã tạo
const int updateThingSpeakInterval = 16 * 1000; // tốc độ truyền dữ liệu
long lastCheck=0;
const int updateInterval = 10000;
char auth []="ab9b7cc68036461f8eabef05cf095bb8";
IPAddress ip(192, 168, 1, 73); // địa chỉ IP
EthernetClient client; // khai báo người dùng
int sensorPin=A0; // chân đọc dữ liệu A0
long previousMillis = 0;
unsigned long currentMillis = 0;
long interval = 250000;
int sensorValue=0; // giá trị ban đầu được gán bằng 0
int doam1=0; // giá trị doam được gán bằng 0
String data1; // trường thứ nhất trên trang thingspeak
String data2; // trường thứ hai trên trang thingspeak
int buzzer_ledPin=11; // khai báo loa ở chân 11 của Arduino
int MotorPin=12; // khai báo động cơ ở chân 12 của Arduino
SimpleTimer timer;

void setup() { // bắt đầu khởi tạo các giao diện
  Serial.begin(9600); // tốc độ truyền dữ liệu cho màn hình
  lcd.begin(20,4); // khai báo LCD 20x4
```

```

while (!Serial) {           // cấu trúc cần có để truyền dữ liệu lên ThingSpeak
;
}
// bắt đầu kết nối
if (Ethernet.begin(mac) == 0) {
Serial.println("Ket noi khong thanh cong!");
Ethernet.begin(mac, ip);
}
for (byte thisByte = 0; thisByte < 4; thisByte++) // Xuất địa chỉ IP
{
Serial.print(Ethernet.localIP()[thisByte], DEC);
Serial.print(".");
}
Serial.println("");
Serial.print("");
Serial.begin (9600);
timer.setInterval(1000L, sendSensor);           //1s đọc cảm biến 1 lần.
lcd.setBacklight(HIGH);                          //nền LCD sáng
lcd.begin(20,4);

lcd.begin(20, 4);
lcd.clear();
lcd.setBacklight(HIGH);    //Màn hình LCD được bật sáng
lcd.setCursor(2,0);        //con trỏ LCD ở ô 3 dòng 1
lcd.print("DANG KHOI DONG"); // xuất ra màn hình LCD dòng text
lcd.setCursor(5,1);        // con trỏ LCD ở ô 6 dòng 2
lcd.print("THIET BI");     // xuất ra màn hình LCD dòng text
pinMode(11, OUTPUT);
// khai báo chân số 11 của arduino làm nhiệm vụ OUTPUT LED_Buzze
pinMode(12, OUTPUT);
// khai báo chân số 9 của arduino làm nhiệm vụ OUTPUT Motor

delay(4000); // cứ 4s sẽ truyền tín hiệu lên LCD 1 lần
lcd.clear();
data1 = "";
data2 = "";
} //kết thúc khởi tạo ban đầu

//khởi tạo hàm reset
void(* resetFunc) (void) = 0;

// nhận dữ liệu và xử lý
void sendSensor(){

```

```

int sensorValue=analogRead(sensorPin);      //đọc giá trị vào
int doam1=map(sensorValue, 0, 1023, 100, 0);
//chuyển đổi giá trị đọc được sang thang đo mới
delay(1000); //1s đọc giá trị 1 lần
}

//khởi tạo vòng lặp đưa dữ liệu lên trang ThingSpeak và màn hình LCD
void loop(){
//dòng lệnh bắt buộc phải có
timer.run ();
currentMillis = millis();
if(currentMillis - previousMillis > interval) {
previousMillis = currentMillis;
}

sensorValue=analogRead(sensorPin);      //đọc giá trị mà cảm biến truyền

int doam1=map(sensorValue,0,1023,100,0);// đổi thang đo từ 0-1023 sang 100-0
int doam2=map(sensorValue,0,1023,0,100);// đổi thang đo từ 0-1023 sang 0-100
String data1="field1="; //tên của data 1 là field1
String data2="field2="; //tên của data 2 là field2
data1+=String(doam1); //gán giá trị doam1 vào data1
data2+= String(doam2); //gán giá trị doam1 vào data1
Serial.println(data1);      //xuất giá trị tại data1 ra màn hình
Serial.println(data2);      //xuất giá trị tại data1 ra màn hình

//Cấu trúc người dùng bắt buộc
if (client.connect(thingSpeakAddress,80)) {
    client.println("POST /update HTTP/1.1");
    client.println("Host: api.thingspeak.com");
    client.println("Connection: close");
    client.println("X-THINGSPEAKAPIKEY: "+apiKey);
    client.println("Content-Type: application/x-www-form-urlencoded");
    client.print("Content-Length: ");
    client.println(data1.length()+data2.length()+1);

    client.println("");
    client.print(data1);
    client.print("&");
    client.print(data2);
}
if (client.connected()) {
    client.stop();

```

```

    }

    //hiển thị dữ liệu lên LCD
    lcd.clear();
    lcd.setCursor(0,0); //con trỏ đặt ở ô 1 hàng 1
    lcd.print("DO AM : "); //xuất dòng chữ DO AM:
    lcd.print(doam1); //gán giá trị doam1 vào
    lcd.print("%"); // xuất ký hiệu %
    lcd.setCursor(0,1); //con trỏ đặt ở ô 1 hàng 2
    lcd.print("Do am:"); //xuất dòng chữ Do am:
    lcd.setCursor(6,1); //con trỏ đặt ở ô 7 hàng 2
    lcd.print(sensorValue); //gán giá trị ban đầu chưa chuyển đổi thang đo vào

    //Các trường hợp của độ ẩm và hiển thị trạng thái trên LCD
    if(doam1<30)
    {
        lcd.setCursor(1,2);
        lcd.print("TRANG THAI CUA DAT");
        lcd.setCursor(8,3);
        lcd.print("KHO");
        digitalWrite(11,HIGH);
        digitalWrite(12,HIGH);
    }
    if(doam1>30 && doam1<70)
    {
        lcd.setCursor(1,2);
        lcd.print("TRANG THAI CUA DAT");
        lcd.setCursor(8,3);
        lcd.print("AM");
        digitalWrite(11,LOW);
        digitalWrite(12,LOW);
    }
    if(doam1>70)
    {
        lcd.setCursor(1,2);
        lcd.print("TRANG THAI CUA DAT");
        lcd.setCursor(8,3);
        lcd.print("UOT");
        digitalWrite(11,LOW);
        digitalWrite(12,LOW);
    }
    delay(1000);

```

}

3.4 TẠO KÊNH DỮ LIỆU

Bước 1: Vào trang web Thingspeak → Log In.

Bước 2: Tạo kênh dữ liệu:

Vào Channel → điền đầy đủ thông tin → Save.

Hình 3.3: Đăng nhập vào ThingSpeak. Ảnh thực nghiệm

Đây là giao diện để tạo kênh dữ liệu. Điền các thông tin cần thiết vào New Channel. Nếu không hiểu nội dung của New Channel thì bạn có thể xem thông tin tại “Help”.

Hình 3.4: Tạo kênh dữ liệu trên ThingSpeak. Ảnh thực nghiệm

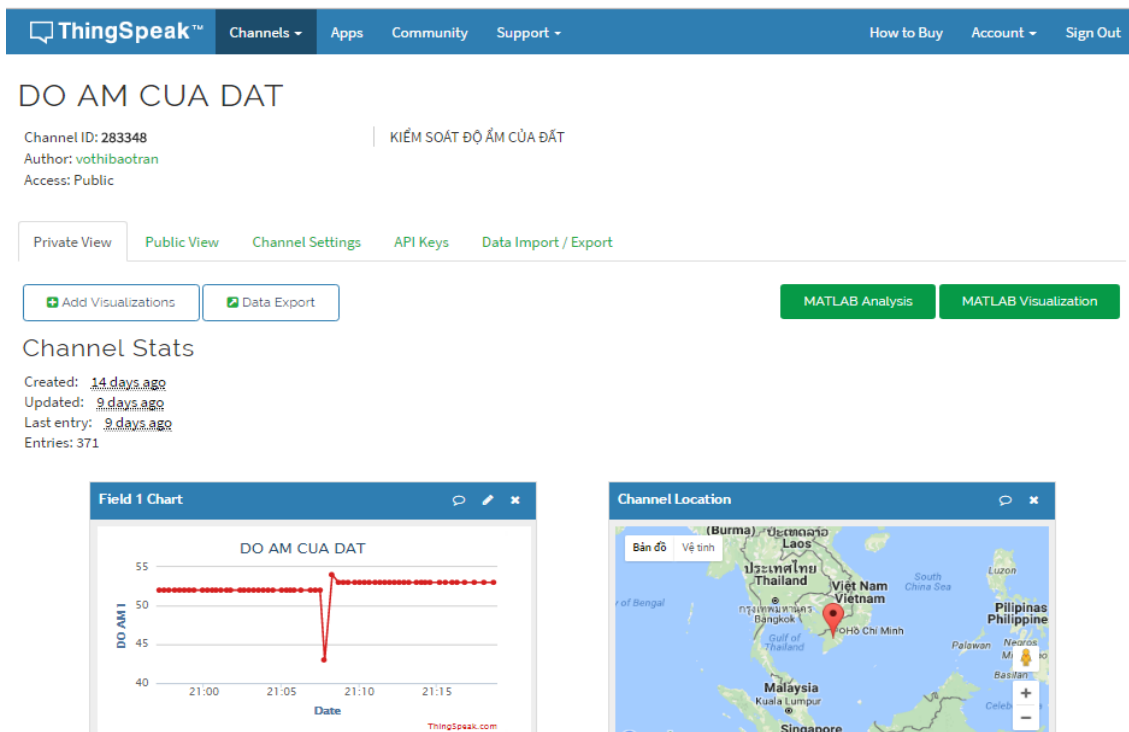
Bước 3: (Khi đã tạo xong kênh dữ liệu)

Cho chạy kênh dữ liệu vừa khởi tạo.

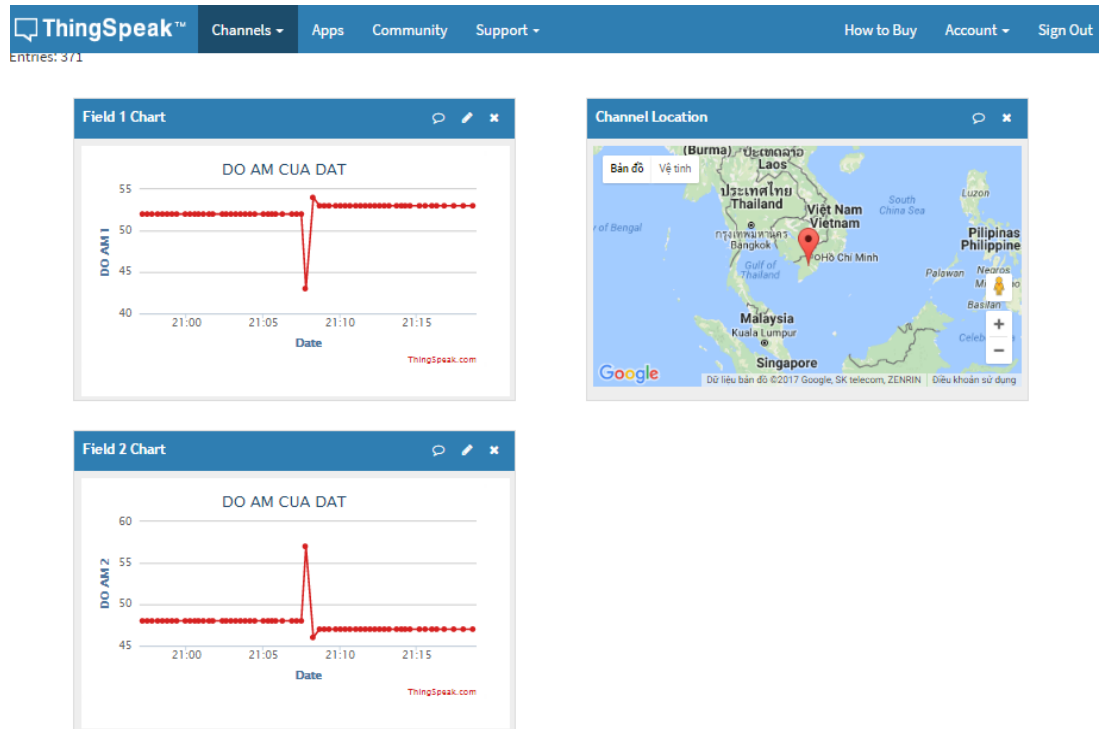
Ta được cung cấp: Channel ID (mã người dùng).

Bước 4: Kiểm tra kết quả.

Khi chương trình thiết kế đã chạy, ta có thể xem kết quả của thiết bị hay cảm biến hoạt động như thế nào.



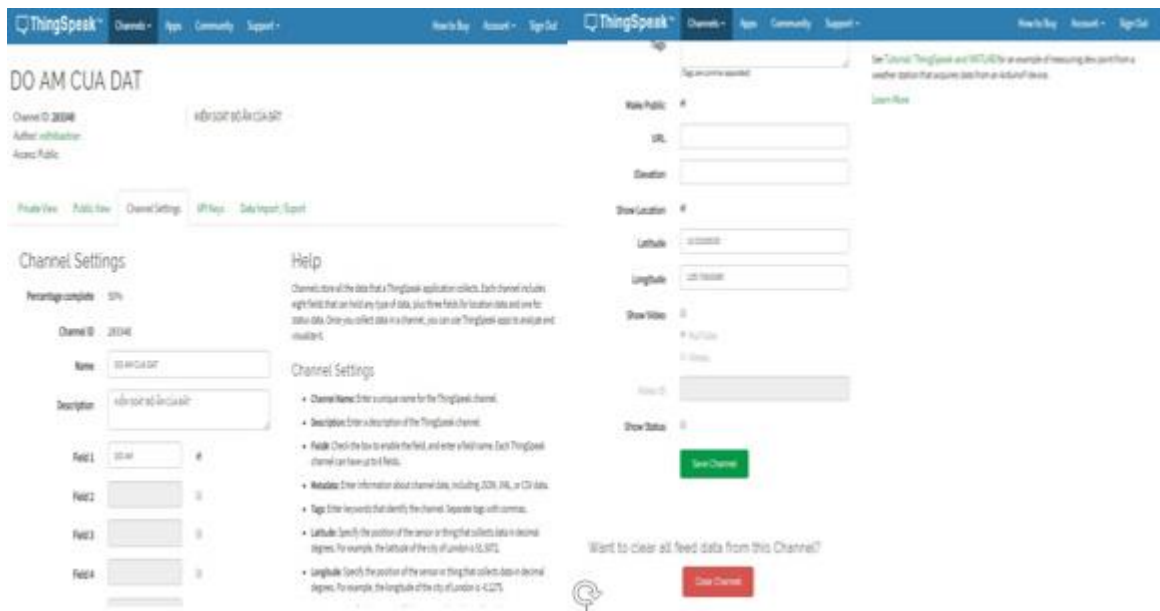
Hình 3.5: Kênh dữ liệu trên ThingSpeak. Ảnh thực nghiệm



Hình 3.6: Kiểm tra kênh dữ liệu ThingSpeak. Ảnh thực nghiệm

3.5 THAY ĐỔI THÔNG TIN HIỂN THỊ TRÊN KÊNH DỮ LIỆU

Channel Setting → sửa thông tin → Save.



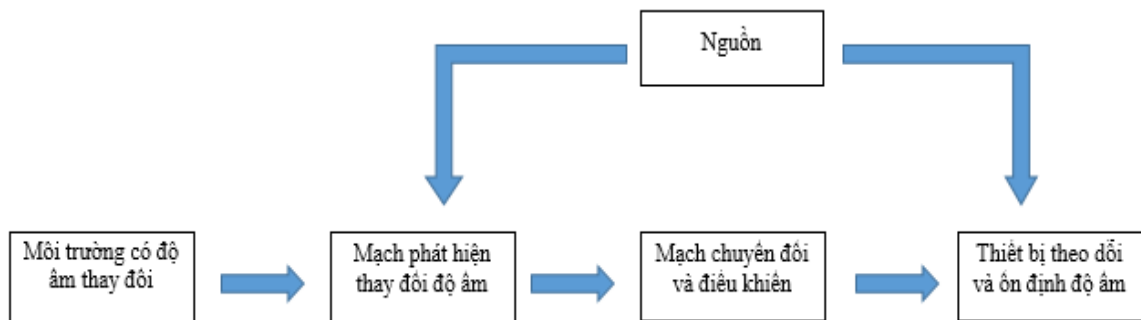
Hình 3.7: Sửa đổi thông tin dữ liệu trên ThingSpeak. Ảnh thực nghiệm

CHƯƠNG 4

THIẾT KẾ MẠCH VÀ THI CÔNG ĐỀ TÀI

4.1 MẠCH ĐO ĐỘ ẨM ĐẤT

4.1.1 Sơ đồ khối của mạch



Hình 4.1: Mạch đo độ ẩm đất. Ảnh: Internet

4.1.2 Nguyên lý mạch phát hiện sự thay đổi độ ẩm đất

Đặt đầu dò của bộ cảm biến và môi trường đất có sự thay đổi độ ẩm, điện trở của 2 đầu dò sẽ thay đổi (hay có sự chênh lệch điện trở giữa hai que điện trở của đầu dò). Qua module của bộ cảm biến sẽ khuếch đại tín hiệu và truyền tín hiệu Analog vào Arduino.

Thời gian thu thập dữ liệu phụ thuộc vào “yêu cầu” của ATmega32 được điều khiển bởi ngôn ngữ lập trình Arduino.

4.1.3 Nguyên lý mạch chuyển đổi và điều khiển

Khi nhận tín hiệu đầu vào Analog từ bộ cảm biến. Arduino thu nhận dữ liệu và chuyển đổi tín hiệu dữ liệu từ Analog sang Digital. Tín hiệu đầu ra Digital có 2 mức năng lượng là 0: mức thấp (0V), 1: mức cao (5V). Đồng thời Arduino còn nạp dữ liệu cho ATmega32, giúp ATmega32 có thể hoạt động bình thường và độc lập khi không có Arduino.

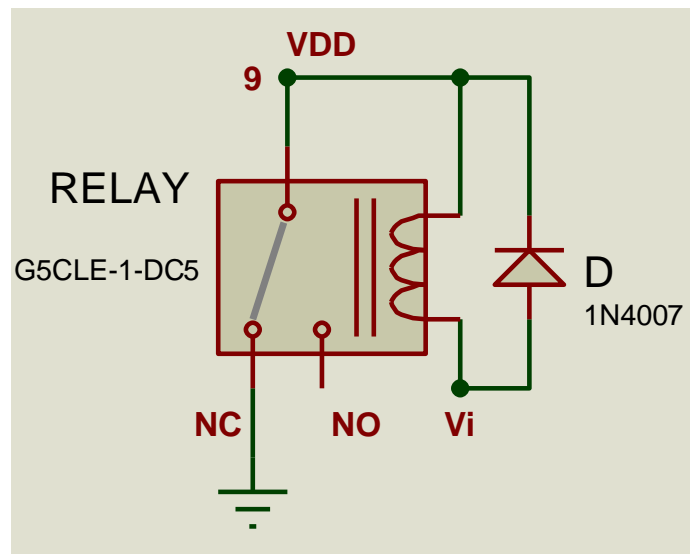
4.1.4 Nguyên lý mạch ngoại vi, thiết bị theo dõi và ổn định độ ẩm đất

Máy bơm nước được nối với relay, relay một đầu nối với nguồn 5v, một đầu được nối với chân 12 của Arduino (hay chân 18 của ATmega32).

Khi đầu ra ở trạng thái thấp, thì relay ở trạng thái NC, máy bơm nước không hoạt động.

Khi đầu ra ở mức cao, kích relay chuyển trạng thái từ NC sang NO. Kích máy bơm hoạt động. Đầu NO được nối với máy bơm nước (máy bơm đã nối nguồn),

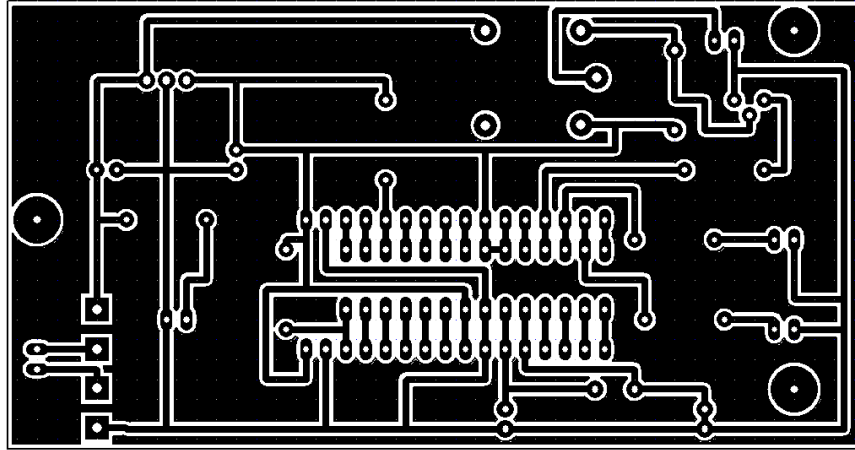
Ngoài ra, ta còn thay thế thiết bị ra bằng đèn, loa,... Tùy thuộc vào mục đích của người sử dụng. Cũng sẽ có nguyên lý tương tự như nguyên lý hoạt động sử dụng máy bơm nước.



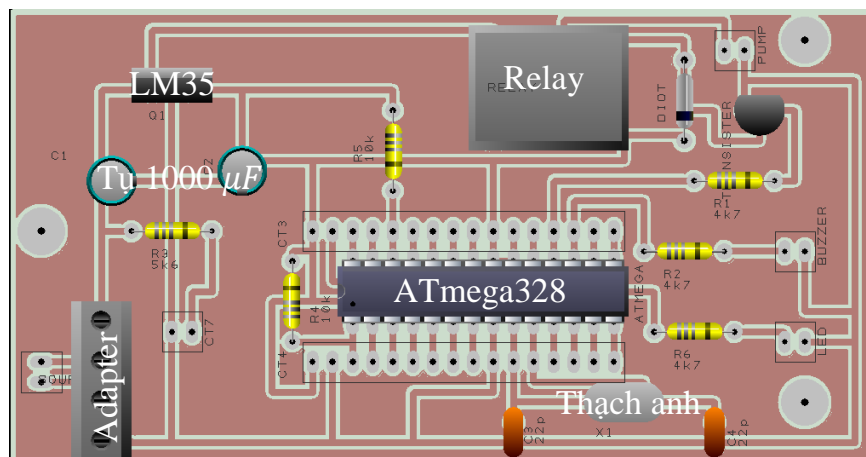
Hình 4.2: Mạch Relay. Ảnh: Internet

4.2 THIẾT KẾ MẠCH VÀ THI CÔNG MẠCH

4.2.1 Mạch in



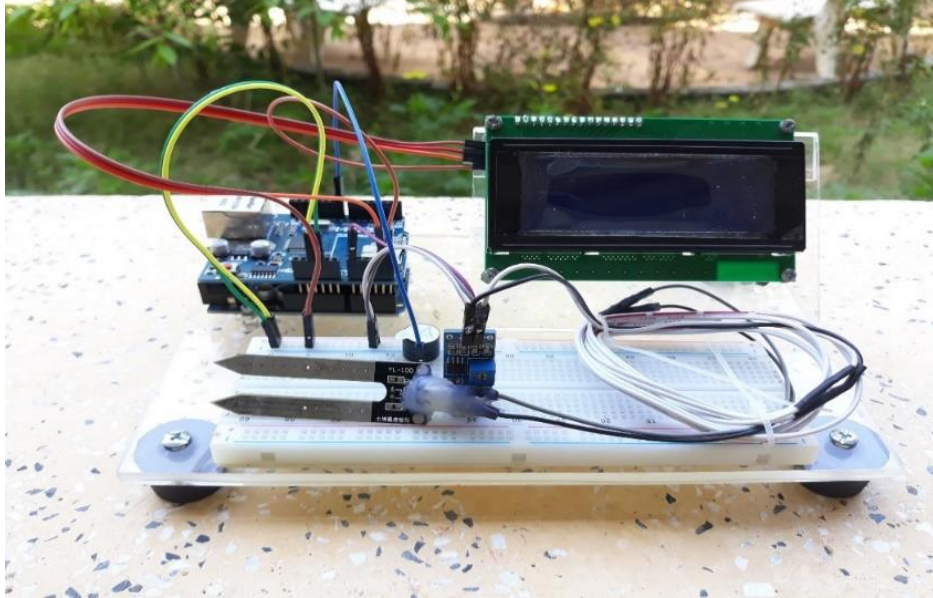
Hình 4.3: Mạch in mạch đo độ ẩm đất. Ảnh: Internet



Hình 4.4: Mạch 3D mạch đo độ ẩm đất. Ảnh: Internet

Chế tạo mạch in là công đoạn quan trọng trong quá trình chế tạo bảng mạch điện tử. Với một bảng mạch điện, bao gồm bộ nguồn riêng, nó trở thành 1 sản phẩm dụng cụ tự động, với “bộ não” ATmega328 sẽ giúp bộ thiết bị linh kiện này trở thành một bộ sản phẩm thông minh giúp đo và ổn định độ ẩm đất.

4.2.2 Ảnh thực tế



Hình 4.5: Mạch đo độ ẩm đất hiển thị lên LCD I^2C . Ảnh thực tế



Hình 4.6: Mặt trước của thiết bị. Ảnh thực tế



Hình 4.7: Mặt sau của thiết bị. Ảnh thực tế



Hình 4.8: Mặt bên của thiết bị. Ảnh thực tế

4.3 ỨNG DỤNG

Chè xanh (trà xanh) là cây dài ngày, chỉ trồng 1 lần nhưng cho thu hoạch 30 – 40 năm, cho nên khâu chăm sóc vườn ươm cây giống là khâu rất quan trọng, quyết định trực tiếp đến tỉ lệ xuất vườn, chất lượng cây giống của vườn ươm.

Tùy thuộc vào từng giai đoạn của vườn ươm mà điều chỉnh độ ẩm thích hợp.

Bảng 4.1: Giai đoạn vườn ươm

Giai đoạn	Thời gian	Mô tả	Độ ẩm đất
1	Từ khi cắm cành đến 15- 20 ngày	Hom trà vừa tách khỏi cây mẹ sống tự lập, chưa ổn định	80%
2	Từ 15 – 20 ngày sau	Vết cắt của hom liền, hom chè hút nước mạnh. 2 ngày tưới 1 lần, 1,5l/1m ² bầu/1 lần tưới	70 – 80%
3	Từ 30 – 60 ngày	Rễ hình thành và phát triển, lượng nước cần tưới thường xuyên. 2-3 ngày tưới 1 lần, 1,5l/1m ² bầu/1 lần tưới	75 -80%
4	Từ 60 – 90 ngày	Bộ rễ giảm phát triển mạnh, đặc biệt là rễ hút, cây bắt đầu hút chất dinh dưỡng. 3 ngày tưới 1 lần, 1,5 - 2l/1m ² bầu/1 lần tưới	75 – 80%
5	Từ 90 – 120 ngày	Giai đoạn sinh trưởng của cây, cần nhiều nước. 5- 6 ngày tưới	70 – 80%

		1 lần, $2l/1m^2$ bầu/1 lần tưới.	
6	120 – 180 ngày	Cây con đã hoàn chỉnh. 5-6 ngày tưới 1 lần, $3l/1m^2$ bầu/1 lần tưới	70 -75%

❖ **Ưu nhược điểm của quá trình**

Ưu điểm:

Giúp cây trồng tăng trưởng nhanh.

Tiết kiệm tài nguyên (nước)

Giúp người nông dân trong vấn đề tưới tiêu.

Nhược điểm:

Yêu cầu sử dụng điện cao.

Tốn chi phí lắp đặt các cơ sở vật chất.

PHẦN KẾT LUẬN

Qua việc thực hiện luận văn “Đo và ổn định độ ẩm đất” đã đạt được các kết quả sau:

Về lý thuyết:

Có cơ hội kiểm tra, nghiên cứu những phần kiến thức lý thuyết đã học về cảm biến, các linh kiện, các mạch điện tử,... Qua đó giúp tôi nắm rõ hơn và vững tin hơn vào kiến thức tôi đã tích lũy được.

Về thực nghiệm:

Đã thiết kế được mạch điện tử thông minh đơn giản phục vụ trong học tập và đời sống.

Đã ứng dụng được trên mô hình trồng cây trước phòng thí nghiệm. Thiết kế thành công mạch “Đo và ổn định độ ẩm đất” để từ đó hoàn thành đề tài luận văn tốt nghiệp.

Đã hoàn thành mạch đo và ổn định độ ẩm đất bằng máy bơm nước.

Đã hoàn thành mạch đo và phát.

Đã đưa được dữ liệu lên trang Thingspeak.

→ Đáp ứng được yêu cầu đã đặt ra.

TÀI LIỆU THAM KHẢO

- [1]. Phạm Quang Huy, Lê Cảnh Trung – Lập trình điều khiển với Arduino – NXB KHOA HỌC VÀ KỸ THUẬT.
- [2]. Phạm Quang Huy, Lê Cảnh Trung – Vi điều khiển và ứng dụng Arduino (dành cho người tự học) –NXB BÁCH KHOA HÀ NỘI.
- [3]. Nguyễn Trí Tuấn (5/2016) – Giáo trình Cảm biến – NXB Đại học Cần Thơ.
- [4]. Thanh Huyền – Kỹ thuật trồng và chăm sóc cây chè.
- [5]. Khoa Nông Nghiệp – Độ ẩm đất với cây trồng.

WEB THAM KHẢO

Datasheet của I²C LCD:

http://category.alldatasheet.com/index.jsp?sSearchword=16x2%20lcd%20datasheet&gclid=CjwKCAjwwbHWBRBWEiwAMIV7E-uR8ExxZrWT2WAYLnw1nOmt0861C7Yoj4gbYCMvB3qKkhsecqp6_BoCaYkQAvD_BwE

Datasheet của Arduino UNO:

<https://datasheet.octopart.com/A000066-Arduino-datasheet-38879526.pdf>

Datasheet của ATmega328:

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

Datasheet của cảm biến đo độ ẩm đất (soil moisture sensor):

<http://forum.researchdesignlab.com/datasheet/sensors/soil%20moisture%20sensor.pdf>

Giới thiệu chuẩn giao tiếp I²C:

<http://arduino.vn/bai-viet/1053-giao-tiep-i2c-voi-nhieu-module>