

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÀ RỊA-VŨNG TÀU

Viện CNTT - Điện - Điện Tử

-----☎-----



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

ĐỒ ÁN TỐT NGHIỆP

TÊN ĐỀ TÀI:

**ĐIỀU KHIỂN MÁY BƠM BẰNG SÓNG WIFI VÀ
SÓNG ĐIỆN THOẠI**

Trình độ đào tạo	: Đại học chính quy
Ngành	: Công Nghệ Kỹ Thuật Điện - Điện Tử
Chuyên ngành	: Kỹ Thuật Điện - Điện Tử
Giảng viên hướng dẫn	: ThS. Phạm Chí Hiếu
Sinh viên thực hiện	: Ngô Văn Hoàng
MSSV	: 13030128
Lớp	: DH13DD

Bà Rịa - Vũng Tàu 7/2017

TRƯỜNG ĐẠI HỌC BÀ RỊA-VŨNG TÀU
KHOA ĐIỆN-ĐIỆN TỬ

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA
VIỆT NAM
Độc lập - Tự do - Hạnh phúc
-----o0o-----

PHIẾU GIAO ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP

(Đính kèm Quy định về việc tổ chức, quản lý các hình thức tốt nghiệp ĐH, CĐ ban hành kèm theo Quyết định số 585/QĐ-ĐHBRVT ngày 16/7/2013 của Hiệu trưởng Trường Đại học BR-VT)

Họ và tên sinh viên: Ngô Văn Hoàng Ngày sinh: 26/09/1995

MSSV : 13030128 Lớp: DH13DD

E-mail : sadprincebrvt95@gmail.com

Trình độ đào tạo : Đại học

Hệ đào tạo : Chính quy

Ngành : Công nghệ kỹ thuật điện-điện tử

Chuyên ngành : Kỹ thuật điện-điện tử

1. Tên đề tài: *Mô hình điều khiển máy bơm nước bằng sóng wifi và sóng điện thoại.*

2. Giảng viên hướng dẫn: Th.S. Phạm Chí Hiếu

3. Ngày giao đề tài:

4. Ngày hoàn thành đồ án/ khoá luận tốt nghiệp: 2/07/2017

Bà Rịa-Vũng Tàu, ngày tháng năm 2017

GIẢNG VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

SINH VIÊN THỰC HIỆN

(Ký và ghi rõ họ tên)

TRƯỞNG BỘ MÔN

(Ký và ghi rõ họ tên)

TRƯỞNG KHOA

(Ký và ghi rõ họ tên)

LỜI CAM ĐOAN

Tôi xin cam đoan đồ án này tổng quát lại kết quả quá trình nghiên cứu của tôi. Các số liệu, hình ảnh, thông tin trong đồ án đều trung thực, do tôi tìm hiểu, tham khảo từ nhiều nguồn tư liệu. Đồ án này không sao chép các đồ án đã có từ trước.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đề tài của mình. Trường đại học BÀ RỊA-VŨNG TÀU không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

Vũng Tàu, ngày ... tháng 07 năm 2017

Người cam đoan:

Ngô Văn Hoàng

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

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

Vũng Tàu, ngày ... tháng 07 năm 2017

Giáo viên hướng dẫn

PHẠM CHÍ HIẾU

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vũng Tàu, ngày ... tháng 07 năm 2017

Giáo viên phản biện

LỜI NÓI ĐẦU

Ngày nay, sự phát triển mạnh mẽ của khoa học công nghệ, cuộc sống của con người đã có những thay đổi ngày càng tốt hơn, với những trang thiết bị hiện đại phục vụ công cuộc công nghiệp hoá, hiện đại hoá đất nước. Đặc biệt góp phần vào sự phát triển đó thì ngành kỹ thuật điện tử đã góp phần không nhỏ trong sự nghiệp xây dựng và phát triển đất nước. Những thiết bị điện, điện tử được phát triển mạnh mẽ và được ứng dụng rộng rãi trong đời sống cũng như sản xuất. Từ những thời gian đầu phát triển vi xử lý đã cho thấy sự ưu việt của nó và cho tới ngày nay tính ưu việt đó ngày càng được khẳng định thêm. Những thành tựu của nó đã có thể biến được những cái tưởng chừng như không thể thành những cái có thể, góp phần nâng cao đời sống vật chất và tinh thần cho con người.

Để góp phần làm sáng tỏ hiệu quả của những ứng dụng trong thực tế của môn vi xử lý, sau một thời gian học tập được các thầy cô trong khoa giảng dạy về các kiến thức chuyên ngành, đồng thời được sự giúp đỡ nhiệt tình của các thầy cô trong khoa Điện-Điện tử, cùng với sự nỗ lực của bản thân, em đã **“thiết kế và chế tạo mô hình điều khiển máy bơm nước bằng song wifi và sóng điện thoại”** nhưng do thời gian, kiến thức và kinh nghiệm của em còn có hạn nên sẽ không thể tránh khỏi những sai sót. Em rất mong được sự giúp đỡ và tham khảo ý kiến của thầy cô và các bạn nhằm đóng góp phát triển thêm đề tài.

LỜI CẢM ƠN

Trước khi bắt đầu đồ án tốt nghiệp, với lòng biết ơn sâu sắc nhất, em xin cảm ơn quý thầy cô Khoa Điện- Điện tử đã tận tình truyền đạt kiến thức cũng như giúp đỡ em trong quá trình học tập tại trường.

Đặc biệt, em xin ghi nhớ sự nhiệt tình của thầy Phạm Chí Hiếu, người trực tiếp hướng dẫn và đã giúp em hoàn thành đồ án này.

Bên cạnh đó, em xin chuyển lời cảm ơn đến các thầy giảng dạy bộ môn vi xử lý đã nhiệt tình giúp đỡ em trong việc thu thập tài liệu, trao đổi thông tin và tạo mọi điều kiện thuận lợi trong quá trình xây dựng mô hình.

Sau cùng, tôi cũng xin cảm ơn những người bạn đã đóng góp ý kiến và hỗ trợ thông tin để hoàn thiện đồ án tốt nghiệp.

Vũng Tàu, ngày 5 tháng 07 năm 2017

Sinh viên thực hiện

Ngô Văn Hoàng

MỤC LỤC

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	3
NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN	4
MỤC LỤC	7
CHƯƠNG 1: TỔNG QUAN	8
1.1 Nhu cầu tự động hóa ở Việt Nam	8
1.2 Mục tiêu của đề tài	8
1.3 Tính tối ưu của đề tài	8
CHƯƠNG 2: THIẾT BỊ VÀ CÁC GIẢI PHÁP CÔNG NGHỆ	9
2.1 Giới thiệu về Mạch Arduino UNO R3	9
2.2.1 Đặc tính nổi bật	15
2.2.2 Thông số kỹ thuật	16
2.3 Module SIM900A	17
2.4 Giới thiệu về màn hình LCD 16x2:	20
2.5 Module chuyển đổi I2C cho LCD10602	39
Chương 3: THIẾT KẾ MẠCH VÀ CHƯƠNG TRÌNH	41
3.1 Thiết kế mạch trên Proteus:	41
3.2 Sơ đồ mạch in:	41
3.3 Phân viết chương trình:	42
Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI	49
4.1 Kết Luận:	49
4.1.1 Ưu điểm của đề tài:	49
4.1.2 Nhược điểm của đề tài:	49
4.2 Hướng phát triển:	49
TÀI LIỆU THAM KHẢO	50

CHƯƠNG 1:TỔNG QUAN

1.1 Nhu cầu tự động hóa ở Việt Nam

Tự động hóa là một lĩnh vực công nghệ rất quan trọng trong sự phát triển của mỗi quốc gia. Khi ngành này phát triển và được ứng dụng rộng rãi thì nó sẽ góp phần cải thiện đáng kể năng suất và chất lượng sản phẩm. Nhưng hiện tại ở nước ta ngành này vẫn còn rất thiếu và yếu về quy mô lẫn năng lực làm chủ công nghệ. Điều đó điều đó là hạn chế rất lớn cản trở sự phát triển về mọi mặt của đất nước. Nhưng nhìn về mặt tích cực thì đó cũng là cơ hội để ngành này khai thác nhu cầu rất lớn từ nền sản xuất còn khá lạc hậu của nước ta.

Và ngành tự động hóa trong nông nghiệp tại nước ta lại càng thiếu và yếu rất nhiều, đòi hỏi ngành phải đi sâu vào giải quyết nhiều vấn đề để nâng cao chất lượng cũng như số lượng của sản xuất nông nghiệp tại nước ta.

1.2 Mục tiêu của đề tài

- Nghiên cứu mô hình điều khiển máy bơm nước sử dụng sóng wifi và sóng điện thoại.
- Nghiên cứu cơ sở lý thuyết để xây dựng mô hình dựa trên các kiến thức đã học về lập trình
- Ứng dụng các công nghệ gần gũi với cuộc sống con người để xây dựng lên hệ thống điều khiển từ xa.

1.3 Tính tối ưu của đề tài

- Tạo tính tư duy cho sinh viên trong quá trình nghiên cứu.
- Có tính linh động và mở rộng cho sinh viên thiết kế mô hình dựa trên cơ sở thực tế.
- Mô hình đơn giản nhưng rất hữu ích.

CHƯƠNG 2: THIẾT BỊ VÀ CÁC GIẢI PHÁP CÔNG NGHỆ

2.1 Giới thiệu về Mạch Arduino UNO R3

Mạch Arduino Uno là dòng mạch Arduino phổ biến, khi mới bắt đầu làm quen, lập trình với Arduino thì mạch Arduino thường nói tới chính là dòng Arduino UNO. Hiện dòng mạch này đã phát triển tới thế hệ thứ 3 (*Mạch Arduino Uno R3*).

Arduino Uno R3 là dòng cơ bản, linh hoạt, thường được sử dụng cho người mới bắt đầu. Bạn có thể sử dụng các dòng Arduino khác như: Arduino Mega, Arduino Nano, Arduino Micro... Nhưng với những ứng dụng cơ bản thì mạch Arduino Uno là lựa chọn phù hợp nhất.



Hình 2.1: Board mạch Arduino



Hình 2.2: Chip Atmega 328

Arduino UNO có thể sử dụng 3 vi điều khiển họ 8bit AVR là: ATmega8 (*Board Arduino Uno r2*), ATmega168, ATmega328 (*Board Arduino Uno r3*). Bộ não này có thể xử lý những tác vụ đơn giản như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, điều khiển động cơ bước, điều khiển động cơ serve, làm một trạm đo nhiệt độ – độ ẩm và hiển thị lên màn hình LCD,... hay những ứng dụng khác.

Mạch Arduino UNO R3 với thiết kế tiêu chuẩn sử dụng vi điều khiển ATmega328. Tuy nhiên nếu yêu cầu phần cứng của bạn không cao hoặc túi tiền không cho phép, bạn có thể sử dụng các loại vi điều khiển khác có chức năng tương đương nhưng rẻ hơn như ATmega8 (bộ nhớ flash 8KB) hoặc ATmega168 (bộ nhớ flash 16KB).

Nguồn sử dụng

Arduino UNO R3 có thể được cấp nguồn 5V thông qua cổng USB hoặc cấp nguồn ngoài với điện áp khuyến dùng là 7-12V DC hoặc điện áp giới hạn là 6-20V. Thường thì cấp nguồn bằng pin vuông 9V là hợp lý nhất nếu bạn không có sẵn nguồn từ cổng USB. Nếu cấp nguồn vượt quá ngưỡng giới hạn trên, bạn sẽ làm hỏng Arduino UNO.

Các chân năng lượng

GND (Ground): cực âm của nguồn điện cấp cho Arduino UNO. Khi bạn dùng các thiết bị sử dụng những nguồn điện riêng biệt thì những chân này phải được nối với nhau.

5V: cấp điện áp 5V đầu ra. Dòng tối đa cho phép ở chân này là 500mA.

3.3V: cấp điện áp 3.3V đầu ra. Dòng tối đa cho phép ở chân này là 50mA.

Vin (Voltage Input): để cấp nguồn ngoài cho Arduino UNO, bạn nối cực dương của nguồn với chân này và cực âm của nguồn với chân GND.

IOREF: điện áp hoạt động của vi điều khiển trên Arduino UNO có thể được đo ở chân này. Và dĩ nhiên nó luôn là 5V. Mặc dù vậy bạn không được lấy nguồn 5V từ chân này để sử dụng bởi chức năng của nó không phải là cấp nguồn.

RESET: việc nhấn nút Reset trên board để reset vi điều khiển tương đương với việc chân RESET được nối với GND qua 1 điện trở 10KΩ.

Lưu ý:

Arduino UNO không có bảo vệ cắm ngược nguồn vào. Do đó bạn phải hết sức cẩn thận, kiểm tra các cực âm – dương của nguồn trước khi cấp cho Arduino UNO. Việc làm chập mạch nguồn vào của Arduino UNO sẽ biến nó thành một miếng nhựa chặn giấy. mình khuyên bạn nên dùng nguồn từ cổng USB nếu có thể.

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. Điều này không được nhà sản xuất khuyến khích.

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 vào/ra 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, bạn phải mắc một điện trở hạn dòng.

Khi các bạn sử dụng mạch Arduino, đặc biệt một số bạn mới bắt đầu tiếp xúc, làm quen thì việc cấp nguồn nên thận trọng. Theo mình thì nên sử dụng nguồn 5V chuẩn qua USB, hoặc sử dụng nguồn 9v cấp cho cổng đầu vào mạch Arduino. Tránh trường hợp hỏng mạch Arduino.

Bộ nhớ sử dụng

Vi điều khiển **Atmega328** tiêu chuẩn sử dụng trên Arduino uno r3 có:

32KB bộ nhớ Flash: những đoạn lệnh bạn lập trình sẽ được lưu trữ trong bộ nhớ Flash của vi điều khiển. Thường thì sẽ có khoảng vài KB trong số này sẽ được dùng cho bootloader nhưng đừng lo, bạn hiếm khi nào cần quá 20KB bộ nhớ này đâu.

2KB cho SRAM (Static Random Access Memory): giá trị các biến bạn khai báo khi lập trình sẽ lưu ở đây. Bạn khai báo càng nhiều biến thì càng cần nhiều bộ nhớ RAM. Tuy vậy, thực sự thì cũng hiếm khi nào bộ nhớ RAM lại trở thành thứ mà bạn phải bận tâm. Khi mất điện, dữ liệu trên SRAM sẽ bị mất.

1KB cho

EEPROM (Electrically Erasable Programmable Read Only Memory): đây giống như một chiếc ổ cứng mini – nơi bạn có thể đọc và ghi dữ liệu của mình vào đây mà không phải lo bị mất khi cúp điện giống như dữ liệu trên SRAM.

Các cổng vào/ra trên Arduino Board



Hình 2.3: Các cổng ra/vào

Mạch Arduino UNO có 14 chân digital dùng để đọc hoặc xuất tín hiệu. Chúng chỉ có 2 mức điện áp 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 từ được cài đặt ngay trong vi điều khiển ATmega328 (mặc định thì các điện trở này không được kết nối).

Một số chân **digital** có các chức năng đặc biệt như sau:

2 chân Serial: 0 (RX) và 1 (TX): dùng để gửi (transmit – TX) và nhận (receive – RX) dữ liệu TTL Serial. Arduino Uno có thể giao tiếp với thiết bị khác thông qua 2 chân này. Kết nối bluetooth thường thấy nói nôm na chính là kết nối Serial không dây. Nếu không cần giao tiếp Serial, bạn không nên sử dụng 2 chân này nếu không cần thiết

Chân PWM (~): 3, 5, 6, 9, 10, và 11: cho phép bạn xuất ra xung PWM với độ phân giải 8bit (giá trị từ 0 \rightarrow 2^8-1 tương ứng với 0V \rightarrow 5V) bằng hàm

analogWrite(). Nói một cách đơn giản, bạn có thể điều chỉnh được điện áp ra ở chân này từ mức 0V đến 5V thay vì chỉ cố định ở mức 0V và 5V như những chân khác.

Chân giao tiếp SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Ngoài các chức năng thông thường, 4 chân này còn dùng để truyền phát dữ liệu bằng giao thức SPI với các thiết bị khác.

LED 13: trên Arduino UNO có 1 đèn led màu cam (kí hiệu chữ L). Khi bấm nút Reset, bạn sẽ thấy đèn này nhấp nháy để báo hiệu. Nó được nối với chân số 13. Khi chân này được người dùng sử dụng, LED sẽ sáng.

Arduino UNO Broad có 6 chân analog (A0 → A5) cung cấp độ phân giải tín hiệu 10bit ($0 \rightarrow 2^{10}-1$) để đọc giá trị điện áp trong khoảng 0V → 5V. Với chân **AREF** trên board, bạn có thể để đưa vào điện áp tham chiếu khi sử dụng các chân analog. Tức là nếu bạn cấp điện áp 2.5V vào chân này thì bạn có thể dùng các chân analog để đo điện áp trong khoảng từ 0V → 2.5V với độ phân giải vẫn là 10bit.

Đặc biệt, Arduino UNO có 2 chân A4 (SDA) và A5 (SCL) hỗ trợ giao tiếp I2C/TWI với các thiết bị khác.

Lập trình cho Arduino

Các thiết bị dựa trên nền tảng Arduino được lập trình bằng ngôn riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung. Và Wiring lại là một biến thể của C/C++. Một số người gọi nó là Wiring, một số khác thì gọi là C hay C/C++. Riêng mình thì gọi nó là “*ngôn ngữ Arduino*”, và đội ngũ phát triển Arduino cũng gọi như vậy. Ngôn ngữ Arduino bắt nguồn từ C/C++ phổ biến hiện nay do đó rất dễ học, dễ hiểu. Nếu học tốt chương trình Tin học 11 thì việc lập trình Arduino sẽ rất dễ thở đối với bạn.

Để lập trình cho Mạch Arduino, nhà phát triển cung cấp một môi trường lập trình Arduino được gọi là **Arduino**

IDE (Intergrated **D**evelopment **E**nvironment) như hình dưới đây.



Hình 2.4: Giao diện lập trình

2.2 Kit RF Thu Phát Wifi ESP8266 ESP-202



Hình 2.5: Kit ESP8266-12

ESP8266-12 là module wifi giá rẻ và được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác.

ESP8266 là một chip tích hợp cao, được thiết kế cho nhu cầu của một thế giới kết nối mới, thế giới Internet of thing (IOT). Nó cung cấp một giải pháp kết nối mạng Wi-Fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng. ESP8266 có xử lý và khả năng lưu trữ mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với một chi phí tối thiểu và một PCB tối thiểu.

2.2.1 Đặc tính nổi bật

- SDIO 2.0, SPI, UART
- 32-pin QFN (Chip esp8266)
- Tích hợp RF switch, balun, 24dBm PA, DCXO, and PMU
- Tích hợp bộ xử lý RISC, trên chip bộ nhớ và giao diện bộ nhớ bên ngoài
- Tích hợp bộ vi xử lý MAC / baseband
- Chất lượng quản lý dịch vụ
- Giao diện I2S cho độ trung thực cao ứng dụng âm thanh
- On-chip thấp học sinh bỏ học điều chỉnh tuyến tính cho tất cả các nguồn cung cấp nội bộ
- Kiến trúc giả miễn phí thế hệ đồng hồ độc quyền
- Tích hợp WEP, TKIP, AES, và các công cụ WAPI

2.2.2 Thông số kỹ thuật

- Wifi 802.11 b/g/n
- Wi-Fi Direct (P2P), soft-AP
- Tích hợp giao thức TCP / IP stack
- Tích hợp TR chuyển đổi, balun, LNA, bộ khuếch đại quyền lực và phù hợp với mạng
- PLLs tích hợp, quản lý, DCXO và các đơn vị quản lý điện năng
- + Công suất đầu ra 19.5dBm ở chế độ 802.11b
- Tích hợp công suất thấp 32-bit CPU có thể được sử dụng như là bộ vi xử lý ứng dụng
- SDIO 1.1 / 2.0, SPI, UART
- STBC, MIMO 1×1 , 2×1 MIMO
- A-MPDU & A-MSDU tập hợp & 0.4ms khoảng bảo vệ
- Thức dậy và truyền tải các gói dữ liệu trong <2ms
- Chế độ chờ tiêu thụ điện năng <1.0mW (DTIM3)

2.3 Module SIM900A



Hình 2.6 Module SIM900A Mini

Module sim900 mini sử dụng nguồn khoảng 3.7V ~ 4.8V, có thể dùng pin lithium nhưng không được quá 4,8V nếu không các linh kiện điện tử sẽ bị cháy. Về kết nối

Module này sử dụng giao diện TTL, có thể được kết nối trực tiếp với MCU, ARM, mà không cần thiết bị chuyển đổi, và không sử dụng các liên kết máy tính như cổng USB, RS232, RS485 và liên kết nối tiếp khác, module sẽ bị cháy.

Sơ đồ chân của module sim900 mini

- VCC: Nguồn vào 5V.
- TXD: Chân truyền Uart TX.
- RXD: Chân nhận Uart RX.
- Headphone: Chân phát âm thanh.
- Microphone: Chân nhận âm thanh (phải gắn thêm Micro từ GND vào chân này thì mới thu được tiếng).
- GND: Chân Mass, cấp 0V.

HÌNH ẢNH CỦA MODULE SIM900A MINI



Hình 2.7 Mặt trước Module sim900A Mini



Hình 2.8 Mặt sau Module sim900A Mini



Hình 2.9 Các chân của Module sim900A mini

Modul sim900a mini sau khi hàn thêm diot vào chân vcc và tụ 2200uF/10V sẽ sử dụng được nguồn 5v từ mạch arduino

- 5V nối với chân 5V của board Arduino.
- GND nối với chân GND của board Arduino.
- TX nối với chân 51/2 của board Arduino MEGA/UNO.
- RX nối với chân 50/3 của board Arduino MEGA/UNO.
- PWR: Đây là chân bật tắt modul sim900a.
- SPK: Chân này cần kết nối nếu bạn muốn xuất âm thanh ra loa thoại.
- MIC: Chân này cần kết nối nếu bạn muốn tạo mic để đàm thoại.

2.4 Giới thiệu về màn hình LCD 16x2:

Hình dáng và kích thước:

Có rất nhiều loại LCD với nhiều hình dáng và kích thước khác nhau, trên hình 1 là loại LCD thông dụng.



Hình 2.10 : Hình dáng của loại LCD thông dụng

Khi sản xuất LCD, nhà sản xuất đã tích hợp chip điều khiển (HD44780) bên trong lớp vỏ và chỉ đưa các chân giao tiếp cần thiết.

Chức năng các chân :

Bảng 2.1 : Chức năng các chân của LCD

Chân	Ký hiệu	Mô tả
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển
3	VEE	Điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.
7 - 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này :

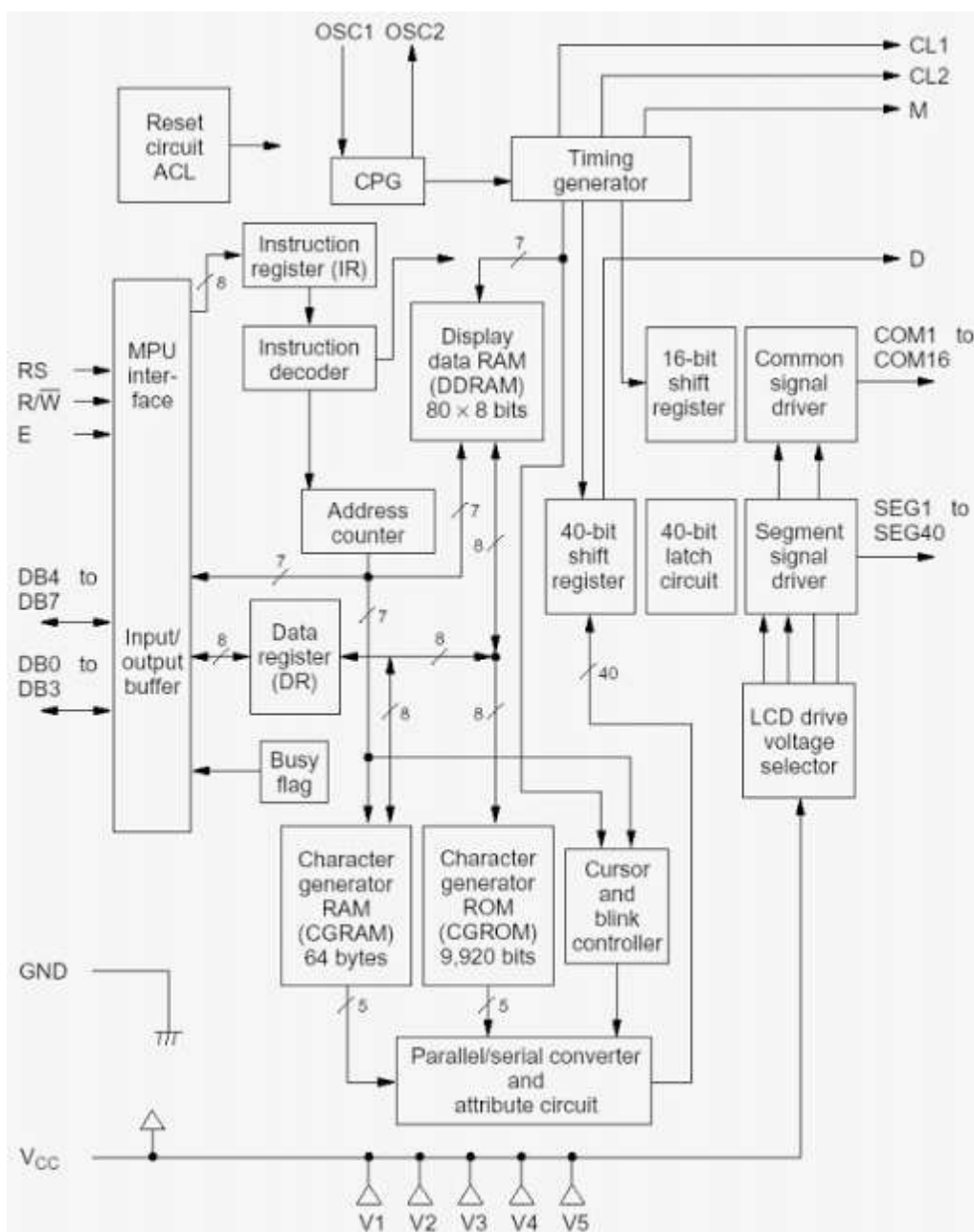
		+ Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7
15	-	Nguồn dương cho đèn nền
16	-	GND cho đèn nền

* Ghi chú : Ở chế độ “đọc”, nghĩa là MPU sẽ đọc thông tin từ LCD thông qua các chân DBx.

Còn khi ở chế độ “ghi”, nghĩa là MPU xuất thông tin điều khiển cho LCD thông qua các chân DBx.

Sơ đồ khối của HD44780:

Để hiểu rõ hơn chức năng các chân và hoạt động của chúng, ta tìm hiểu sơ qua chip HD44780 thông qua các khối cơ bản của nó.



Hình 2.11 : Sơ đồ khối của HD44780

Các thanh ghi :

Chip HD44780 có 2 thanh ghi 8 bit quan trọng : Thanh ghi lệnh IR (Instructor Register) và thanh ghi dữ liệu DR (Data Register)

- Thanh ghi IR : Để điều khiển LCD, người dùng phải “ra lệnh” thông qua tám đường bus DB0-DB7. Mỗi lệnh được nhà sản xuất LCD đánh địa chỉ rõ ràng. Người dùng chỉ việc cung cấp địa chỉ lệnh bằng cách

nạp vào thanh ghi IR. Nghĩa là, khi ta nạp vào thanh ghi IR một chuỗi 8 bit, chip HD44780 sẽ tra bảng mã lệnh tại địa chỉ mà IR cung cấp và thực hiện lệnh đó.

VD : Lệnh “hiển thị màn hình” có địa chỉ lệnh là 00001100
(DB7...DB0)

Lệnh “hiển thị màn hình và con trỏ” có mã lệnh là 00001110

- Thanh ghi DR : Thanh ghi DR dùng để chứa dữ liệu 8 bit để ghi vào vùng RAM DDRAM hoặc CGRAM

(ở chế độ ghi) hoặc dùng để chứa dữ liệu từ 2 vùng RAM này gửi ra cho MPU (ở chế độ đọc). Nghĩa là, khi MPU ghi thông tin vào DR, mạch nội bên trong chip sẽ tự động ghi thông tin này vào DDRAM hoặc CGRAM. Hoặc khi thông tin về địa chỉ được ghi vào IR, dữ liệu ở địa chỉ này trong vùng RAM nội của HD44780 sẽ được chuyển ra DR để truyền cho MPU. => Bằng cách điều khiển chân RS và R/W chúng ta có thể chuyển qua lại giữa 2 thanh ghi này khi giao tiếp với MPU. Bảng sau đây tóm tắt lại các thiết lập đối với hai chân RS và R/W theo mục đích giao tiếp.

Bảng 2.2 : Chức năng chân RS và R/W theo mục đích sử dụng

RS	R/W	Chức năng
0	0	Ghi vào thanh ghi IR để ra lệnh cho LCD
0	1	Đọc cờ bận ở DB7 và giá trị của bộ đếm địa chỉ ở DB0-DB6
1	0	Ghi vào thanh ghi DR
1	1	Đọc dữ liệu từ DR

Cờ báo bận BF: (Busy Flag)

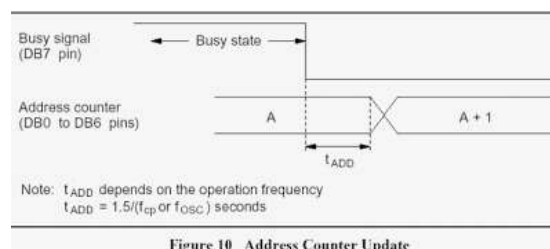
Khi thực hiện các hoạt động bên trong chip, mạch nội bên trong cần một khoảng thời gian để hoàn tất. Khi đang thực thi các hoạt động bên trong chip như thế, LCD bỏ qua mọi giao tiếp với bên ngoài và bật cờ BF (thông qua chân DB7 khi có thiết lập RS=0, R/W=1) lên để báo cho MPU biết nó đang “bận”. Dĩ nhiên, khi xong việc, nó sẽ đặt cờ BF lại mức 0.

Bộ đếm địa chỉ AC : (Address Counter)

Như trong sơ đồ khối, thanh ghi IR không trực tiếp kết nối với vùng RAM (DDRAM và CGRAM) mà thông qua bộ đếm địa chỉ AC. Bộ đếm này lại nối với 2 vùng RAM theo kiểu rẽ nhánh. Khi một địa chỉ lệnh được nạp vào thanh ghi IR, thông tin được nối trực tiếp cho 2 vùng RAM nhưng việc chọn lựa vùng RAM tương tác đã được bao hàm trong mã lệnh.

Sau khi ghi vào (đọc từ) RAM, bộ đếm AC tự động tăng lên (giảm đi) 1 đơn vị và nội dung của AC được xuất ra cho MPU thông qua DB0-DB6 khi có thiết lập RS=0 và R/W=1 (xem bảng tóm tắt RS - R/W).

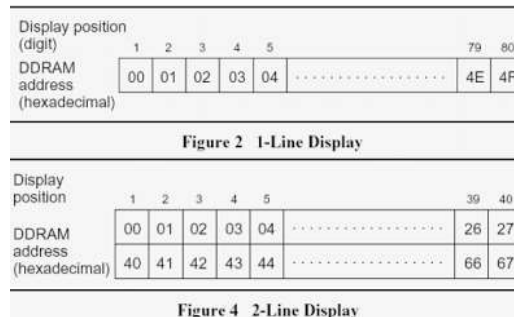
Lưu ý: Thời gian cập nhật AC không được tính vào thời gian thực thi lệnh mà được cập nhật sau khi cờ BF lên mức cao (not busy), cho nên khi lập trình hiển thị, bạn phải delay một khoảng t_{ADD} khoảng 4 μ S-5 μ S (ngay sau khi BF=1) trước khi nạp dữ liệu mới. Xem thêm hình bên dưới.



Hình 2.12 : Giản đồ xung cập nhật AC

Vùng RAM hiển thị DDRAM : (Display Data RAM)

Đây là vùng RAM dùng để hiển thị, nghĩa là ứng với một địa chỉ của RAM là một ô kí tự trên màn hình và khi bạn ghi vào vùng RAM này một mã 8 bit, LCD sẽ hiển thị tại vị trí tương ứng trên màn hình một kí tự có mã 8 bit mà bạn đã cung cấp. Hình sau đây sẽ trình bày rõ hơn mối liên hệ này :



Hình 2.13 : Mối liên hệ giữa địa chỉ của DDRAM và vị trí hiển thị của LCD

Vùng RAM này có 80x8 bit nhớ, nghĩa là chứa được 80 kí tự mã 8 bit. Những vùng RAM còn lại không dùng cho hiển thị có thể dùng như vùng RAM đa mục đích.

Lưu ý là để truy cập vào DDRAM, ta phải cung cấp địa chỉ cho AC theo mã HEX

Vùng ROM chứa kí tự CGROM: Character Generator ROM

Vùng ROM này dùng để chứa các mẫu kí tự loại 5x8 hoặc 5x10 điểm ảnh/kí tự, và định địa chỉ bằng 8 bit. Tuy nhiên, nó chỉ có 208 mẫu kí tự 5x8 và 32 mẫu kí tự kiểu 5x10 (tổng cộng là 240 thay vì 2⁸ = 256 mẫu kí tự). Người dùng không thể thay đổi vùng ROM này.

Table 2 Example of Correspondence between EPROM Address Data and Character Pattern (5 × 8 Bits)

EPROM Address								Data								
A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	1	0	1	1	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
0	1	1	0	0	1	0	0	1	1	1	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Character code: Line position

Notes: 1. EPROM addresses A11 to A4 correspond to a character code.
 2. EPROM addresses A3 to A0 specify a line position of the character pattern.
 3. EPROM data D4 to D0 correspond to character pattern data.
 4. EPROM data D5 to D7 must be specified as 0.
 5. A lit display position (black) corresponds to a 1.
 6. Line 0 and the following lines must be blanked with 0s for a 5 × 8-dot character fonts.

Hình 2.14 : Mối liên hệ giữa địa chỉ của ROM và dữ liệu tạo mẫu kí tự.

Như vậy, để có thể ghi vào vị trí thứ x trên màn hình một kí tự y nào đó, người dùng phải ghi vào vùng DDRAM tại địa chỉ x (xem bảng mối liên hệ giữa DDRAM và vị trí hiển thị) một chuỗi mã kí tự 8 bit trên CGROM. Chú ý là trong bảng mã kí tự trong CGROM ở hình bên dưới có mã ROM A00.

Ví dụ : Ghi vào DDRAM tại địa chỉ “01” một chuỗi 8 bit “01100010” thì trên LCD tại ô thứ 2 từ trái sang (dòng trên) sẽ hiển thị kí tự “b”.

Bảng 2.3 : Bảng mã kí tự (ROM code A00)

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Code	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000			0	@P`P									-	タ	ミ	αp
xxxx0001	(2)	!	1	AQa9									。	ア	チ	4 äq
xxxx0010	(3)	"	2	BRbr									「	イ	ツ	× p θ
xxxx0011	(4)	#	3	CScs									」	ウ	テ	モ ε ∞
xxxx100	(5)	\$	4	DTdt									、	エ	ト	μ Ω
xxxx101	(6)	%	5	EUeu									・	オ	ナ	1 0 Ü
xxxx110	(7)	&	6	FUfu									ヲ	カ	ニ	ヨ ρ Σ
xxxx111	(8)	'	7	GWgw									ア	キ	ヌ	ラ q π
xxxx500	(1)	<	8	HXhx									イ	ク	ネ	リ J X
xxxx1001	(2)	>	9	IYiy									ウ	ケ	ル	・ y
xxxx1010	(3)	*	:	JZjz									エ	コ	ン	レ j 千
xxxx1011	(4)	+	;	K[k<									オ	サ	ヒ	ロ * 万
xxxx1100	(5)	,	<	L¥ll									カ	シ	フ	ワ 4 円
xxxx1101	(6)	-	=	M]m}									ユ	ス	ヘ	ン も ÷
xxxx1110	(7)	.	>	N^n→									ヨ	セ	ホ	° 万
xxxx1111	(8)	/	?	0_o←									ウ	ツ	マ	° 0

Note: The user can specify any pattern for character-generator RAM.

Vùng RAM chứa kí tự đồ họa CGRAM : (Character Generator RAM)

Như trên bảng mã kí tự, nhà sản xuất dành vùng có địa chỉ byte cao là 0000 để người dùng có thể tạo các mẫu kí tự đồ họa riêng. Tuy nhiên dung lượng vùng này rất hạn chế: Ta chỉ có thể tạo 8 kí tự loại 5x8 điểm ảnh, hoặc 4 kí tự loại 5x10 điểm ảnh. Để ghi vào CGRAM, hãy xem hình 6 bên dưới.

Table 5 Relationship between CGRAM Addresses, Character Codes (DDRAM) and Character Patterns (CGRAM Data)

For 5 × 8 dot character patterns

Character Codes (DDRAM data)		CGRAM Address		Character Patterns (CGRAM data)	
7 6 5 4 3 2 1 0		5 4 3 2 1 0		7 6 5 4 3 2 1 0	
High	Low	High	Low	High	Low
			0 0 0		1 1 1 1 0
			0 0 1		1 0 0 0 1
			0 1 0		1 0 0 0 1
			0 1 1		1 1 1 1 0
0 0 0 0 *	0 0 0	0 0 0	1 0 0		1 0 1 0 0
			1 0 1		1 0 0 1 0
			1 1 0		1 0 0 0 1
			1 1 1		0 0 0 0 0
			0 0 0		1 0 0 0 1
			0 0 1		0 1 1 1 0
			0 1 0		1 1 1 1 0
0 0 0 0 *	0 0 1	0 0 1	1 0 0		0 0 1 0 0
			1 0 1		0 0 1 0 0
			1 1 0		0 0 1 0 0
			1 1 1		0 0 0 0 0
			0 0 0		1 0 0 0 1
			0 0 1		0 1 1 1 0
			0 1 0		1 1 1 1 0
0 0 0 0 *	1 1 1	1 1 1	1 0 0		0 0 1 0 0
			1 0 1		0 0 1 0 0
			1 1 0		0 0 1 0 0
			1 1 1		0 0 0 0 0

Note:

- Character code bits 0 to 2 correspond to CGRAM address bits 3 to 5 (3 bits: 8 types).
- CGRAM address bits 0 to 2 designate the character pattern line position. The 8th line is the cursor position and its display is formed by a logic OR with the cursor. Maintain the 8th line data, corresponding to the cursor display position, at 0 as the cursor display if the 8th line data is 1. 1 bits will light up the 8th line regardless of the cursor presence.
- Character pattern row positions correspond to CGRAM data bits 0 to 4 (bit 4 being at the left).
- As shown Table 5, CGRAM character patterns are selected when character code bits 4 to 7 are all 0. However, since character code bit 3 has no effect, the R display example above can be selected by either character code 00H or 08H.
- 1 for CGRAM data corresponds to display selection and 0 to non-selection.

* Indicates no effect.

Hình 2.15 : Mối liên hệ giữa địa chỉ của CGRAM, dữ liệu của CGRAM, và mã kí tự.

Tập lệnh của LCD :

Trước khi tìm hiểu tập lệnh của LCD, sau đây là một vài chú ý khi giao tiếp với LCD :

- * Tuy trong sơ đồ khối của LCD có nhiều khối khác nhau, nhưng khi lập trình điều khiển LCD ta chỉ có thể tác động trực tiếp được vào 2 thanh ghi DR và IR thông qua các chân DBx, và ta phải thiết lập chân RS, R/W phù hợp để chuyển qua lại giữ 2 thanh ghi này. (xem bảng 2)
- * Với mỗi lệnh, LCD cần một khoảng thời gian để hoàn tất, thời gian này có thể khá lâu đối với tốc độ của MPU, nên ta cần kiểm tra cờ BF hoặc đợi (delay) cho LCD thực thi xong lệnh hiện hành mới có thể ra lệnh tiếp theo.
- * Địa chỉ của RAM (AC) sẽ tự động tăng (giảm) 1 đơn vị, mỗi khi có lệnh ghi vào RAM. (Điều này giúp chương trình gọn hơn)
- * Các lệnh của LCD có thể chia thành 4 nhóm như sau :

- Các lệnh về kiểu hiển thị. VD : Kiểu hiển thị (1 hàng / 2 hàng), chiều dài dữ liệu (8 bit / 4 bit), ...
- Chỉ định địa chỉ RAM nội.
- Nhóm lệnh truyền dữ liệu trong RAM nội.
- Các lệnh còn lại .

Bảng 2.4 : Tập lệnh của LCD

Tên lệnh	Hoạt động
Clear Display	<p>Mã lệnh : DB_x = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DB_x = 0 0 0 0 0 0 0 1</p> <p>Lệnh Clear Display (xóa hiển thị) sẽ ghi một khoảng trống-blank (mã hiện kí tự 20H) vào tất cả ô nhớ trong DDRAM, sau đó trả bộ đếm địa AC=0, trả lại kiểu hiển thị gốc nếu nó bị thay đổi. Nghĩa là : Tắt hiển thị, con trỏ dời về góc trái (hàng đầu tiên), chế độ tăng AC.</p>
Return home	<p>Mã lệnh : DB_x = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DB_x = 0 0 0 0 0 0 1 *</p> <p>Lệnh Return home trả bộ đếm địa chỉ AC về 0, trả lại kiểu hiển thị gốc nếu nó bị thay đổi. Nội dung của DDRAM không thay đổi.</p>
Entry mode set	<p>Mã lệnh : DB_x = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DB_x = 0 0 0 0 0 1 [I/D] [S]</p> <p>I/D : Tăng (I/D=1) hoặc giảm (I/D=0) bộ đếm địa chỉ hiển thị AC 1 đơn vị mỗi khi có hành động ghi hoặc đọc vùng DDRAM. Vị trí con trỏ cũng di chuyển theo sự tăng giảm này.</p> <p>S : Khi S=1 toàn bộ nội dung hiển thị bị dịch sang phải (I/D=0) hoặc sang trái (I/D=1) mỗi khi có hành động ghi vùng DDRAM. Khi S=0: không dịch nội dung hiển thị. Nội dung hiển thị không dịch khi đọc DDRAM hoặc đọc/ghi vùng CGRAM.</p>
Display on/off control	<p>Mã lệnh : DB_x = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DB_x = 0 0 0 0 1 [D] [C] [B]</p> <p>D: Hiển thị màn hình khi D=1 và ngược lại. Khi tắt hiển thị, nội dung</p>

	<p>DDRAM không thay đổi.</p> <p>C: Hiển thị con trỏ khi C=1 và ngược lại.</p> <p>B: Nhấp nháy kí tự tại vị trí con trỏ khi B=1 và ngược lại.</p> <p>Chu kì nhấp nháy khoảng 409,6ms khi mạch dao động nội LCD là 250kHz.</p>															
<p>Cursor or display shift</p>	<p>Mã lệnh : DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 0 0 1 [S/C] [R/L] * *</p> <p>Lệnh Cursor or display shift dịch chuyển con trỏ hay dữ liệu hiển thị sang trái mà không cần hành động ghi/đọc dữ liệu. Khi hiển thị kiểu 2 dòng, con trỏ sẽ nhảy xuống dòng dưới khi dịch qua vị trí thứ 40 của hàng đầu tiên. Dữ liệu hàng đầu và hàng 2 dịch cùng một lúc. Chi tiết sử dụng xem bảng bên dưới:</p> <table border="1"> <thead> <tr> <th>S/C</th> <th>R/L</th> <th>Hoạt động</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).</td> </tr> <tr> <td>0</td> <td>1</td> <td>Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).</td> </tr> <tr> <td>1</td> <td>0</td> <td>Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.</td> </tr> <tr> <td>1</td> <td>1</td> <td>Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.</td> </tr> </tbody> </table>	S/C	R/L	Hoạt động	0	0	Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).	0	1	Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).	1	0	Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.	1	1	Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.
S/C	R/L	Hoạt động														
0	0	Dịch vị trí con trỏ sang trái (Nghĩa là giảm AC một đơn vị).														
0	1	Dịch vị trí con trỏ sang phải (Tăng AC lên 1 đơn vị).														
1	0	Dịch toàn bộ nội dung hiển thị sang trái, con trỏ cũng dịch theo.														
1	1	Dịch toàn bộ nội dung hiển thị sang phải, con trỏ cũng dịch theo.														
<p>Function set</p>	<p>Mã lệnh : DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DBx = 0 0 1 [DL] [N] [F] * *</p> <p>DL: Khi DL=1, LCD giao tiếp với MPU bằng giao thức 8 bit (từ bit DB7 đến DB0). Ngược lại, giao thức giao tiếp là 4 bit (từ bit DB7 đến bit DB0). Khi chọn giao thức 4 bit, dữ liệu được truyền/nhận 2 lần liên tiếp. với 4 bit cao gửi/nhận trước, 4 bit thấp gửi/nhận sau.</p> <p>N : Thiết lập số hàng hiển thị. Khi N=0 : hiển thị 1 hàng, N=1: hiển thị 2 hàng.</p> <p>F : Thiết lập kiểu kí tự. Khi F=0: kiểu kí tự 5x8 điểm ảnh, F=1: kiểu kí tự 5x10 điểm ảnh.</p>															
<p>Set</p>	<p>Mã lệnh : DBx = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p>															

CGRAM address	$DBx = 0 \quad 1 \quad [ACG][ACG][ACG][ACG][ACG][ACG]$ <p>Lệnh này ghi vào AC địa chỉ của CGRAM. Kí hiệu [ACG] chỉ 1 bit của chuỗi dữ liệu 6 bit. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ CGRAM tại địa chỉ đã được chỉ định.</p>
Set DDRAM address	<p>Mã lệnh : $DBx = DB7 \quad DB6 \quad DB5 \quad DB4 \quad DB3 \quad DB2 \quad DB1 \quad DB0$</p> $DBx = 1 \quad [AD] \quad [AD] \quad [AD] \quad [AD] \quad [AD] \quad [AD] \quad [AD]$ <p>Lệnh này ghi vào AC địa chỉ của DDRAM, dùng khi cần thiết lập tọa độ hiển thị mong muốn. Ngay sau lệnh này là lệnh đọc/ghi dữ liệu từ DDRAM tại địa chỉ đã được chỉ định.</p> <p>Khi ở chế độ hiển thị 1 hàng: địa chỉ có thể từ 00H đến 4FH. Khi ở chế độ hiển thị 2 hàng, địa chỉ từ 00h đến 27H cho hàng thứ nhất, và từ 40h đến 67h cho hàng thứ 2.</p>
Read BF and address	<p>Mã lệnh : $DBx = DB7 \quad DB6 \quad DB5 \quad DB4 \quad DB3 \quad DB2 \quad DB1 \quad DB0$</p> $DBx = [BF] \quad [AC] \quad [AC] \quad [AC] \quad [AC] \quad [AC] \quad [AC] \quad [AC]$ <p>(RS=0,R/W=1)</p> <p>Như đã đề cập trước đây, khi cờ BF bật, LCD đang làm việc và lệnh tiếp theo (nếu có) sẽ bị bỏ qua nếu cờ BF chưa về mức thấp. Cho nên, khi lập trình điều khiển, phải kiểm tra cờ BF trước khi ghi dữ liệu vào LCD.</p> <p>Khi đọc cờ BF, giá trị của AC cũng được xuất ra các bit [AC]. Nó là địa chỉ của CG hay DDRAM là tùy thuộc vào lệnh trước đó.</p>
Write data to CG or DDRAM	<p>Mã lệnh : $DBx = DB7 \quad DB6 \quad DB5 \quad DB4 \quad DB3 \quad DB2 \quad DB1 \quad DB0$</p> $DBx = \quad \quad \quad [Write \ data] \quad \quad \quad (RS=1,$ <p>R/W=0)</p> <p>Khi thiết lập RS=1, R/W=0, dữ liệu cần ghi được đưa vào các chân DBx từ mạch ngoài sẽ được LCD chuyển vào trong LCD tại địa chỉ được xác định từ lệnh ghi địa chỉ trước đó (lệnh ghi địa chỉ cũng xác định luôn vùng RAM cần ghi)</p>

	Sau khi ghi, bộ đếm địa chỉ AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode.
Read data from CG or DDRAM	<p>Mã lệnh : DB_x = DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0</p> <p>DB_x = [Read data] (RS=1, R/W=1)</p> <p>Khi thiết lập RS=1, R/W=1, dữ liệu từ CG/DDRAM được chuyển ra MPU thông qua các chân DB_x (địa chỉ và vùng RAM đã được xác định bằng lệnh ghi địa chỉ trước đó).</p> <p>Sau khi đọc, AC tự động tăng/giảm 1 tùy theo thiết lập Entry mode, tuy nhiên nội dung hiển thị không bị dịch bất chấp chế độ Entry mode.</p>

Giao tiếp giữa LCD và MPU :**Đặc tính điện của các chân giao tiếp :**

LCD sẽ bị hỏng nghiêm trọng, hoặc hoạt động sai lệch nếu bạn vi phạm khoảng đặc tính điện sau đây:

Bảng 2.5 : Maximun Rating

Chân cấp nguồn (Vcc-GND)	Min:-0.3V , Max+7V
Các chân ngõ vào (DBx,E,...)	Min:-0.3V , Max:(Vcc+0.3V)
Nhiệt độ hoạt động	Min:-30C , Max:+75C
Nhiệt độ bảo quản	Min:-55C , Max:+125C

Đặc tính điện làm việc điển hình: (Đo trong điều kiện hoạt động Vcc = 4.5V đến 5.5V, T = -30 đến +75C)

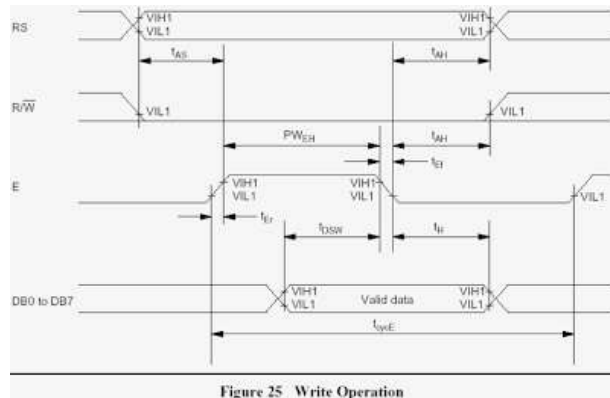
Bảng 2.6: Miền làm việc bình thường

Chân cấp nguồn Vcc-GND	2.7V đến 5.5V
Điện áp vào mức cao VIH	2.2V đến Vcc
Điện áp vào mức thấp VIL	-0.3V đến 0.6V
Điện áp ra mức cao (DB0-DB7)	Min 2.4V (khi IOH = -0.205mA)
Điện áp ra mức thấp (DB0-DB7)	Max 0.4V (khi IOL = 1.2mA)
Dòng điện ngõ vào (input leakage current) ILI	-1uA đến 1uA (khi VIN = 0 đến Vcc)
Dòng điện cấp nguồn ICC	350uA(typ.) đến 600uA
Tần số dao động nội fOSC	190kHz đến 350kHz (điển hình là 270kHz)

Sơ đồ nối mạch điển hình:

- Sơ đồ mạch kết nối giữa mô đun LCD và VDK 89S52 (8 bit).
- Sơ đồ mạch kết nối giữa mô đun LCD và VDK (4 bit).

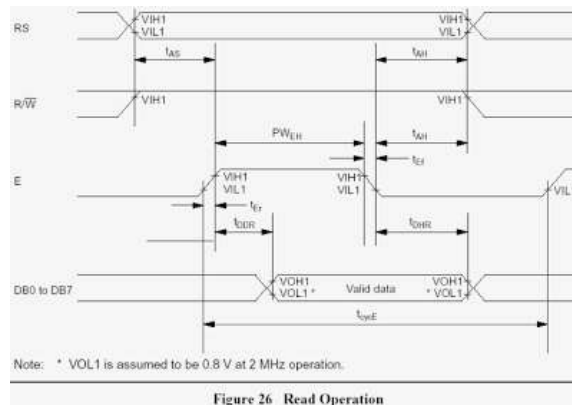
Bus Timing:



Hình 2.16 Bus Timing

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	t_{cycE}	500	—	—	ns
Enable pulse width (high level)	PW_{EH}	230	—	—	
Enable rise/fall time	t_{Er}, t_{Ef}	—	—	20	
Address set-up time (RS, R/W to E)	t_{AS}	40	—	—	
Address hold time	t_{AH}	10	—	—	
Data set-up time	t_{DSW}	80	—	—	
Data hold time	t_{Hf}	10	—	—	

Hình 2.17



Hình 2.18

Item	Symbol	Min	Typ	Max	Unit
Enable cycle time	t_{cyE}	500	—	—	ns
Enable pulse width (high level)	PW_{EH}	230	—	—	
Enable rise/fall time	$t_{Er} t_{Ef}$	—	—	20	
Address set-up time (RS, R/W to E)	t_{AS}	40	—	—	
Address hold time	t_{AH}	10	—	—	
Data delay time	t_{DDR}	—	—	160	
Data hold time	t_{DHR}	5	—	—	

Hình 2.19

Khởi tạo LCD:

Khởi tạo là việc thiết lập các thông số làm việc ban đầu. Đối với LCD, khởi tạo giúp ta thiết lập các giao thức làm việc giữa LCD và MPU. Việc khởi tạo chỉ được thực hiện 1 lần duy nhất ở đầu chương trình điều khiển LCD và bao gồm các thiết lập sau :

- Display clear : Xóa/không xóa toàn bộ nội dung hiển thị trước đó.
- Function set : Kiểu giao tiếp 8bit/4bit, số hàng hiển thị 1hàng/2hàng, kiểu kí tự 5x8/5x10.
- Display on/off control: Hiển thị/tắt màn hình, hiển thị/tắt con trỏ, nhấp nháy/không nhấp nháy.
- Entry mode set : các thiết lập kiểu nhập kí tự như: Dịch/không dịch, tự tăng/giảm (Increment).

Mạch khởi tạo bên trong chip HD44780:

Mỗi khi được cấp nguồn, mạch khởi tạo bên trong LCD sẽ tự động khởi tạo cho nó. Và trong thời gian khởi tạo này cờ BF bật lên 1, đến khi việc khởi tạo hoàn tất cờ BF còn giữ trong khoảng 10ms sau khi Vcc đạt đến 4.5V (vì 2.7V thì LCD đã hoạt động). Mạch khởi tạo nội sẽ thiết lập các thông số làm việc của LCD như sau:

- Display clear : Xóa toàn bộ nội dung hiển thị trước đó.
- Function set: DL=1 : 8bit; N=0 : 1 hàng; F=0 : 5x8
- Display on/off control: D=0 : Display off; C=0 : Cursor off; B=0 : Blinking off.
- Entry mode set: I/D =1 : Tăng; S=0 : Không dịch.

Như vậy sau khi mở nguồn, bạn sẽ thấy màn hình LCD giống như chưa mở nguồn do toàn bộ hiển thị tắt. Do đó, ta phải khởi tạo LCD bằng lệnh.

Khởi tạo bằng lệnh: (chuỗi lệnh)

Việc khởi tạo bằng lệnh phải tuân theo lưu đồ sau của nhà sản xuất :

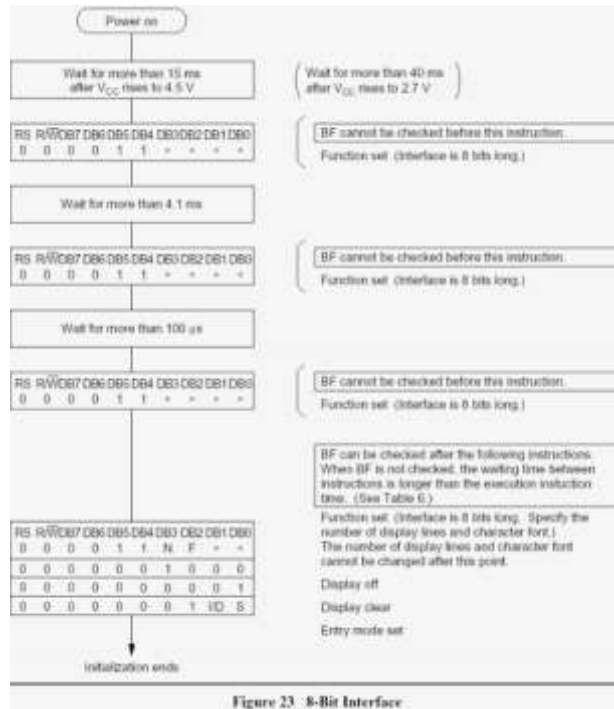


Figure 23 - 8-Bit Interface

Hình 2.20 Hình chuỗi lệnh

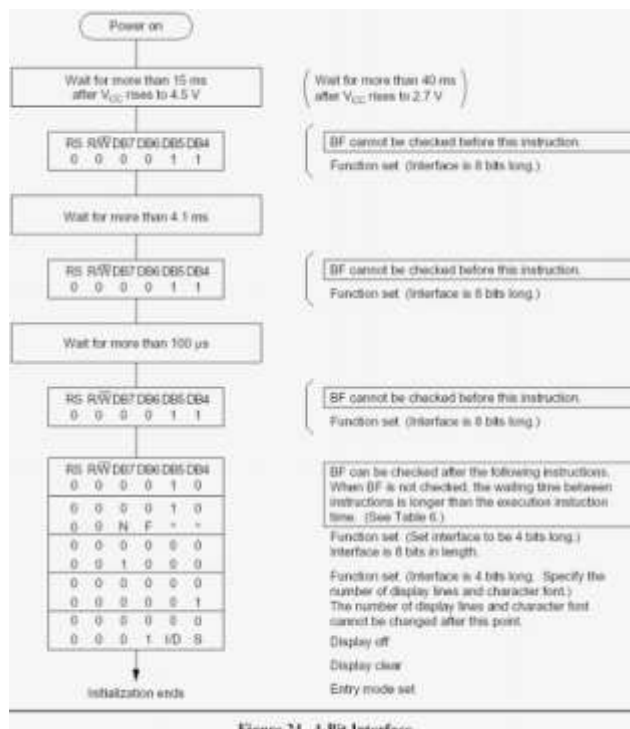


Figure 24 - 4-Bit Interface

Hình 2.21 Chuỗi lệnh

Như đã đề cập ở trên, chế độ giao tiếp mặc định của LCD là 8bit (tự khởi tạo lúc mới bật điện lên). Và khi kết nối mạch theo giao thức 4bit, 4 bit thấp từ DB0-DB3 không được kết nối đến LCD, nên lệnh khởi tạo ban đầu (lệnh chọn giao thức giao tiếp – function set 0010****) phải giao tiếp theo chế độ 8 bit (chỉ gửi 4 bit cao một lần, bỏ qua 4 bit thấp). Từ lệnh sau trở đi, phải gửi/nhận lệnh theo 2 nibble.

Lưu ý là sau khi thiết lập function set, bạn không thể thay đổi function set ngoại trừ thay đổi giao thức giao tiếp (4bit/8bit).

2.5 Module chuyển đổi I2C cho LCD10602



Hình 2.22 Module I2C

I2C sử dụng hai đường truyền tín hiệu:

- Một đường xung nhịp đồng hồ(SCL) chỉ do Master phát đi (thông thường ở 100kHz và 400kHz. Mức cao nhất là 1Mhz và 3.4MHz).
- Một đường dữ liệu(SDA) theo 2 hướng.

Có rất nhiều thiết bị có thể cùng được kết nối vào một bus I2C, tuy nhiên sẽ không xảy ra chuyện nhầm lẫn giữa các thiết bị, bởi mỗi thiết bị sẽ được nhận ra bởi một địa chỉ duy nhất với một quan hệ chủ/tớ tồn tại trong suốt thời gian kết nối. Mỗi thiết bị có thể hoạt động như là thiết bị nhận hoặc truyền dữ liệu hay có thể vừa truyền vừa nhận. Hoạt động truyền hay nhận còn tùy thuộc vào việc thiết bị đó là chủ (master) hay tớ (slave).

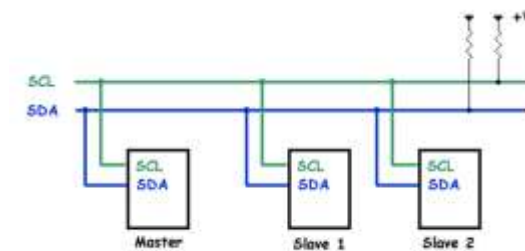
Một thiết bị hay một IC khi kết nối với bus I2C, ngoài một địa chỉ (duy nhất) để phân biệt, nó còn được cấu hình là thiết bị chủ hay tớ. Tại sao lại có sự phân biệt này ? Đó là vì trên một bus I2C thì quyền điều khiển thuộc về thiết bị chủ. Thiết bị chủ nắm vai trò tạo xung đồng hồ cho toàn hệ thống, khi giữa hai thiết bị chủ-tớ giao tiếp thì thiết bị chủ có nhiệm vụ tạo xung đồng hồ và quản lý địa chỉ của thiết bị tớ trong suốt quá trình giao tiếp. Thiết bị chủ giữ vai trò chủ động, còn thiết bị tớ giữ vai trò bị động trong việc giao tiếp.

Về lý thuyết lẫn thực tế I2C sử dụng 7 bit để định địa chỉ, do đó trên một bus có thể có tới 2^7 địa chỉ tương ứng với 128 thiết bị có thể kết nối, nhưng chỉ có

112 , 16 địa chỉ còn lại được sử dụng vào mục đích riêng. Bit còn lại quy định việc đọc hay ghi dữ liệu (1 là write, 0 là read)

Điểm mạnh của I2C chính là hiệu suất và sự đơn giản của nó: một khối điều khiển trung tâm có thể điều khiển cả một mạng thiết bị mà chỉ cần hai lõi ra điều khiển.

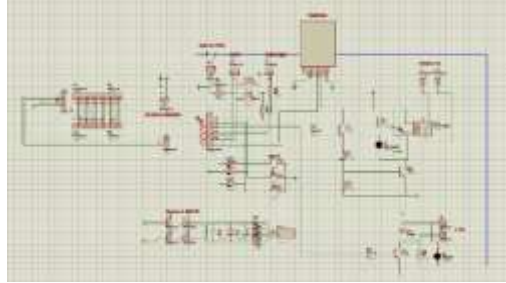
Ngoài ra I2C còn có chế độ 10bit địa chỉ tương đương với 1024 địa chỉ, tương tự như 7 bit, chỉ có 1008 thiết bị có thể kết nối, còn lại 16 địa chỉ sẽ dùng để sử dụng mục đích riêng.



Hình 2.23 Chế độ bit

Chương 3: THIẾT KẾ MẠCH VÀ CHƯƠNG TRÌNH

3.1 Thiết kế mạch trên Proteus:

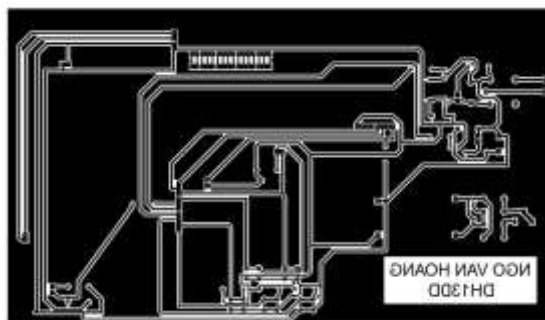


Hình 3.1 Sơ đồ mạch nguyên lí

3.2 Sơ đồ mạch in:



Hình 3.2 Sơ đồ mạch in



Hình 3.3 Mạch in

3.3 Phần viết chương trình:

```
#include "SIM900.h"
#include <SoftwareSerial.h>
#include "sms.h"
#include "call.h"
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x3F,16,2);
CallGSM call;
MSGSMS sms;
char number[20];
byte stat = 0;
char value_str[5];
int trangthai = 0;
int button1 = 4;
int dongco = 7;
int loa = 8;
int cambien = 6;
int button2 = 5;
int mode = 11;
int Up = 10;
int Down = 9;
int x=0;
int y=0;
int z=0;
int dem=0;
int modeState;
int lastmodeState;
int UpState;
int DownState;
int lastUpState;
```

```
int lastDownState;
int button1Status;
int button2Status;
int cambienStatus;

void setup()
{
  lcd.init();
  lcd.backlight();
  pinMode(7, OUTPUT);
  pinMode(button1, INPUT); //Cài đặt chân a0 ở trạng thái đọc dữ liệu
  pinMode(dongco,OUTPUT);
  pinMode(cambien, INPUT);
  pinMode(button2, INPUT);
  pinMode(loa,OUTPUT);
  pinMode(mode, INPUT);
  pinMode(Up, INPUT);
  Serial.begin(9600);
  Serial.println("GSM Shield testing.");
  if (gsm.begin(2400))
    Serial.println("\nstatus=READY");
  else Serial.println("\nstatus=IDLE");
}

void setmode()
{
  UpState = digitalRead(Up);
  if ((UpState != lastUpState)&(UpState ==1))
  {
    digitalWrite(loa,1);
  }
}
```

```
delay(50);
digitalWrite(loi,0);
    y=y+1;
    lcd.clear();
    lcd.setCursor(3, 1);
    lcd.print(y);
    lcd.setCursor(5, 1);
    lcd.print("Phut");
}
lastUpState = UpState;
DownState = digitalRead(Down);
if ((DownState != lastDownState)&(DownState ==1))
{
    digitalWrite(loi,1);
    delay(50);
    digitalWrite(loi,0);
    y=y-1;
    if (y<0) y=0;
    lcd.clear();
    lcd.setCursor(3, 1);
    lcd.print(y);
    lcd.setCursor(5, 1);
    lcd.print("Phut");
}
lastDownState = DownState;
z=y*60;}
void loop()
{
    if (x % 2 ==0)
    {
```

```
lcd.setCursor(1, 0);
lcd.print("SETUP Time Chay");
lcd.setCursor(3, 1);
lcd.print(y);
lcd.setCursor(5, 1);
lcd.print("Phut");
setmode();
}
button1Status = digitalRead(button1);
if (button1Status == 1) trangthai = 1;
modeState = digitalRead(mode);
if ((modeState != lastmodeState)&(modeState == 1))
{
digitalWrite(10a,1);
delay(50);
digitalWrite(10a,0);
x=x+1;
}
lastmodeState = modeState;
if (x % 2 !=0)
{
stat = call.CallStatusWithAuth(number, 1, 5);
if (stat == CALL_INCOM_VOICE_AUTH) {
call.HangUp();
if (trangthai == 0)
{
//sms.SendSMS(number, "Da bat");
trangthai = 1;
} else
{
```

```
    if (trangthai == 1)
    {
        //sms.SendSMS(number, "Da tat");
        trangthai = 0;
    }
}
if (trangthai == 0)
{cambienStatus = digitalRead(cambien);
  if (cambienStatus == 0)
  {lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("MAY BOM TAT");
    lcd.setCursor(2, 1);
    lcd.print("BON HET NUOC");
    int buttonStatus = digitalRead(loa);
    digitalWrite(loa,1);
    delay(50);
    digitalWrite(loa,0);
    delay(50);}
  else
  {
    lcd.clear();
    lcd.setCursor(2, 0);
    lcd.print("MAY BOM TAT");}}
else
{
  if (trangthai == 1)
  {
    lcd.setCursor(2, 0);
```

```
lcd.print("MAY BOM CHAY");
for (dem = 0; dem <= z; dem++) {
  cambienStatus = digitalRead(cambien);
  if (cambienStatus == 0)
  { digitalWrite(7, LOW);
    break;
  }
  else
  {
    button2Status = digitalRead(button2);
    if (button2Status == 1) break;
  }
  digitalWrite(7, HIGH);
  lcd.clear();
  lcd.setCursor(2, 0);
  lcd.print("MAY BOM CHAY");
  lcd.setCursor(3, 1);
  lcd.print(dem/60);
  lcd.setCursor(4, 1);
  lcd.print("Phut");
  lcd.setCursor(9, 1);
  lcd.print(dem%60);
  lcd.setCursor(11, 1);
  lcd.print("s");
  delay(500);

  stat = call.CallStatusWithAuth(number, 1, 5);
  if (stat == CALL_INCOM_VOICE_AUTH) {
    call.HangUp();
    break;          }
  }
}
```



```
    trangthai = 0;
    lcd.clear();
    digitalWrite(7, LOW);
  }
}}
```

Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

4.1 Kết Luận:

4.1.1 Ưu điểm của đề tài:

- Mạch nhỏ gọn.
- Đáp ứng được yêu cầu của đề tài.
- Hiện thị rõ ràng.
- Tiết kiệm được công sức con người.

4.1.2 Nhược điểm của đề tài:

- Còn hạn chế về việc điều khiển bằng tin nhắn
- Phần điều khiển vẫn chưa được thực hiện đồng đều
- Chưa khắc phục được phần nút nhấn bị trễ

4.2 Hướng phát triển:

- Mô hình có thể áp dụng để xây dựng hệ thống bơm nước vào các bồn chứa nước cho sinh hoạt hằng ngày.

- Áp dụng trong nông nghiệp để tưới tiêu.
- Thiết kế SmartHome – Ngôi nhà thông minh với các thiết bị được điều khiển qua điện thoại...

Do thời gian và kiến thức còn hạn hẹp nên không thể tránh khỏi những thiếu sót trong quá trình thực hiện đề tài. Rất mong nhận được những góp ý, những đánh giá quý báu của quý thầy cô và các bạn.

TÀI LIỆU THAM KHẢO

1. **Kỹ thuật điện tử.** (1999)
Đỗ Xuân Thụ. – NXB giáo dục.
2. **Giáo trình cảm biến.** (2000)
Phan Quốc Phô, Nguyễn Đức Chiến. – NXB Khoa học và kỹ thuật.
3. **Vi điều khiển cấu trúc lập trình và ứng dụng.** (2008)
Kiều Xuân Thực, Vũ Thị Hương, Vũ Trung Kiên – NXB Giáo Dục.
4. **Website** <http://alldatasheet.com/>
5. **Website** <http://arduino.vn/>
6. **Website** <http://codientu.org/>
7. **Website** <http://webdien.com/>
8. **Website** <http://www.tailieu.vn/>
9. **Website** <http://wikipedia.com/>