

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP HỒ CHÍ MINH**  
**KHOA ĐIỆN – ĐIỆN TỬ**



**HCMUTE**

**ĐỒ ÁN MÔN HỌC 1**

**ĐIỀU KHIỂN THIẾT BỊ TỪ XA**  
**QUA TIN NHẮN SMS.**  
**ĐIỀU KHIỂN GIẾNG TRỜI TỰ ĐỘNG**

**SVTH : NGUYỄN BẢO TÍNH**

**MSSV : 20161383**

**KHÓA : 2020**

**NGÀNH: CNKT ĐIỆN TỬ - VIỄN THÔNG**

**GVHD : ThS. NGUYỄN NGÔ LÂM**

Tp. Hồ Chí Minh, tháng 06 năm 2023

## **NHIỆM VỤ ĐỒ ÁN MÔN HỌC**

Họ và tên sinh viên: Nguyễn Bảo Tính

MSSV:20161383

Ngành:Công Nghệ Kỹ Thuật Điện tử - Viễn thông

Lớp: 20161VMVT1

Giảng viên hướng dẫn: ThS. Nguyễn Ngô Lâm

Ngày nhận đề tài:

Ngày nộp đề tài:

1. Tên đề tài: Điều khiển thiết bị qua tin nhắn SMS,điều khiển giếng trời tự động.

2. Tài liệu, số liệu:

Kiến thức cơ bản về các môn Mạch điện, Điện tử cơ bản, Vi xử lý, Arduino.

3. Nội dung thực hiện đề tài:

- Thiết kế hệ thống
- Mô phỏng mạch trên Proteus
- Lập trình cho hệ thống
- Chỉnh sửa và kiểm tra mạch
- Viết báo cáo
- Sản phẩm.

Mạch nhận tin nhắn điều khiển từ điện thoại và điều khiển bật tắt 4 relay để điều khiển hoạt động của 4 thiết bị, tự động đóng giếng trời khi trời mưa và thông báo mưa đến điện thoại.

GIẢNG VIÊN HƯỚNG DẪN

## PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Nguyễn Bảo Tính

MSSV:20161383

Ngành:Công Nghệ Kỹ Thuật Điện tử - Viễn thông

Lớp: 20161VMVT1

Giảng viên hướng dẫn: ThS. Nguyễn Ngô Lâm

Tên đề tài: Điều khiển thiết bị qua tin nhắn SMS,điều khiển giếng trời tự động

**NHẬN XÉT**

Về nội dung đề tài & khối lượng thực hiện:

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

Ưu điểm:

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

Khuyết điểm:

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

Đề nghị cho bảo vệ hay không?

.....

Đánh giá loại:

.....

6. Điểm:.....(Bằng chữ:..... )

.....

Tp. Hồ Chí Minh, ngày tháng năm

Giáo viên hướng dẫn

## LỜI CẢM ƠN

Đầu tiên, em xin được gửi lời cảm ơn đến thầy Nguyễn Ngô Lâm đã tận tình hướng dẫn, tạo điều kiện và giúp đỡ cho em trong suốt quá trình thực hiện đồ án. Em xin gửi lời cảm ơn chân thành đến Thầy ạ.

Em cũng muốn gửi lời cảm ơn đến các bạn bè của mình đã hỗ trợ, đóng góp ý kiến và chia sẻ kinh nghiệm để giúp em hoàn thành tốt đề tài. Em đã cố gắng hết sức, nhưng vì lượng kiến thức chưa đủ nên không thể tránh khỏi những thiếu sót. Vì vậy, em mong nhận được sự góp ý quý báu từ Thầy để em có thể hoàn thiện và cải thiện báo cáo đồ án môn học 1 cũng như tích lũy kinh nghiệm để hoàn thành tốt báo cáo đồ án 2 và đồ án tốt nghiệp sau này.

Cuối cùng, em xin chân thành cảm ơn Thầy và kính chúc sức khỏe, may mắn và thành công trong sự nghiệp trồng người cao quý của mình.

Em xin chân thành cảm ơn!

# MỤC LỤC

NHIỆM VỤ ĐỒ ÁN MÔN HỌC .....	
PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN .....	
LỜI CẢM ƠN .....	
MỤC LỤC .....	
DANH MỤC CÁC TỪ VIẾT TẮT .....	
DANH MỤC CÁC BẢNG BIỂU .....	
DANH MỤC CÁC HÌNH ẢNH .....	
CHƯƠNG 1: TỔNG QUAN.....	2
1.1.GIỚI THIỆU .....	2
1.2.ĐỐI TƯỢNG NGHIÊN CỨU .....	2
1.3.ĐỐI TƯỢNG NGHIÊN CỨU .....	2
1.4.PHẠM VI NGHIÊN CỨU .....	3
1.5.BỐ CỤC ĐỒ ÁN .....	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	4
2.1. TỔNG QUAN VỀ SMS :.....	4
2.1.1. Giới thiệu về SMS.....	4
2.1.2. Ưu điểm của SMS :.....	5
2.1.3. SMS gateway: .....	5
2.2.MODULE SIM800L: .....	6
2.3. MẠCH GIẢM ÁP DC-DC BUCK LM2596 3A .....	8
2.4.RAIN WATER MODULE.....	9
2.5.MODULE 4 RELAY .....	11
2.6.MODULE L298N .....	12
2.7.ARDUINO UNO R3 .....	13
CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG.....	16
3.1.YÊU CẦU VÀ SƠ ĐỒ KHỐI HỆ THỐNG.....	16
3.1.1.Yêu cầu của hệ thống.....	16
3.1.2.Sơ đồ khối và chức năng mỗi khối.....	16
3.1.3.Hoạt động của hệ thống .....	17
3.2.THIẾT KẾ HỆ THỐNG PHẦN CỨNG .....	18
3.2.1.Khối thu phát tín hiệu .....	18
3.2.2.Khối hiển thị .....	19
3.2.3.Khối cảm biến mưa.....	20
3.2.4.Khối điều khiển trực tiếp .....	20

3.2.5.Khối điều khiển giếng trời .....	21
3.2.6.Khối xử lý trung tâm .....	22
3.2.7.Khối nguồn.....	23
3.3. CHỨC NĂNG VÀ HOẠT ĐỘNG CỦA PHẦN MỀM .....	23
3.4.LƯU ĐỒ.....	24
3.4.1.Lưu đồ chương trình chính .....	24
3.4.2.Lưu đồ chương trình con.....	25
CHƯƠNG 4: KẾT QUẢ THỰC HIỆN .....	28
4.1. Kết quả phân cứng.....	28
4.2. Kết quả hoạt động hệ thống .....	28
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	31
5.1.KẾT LUẬN .....	31
5.2.HẠN CHẾ VÀ HƯỚNG PHÁT TRIỂN .....	31
5.2.1.Hạn chế .....	31
5.2.2.Hướng phát triển .....	31
TÀI LIỆU THAM KHẢO.....	32
PHỤ LỤC.....	33

## DANH MỤC CÁC TỪ VIẾT TẮT

SMS	Short Messaging Service	Dịch vụ tin nhắn ngắn
DC	Direct Current	Dòng điện một chiều
UART	Universal Asynchronous Receiver / Transmitter	Truyền dữ liệu nối tiếp bất đồng bộ
I2C	Inter-Integrated Circuit	Vì mạch tích hợp truyền thông nối tiếp
I/O	Input/Output	Ngõ vào/ngõ ra
IC	Integrated Circuit	Mạch tích hợp
ADC hay A/D	Analog Digital Converter	Chuyển đổi tín hiệu tương tự sang tín hiệu số
CDMA	Code Division Multiple Access	đa truy nhập (đa người dùng) phân chia theo mã
TDMA	Time Division Multiple Access	Đa kết nối phân chia theo khe thời gian
GSM	Global System for Mobile Communications	Hệ thống thông tin di động toàn cầu
TCP/IP	TCP (Transmission Control Protocol) và IP (Internet Protocol)	Giao thức điều khiển truyền vận
HTTP	HyperText Transfer Protocol	Giao thức truyền tải siêu văn bản

## **DANH MỤC CÁC BẢNG BIỂU**

Bảng 1 Thông số kỹ thuật module SIM800L .....	7
Bảng 2 Thông số các chân module SIM800L .....	8
Bảng 3 Thông số kỹ thuật mạch giảm áp LM2596 .....	9
Bảng 4 Thông số kỹ thuật mạch giảm áp LM2596 .....	9
Bảng 5 Thông số kỹ thuật module cảm biến mưa .....	10
Bảng 6 Thông số các chân module cảm biến mưa .....	11
Bảng 7 Thông số kỹ thuật module relay 4 kênh .....	11
Bảng 8 Thông số kỹ thuật mạch cầu L298N .....	13
Bảng 9 Thông số các chân mạch cầu L298N .....	13
Bảng 10 Thông số kỹ thuật Arduino Uno R3 .....	14
Bảng 11 Thông số các chân Arduino Uno R3 .....	15
Bảng 12 Sơ đồ kết nối chân module sim800l với Arduino Uno R3 .....	18
Bảng 13 Sơ đồ kết nối chân Mạch relay 4 kênh với Arduino Uno R3 .....	19
Bảng 14 Sơ đồ kết nối nút nhấn với Arduino Uno R3.....	21
Bảng 15 Sơ đồ kết nối chân của khối điều khiển giếng trời với Arduino Uno R3 .....	21



## DANH MỤC CÁC HÌNH ẢNH

Hình 1: Nguyên lý SMS Gateway .....	5
Hình 2: Module SIM800L .....	6
Hình 3: Các chân của module sim800l .....	7
Hình 4: Mạch giảm áp DC-DC LM2596 .....	8
Hình 5: Sơ đồ nguyên lý mạch giảm áp LM2596 .....	9
Hình 6: Module cảm biến mưa .....	10
Hình 7: Module relay 4 kênh .....	11
Hình 8: Sơ đồ nguyên lý relay 4 kênh .....	11
Hình 9 Mạch cầu H L298N .....	12
Hình 10: Arduino Uno R3 .....	14
Hình 11: Sơ đồ khối của hệ thống .....	16
Hình 12: Mô phỏng module sim800l trên proteus .....	18
Hình 13: Mô phỏng relay 4 kênh trên proteus .....	19
Hình 14: Mô phỏng cảm biến mưa trên Proteus .....	20
Hình 15: Mô phỏng kết nối nút nhấn với Arduino Uno R3 trên proteus .....	20
Hình 16: Mô phỏng khối điều khiển giếng trời .....	21
Hình 17 Mô phỏng Arduino Uno R3 trên proteus .....	22
Hình 18 Mô phỏng nguồn trên proteus .....	23
Hình 19: Giao diện phần mềm Arduino IDE .....	24
Hình 20: Lưu đồ chương trình chính .....	25
Hình 21: Lưu đồ xử lý tin nhắn .....	26
Hình 22: Lưu đồ nhận dữ liệu .....	26
Hình 23: Lưu đồ xử lý nút nhấn .....	27
Hình 24: Lưu đồ xử lý giếng trời .....	27
Hình 25: Hình ảnh thực tế phần cứng .....	28
Hình 26: Hình ảnh hoạt động của thiết bị qua tin nhắn SMS .....	29
Hình 27: Hình ảnh hoạt động của thiết bị với nút nhấn .....	29
Hình 28: Hình ảnh hoạt động của thiết bị báo trời mưa .....	30

# CHƯƠNG 1: TỔNG QUAN

## 1.1. GIỚI THIỆU

Hiện nay, xu hướng hiện đại hoá các hệ thống trong công nghiệp và đời sống đang diễn ra rất mạnh mẽ. Các thiết bị đang được số hoá để đáp ứng chất lượng cho hệ thống và dễ dàng điều khiển hoặc sử dụng. Trong cách mạng công nghiệp lần thứ 4, các thiết bị điện tử thông minh đang phát triển mạnh mẽ và mang lại hiệu quả cao trong hầu hết các lĩnh vực.

Các tính năng của điện thoại hiện nay không chỉ bao gồm các chức năng cơ bản như nghe gọi, nhắn tin, chụp ảnh, chơi game, xem phim, mà còn có khả năng giám sát và điều khiển các thiết bị từ xa thông qua tin nhắn SMS. Điều này đã mở ra nhiều tiềm năng trong việc ứng dụng công nghệ để điều khiển các thiết bị trong cuộc sống hàng ngày và trong các hệ thống công nghiệp. Cùng với sự phát triển của công nghệ khoa học kỹ thuật nói chung và các thiết bị liên lạc như điện thoại di động nói riêng, điện thoại có khả năng giám sát và điều khiển thiết bị từ khoảng cách xa thông qua tin nhắn SMS. Chính vì thực tiễn này em đã chọn đề tài “ĐIỀU KHIỂN THIẾT BỊ TỪ XA QUA TIN NHẮN SMS VÀ TỰ ĐỘNG ĐIỀU KHIỂN GIẾNG TRỜI”

## 1.2. MỤC TIÊU NGHIÊN CỨU

Nghiên cứu cơ sở lý thuyết về module sim800L điều khiển hiển thị các thiết bị, module relay, mạch giảm áp và arduino uno và module cảm biến mưa.

Nghiên cứu, thực hành các thao tác kỹ thuật điện tử cơ bản (lắp ráp, test mạch, mô phỏng, thiết kế...).

Ngoài ra khi nghiên cứu đề tài em muốn tạo ra sản phẩm có ích trong một số lĩnh vực trong đời sống. Có thể đạt được độ chính xác cao trên 80%.

## 1.3. ĐỐI TƯỢNG NGHIÊN CỨU

- Arduino Uno
- Module SIM800L
- Module giảm áp
- Module 4 relay

- Module cảm biến mưa
- Điện thoại và SMS

Kết nối các module sim, module relay và module cảm biến mưa với board Arduino để điều khiển các thiết bị thông qua Relay.

#### **1.4. PHẠM VI NGHIÊN CỨU**

Nghiên cứu cơ sở lý thuyết về mạch điều khiển thiết bị từ xa bằng cách nhắn tin từ điện thoại. Nghiên cứu cơ sở lý thuyết về mạch tự động điều khiển giếng trời và thông báo qua tin nhắn SMS. Mục đích của đề tài là lựa chọn thuật toán, phương pháp có độ chính xác để nhận diện kết nối module sim từ mạch và điện thoại cần điều khiển để sản phẩm có thể thành công.

Tìm hiểu các lý thuyết có liên quan như môi trường ARDUINO IDE, ngôn ngữ C++.

#### **1.5. BỐ CỤC ĐỒ ÁN**

Chương 1: Tổng quan: Nêu tính cấp thiết của đề tài, xu hướng và tình hình khoa học và công nghệ hiện nay. Sự phát triển công nghiệp và đời sống hằng ngày và từ đó đưa ra lý do chọn đề tài và xác định mục tiêu cho đề tài.

Chương 2: Cơ sở lý thuyết: Trình bày tổng quan về các thành phần và chức năng của từng loại phần cứng có trong hệ thống, dẫn dắt chi tiết cụ thể để xây dựng hoàn chỉnh về mô hình.

Chương 3: Thiết kế và xây dựng hệ thống: Từ yêu cầu đề tài, trình bày về sơ đồ hệ thống. Nêu ra các phương pháp xử lý dữ liệu rồi từ đó thiết kế mô hình.

Chương 4: Kết quả thực hiện: Trình bày về kết quả của từng khối nhỏ và kết quả điều khiển thiết bị qua tin nhắn. Đưa ra các hiển thị thiết bị như mong muốn đã lập trình.

Chương 5: Kết luận và hướng phát triển: Dựa vào kết quả có được từ chương 4, đưa ra kết luận tổng quan về những gì đạt được và chưa đạt được của đề tài. Từ đó đưa ra hướng phát triển để cải thiện hệ thống.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1.TỔNG QUAN VỀ SMS :

#### 2.1.1.Giới thiệu về SMS

SMS là từ viết tắt của Short Message Service. Đó là một công nghệ cho phép gửi và nhận các tin nhắn giữa các điện thoại với nhau. SMS xuất hiện đầu tiên ở Châu Âu vào năm 1992. Ở thời điểm đó, nó bao gồm cả các chuẩn về GSM (Global System for Mobile Communication). Một thời gian sau đó, nó phát triển sang công nghệ wireless như CDMA và TDMA. Các chuẩn GSM và SMS có nguồn gốc phát triển bởi ETSI (European Telecommunication Standards Institute). Ngày nay 3GPP (Third Generation Partnership Project) đang giữ vai trò kiểm soát về sự phát triển và duy trì các chuẩn GSM và SMS.

Như chính tên đầy đủ của SMS là Short Message Service, dữ liệu có thể được lưu giữ bởi một SMS là rất giới hạn. Một SMS có thể chứa tối đa là 140 byte (1120 bit) dữ liệu. Vì vậy, một SMS có thể chứa:

- 160 ký tự nếu mã hóa ký tự 7 bit được sử dụng (phù hợp với mã hóa các ký tự latin như alphabet của tiếng Anh)
- 70 ký tự nếu như mã hóa ký tự 16 bit Unicode UCS2 được sử dụng (dùng cho các ký tự không phải mã latin như chữ Trung Quốc...)

SMS dạng text hỗ trợ nhiều ngôn ngữ khác nhau. Nó có thể hoạt động tốt với nhiều ngôn ngữ mà có hỗ trợ mã Unicode, bao gồm Arabic, Trung Quốc, Nhật Bản, Hàn Quốc...

Bên cạnh gửi tin nhắn dạng text thì tin nhắn còn có thể mang dữ liệu dạng binary. Nó cho phép gửi nhạc chuông, hình ảnh cùng nhiều tiện ích khác...tới điện thoại khác. Nội dung của 1 tin nhắn SMS khi được gửi đi chia làm 5 phần như sau:

Instructions to air interface	Instructions to SMSC	Instructions to handset	Instructions to SIM (optional)	Message Body
-------------------------------	----------------------	-------------------------	--------------------------------	--------------

- Instructions to air interface: chỉ thị dữ liệu kết nối với air interface (giao diện không khí).
- Instructions to SMSC: chỉ thị dữ liệu kết nối với trung tâm tin nhắn SMSC.

- Instructions to handset: chỉ thị dữ liệu kết nối bắt tay
- Instructions to SIM (optional): chỉ thị dữ liệu kết nối, nhận biết SIM.
- Message body: nội dung tin nhắn SMS

### 2.1.2.Ưu điểm của SMS :

- Tin nhắn có thể được gửi và đọc tại bất kỳ thời điểm nào
- Tin nhắn SMS có thể được gửi tới các điện thoại dù chúng đang bị tắt nguồn.
- Ít gây phiền phức trong khi bạn vẫn có thể giữ liên lạc với người khác.
- Được sử dụng trên các điện thoại di động khác nhau và có thể gửi cùng mạng hoặc khác mạng đều được.
- Phù hợp với các ứng dụng wireless sử dụng cùng với nó như: chức năng SMS được hỗ trợ 100% bởi các điện thoại sử dụng công nghệ GSM; có thể gửi nhạc chuông, hình ảnh...; hỗ trợ chi trả các dịch vụ trực tuyến download nhạc chuông....

### 2.1.3.SMS gateway:

Một khó khăn của SMS là các SMSC được phát triển, xây dựng bởi các công ty sử dụng giao thức truyền thông riêng của họ và hầu hết các giao thức này thuộc quyền sở hữu riêng. Ví dụ như Nokia có một giao thức SMSC là CIMD, nhà điều hành CMG lại có giao thức SMSC là EMI. Chúng ta không thể kết nối hai SMSC nếu chúng không có cùng giao thức SMSC. Để giải quyết vấn đề này, một SMS gateway được đặt giữa hai giao thức SMSC khác nhau. Gateway này hoạt động ở hai sóng mang khác nhau để có thể gửi SMS cho nhau mà không gặp bất kỳ trở ngại nào.



Hình 1: Nguyên lý SMS Gateway

## 2.2. MODULE SIM800L:<sup>[2]</sup>

SIM800L là một module GSM/GPRS thu nhỏ được sử dụng để kết nối thiết bị với mạng di động để truyền dữ liệu. Module này có kích thước nhỏ gọn và tiêu thụ điện năng thấp, vì vậy nó thường được sử dụng trong các ứng dụng IoT và các thiết bị di động như máy định vị GPS, đồng hồ thông minh, hệ thống an ninh, và các thiết bị giám sát.

SIM800L hỗ trợ nhiều chuẩn kết nối, bao gồm GSM, GPRS, và SMS. Nó cũng có thể kết nối với mạng Wi-Fi thông qua một module chuyển đổi, nhưng điều này yêu cầu thêm phần cứng và phần mềm. Module này có thể được điều khiển bằng AT command thông qua các chân GPIO của thiết bị điều khiển.

SIM800L có một số tính năng nâng cao như bộ định vị toàn cầu GPS, bộ nhớ lưu trữ dữ liệu và hỗ trợ các giao thức như TCP/IP và HTTP. Nó cũng có thể được cấu hình để gửi và nhận dữ liệu từ máy chủ từ xa thông qua GPRS.

Tuy nhiên, khi sử dụng SIM800L, cần lưu ý rằng nó cần được cấp nguồn điện ổn định và đủ lớn để hoạt động tốt. Ngoài ra, vì SIM800L sử dụng sóng RF để kết nối với mạng di động, nó có thể bị ảnh hưởng bởi các tín hiệu nhiễu RF khác trong môi trường xung quanh.



Hình 2: Module SIM800L

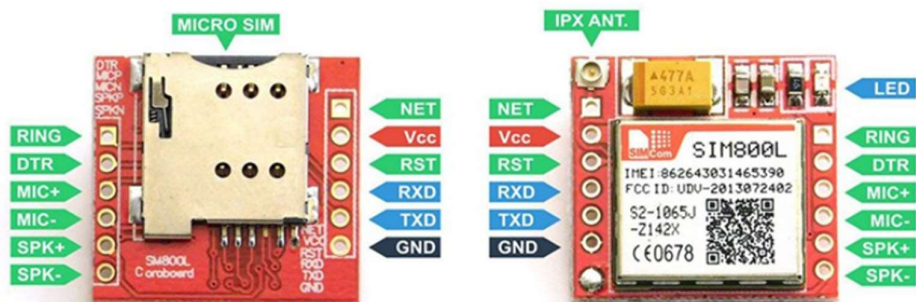
Một số đặc trưng của module sim

- Hỗ trợ 4 băng tần: GSM850, EGSM900, DCS1800 và PCS1900
- Kết nối với bất kỳ mạng GSM toàn cầu nào với bất kỳ SIM 2G nào
- Thực hiện và nhận cuộc gọi thoại bằng loa ngoài 8Ω và micrô điện tử
- Gửi và nhận tin nhắn SMS

- Gửi và nhận dữ liệu GPRS (TCP / IP, HTTP, v.v.)
- Quét và nhận các chương trình phát sóng radio FM
- Truyền điện:
- Loại 4 (2W) cho GSM850
- Loại 1 (1W) cho DCS1800
- Bộ lệnh AT dựa trên nối tiếp
- Đầu nối FL cho ăng ten di động
- Chấp nhận thẻ Micro SIM

*Bảng 1 Thông số kỹ thuật module SIM800L*

STT		Thông số
1	Điện áp hoạt động	3.8V – 4.4V
2	Dòng khi ở chế độ chờ	10mA
3	Dòng khi hoạt động	100mA – 1A
4	Khe cắm SIM	MICROSIM
5	Hỗ trợ 4 băng tần	GSM850MHz,EGSM900MHz, DSC1800Mhz, PCS1900MHz
6	Chân	12
7	Kích thước	25 x 22 mm



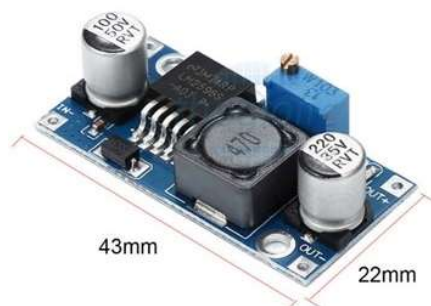
*Hình 3: Các chân của module sim800l*

*Bảng 2 Chức năng các chân module SIM800L*

Chân	Tên	Chức năng
1	VCC	Nguồn vào 3.7 – 4.2V.
2	TXD	Chân truyền Uart TX.
3	RXD	Chân nhận Uart RX.
4	DTR	Chân UART DTR, thường không xài.
5	SPKP, SPKN	Ngõ ra âm thanh, nối với loa để phát âm thanh.
6	MICP, MICN	Ngõ vào âm thanh, phải gắn thêm Micro để thu âm thanh.
7	Reset	Chân khởi động lại Sim800L (thường không xài).
8	RING	Báo có cuộc gọi đến
9	GND	Chân Mass, cấp 0V.

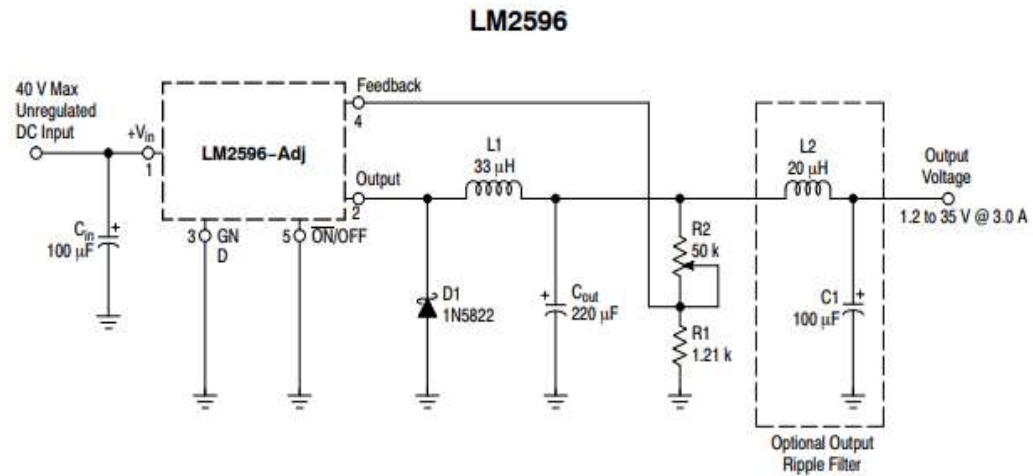
### **2.3.MẠCH GIẢM ÁP DC-DC BUCK LM2596 3A:<sup>[5]</sup>**

Mạch giảm áp DC-DC LM2596 là một thiết bị được sử dụng để giảm áp đầu vào xuống mức thấp hơn trong các ứng dụng điện tử. Nó có thể điều chỉnh đầu ra từ 1.23V đến 37V với dòng điện tối đa là 3A và đầu vào từ 4.5V đến 40V. Mạch LM2596 được thiết kế để cung cấp độ ổn định cao và hiệu quả chuyển đổi tối đa 92%, giúp tiết kiệm năng lượng và kéo dài tuổi thọ của pin hoặc nguồn cấp. Nó cũng có tính năng bảo vệ quá tải, quá nhiệt và ngắn mạch để đảm bảo an toàn khi sử dụng. Mạch LM2596 được ứng dụng rộng rãi trong các thiết bị điện tử gia đình, thiết bị đo lường, xe hơi, công nghiệp và các ứng dụng DIY.



*Hình 4: Mạch giảm áp DC-DC LM2596*





Hình 5: Sơ đồ nguyên lý mạch giảm áp LM2596

Bảng 3 Thông số kỹ thuật mạch giảm áp DC-DC LM2596

STT		Thông số
1	Điện áp đầu vào	4V – 35V
2	Điện áp đầu ra	1.23V – 30V
3	Dòng đầu ra	3A
4	Hiệu suất chuyển đổi	Tối đa 92%
5	Tần số hoạt động module hạ áp	150Hz
6	Nhiệt độ hoạt động	-40 °C đến + 85 °C
7	Công suất cực đại	26W

Bảng 4 Chức năng các chân mạch giảm áp DC-DC LM2596

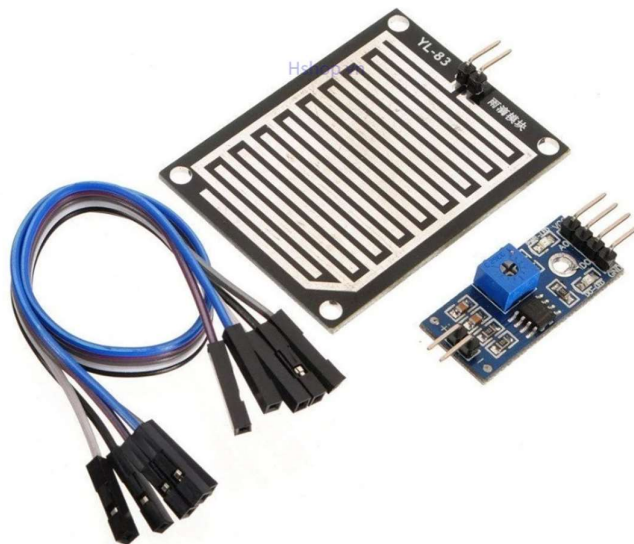
Chân	Tên	Mô tả
1	INPUT +	Ngõ vào dương
2	INPUT -	Ngõ vào âm
3	OUTPUT +	Ngõ ra dương
4	OUTPUT -	Ngõ ra âm

## 2.4.MODULE CẢM BIẾN MƯA<sup>[10]</sup>

Cảm biến mưa sử dụng để phát hiện mực nước, trời mưa, hay các môi trường có nước. Mạch cảm biến mưa được đặt ngoài trời để kiểm tra trời có mưa không, qua đó

truyền tín hiệu điều khiển đóng / ngắt rơ le. Mạch cảm biến mưa gồm 2 bộ phận: Bộ phận cảm biến mưa được gắn ngoài trời, Bộ phận điều chỉnh độ nhạy cần được che chắn.

Sử dụng nguyên lý điện trở và dòng điện để phát hiện mưa hoặc nước. Module cảm biến mưa bao gồm hai điện cực, điện cực dương và điện cực âm. Khi không có mưa hoặc nước, điện trở giữa hai điện cực là rất lớn, không đủ để dòng điện chạy qua. Khi mưa hoặc nước rơi vào trên hai điện cực, điện trở giữa hai điện cực sẽ giảm đáng kể, dòng điện sẽ chạy qua và được đo bởi một mạch điện tử. Dữ liệu đo được sẽ được chuyển đến một vi xử lý hoặc một module điều khiển để xử lý và đưa ra kết quả. Dựa trên kết quả đo được, các hệ thống tự động hoặc các thiết bị khác có thể được kích hoạt để thực hiện các thao tác như đóng mở cửa, bật tắt đèn, v.v.



Hình 6: Module cảm biến mưa

Bảng 5 Thông số kỹ thuật module cảm biến mưa

STT		Thông số
1	Điện áp sử dụng	5VDC
2	Kích thước tấm cảm biến mưa	54 x 40 mm
3	Kích thước board PCB	30 x 16 mm
4	Tín hiệu đầu ra	Digital D0 và Analog A0

*Bảng 6 Chức năng các chân module cảm biến mưa*

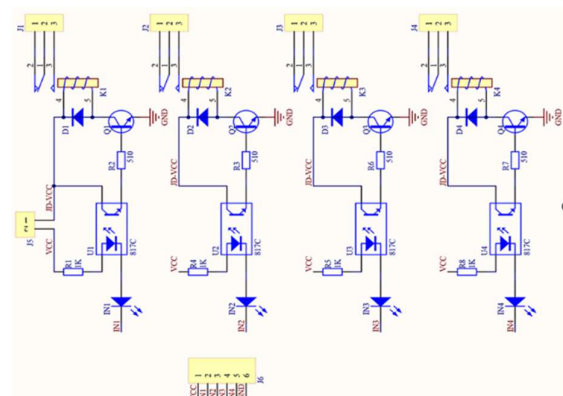
Chân	Tên	Mô tả
1	VCC	Chân cấp nguồn 5V
2	GND	Chân nối đất 0V
3	D0	Ngõ ra digital
4	A0	Ngõ ra analog

## 2.5.MODULE 4 RELAY <sup>[4]</sup>

Module 4 relay 5V với Opto cách ly gồm 4 rơ le hoạt động tại điện áp 5VDC, chịu được hiệu điện thế lên đến 250VAC 10A. Module 4 relay 5V với Opto cách ly được thiết kế chắc chắn, khả năng cách điện tốt. Trên module đã có sẵn mạch kích relay sử dụng transistor và IC cách ly quang giúp cách ly hoàn toàn mạch điều khiển (vi điều khiển) với rơ le bảo đảm vi điều khiển hoạt động ổn định. Có sẵn header rất tiện dụng khi kết nối với vi điều khiển.



*Hình 7: Module relay 4 kênh*



*Hình 8: Sơ đồ nguyên lý relay 4 kênh*

Module 4 relay 5V với Opto cách ly sử dụng chân kích mức Thấp (0V), khi có tín hiệu 0V vào chân IN thì relay sẽ nhảy qua thường Hở của Relay. Ứng dụng với relay module khá nhiều bao gồm cả điện DC hay AC.

*Bảng 7 Thông số kỹ thuật module relay 4 kênh*

STT		Thông số
1	Điện áp sử dụng	5VDC
2	Dòng tiêu thụ	~200mA/1Relay

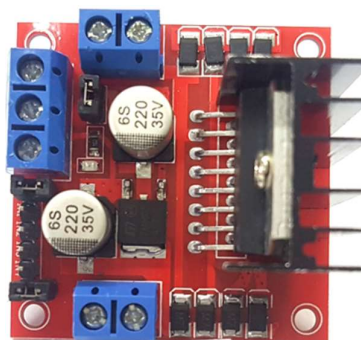
3	Tín hiệu kích	Low 0V
4	Nguồn nuôi relay	5VDC
5	Tiếp điểm đóng ngắt	Tối đa 250VAC-10A hoặc 30VDC-10A
6	Kích thước	76mm x 56mm x 18.5mm

*Bảng 8 Chức năng các chân module relay 4 kênh*

Chân	Tên	Mô tả
1	VCC	Chân cấp nguồn 5V
2	GND	Chân nối đất 0V
3,4,5,6	IN1,IN2,IN3,IN4	Ngõ vào điều khiển 4 relay
7-18	NO1,NO2,NO3,NO4,NC1,NC2,NC3,NC4,COM1,COM2,COM3,COM4.	Các ngõ ra

## 2.6. MODULE L298N<sup>[3]</sup>

Module L298 là một module điều khiển động cơ hai chiều, được sử dụng rộng rãi trong các ứng dụng điện tử để điều khiển động cơ DC hoặc bước. Nó được thiết kế để cung cấp công suất cao và hiệu suất ổn định trong quá trình hoạt động. Module L298 có thể điều khiển động cơ DC hai chiều hoặc một động cơ bước bằng cách sử dụng tín hiệu điều khiển từ một vi điều khiển hoặc các thiết bị khác. Nó có thể cung cấp dòng điện đến động cơ lên đến 2A cho mỗi kênh và điện áp đầu vào từ 7V đến 46V.



*Hình 9 Mạch cầu H L298N*

Module L298 hoạt động dựa trên nguyên lý của mạch cầu H, được sử dụng để điều khiển động cơ hai chiều hoặc động cơ bước. Nó có hai kênh điều khiển độc lập, mỗi kênh được kết nối với một mạch cầu H để điều khiển một động cơ. Module này là một giải pháp tuyệt vời để điều khiển động cơ một chiều hoặc động cơ bước với độ chính xác cao và hiệu suất ổn định.

*Bảng 9 Thông số kỹ thuật mạch cầu L298N*

STT		Thông số
1	IC chính	L298-Dual Full Bridge Driver
2	Điện áp đầu vào	5V – 30V
3	Tín hiệu kích	High 5V hoặc Low 0V
4	Công suất tối đa	25W/1 cầu
5	Dòng tối đa cho mỗi cầu	2A
6	Kích thước	43mm x 43mm x 27mm

*Bảng 10 Chức năng các chân mạch cầu L298N*

Chân	Tên	Mô tả
1	VCC	Chân cấp nguồn cho module
2	VS	Chân cấp nguồn cho động cơ
3	GND	Chân nối đất
4,5,6,7	IN1,IN2,IN3,IN4	Chân điều khiển động cơ
8,9	ENA,ENB	Chân điều chỉnh tốc độ động cơ 1, động cơ 2

## 2.7.ARDUINO UNO R3 <sup>[6]</sup>

Arduino Uno R3 là một board phát triển điện tử mã nguồn mở (open-source) được thiết kế dựa trên vi điều khiển ATmega328P của hãng Atmel. Board này được phát triển bởi Massimo Banzi và David Cuartielles vào năm 2005 và được sản xuất bởi công ty Arduino LLC. Arduino Uno R3 là một trong những board phổ biến và thường được sử dụng nhất trong cộng đồng Arduino.

Board Arduino Uno R3 có kích thước nhỏ gọn, dễ dàng sử dụng và lập trình. Có thể được lập trình bằng ngôn ngữ lập trình Arduino và có thể được sử dụng để phát triển nhiều ứng dụng điện tử như hệ thống nhúng (embedded system), điều khiển thiết bị, robot, đo lường và điều khiển các tín hiệu, ...



Hình 10: Arduino Uno R3

Bảng 11 Thông số kỹ thuật Arduino Uno R3

STT		Thông số
1	Vi điều khiển	Atmega328 họ 8 bit
2	Điện áp hoạt động	5VDC
3	Tần số hoạt động	16MHz
4	Dòng tiêu thụ	30mA
5	Điện áp vào khuyến dùng	7-12VDC
6	Điện áp vào giới hạn	6-20VDC
7	Số chân Digital I/O	14 chân
8	Số chân analog	6 chân
9	Dòng tối đa trên mỗi chân I/O	30mA
10	Dòng ra tối đa (5V)	500mA
11	Dòng ra tối đa(3.3V)	50mA
12	Bộ nhớ flash	32Kb

13	SRAM	2Kb
14	EEPROM	1Kb

*Bảng 12 Thông số các chân Arduino Uno R3*

Stt	Ký hiệu	Chức năng
1	IOREF	Điện áp tham chiếu của board
2	RESET	Reset board
3	3.3V	Cấp điện áp DC 3.3V
4	5V	Cấp điện áp DC 5V
6	GND	Chân nối đất
7	Vin	Cấp nguồn vào cho board
8-13	A0-A5	Ngõ vào analog 0 - 5
14	D0/RX	Ngõ vào digital 0, giao tiếp serial
15	D1/TX	Ngõ vào digital 1, giao tiếp serial
16	D2	Ngõ vào digital 2
17	D3~	Ngõ vào digital 3, hỗ trợ PWM
18	D4	Ngõ vào digital 4
19	D5~	Ngõ vào digital 5, hỗ trợ PWM
20	D6~	Ngõ vào digital 6, hỗ trợ PWM
21	D7	Ngõ vào digital 7
22	D8	Ngõ vào digital 8
23	D9~	Ngõ vào digital 9, hỗ trợ PWM
24	D10~	Ngõ vào digital 10, hỗ trợ PWM
25	D11~	Ngõ vào digital 11, hỗ trợ PWM
26	D12	Ngõ vào digital 12
27	D13	Ngõ vào digital 13
28	AREF	Cấp điện áp tham chiếu cho các chân Analog

## CHƯƠNG 3: THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

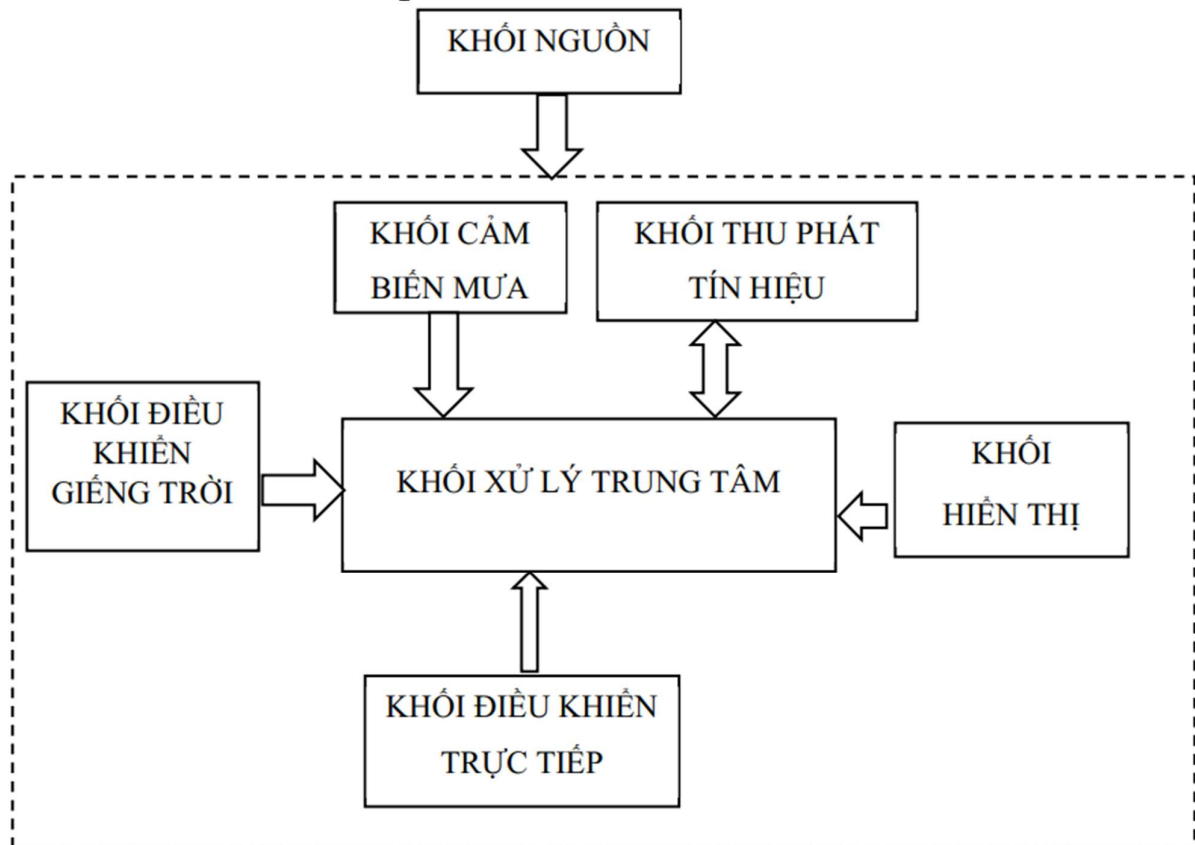
### 3.1.YÊU CẦU VÀ SƠ ĐỒ KHỐI HỆ THỐNG

#### 3.1.1.Yêu cầu của hệ thống

Hệ thống có các chức năng sau:

- Điều khiển 4 thiết bị riêng biệt
- Điều khiển bật tắt được bằng tin nhắn.
- Trả tin nhắn trạng thái hiển thị của thiết bị.
- Điều khiển bật tắt bằng nút nhấn
- Tự động đóng giếng trời khi trời mưa
- Thông báo bằng tin nhắn SMS khi trời mưa
- Điều khiển giếng trời bằng nút nhấn

#### 3.1.2.Sơ đồ khối và chức năng mỗi khối



Hình 11: Sơ đồ khối của hệ thống



Chức năng từng khối:

- Khối thu phát tín hiệu : Là nơi nhận tín hiệu tắt bật thiết bị từ điện thoại qua tin nhắn SMS.
- Khối xử lý trung tâm: Trung tâm xử lý các yêu cầu từ các khối nút nhấn và khối nhận tín hiệu điều khiển để đưa ra khối hiển thị.
- Khối hiển thị: Là nơi thi hành các yêu cầu của người dùng .
- Khối điều khiển trực tiếp: dùng để chuyển đổi các trạng thái relay và đóng mở giếng trời
- Khối điều khiển giếng trời: Nhận tín hiệu từ khối xử lý trung tâm để điều khiển đóng mở giếng trời
- Khối cảm biến mưa: truyền tín hiệu đến khối xử lý trung tâm để xử lý dữ liệu điều khiển đóng mở giếng trời và thông báo mưa qua tin nhắn sms.
- Khối nguồn: Cung cấp nguồn cho toàn bộ hệ thống hoạt động.

### **3.1.3.Hoạt động của hệ thống**

Khi trời mưa thì hệ thống sẽ tự động đóng giếng trời và thông báo đến điện thoại qua tin nhắn SMS. Khi người sử dụng muốn trao đổi thông tin với hệ thống thì sẽ phải truyền đi dữ liệu qua sóng GSM đến khối truyền nhận tín hiệu điều khiển. Từ đây khối truyền nhận sẽ gửi thông tin được nhận và cần được điều khiển cho khối xử lý trung tâm , sau đó khối xử lý trung tâm sẽ nhận tín hiệu và thực thi bằng các lệnh đã được lập trình và cho xuất ra khối hiển thị .

Cú pháp nhắn tin :

Thuê bao gửi tin nhắn điều khiển đến Module Sim800L theo cú pháp:

# <nội dung điều khiển>.

Trong đó:<nội dung điều khiển> :

- 1on\_1off : bật\_tắt thiết bị 1
- 2on\_2off : bật\_tắt thiết bị 2
- 3on\_3off : bật\_tắt thiết bị 3
- 4on\_4off : bật\_tắt thiết bị 4

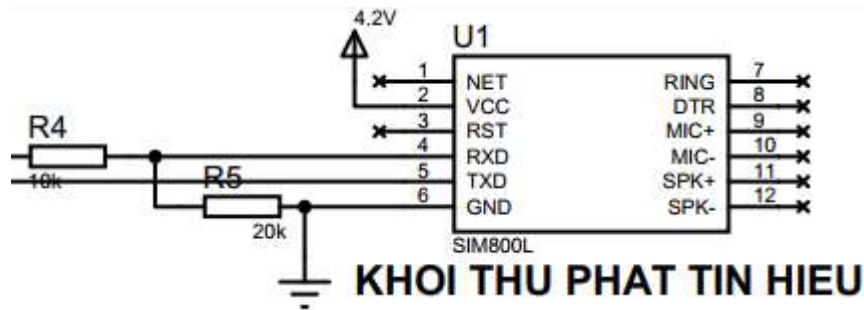
Sau khi đã thực hiện điều khiển thiết bị thì ta sẽ nhận được tin nhắn từ module sim800L về trạng thái của thiết bị lúc bấy giờ.

## 3.2. THIẾT KẾ HỆ THỐNG PHẦN CỨNG

### 3.2.1. Khối thu phát tín hiệu

Chúng ta sẽ kết nối Module Sim800L với khối xử lý trung tâm ( Arduino ) để truyền tín hiệu điều khiển được ra lệnh từ tin nhắn điện thoại.

Nối các chân theo thứ tự như sau:



Hình 12: Mô phỏng module sim800l trên proteus

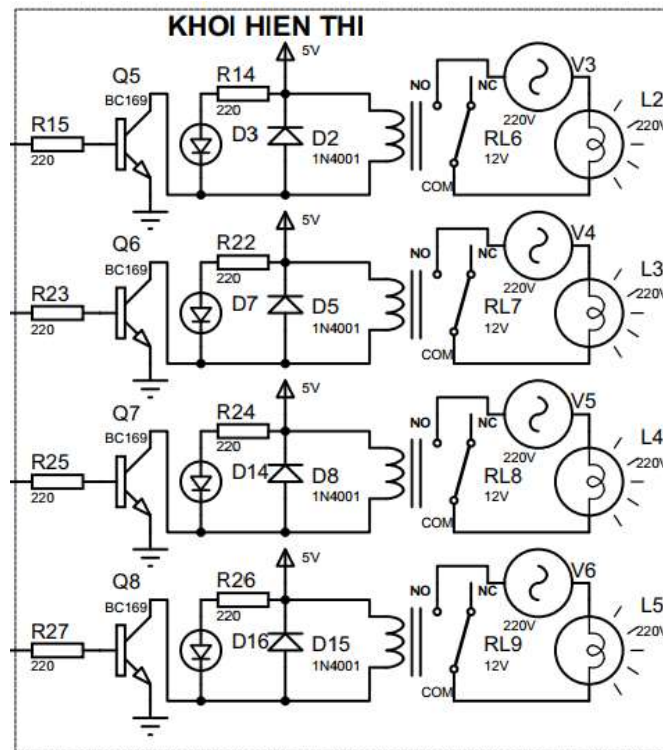
Bảng 13 Sơ đồ kết nối chân module sim800l với Arduino Uno R3

Module Sim800l	Arduino Uno R3	
TX	Chân 10	
RX	Chân 11	
VCC		Nguồn dương 4.2VCD
GND		Nối nguồn âm GND

- ❖ Lưu ý: Chúng ta không thể kết nối trực tiếp chân Rx trên module với chân TX của Arduino vì Arduino Uno sử dụng GPIO 5V trong khi module SIM800L sử dụng logic mức 3,3V và không chịu được mức 5V. Điều này có nghĩa là tín hiệu Tx đến từ Arduino Uno phải được giảm xuống còn 3,3V để không làm hỏng module SIM800L. Có một số cách để làm điều này nhưng cách dễ nhất là sử dụng một bộ chia áp bằng điện trở đơn giản. Một điện trở 10K giữa SIM800L Rx và Arduino Uno R3, và 20K giữa SIM800L Rx và GND.

### 3.2.2. Khôi hiển thị

Ở mạch này em sử dụng Module 4 relay để điều khiển thiết bị là 4 đèn 200VAC



Hình 13: Mô phỏng relay 4 kênh trên proteus

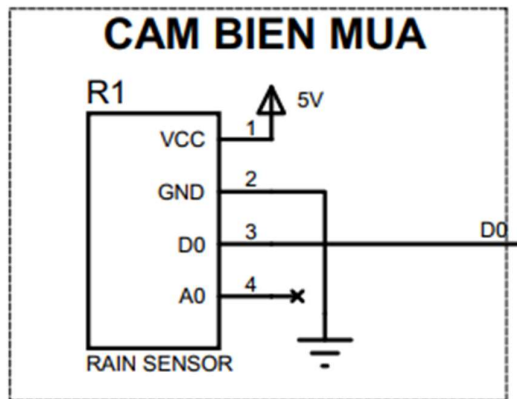
Bảng 14 Sơ đồ kết nối chân Mạch relay 4 kênh với Arduino Uno R3

Relay 4 kênh	Arduino Uno R3	
IN1	Chân 9	
IN2	Chân 8	
IN3	Chân 7	
IN4	Chân 6	
VCC		Nguồn dương 12VDC
GND		Nối nguồn âm GND

Các ngõ ra ta sử dụng hai loại chân là NO và COM ,nối với 1 chân của thiết bị cần điều khiển và chân còn lại vào nguồn cấp cho thiết bị cần điều khiển.

### 3.2.3. Khôi cảm biến mưa

Chúng ta nối chân D0 của cảm biến mưa vào chân số 8 của Arduino Uno R3



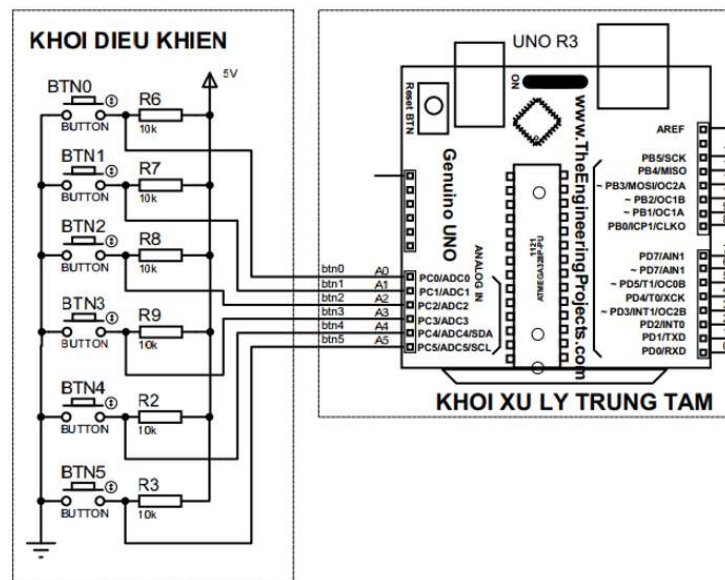
Hình 14: Mô phỏng cảm biến mưa trên Proteus

Chân D0 của cảm biến nối với chân 12 của Arduino và cấp nguồn cho module cảm biến mưa bằng Arduino với VCC, GND cảm biến lần lượt nối với chân 5V, GND của Arduino.

### 3.2.4. Khôi điều khiển trực tiếp

Do ta cần điều khiển 4 thiết bị và giếng trời nên sẽ cần 6 nút nhấn và mỗi nút nhấn sẽ đi kèm 1 điện trở kéo lên nguồn.

Ta sẽ kết nối 6 nút nhấn với các PIN của Arduino theo sơ đồ sau:



Hình 15: Mô phỏng kết nối nút nhấn với Arduino Uno R3 trên proteus

Với dòng ra tối đa cho mỗi I/O của Arduino là 500mA và cấp nguồn 5VDC cho nút nhấn, vì thế ta sẽ chọn điện trở có giá trị  $R = U/I = 5/500 = 10k\Omega$

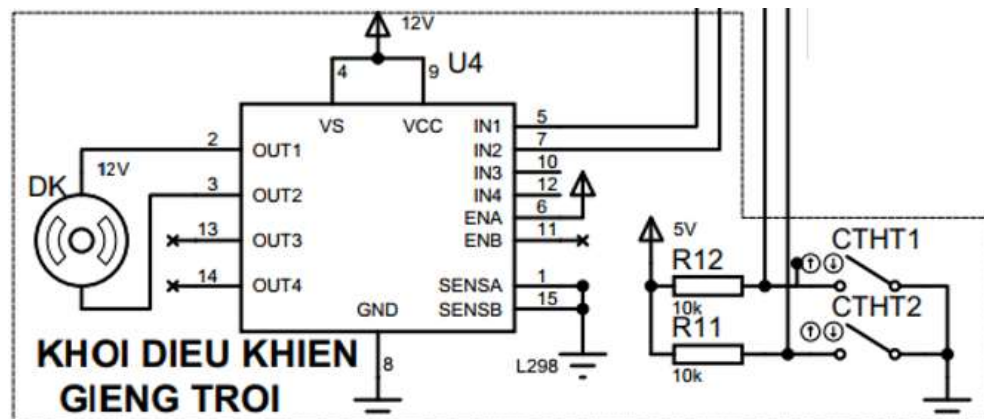
*Bảng 15 Sơ đồ kết nối nút nhấn với Arduino Uno R3*

Nút nhấn	Arduino Uno R3
BTN0	Chân A0
BTN1	Chân A1
BTN2	Chân A2
BTN3	Chân A3
BTN4	Chân A4
BTN5	Chân A5

BTN0-BTN3 dùng để điều khiển 4 relay và BTN4 ,BTN5 dùng điều khiển đóng mở giếng trời

### 3.2.5. Khối điều khiển giếng trời

Chúng ta dùng mạch cầu H L28N để điều khiển một motor quay, thông qua cơ chế hoạt động cơ để đóng mở giếng trời và quản lý bởi hai công tắc hành trình.



*Hình 16: Mô phỏng khối điều khiển giếng trời*

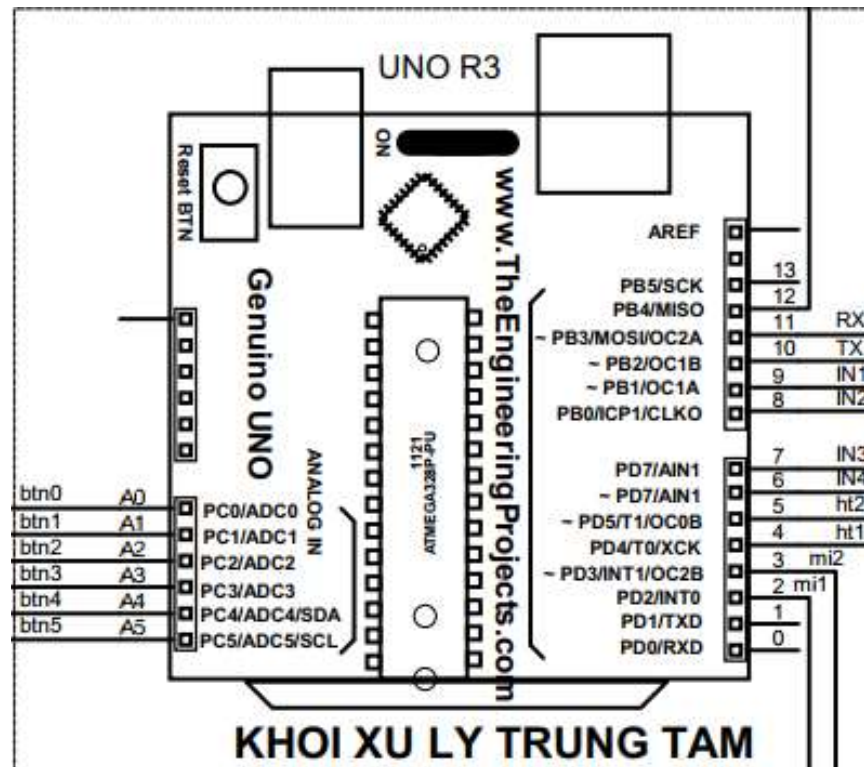
*Bảng 16 Sơ đồ kết nối chân của khối điều khiển giếng trời với Arduino Uno R3*

L28N	Arduino Uno R3	
OUT1, OUT2		Motor
IN1	Chân 2	
IN2	Chân 3	
VS	Trong L28N, VS nối với VCC thông qua 1 jumper	

VCC	Nối nguồn 12VDC	
GND	Nối đất GND	
	Chân 4	Công tắc hành trình 1
	Chân 5	Công tắc hành trình 2

### 3.2.6. Khối xử lý trung tâm

Hiện nay trên thị trường có rất nhiều loại board Arduino như: Mega, Uno, Nano,...với nhiều tính năng khác nhau. Em chọn board arduino uno để nghiên cứu sử dụng làm khối xử lý trung tâm



Hình 17 Mô phỏng Arduino Uno R3 trên proteus

Các chân A0-A5 nối với khối điều khiển trực tiếp

Các chân 2-5 nối với khối điều khiển giằng trôi

Các chân 6-9 nối với khối hiển thị

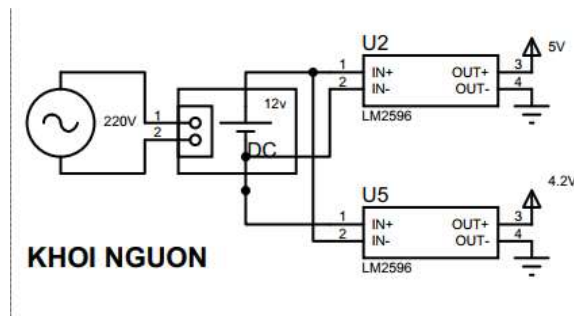
Chân 12 nối với khối cảm biến mưa

Chân 10, 11 nối với khối thu phát tín hiệu

### 3.2.7. Khôi nguồn

Hệ thống gồm các module như Module L298N với điện áp 12V, Module Relay, Công tắc, module cảm biến mưa với điện áp 5V, module Sim800L với điện áp 4.2 V. Do đó em sử dụng nguồn 12VDC và 2 module LM2596 để cung cấp các mức điện áp 12V, 5V, 4.2V.

Module L298N có dòng tối đa 2A, Module Relay 4 kênh khoảng 800mA và module SIM800L khoảng 1A, do đó em dùng nguồn khoảng 3A để có thể cung cấp đủ dòng cho hệ thống hoạt động.



Hình 18 Mô phỏng nguồn trên proteus

Với các thông số trên, em lựa chọn nguồn tổ ong với đầu vào AC110V/220V phù hợp với lưới điện gia đình và ngõ ra DC 12V-3A để cung cấp cho hệ thống

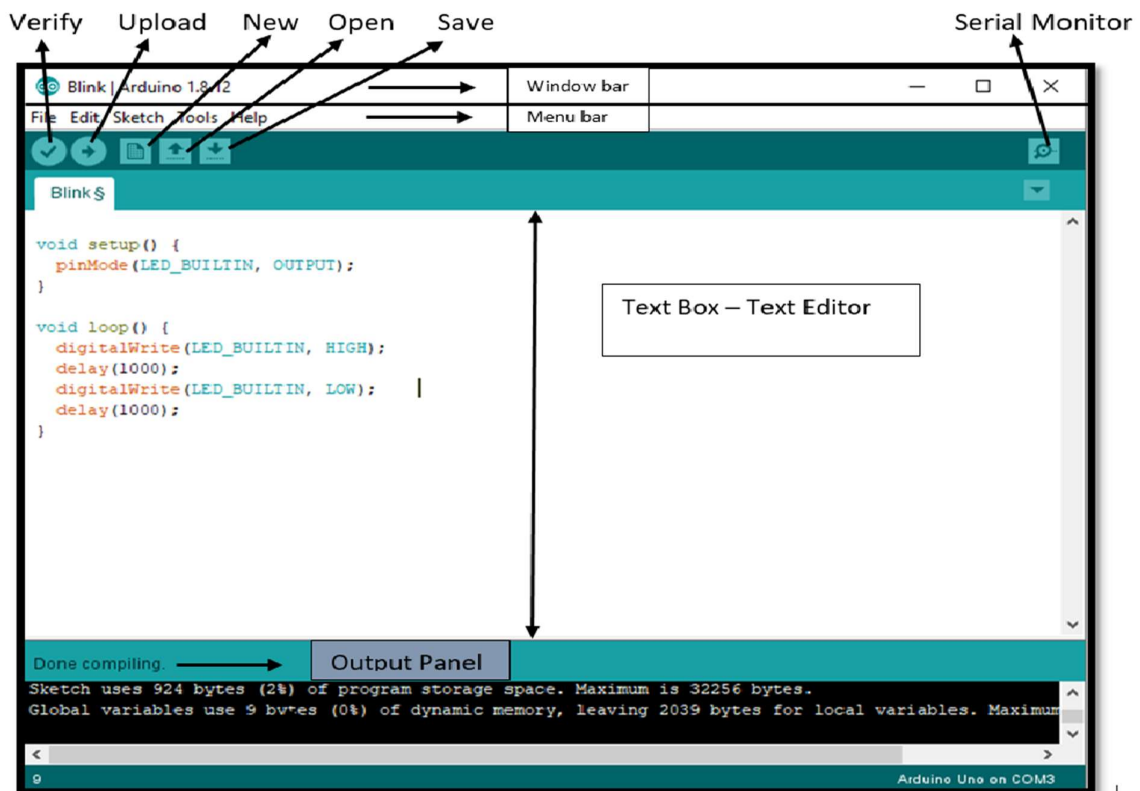
### 3.3. CHỨC NĂNG VÀ HOẠT ĐỘNG CỦA PHẦN MỀM

Trong đề tài em sử dụng phần mềm Arduino IDE để lập trình và giải quyết các vấn đề. Arduino IDE là một phần mềm mã nguồn mở chủ yếu được sử dụng để viết và biên dịch mã vào module Arduino. Đây là một phần mềm Arduino chính thức, giúp cho việc biên dịch mã trở nên dễ dàng mà ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được. Nó có các phiên bản cho các hệ điều hành như MAC, Windows, Linux và chạy trên nền tảng Java đi kèm với các chức năng và lệnh có sẵn đóng vai trò quan trọng để gỡ lỗi, chỉnh sửa và biên dịch mã trong môi trường. Có rất nhiều các module Arduino như Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro và nhiều module khác.

Mỗi module chứa một bộ vi điều khiển trên bo mạch được lập trình và chấp nhận thông tin

dưới dạng mã.Mã chính, còn được gọi là sketch, được tạo trên nền tảng IDE sẽ tạo ra một file Hex, sau đó được chuyển và tải lên trong bộ điều khiển trên bo.

Môi trường IDE chủ yếu chứa hai phần cơ bản: Trình chỉnh sửa và Trình biên dịch, phần đầu sử dụng để viết mã được yêu cầu và phần sau được sử dụng để biên dịch và tải mã lên module Arduino.



Hình 19: Giao diện phần mềm Arduino IDE

### 3.4.LƯU ĐỒ

#### 3.4.1.Lưu đồ chương trình chính

Đầu tiên, khi vừa khởi động hoặc reset, vi điều khiển sẽ tiến hành khởi tạo các biến, thiết lập ngõ vào, ngõ ra. Sau đó vi điều khiển sẽ kết nối và khởi động Module SIM800L vào trạng thái hoạt động. Tiếp theo sẽ nhận dữ liệu từ module sim800l và xử lý

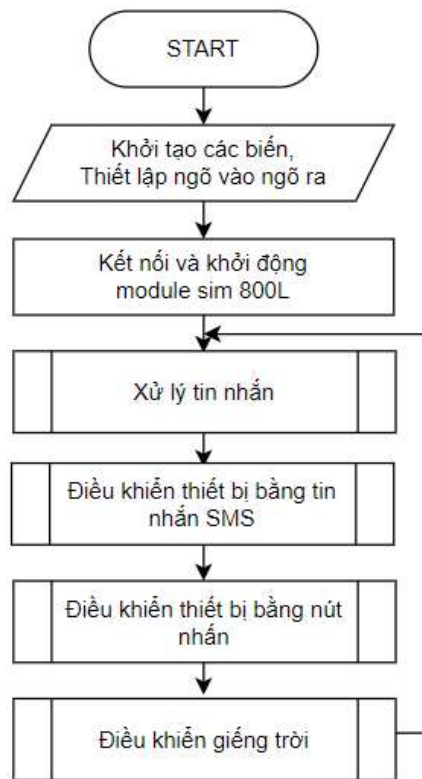


chúng.

Sau khi nhận tin nhắn từ module, vi điều khiển sẽ tiến hành xử lý nội dung tin nhắn để điều khiển thiết bị. Vi điều khiển tiến hành điều khiển module gửi tin nhắn trả lời báo kết quả điều khiển và kết thúc quá trình điều khiển.

Tiếp theo, vi điều khiển kiểm tra các công tắc hành trình và đọc dữ liệu từ cảm biến mưa để điều khiển giếng trời và kết thúc quá trình

Cuối cùng, khi thực hiện xong quá trình, vi điều khiển quay lại ban đầu tiếp tục chờ có tin nhắn mới.



Hình 20: Lưu đồ chương trình chính

#### 3.4.2. Lưu đồ chương trình con

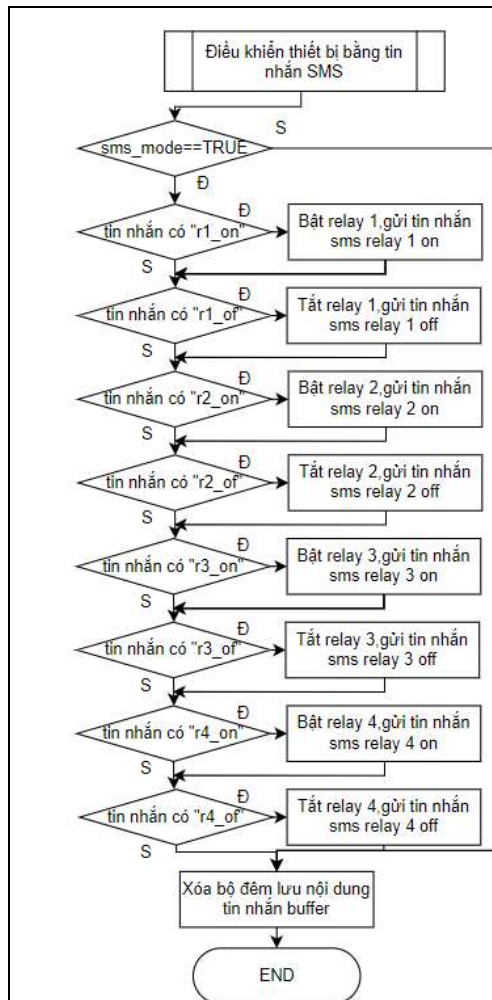
- Nhận dữ liệu: Thiết lập bộ nhớ lưu dữ liệu, nếu có dữ liệu đến sẽ nhận ký tự, chưa có ký tự kết thúc thì lưu ký tự và tăng địa chỉ lưu, nếu có ký tự kết thúc rồi thì chuyển SMS\_MODE sang giá trị TRUE

- Xử lý tin nhắn: Kiểm tra giá trị biến SMS\_MODE, nếu điều kiện TRUE sẽ lấy nội dung tin nhắn, tiếp theo kiểm tra cú pháp, nếu đúng sẽ xử lý nội dung và điều khiển thiết bị, sau

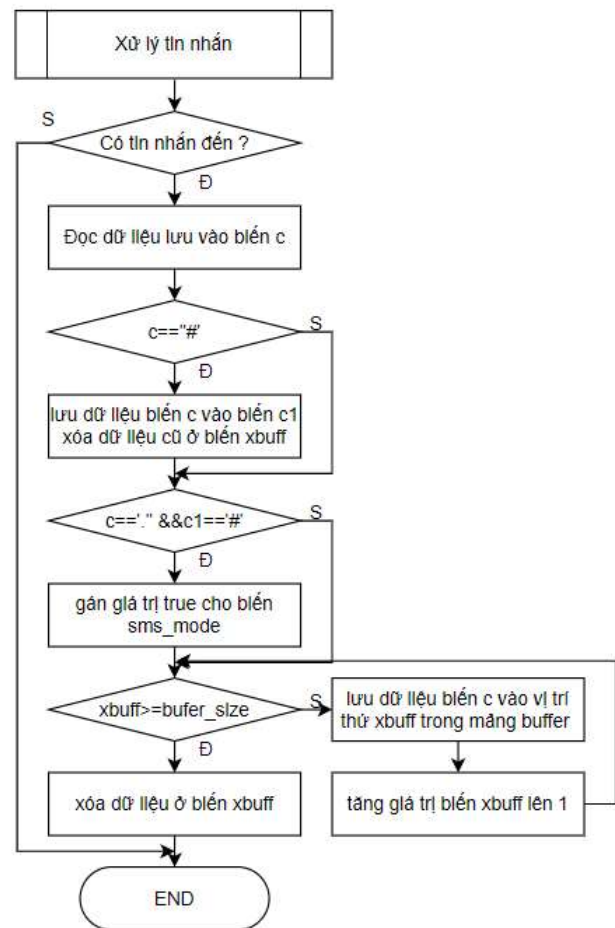
đó gửi trạng thái thiết bị đến số điện thoại được cài đặt sẵn và kết thúc quá trình

- Xử lý nút nhấn: Kiểm tra lần lượt các nút nhấn, nếu điều kiện đúng thì chuyển đổi trạng thái thiết bị và gửi tin nhắn trạng thái hiện tại của thiết bị đến số điện thoại được cài đặt sẵn

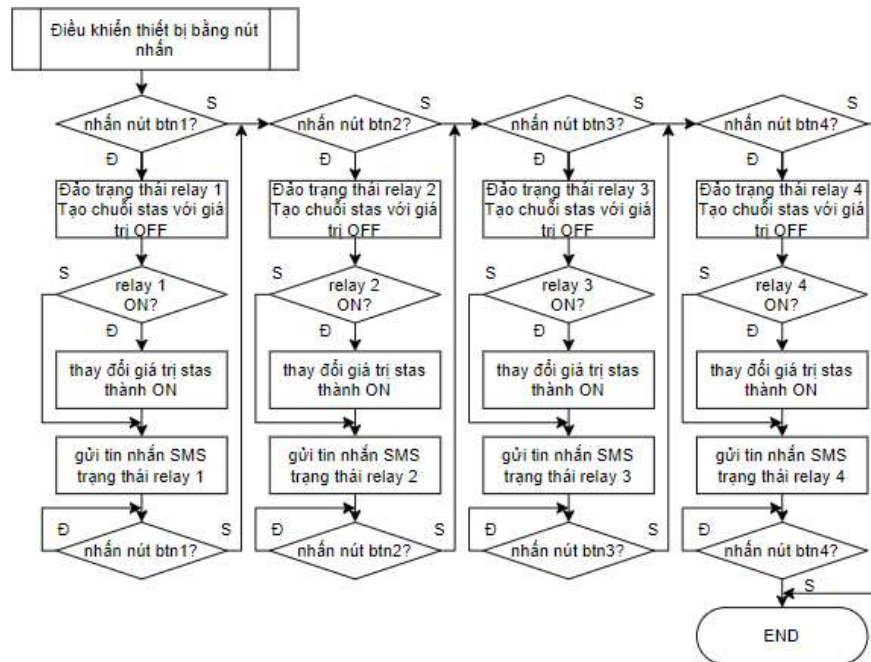
-Xử lý giếng trời: Kiểm tra nếu cảm biến mưa có mưa(mức thấp) thì đóng giếng trời và gửi tin nhắn thông báo mưa,tiếp theo kiểm tra các công tắc hành trình 1 và 2 tương ứng hai chu trình đóng mở, nếu công tắc hành trình 2 chưa có tín hiệu thì có thể đóng giếng trời, ngược lại thì không thể đóng giếng trời tương tự với chu trình mở giếng trời với công tắc hành trình 1.



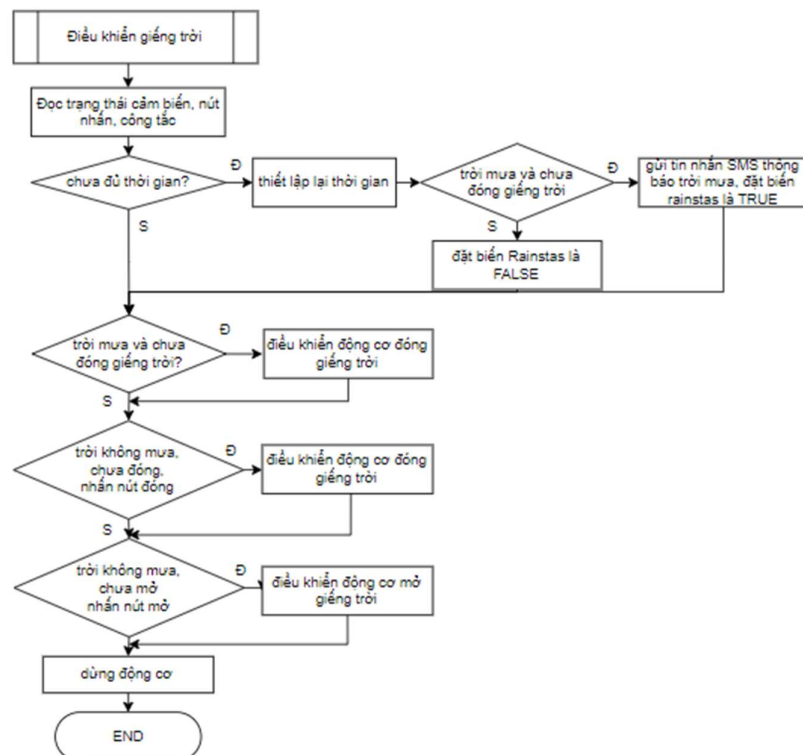
Hình 21: Lưu đồ xử lý tin nhắn



Hình 22: Lưu đồ nhận dữ liệu



Hình 23: Lưu đồ xử lý nút nhấn



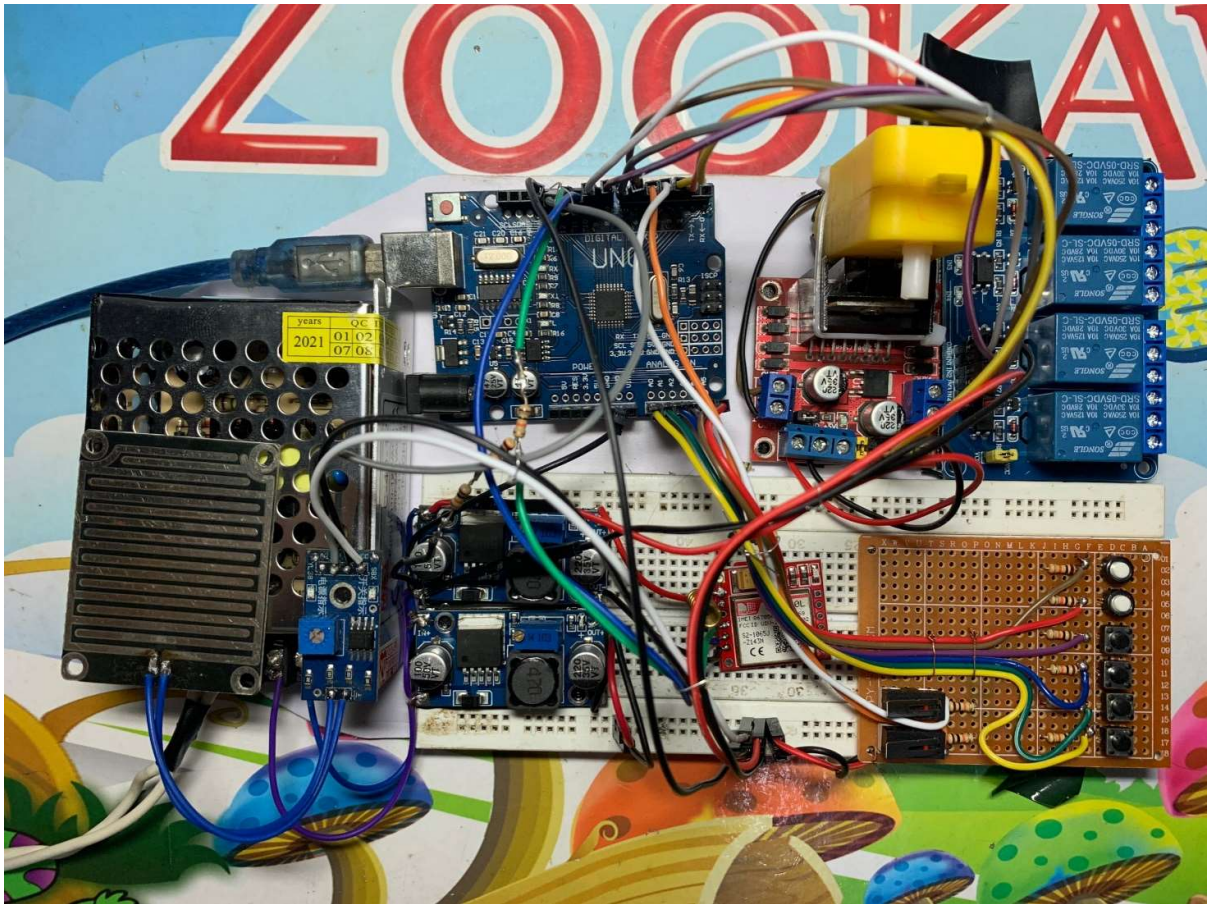
Hình 24: Lưu đồ xử lý giếng trời

## CHƯƠNG 4: KẾT QUẢ THỰC HIỆN

### 4.1. Kết quả phân cứng

Qua quá trình thiết kế phần cứng, chọn lựa linh kiện, em đã tiến hành kiểm tra các kết nối các module của các khối và cho ra kết quả như các hình bên dưới.

Sau khi kiểm tra các module hoạt động ổn định. Em tiến hành kết nối các module và linh kiện lại với nhau và thu được sản phẩm.



Hình 25: Hình ảnh thực tế phần cứng

### 4.2. Kết quả hoạt động hệ thống

Sau khi lắp mạch và hoàn thiện kết nối các linh kiện xong, em sẽ cho mạch hoạt động các chức năng như: điều khiển bật tắt thiết bị bằng nút nhấn, điều khiển thiết bị qua tin nhắn SMS, kiểm tra thông báo cảm biến mưa.

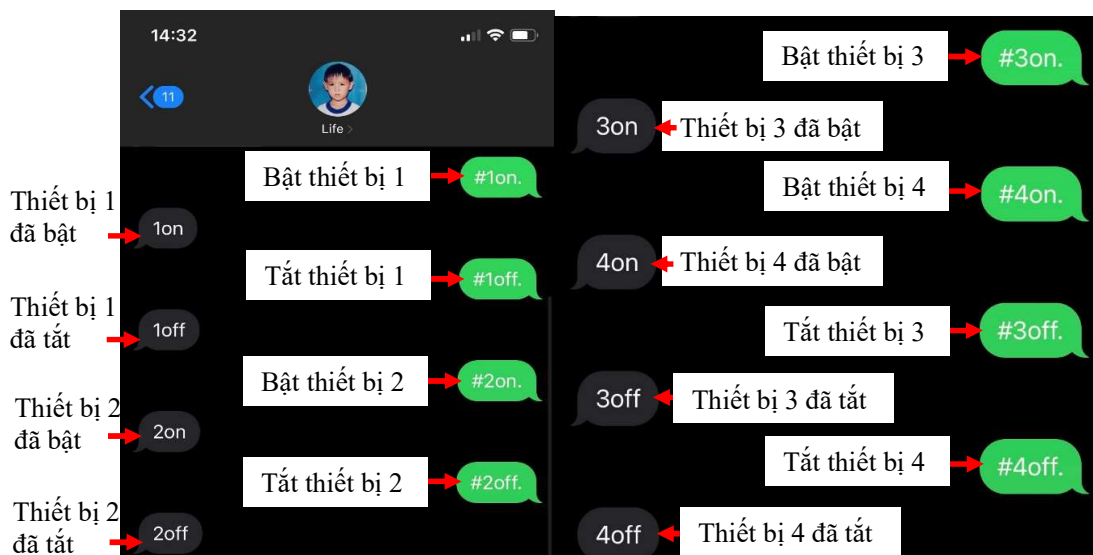
Khi điều khiển bằng tin nhắn SMS, ta có các cú pháp tương ứng với 4 relay:

- Tắt thiết bị 1 : #1off.
- Tắt thiết bị 2: #2off.
- Tắt thiết bị 3: #3off.



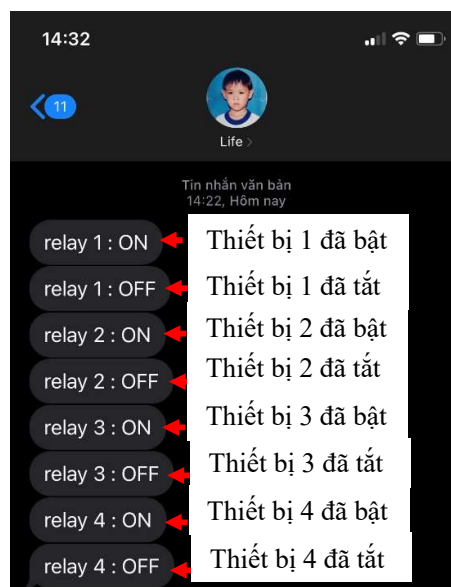
- Tắt thiết bị 4: #4off.
- Bật thiết bị 1 : #1on.
- Bật thiết bị 2: #2on.
- Bật thiết bị 3: #3on.
- Bật thiết bị 4: #4on.

Thiết bị  
đã bật



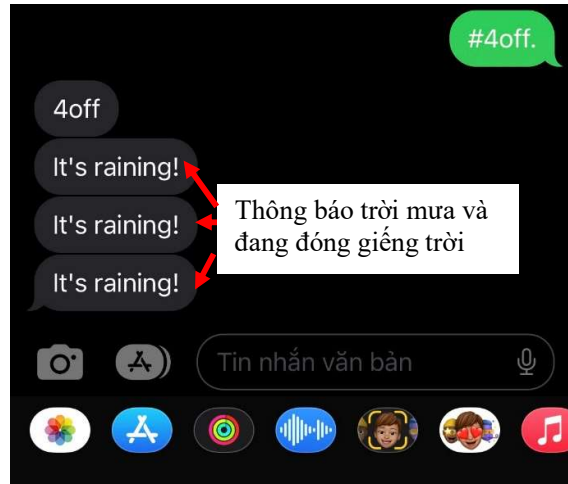
Hình 26: Hình ảnh hoạt động của thiết bị qua tin nhắn SMS

Tương tự, khi nhấn nút tương ứng thì các relay sẽ bật/tắt và gửi tin nhắn báo trạng thái relay về điện thoại.



Hình 27: Hình ảnh hoạt động của thiết bị với nút nhấn

Khi trời mưa, hệ thống sẽ tự đóng giăng trời và thông báo đến điện thoại.



Hình 28: Hình ảnh hoạt động của thiết bị báo trời mưa

Để mô tả rõ hơn hoạt động của sản phẩm em có đính kèm video clip được đính kèm link sau: <https://youtu.be/GuN0BV03Gyc>

## **CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

### **5.1.KẾT LUẬN**

Với sự hỗ trợ tận tình từ giáo viên hướng dẫn về cơ bản đã hoàn thành được những mục tiêu đề ra là thiết kế mạch hiển thị thời gian thực lên led ma trận điều khiển bằng remote hồng ngoại. Trong quá trình thực hiện, tuy gặp không ít khó khăn từ phần cứng đến phần mềm. Mặc dù gặp không ít khó khăn như thế nhưng em đã cố gắng nghiên cứu vượt qua và tích lũy cho mình một số kinh nghiệm mới, kiến thức mới để hoàn thành đề tài.

### **5.2.HẠNH CHẾ VÀ HƯỚNG PHÁT TRIỂN**

#### **5.2.1.Hạ chế**

- Chi phí phần cứng khá cao.
- Hệ thống hoạt động ở vùng có phủ sóng điện thoại di động.
- Hệ thống được lập trình chỉ điều khiển thiết bị dùng tin nhắn SMS dạng text.
- Phải tốn chi phí gửi SMS
- Chưa thiết lập được password để xác định đúng tin nhắn được nhận

#### **5.2.2.Hướng phát triển**

- Cải thiện thuật toán để củng cố mô hình hoàn thiện
- Mở rộng số thiết bị cần điều khiển.
- Lập trình có thể thay đổi pass word để tăng tính bảo mật.
- Mở rộng chức năng cảnh báo sự cố và chống trộm cho hệ thống.
- Có thể điều khiển thiết bị bằng gọi điện thoại trực tiếp

Để khắc phục được các nhược điểm đã nêu ra, tăng tốc độ xử lý và đạt kết quả chính xác cao thì cần phải thay thế khối xử lý trung tâm bằng arduino là một máy tính nhúng được thiết kế trên nền tảng máy tính trí tuệ nhân tạo. Arduino được sử dụng rộng rãi cho mọi ứng dụng lĩnh vực nhằm đem đến những sản phẩm hoàn thiện bằng những thư viện, lệnh, code đơn giản để lập trình chúng.

## TÀI LIỆU THAM KHẢO

- [1] <http://kdientu.duytan.edu.vn/media/49669/tt003-dieu-khien-thiet-bi-qua-sms.pdf>
- [2] <https://datasheet-pdf.com/PDF/SIM800L-Datasheet-SIMCom-989664>
- [3] <https://components101.com/modules/l293n-motor-driver-module>
- [4] [https://www.alibaba.com/product-detail/4-Relay-Module-4-Channel-Relay\\_62059284116.html?spm=a2700.7735675.0.0.7f79dLO7dLO7Ua&s=p](https://www.alibaba.com/product-detail/4-Relay-Module-4-Channel-Relay_62059284116.html?spm=a2700.7735675.0.0.7f79dLO7dLO7Ua&s=p)
- [5] <https://datasheetpdf.com/pdf/1424873/CYStech/LM2596/1>
- [6] <https://www.iostream.vn/article/thong-so-ky-thuat-arduino-uno-r-3-cac-bien-the-va-luu-y-gnlnL>
- [7] <https://huynhnhattung.com/sim800l-giao-tiep-arduino-nhan-tin-goi-dien-sim800l-relay-arduino/>
- [8] <https://lastminuteengineers.com/sim800l-gsm-module-arduino-tutorial/>
- [9] <https://github.com/ahmadlogs/arduino-ide-examples/blob/main/sim800l-registered-phone/sim800l-registered-phone.ino>
- [10] <https://nshopvn.com/product/cam-bien-mua/>



## PHỤ LỤC

```
//Control syntax: #1on.
#include<SoftwareSerial.h>
#define btn0      A0 // Positive low
#define btn1      A1 // Positive low
#define btn2      A2 // Positive low
#define btn3      A3 // Positive low
#define btn4      A4 //opening skylight
#define btn5      A5 //closing skylight
#define mi1       2 //analog
#define mi2       3 //digital
#define ht1       4 //LOW if the skylight is fully open
#define ht2       5 //LOW if the skylight is fully close
#define relay4    6 //device 1
#define relay3    7 //device 2
#define relay2    8 //device 3
#define relay1    9 //device 4
SoftwareSerial mySerial(10,11); // TX, RX
#define rainmodule 12 //Positive low

int answer;
int xbuff=0;
int sms_mode = 0;
const int buffer_size = 160;           // Buffer size you can adjust this
size
char c=0;
char c1=0;
char aux_string[30];
char buffer[buffer_size];
char command[30];
char *ptr1=0;
char phonenumber[]="+84935014779";      // Your phone
phonenumber0868211855
char r1_on[] = "1on";
char r1_of[] = "1off";
char r2_on[] = "2on";
char r2_of[] = "2off";
char r3_on[] = "3on";
```

```

char r3_of[] = "3off";
char r4_on[] = "4on";
char r4_of[] = "4off";
bool Rainstas = false; // variable to remember if rain notification has
been displayed
unsigned long previousMillis = 0; // variable to remember the last time
we checked
const long interval = 100; // check every 0.1 second

//*****
*****

void setup() {
  mySerial.begin(9600);
  Serial.begin(9600);

  pinMode(relay1,OUTPUT);
  pinMode(relay2,OUTPUT);
  pinMode(relay3,OUTPUT);
  pinMode(relay4,OUTPUT);
  pinMode(btn0,INPUT);
  pinMode(btn1,INPUT);
  pinMode(btn2,INPUT);
  pinMode(btn3,INPUT);

  pinMode(mi2,OUTPUT);
  pinMode(mi1,OUTPUT);
  pinMode(rainmodule,INPUT);
  pinMode(ht1,INPUT);
  pinMode(ht2,INPUT);
  pinMode(btn4,INPUT);
  pinMode(btn5,INPUT);

  digitalWrite(relay1,HIGH);
  digitalWrite(relay2,HIGH);
  digitalWrite(relay3,HIGH);
  digitalWrite(relay4,HIGH);
  digitalWrite(mi2,LOW);
  digitalWrite(mi1,LOW);
  Connect_sim800l(); // Khởi động module sim800L

```

```

    Init_sim800l();// thiết lập lệnh cần thiết để gửi nhận tin nhắn cho
module sim800L
    Erase_buffer();// xóa dữ liệu bộ đệm tin nhắn
}
//*****
*****

void loop() {
    Receive_uart();// Nhận dữ liệu từ các module
    Control_devide_sms();
    Control_devide_btn();
    Control_skylight();
}
//*****
*****

void Receive_uart() {
    while(mySerial.available()>0) { //Khi có dữ liệu đến
        c = mySerial.read();
        if(c=='#'){
            c1=c;
            xbuff=0;
        }
        if(c=='.' && c1=='#'){
            sms_mode=true;
        }
        if (xbuff >= buffer_size){
            xbuff=0;
        }
        else{
            buffer[xbuff++]=c;
        }
    }
}
//*****
*****

void Control_devide_btn(){
    if(digitalRead(btn0) == LOW) {
        digitalWrite(relay1, !digitalRead(relay1));
        String stas = "OFF";
        if(!digitalRead(relay1)){ stas = "ON"; }
        send_SMS("relay 1 : " + stas );
    }
}

```

```

    while(digitalRead(btn0) == LOW);
}
if(digitalRead(btn1) == LOW) {
    digitalWrite(relay2, !digitalRead(relay2));
    String stas = "OFF";
    if(!digitalRead(relay2)){ stas = "ON"; }
    send_SMS("relay 2 : " + stas );
    while(digitalRead(btn1) == LOW);
}
if(digitalRead(btn2) == LOW) {
    digitalWrite(relay3, !digitalRead(relay3));
    String stas = "OFF";
    if(!digitalRead(relay3)){ stas = "ON"; }
    send_SMS("relay 3 : " + stas );
    while(digitalRead(btn2) == LOW);
}
if(digitalRead(btn3) == LOW) {
    digitalWrite(relay4, !digitalRead(relay4));
    String stas = "OFF";
    if(!digitalRead(relay4)){ stas = "ON"; }
    send_SMS("relay 4 : " + stas );
    while(digitalRead(btn3) == LOW);
}
}
}
//*****
*****
void Control_devide_sms(){
    if (sms_mode == true){
        ptr1=strstr(buffer,r1_on);
        turn on device 1
        if (strcmp(ptr1,r1_on,3)==0){
            digitalWrite(relay1,LOW);
            send_SMS(r1_on);
        }
        ptr1=strstr(buffer,r1_of);
        if (strcmp(ptr1,r1_of,3)==0){
            digitalWrite(relay1,HIGH);
            send_SMS(r1_of);
        }
        ptr1=strstr(buffer,r2_on);
        // Checking for

```

```

    if (strncmp(ptr1,r2_on,3)==0){
        digitalWrite(relay2,LOW);
        send_SMS(r2_on);
    }
    ptr1=strstr(buffer,r2_of);
    if (strncmp(ptr1,r2_of,3)==0){
        digitalWrite(relay2,HIGH);
        send_SMS(r2_of);
    }
    ptr1=strstr(buffer,r3_on);
    if (strncmp(ptr1,r3_on,3)==0){
        digitalWrite(relay3,LOW);
        send_SMS(r3_on);
    }
    ptr1=strstr(buffer,r3_of);
    if (strncmp(ptr1,r3_of,3)==0){
        digitalWrite(relay3,HIGH);
        send_SMS(r3_of);
    }
    ptr1=strstr(buffer,r4_on);
    if (strncmp(ptr1,r4_on,3)==0){
        digitalWrite(relay4,LOW);
        send_SMS(r4_on);
    }
    ptr1=strstr(buffer,r4_of);
    if (strncmp(ptr1,r4_of,3)==0){

        digitalWrite(relay4,HIGH);
        send_SMS(r4_of);
    }
    Erase_buffer();
}
}
//*****
*****

void Control_skylight() {
    bool Raining = (digitalRead(rainmodule) == LOW);
    bool ht1_stas = (digitalRead(ht1) == LOW);
    bool ht2_stas = (digitalRead(ht2) == LOW);

```

```

bool btn4_pressed = (digitalRead(btn4) == LOW);
bool btn5_pressed = (digitalRead(btn5) == LOW);

unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;
    if (Raining && !ht2_stas) {
        if (!Rainstas) {
            send_SMS("It's raining!");
            Rainstas = true;
        }
    }
    else { Rainstas = false; }
}
if (Raining && !ht2_stas) {
    analogWrite(mi2, 70);
    digitalWrite(mi1, LOW);
} else if (!Raining && !ht2_stas && btn5_pressed) {
    analogWrite(mi2, 70);
    digitalWrite(mi1, LOW);
} else if (!Raining && !ht1_stas && btn4_pressed) {
    analogWrite(mi2, 180);
    digitalWrite(mi1, HIGH);
} else {
    analogWrite(mi2, 0);
    digitalWrite(mi1, LOW);
}
}
//*****
*****

int sendAT(char* ATcommand, char* expected_answer, unsigned int timeout)
{
    unsigned long start_time = millis();
    mySerial.println(ATcommand);
    Serial.println(ATcommand);
    while ((millis() - start_time) < timeout) {
        if (mySerial.find(expected_answer)) {
            Serial.println(expected_answer);
            return 1;
        }
    }
}

```

```

    }
    Serial.println(expected_answer);
    return 0;
}

//*****
*****

void send_SMS(const String msg) {
    sprintf(command,"AT+CMGS=\"%s\"",phonenumber);
    answer = sendAT(command,"Sending",1000);
    mySerial.println(msg);
    mySerial.write(26);// Ctrl + Z
}

//*****
*****

void Connect_sim8001() {
    answer = sendAT("AT", "OK", 2000);
    if (answer == 0) {
        unsigned long start_time = millis();
        while ((millis() - start_time) < 6000) { // Wait for response with
timeout
            answer = sendAT("AT", "OK", 2000);
            if (answer == 1) {
                break;
            }
        }
    }
}

//*****
*****

void Init_sim8001() {
    answer = 0;
    while (answer == 0) {
        answer = sendAT("AT+CREG?", "+CREG: 0,1",500); // Checking status of
Connecting to the network
    }
    answer = 0;
    while (answer == 0) {
        answer = sendAT("AT+CMGF=1", "OK",1000); // Set SMS into text mode
    }
}

```

```

    answer = 0;
    while (answer == 0) {
        answer = sendAT("AT+CNMI=2,2,2,0,0", "OK",1000); // Set Message mode
when receive new SMS
    }
    sprintf(aux_string,"ATD%s;",phonenumber);
    answer = 0;
    while (answer == 0) {
        answer = sendAT(aux_string, "OK", 2000); // Set Message mode when
receive new SMS
    }
    delay(10000);
    answer = 0;
    while (answer == 0) {
        answer = sendAT("ATH", "OK",2000); // Set Message mode when receive
new SMS
    }
}
//*****
*****

void Erase_buffer() {
    memset(buffer, 0, buffer_size);
    memset(ptr1, 0, sizeof(ptr1));
    memset(command, 0, sizeof(command));
    xbuff = 0;
    c = 0;
    c1 = 0;
    sms_mode = false;
}
//*****
*****

```