



TDD-FDP-CCE

Design Project (Technical Design Document)

Project Name	Hệ thống cảm nhận rung chấn, rung lắc cảnh báo thiên tai động đất.		
Student	[1] Nguyen Tran Thanh Trung [2] Vu Quoc Viet	ID	20119302 20119308
Major	Embedded Systems	Supervisor	Assoc. Prof. Phan Van Ca

Mục Lục

1	Nhận diện vấn đề cần giải quyết.....	6
2	Mục tiêu thiết kế.....	6
3	Đặc tả các yêu cầu.....	7
3.1	Yêu cầu người dùng.....	7
3.2	Yêu cầu kỹ thuật.....	7
3.2.1	Chức năng kỹ thuật.....	7
3.2.2	Thông số kỹ thuật.....	8
4	Thiết kế kiến trúc hệ thống.....	9
4.1	Cơ sở lý thuyết.....	9
4.1.1	Microcontroller ESP32.....	9
4.1.2	Module Buck hạ áp 5V/3A.....	13
4.1.3	Module SW420.....	15
4.1.4	Module I2C và LCD1602.....	15
4.2	Sơ đồ khối hệ thống.....	23
5	Thiết kế chi tiết.....	24
5.1	Thiết kế phần cứng.....	24
5.2	Lưu đồ giải thuật khối xử lý trung tâm.....	25
6	Thi công mô hình.....	Error! Bookmark not defined.
7	Kết luận.....	30
7.1	Kết quả và đánh giá.....	30
7.2	Hướng phát triển đề tài.....	30
7.3	Kết luận.....	31
8	Phụ Lục.....	Error! Bookmark not defined.
8.1	Bảng phân công.....	Error! Bookmark not defined.
8.2	Danh mục vật tư.....	Error! Bookmark not defined.
8.3	Biểu đồ hoạt động.....	Error! Bookmark not defined.
8.4	Hướng dẫn sử dụng.....	Error! Bookmark not defined.

Revision History

Version	Date	Content of revision	Author	Supervisor
1.0	15/12/2022			
1.1				
1.2				
1.3				
1.4				
1.5				
2.0				

■ Terms and abbreviations

[UART]	Universal asynchronous receiver-transmitter.
[USB]	Universal Serial Bus.
[IDE]	Integrated development environment.

■ References

- [1] DTTC, "Buck hạ áp DC-DC 5V/3A", [Online]. Available: <https://linhkienthanhcong.com/mach-buck-ha-ap-dc-dc-5v-3a-tich-hop-cong-usb>.
- [2] DEVIOT, "Tổng quan về ESP32", [Online]. Available: <https://deviot.vn/tutorials/esp32.66047996/tong-quan-ve-esp32.18482631>
- [3] Adrunio kit, "Tổng quan về LCD1602 và giao tiếp I2C", [Online]. Available: <https://arduinoakit.vn/giao-tiep-i2c-lcd-arduino/>
- [4] Robocon, "Module cảm biến rung SW-420", [Online]. Available: <http://robocon.vn/detail/mdl78-module-cam-bien-rung-sw-420.html#:~:text=SW%2D420%20%C3%A0%20m%E1%BB%99t%20trong,khi%20kh%C3%B4ng%20c%C3%B3%20rung%20%C4%91%E1%BB%99ng>.
- [5] ALLDATASHEET, " Datasheet LCD16X2 (PDF)", [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/123205/VISHAY/LCD160G032A.html>
- [6] ALLDATASHEET, " Datasheet ESP32 DEV KIT V1 (PDF)", [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1424862/ESPRESSIF/ESP32-C3-DEVKITM-1.html>

■ List of Tables & Figures

Mục lục bảng:

Bảng 8.1 Bảng phân công.....	32
Bảng 8.2 Danh mục vật tư.....	33
Bảng 8.3 Biểu đồ hoạt động.....	33

Mục lục hình ảnh:

Hình 4.1. Hình ảnh thực tế của Microcontroller ESP32 DEV KIT V1.....	10
Hình 4.2: Sơ đồ chân kết nối Microcontroller ESP32 DEV KIT V1.....	12
Hình 4.3: Sơ đồ nguyên lý mạch Microcontroller ESP32.....	12
Hình 4.4: Hình ảnh thực tế của module Buck hạ áp DC-DC 5V/3A.....	13
Hình 4.5. Sơ đồ chân của Module Buck hạ áp DC-DC 5V/3A.....	14
Hình 4.6 Sơ đồ chân của SW-420.....	15
Hình 4.7 Sơ đồ nguyên lý của cảm biến rung SW-420.....	16
Hình 4.8 Hình ảnh thực tế Module I2C.....	18
Hình 4.9 Sơ đồ chân Module I2C.....	19
Hình 4.10 Hình ảnh thực tế của LCD16x2 đã được tích hợp I2C.....	20
Hình 4.11 Sơ đồ chân của LCD16x2.....	21
Hình 4.12 Sơ đồ khối của hệ thống.....	23
Hình 4.13 Sơ đồ nối dây hệ thống báo động rung chấn.....	24
Hình 4.14 Lưu đồ giải thuật của hệ thống.....	25
Hình 5.1 Mô hình sản phẩm khi bật nguồn.....	28
Hình 5.2 Mô hình sản phẩm khi phát ra cảnh báo.....	29

1. Nhận diện vấn đề:

Môi trường tự nhiên ngày càng bị con người tác động và tàn phá làm ô nhiễm khiến cho trái đất xảy ra hiện tượng nhà kính, con người chính là nguyên nhân lớn nhất dẫn đến sự nóng lên toàn cầu và biến đổi khí hậu. Ngày càng có nhiều thiên tai xảy ra trên chính mái nhà của chúng ta, giông bão, động đất, sóng thần,... Khiến cho hàng nghìn hàng triệu người thiệt mạng mỗi năm. Đây là vấn đề lớn nhiều quốc gia phải đau đầu đối mặt tìm mọi phương pháp để bảo vệ tính mạng người dân, sự an toàn của nhân loại. Vì vậy nhóm em sẽ tìm hiểu và phát triển đề tài “Hệ thống cảm nhận rung chấn, rung lắc cảnh báo thiên tai động đất” triển khai được một hệ thống cảm nhận được rung chấn từ mặt đất giúp cho con người kịp thời di tản tránh được các thiên tai một cách sớm nhất có thể, giảm thiệt hại về tài sản và tính mạng con người.

2. Mục tiêu thiết kế

Các rung chấn từ môi trường xung quanh sẽ được cảm biến SW-420 cảm nhận và gửi dữ liệu từ môi trường vào ESP32 là vi điều khiển trung tâm để nhận biết và điều khiển Buzzer báo động và đèn LED sẽ nhấp nháy cảnh báo. Xuất ra kí tự cảnh báo trên màn hình LCD. Từ các thông số đó ta sẽ thiết kế được một hệ thống cảm nhận rung chấn từ động đất thiên tai cảnh báo cho người sử dụng.

3. Đặc tả yêu cầu

3.1 Yêu cầu người dùng

Trong trường hợp thiết bị phát hiện có rung chấn (từ cấp 4 trở lên theo thang đo MKS) thì sẽ phát âm thanh cảnh báo.

- Thiết bị sẽ phát hiện trực tiếp các chấn động của trận động đất và ngay lập tức thông báo qua màn hình LCD của sự xuất hiện rung chấn của trận động đất hoặc thiên tai .
- Khi bật thiết bị, cảm biến đo rung chấn sẽ hoạt động và gửi các mức độ rung chấn mà cảm biến đo được vào bộ xử lý trung tâm sau đó sẽ điều khiển các module hoạt động một cách chính xác.
- Hệ thống báo động có nguồn cung cấp độc lập hoạt động tốt trong các tình huống khẩn cấp, mất điện.
- Giao diện hiển thị thông báo ngắn gọn, đọc hiểu dễ dàng để mọi người lập tức di tản.
- Có âm thanh cảnh báo to, rõ, dồn dập liên hồi có đèn LED nhấp nháy cảnh báo để mọi người nhận biết được nguy hiểm.
- Thiết bị hoạt động liên tục cho đến khi không còn rung chấn thì tự động tắt còi và đèn nhấp nháy cảnh báo.
- Hệ thống có nút nhấn để ngắt khi có va chạm do người dùng mà không phải rung chấn từ động đất hay thiên tai khiến hệ thống hiểu nhầm.

3.2 Yêu cầu kỹ thuật

3.2.1 Chức năng kỹ thuật

- Nghiên cứu và phát triển mô hình phát hiện động đất, thiên tai và phát ra cảnh báo cho người dân di tản kịp thời trong bán kính 10km.
- Ứng dụng hệ điều hành thời gian thực và các ứng dụng gần gũi với người dùng để tạo nên hệ thống cảnh báo thiệt hại của thiên tai, động đất.
- Dùng ESP32 làm công cụ lập trình và thực thi các yêu cầu người dùng đề ra.
- Chức năng chính của thiết bị là đo rung chấn, phát âm thanh cùng với đèn nhấp nháy cảnh báo và hiển thị cảnh báo trên màn hình LCD
- Cảm biến rung được kết nối với module ESP32 lắp đặt trong các công trình hoặc tòa nhà để phát hiện được các tín hiệu rung chấn sớm nhất có thể gây ra thiệt hại bởi thiên tai.

3.2.2 Thông số kỹ thuật

- Điện áp làm việc: 5V DC - 2A.
- Điện áp đầu vào: Pin 9V.
- Nhiệt độ hoạt động: -10°C - 60°C.
- Kích thước: 160 x 93 x 50 mm.
- Trọng lượng: ~500g.
- Thời gian đáp ứng: 0.2s - 0.5s.
- Dải đo độ rung chấn: cấp 1 - cấp 12 (1 - trên 8 độ Richter)
- Độ chính xác: ± 2 Richter.

4. Thiết kế kiến trúc hệ thống:

4.1. Cơ sở lý thuyết

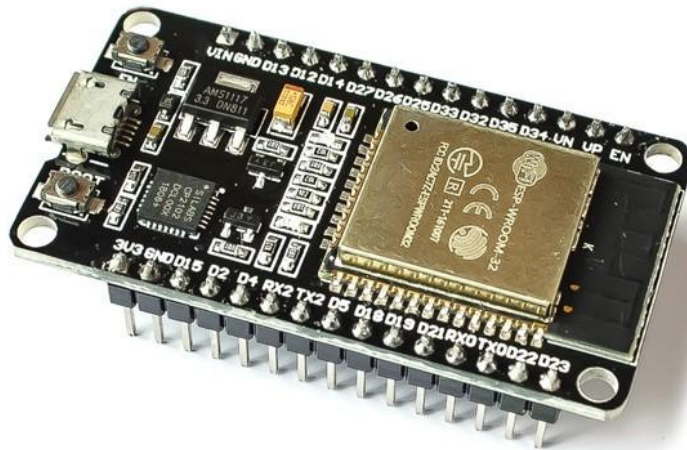
4.1.1. Microcontroller ESP32

Giới thiệu chung

- ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.
- Điểm mạnh về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại nhận tiếng ồn thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.
- Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng, tiện lợi.

ESP32 DevKit - Bo phát triển ESP32

- Espressif Systems đã phát hành một số module dựa trên ESP32 và một trong những lựa chọn phổ biến là module ESP-WROOM-32. Nó bao gồm SoC ESP32, bộ dao động tinh thể 40 MHz, IC Flash 4 MB và một số linh kiện thụ động.
- Ưu điểm của module ESP-WROOM-32 là PCB có các cạnh đục. Nhờ đó các nhà sản xuất bên thứ ba có thể lấy module ESP-WROOM-32 và thiết kế một bo break-out cho module này.
- Một trong những bo như vậy là ESP32 DevKit Board. Nó chứa ESP-WROOM-32 làm module chính và một số phần cứng bổ sung để dễ dàng lập trình ESP32 và tạo kết nối với các chân GPIO.



Hình 4.1. Hình ảnh thực tế của Microcontroller ESP32 DEV KIT V1

Cấu hình chi tiết của Microcontroller ESP32 DEV KIT V1

❖ CPU

- Xtensa Dual-Core LX6 microprocessor.
- Chạy hệ 32 bit
- Tốc độ xử lý 160MHz up to 240 MHz
- Tốc độ xung nhịp đọc flash chip 40MHz → 80MHz (tùy chỉnh khi lập trình)
- RAM: 520 KByte SRAM
- 520 KB SRAM liền chip – (trong đó 8 KB RAM RTC tốc độ cao – 8 KB RAM RTC tốc độ thấp (dùng ở chế độ DeepSleep).

❖ Hỗ trợ 2 giao tiếp không dây

- Wi-Fi: 802.11 b/g/n/e/i
- Bluetooth: v4.2 BR/EDR and BLE

❖ Các loại giao tiếp với GPIO vật lý

- 8-bit DACs(digital to analog) 2 cổng
- Analog(ADC) 12-bit 16 cổng.
- I²C – 2 cổng
- UART – 3 cổng
- SPI – 3 cổng (1 cổng cho chip FLASH)
- I²S – 2 cổng

- SD card /SDIO/MMC host
- Slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN bus 2.0
- IR (TX/RX)
- Băm xung PWM (tất cả các chân)
- Ultra low power analog pre-amplifier'

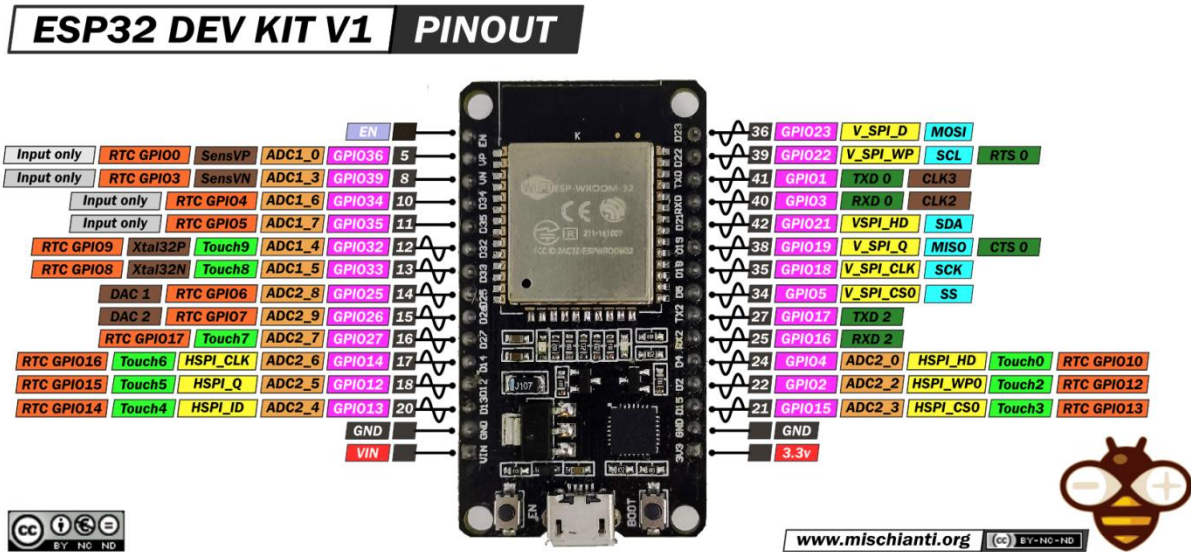
❖ *Bảo mật*

- IEEE 802.11 standard security features all supported, including WFA, WPA/WPA2 and WAPI
- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration: AES, SHA-2, RSA, elliptic curve cryptography (ECC), random number generator (RNG)

❖ *Nguồn điện hoạt động*

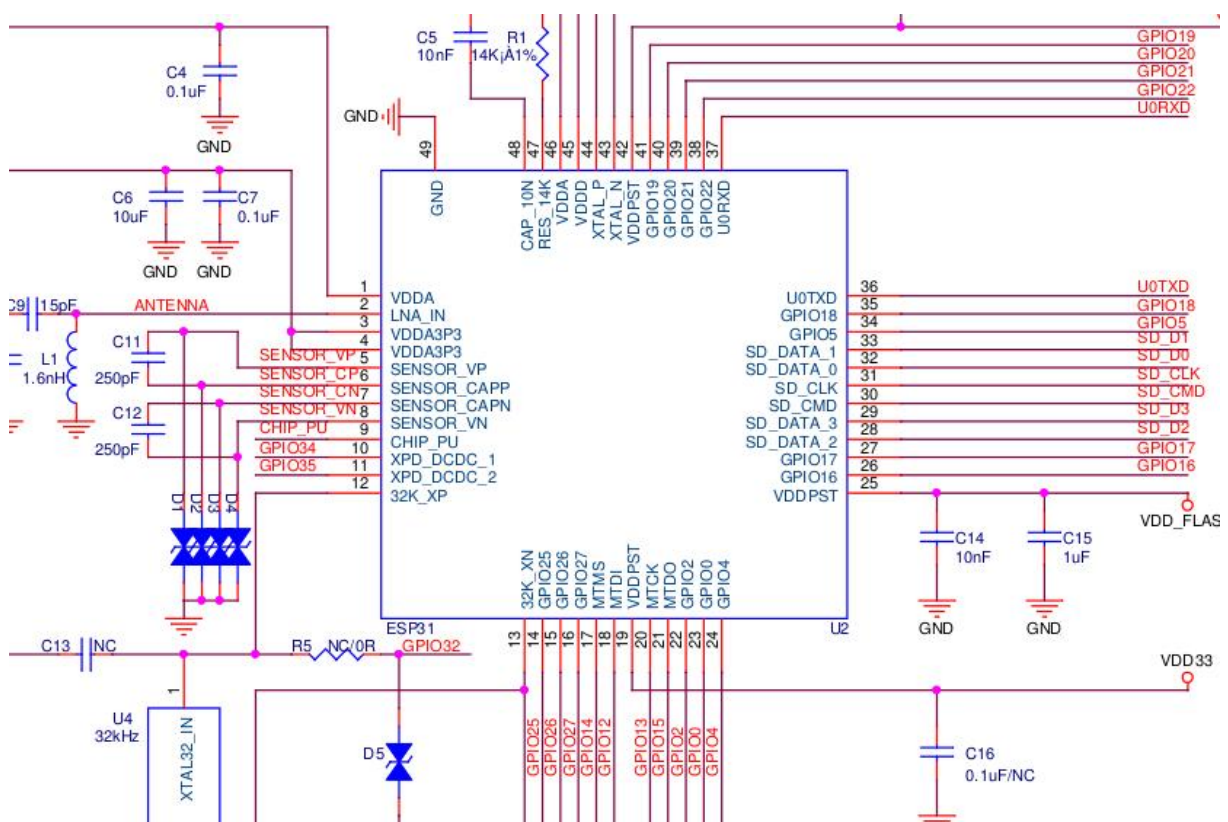
- Nhiệt độ hoạt động -40 + 85C
- Điện áp hoạt động: 2.2-3.6V
- Số cổng GPIOs : 36

Sơ đồ chân



Hình 4.2 Sơ đồ chân kết nối Microcontroller ESP32 DEV KIT V1.

Sơ đồ nguyên lí



Hình 4.3 Sơ đồ nguyên lý mạch Microcontroller ESP32

Thông số kỹ thuật

- Vi điều khiển: **ESP32 DEV KIT V1**.
- Điện áp cấp 5VDC qua cổng Micro USB sẽ được chuyển đổi thành 3.3VDC qua IC nguồn và cấp cho Vi điều khiển chính.
- GPIO giao tiếp mức: 3V3.
- Tích hợp mạch nạp và giao tiếp UART CP2102.
- Tích hợp Led Status, nút BOOT và RESET
- Kích thước: 55mm x 28mm.

4.1.2. Module Buck hạ áp DC-DC 5V/3A

Giới thiệu chung

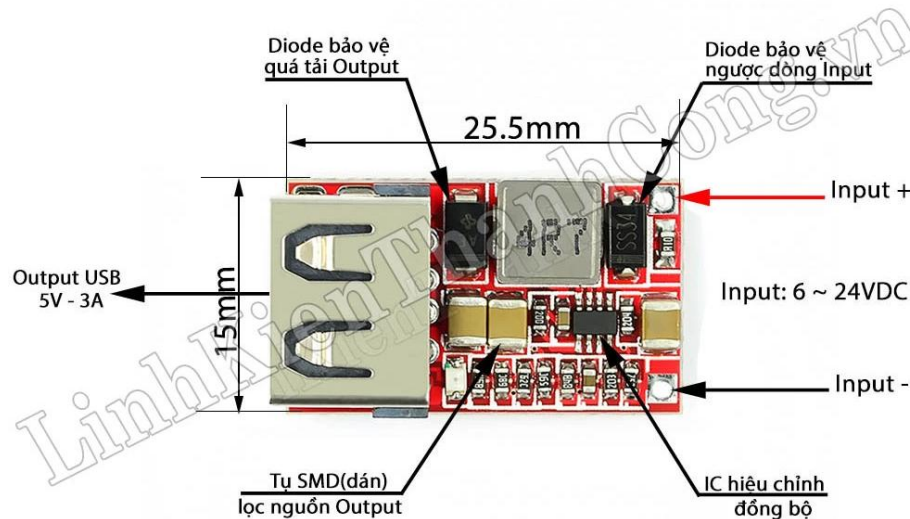
- Mạch BUCK là mạch chuyên để hạ áp DC-DC xuống mức thấp hơn để phù hợp với các nhu cầu sử dụng trong các thành phần điện tử.
- Sản phẩm có kích thước nhỏ gọn và rất chắc chắn, các linh kiện bố trí rất gọn gàng và hợp lý tạo nên một sản phẩm chất lượng khi tới tay người dùng.
- Mạch có chức năng chuyển đổi điện áp đầu vào từ khoảng 6V đến 24VDC xuống 5VDC ổn định với dòng điện ngõ ra lên đến 3A, rất phù hợp cho việc sạc pin cho điện thoại di động hoặc cấp nguồn cho các thiết bị điện tử khác.
- Tùy thuộc vào mục đích sử dụng mà chúng ta sẽ chọn cho mình một bộ sản phẩm phù hợp.



LinhKienThanhCong

Hình 4.4 Hình ảnh thực tế của module Buck hạ áp DC-DC 5V/3A

Sơ đồ chân



Hình 4.5 Sơ đồ chân của Module Buck hạ áp DC-DC 5V/3A.

Chức năng các chân:

- Input: Nguồn vào được cấp từ nguồn pin 9V.
- Output USB: Cho ra nguồn 5V/3A. sẽ được cấp vào cho Microcontroller ESP32.

Thông số kỹ thuật

- Điện áp đầu vào: 6 - 24VDC.
- Điện áp ra: 5.1 - 5.2VDC.
- Hiệu năng: 97.5%.
- Độ gợn sóng ngõ ra: 10mV (MAX).
- Nhiệt độ hoạt động: -40°C ~ 85°C.
- Sai số điện áp: $\pm 0,5\%$.
- Đầu ra: Cổng USB.

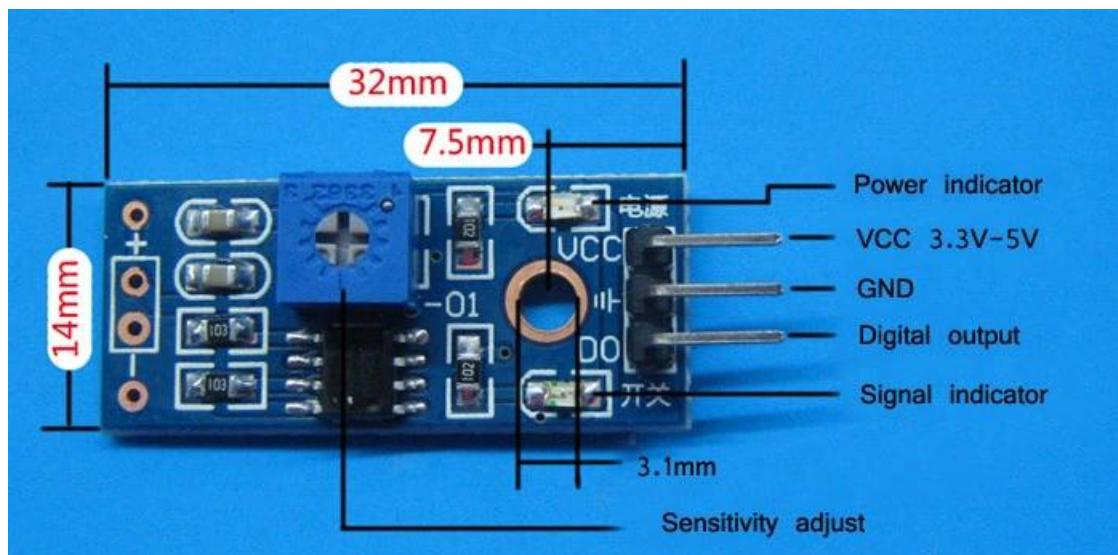
4.1.3. Module SW-420

Giới thiệu chung

Module cảm biến rung sử dụng bộ cảm biến SW-420 được dùng để phát hiện sự rung động trong một ngưỡng xác định từ mọi góc độ. (Ngưỡng rung có thể điều chỉnh bằng biến trở trên board). Kết quả đầu ra logic cao khi cảm biến không được kích hoạt, ngõ ra mức thấp khi được kích hoạt. Board sử dụng điện áp 5V.

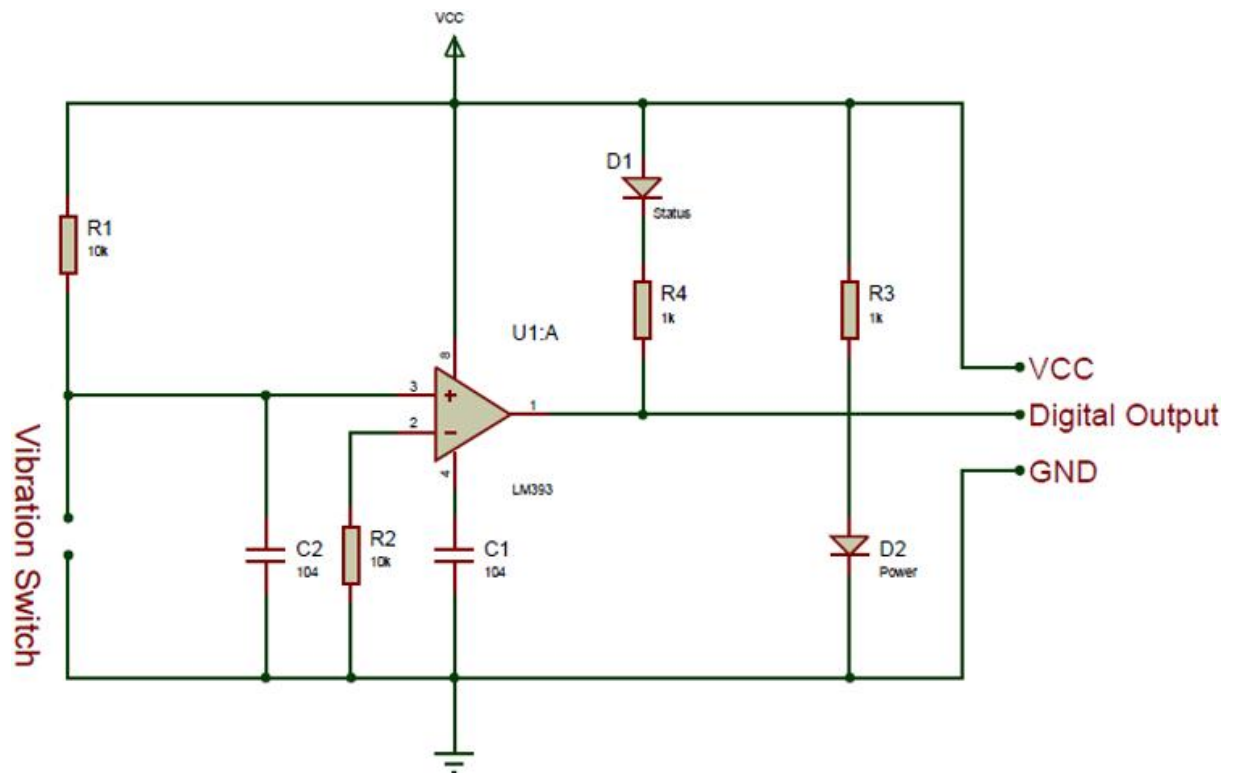
Module được thiết kế nhỏ gọn và dễ dàng sử dụng, tương thích với board IoT Maker UnoX, các Arduino. Mục đích dùng để sử dụng trong các ứng dụng như: thiết bị cảnh báo rung chấn, thiết bị chống trộm, cảnh báo động đất, xe thông minh,...

Sơ đồ chân



Hình 4.6 Sơ đồ chân của SW-420

Sơ đồ nguyên lý



Hình 4.7 Sơ đồ nguyên lý của cảm biến rung SW-420

Kết nối chân

Pin	Chức năng	Mô tả
V _{CC}	Cấp nguồn	Cấp điện áp từ 3.3V – 5V
GND	Nối Mass	Nối Mass
D0	Ngõ ra số	Ngõ ra tín hiệu Digital cảm biến

- Chân V_{CC} được nối vào nguồn 3V3 trên kit ESP32.
- Chân GND được nối vào chân GND trên kit ESP32.
- Chân D0 được nối vào GPIO(14) của vi điều khiển ESP32.

Thông số kỹ thuật

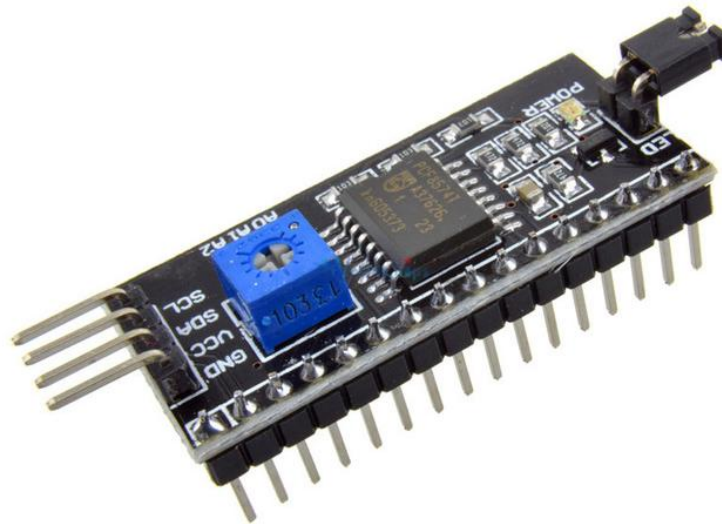
- Điện áp hoạt động: 3.3V – 5V.
- Tín hiệu ngõ ra: Digital, 15mA.
- Điện áp đầu ra ổn định mức 0/1.
- Trạng thái ngõ ra mặc định: LOW.
- Kích thước Board: 3.2cm x 1.4 cm.

4.1.4. Module I2C và LCD 16x2

Giới thiệu chung

❖ Module I2C

- Module I2C là module chuyển đổi từ giao tiếp song song sang giao tiếp I2C cho LCD 1602, LCD 2004.
- Thông thường giao tiếp LCD 1602, LCD 2004 với Arduino cần tới 7 pin IO. Với việc gắn thêm Module I2C vào sẽ giúp tiết kiệm Pin kết nối (chỉ 4 pin) khi giao tiếp LCD với Arduino, Raspberry, Microcontroller. Đồng thời đem lại sự tiện dụng khi thực hành giao tiếp LCD.



Hình 4.8 Hình ảnh thực tế Module I2C.

Sơ đồ chân



Hình 4.9 Sơ đồ chân Module I2C.

Kết nối chân

Pin	Chức năng	Mô tả
V _{CC}	Cấp nguồn	Cấp điện áp 3V3 từ vi điều khiển ESP32
GND	Nối Mass	Nối chân GND của vi điều khiển ESP32
SDA	Chân truyền dữ liệu	Kết nối GPIO(21)
SCL	Chân truyền dữ liệu	Kết nối GPIO(22)

Thông số kỹ thuật

- Tương thích với màn hình: LCD 1602, LCD2004.
- Nguồn cung cấp: +5V DC.
- Hỗ trợ giao thức: I2C.
- Điều chỉnh độ sáng đèn nền và độ tương phản qua biến trở.
- Chân kết nối: SDA, SCL, VCC, GND.
- Kích thước: 41.5mm x 19mm x 16mm.

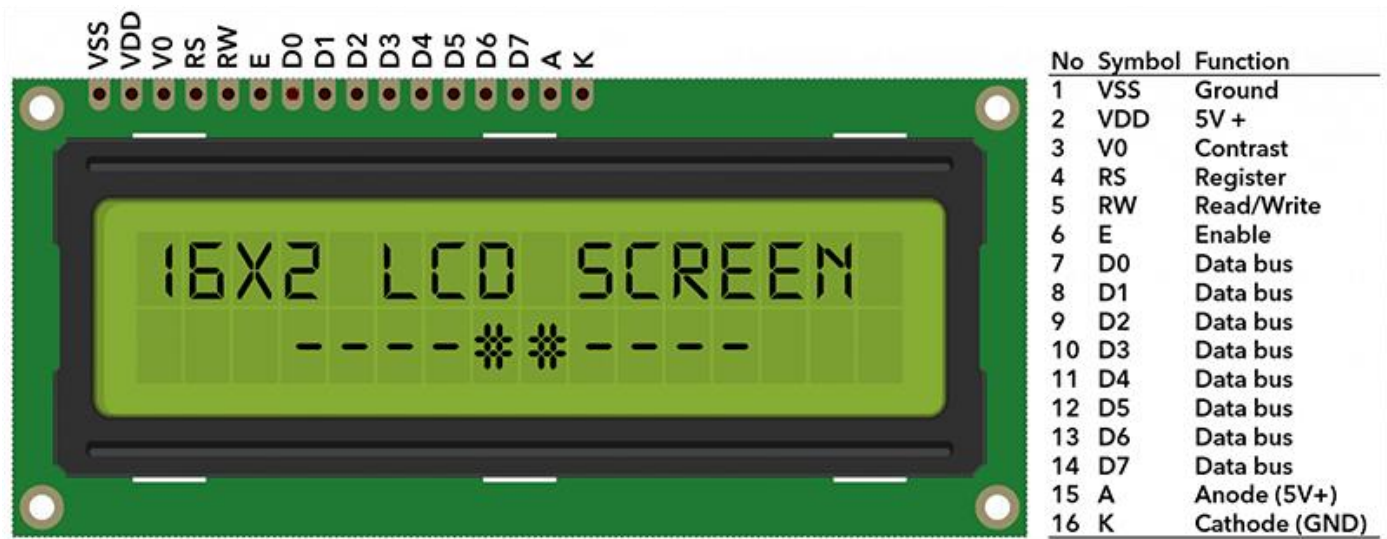
❖ Module LCD 16x2

- Hiện giờ, thiết bị hiển thị LCD 1602 (Liquid Crystal Display) được dùng trong rất nhiều các ứng dụng của vi điều khiển. LCD 1602 có rất nhiều ưu điểm so với các dạng hiển thị khác như: khả năng hiển thị kí tự đa dạng (kí tự đồ họa, chữ, số,); đưa vào mạch ứng dụng theo nhiều giao thức giao tiếp khác nhau dễ dàng , tiêu tốn rất ít tài nguyên hệ thống, giá thành rẻ,...



Hình 4.10 Hình ảnh thực tế của LCD16x2 đã được tích hợp I2C.

Sơ đồ chân



Hình 4.11 Sơ đồ chân của LCD16x2.

Chức năng từng chân

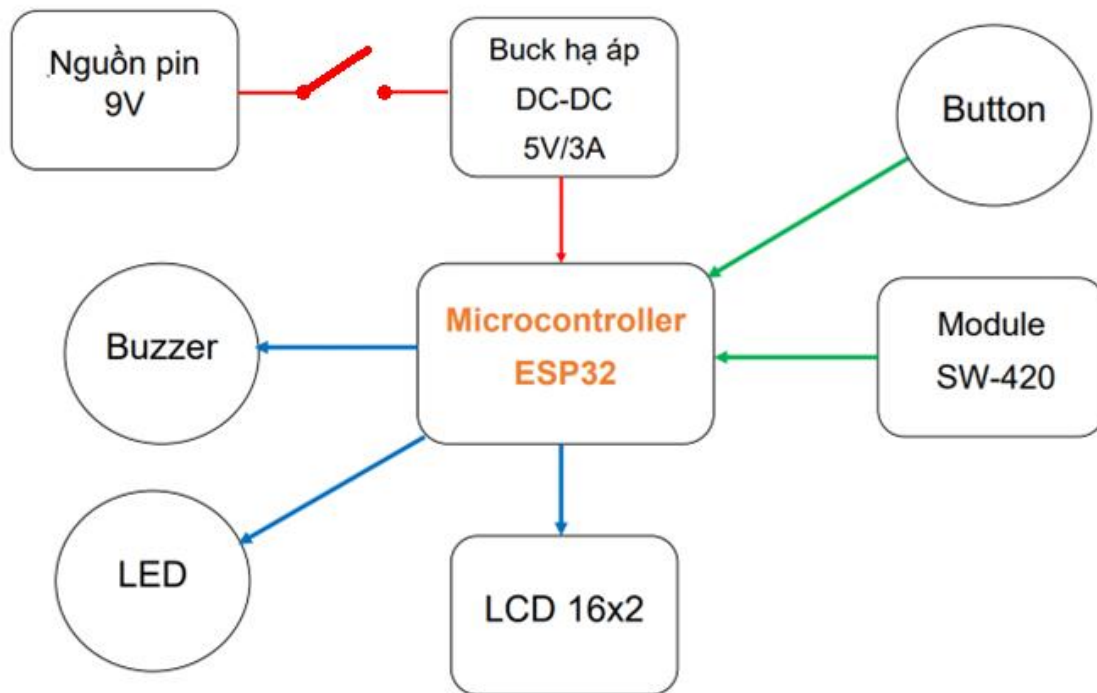
- Chân số 1 - VSS : chân nối đất cho LCD được nối với GND của mạch điều khiển
- Chân số 2 - VDD : chân cấp nguồn cho LCD, được nối với VCC=5V của mạch điều khiển
- Chân số 3 - V0 : điều chỉnh độ tương phản của LCD
- Chân số 4 - RS : chân chọn thanh ghi, được nối với logic "0" hoặc logic "1":
 - + Logic "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
- Chân số 5 - R/W : chân chọn chế độ đọc/ghi (Read/Write), được nối với logic "0" để ghi hoặc nối với logic "1" đọc
- Chân số 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 này như sau:
 - + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào thanh ghi bên trong 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.
- Chân số 7 đến 14 - D0 đến D7: 8 đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này là: Chế độ 8 bit (dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7) và Chế độ 4 bit (dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7).
- Chân số 15 - A : nguồn dương cho đèn nền.

- Chân số 16 - K : nguồn âm cho đèn nền mức thấp.

Thông số kĩ thuật

- Điện áp MAX : 7V
- Điện áp MIN : - 0,3V
- Điện áp ra mức thấp : <0.4V
- Điện áp ra mức cao : > 2.4
- Hoạt động ổn định : 2.7-5.5V
- Dòng điện cấp nguồn : 350uA - 600uA
- Nhiệt độ hoạt động : - 30 - 75 độ C

4.2. Sơ đồ khối hệ thống

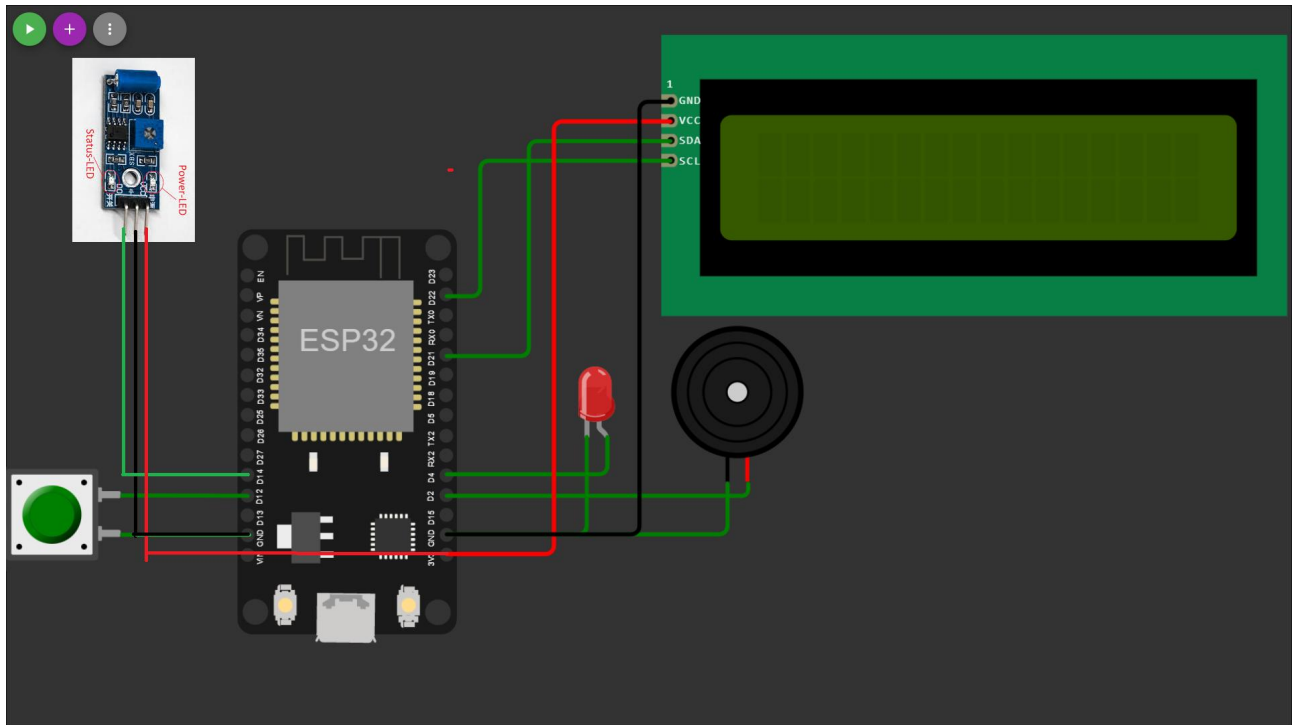


Hình 4.12 Sơ đồ khối của hệ thống.

- ESP32 DEV KIT V1: Vi điều khiển trung tâm cho hệ thống.
- Module SW-420: Cảm biến phát hiện rung chấn để cảnh báo thiên tai.
- Module LCD16X2: Hiển thị cảnh báo cho người sử dụng khi phát hiện rung chấn.
- Buck hạ áp 5V/3A: Hạ từ nguồn pin 9V xuống 5V để cấp nguồn ổn định cho vi điều khiển hoạt động được trong tình huống khẩn cấp, mất điện.
- BUZZER sẽ phát động âm thanh báo động di tản.
- LED nhấp nháy báo động tình trạng khẩn cấp khi có rung chấn.

5. Thiết kế chi tiết

5.1 Thiết kế phần cứng

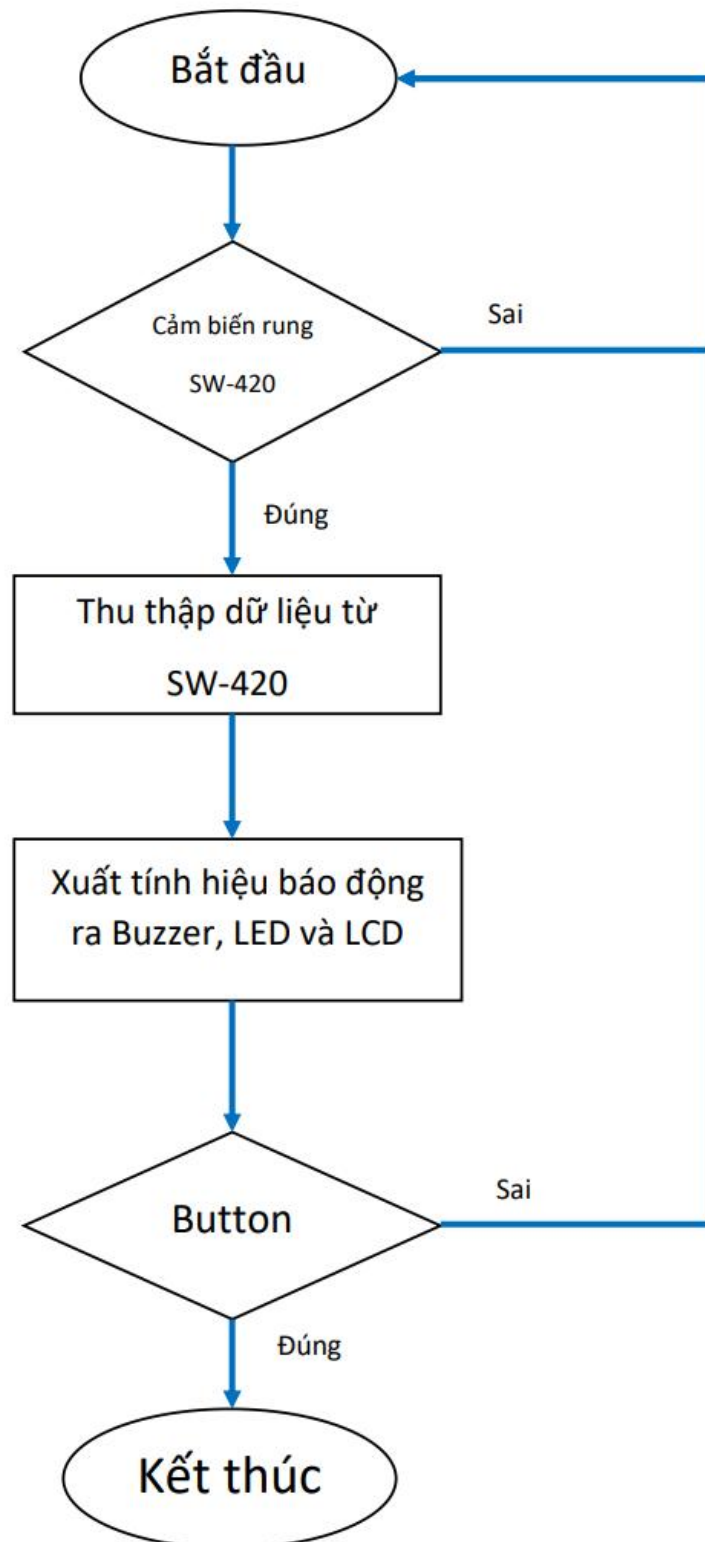


Hình 4.13 Sơ đồ nối dây hệ thống báo động rung chấn.

Giải thích nguyên lý hoạt động

Khi cấp nguồn cho hệ thống vi điều khiển trung tâm sẽ đọc dữ liệu từ cảm biến rung chấn SW-420 một cách liên tục để phát hiện rung chấn từ thiên tai một cách nhanh nhất, nếu có rung chấn vi điều khiển sẽ phát tín hiệu báo động ra Buzzer đóng vai trò là một còi báo động cùng với LED nhấp nháy cảnh báo di tản trong 10 giây, đồng thời tín hiệu **WARNING!!!** sẽ được xuất ra màn hình LCD trên hệ thống để cảnh báo về thiên tai, động đất một cách sớm nhất cho người dùng. Nếu trong quá trình sử dụng người có va chạm làm hệ thống hiểu nhầm là rung chấn và báo động ta có thể ngắt báo động đó bằng cách nhấn Button nó được thiết kế để ngắt hệ thống khi có sự cố không phải là rung chấn thật từ thiên tai hay động đất.

5.2 Lưu đồ giải thuật của hệ thống



Hình 4.14 Lưu đồ giải thuật của hệ thống.

- **Thiết kế phần mềm**

Arduino IDE [5] (Arduino Integrated Development Environment) là một trình soạn thảo văn bản, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino. 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. 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. Môi trường này hỗ trợ cả ngôn ngữ C và C++.

- **Code sử dụng cho phần cứng**

```
#include <LiquidCrystal_I2C.h>
int lcdColumns = 16;
int lcdRows = 2;
LiquidCrystal_I2C lcd(0x27, lcdColumns, lcdRows);

TaskHandle_t Task1;
TaskHandle_t Task2;
const int buzzer = 2;
const int led2 = 4;
const int btn1 = 12;
int state1=0;
int state2=0;
int value = digitalRead(14);
void IRAM_ATTR isr(){
    state1=0;
    digitalWrite(buzzer, state1);
    state2=0;
    digitalWrite(led2, state2);
    delay(10);
}
void setup() {
    Serial.begin(921600);
    pinMode(btn1, INPUT_PULLUP);
    pinMode(buzzer, OUTPUT);
    pinMode(led2, OUTPUT);
    attachInterrupt(btn1, isr, HIGH);
    lcd.init();
    lcd.backlight();
    xTaskCreatePinnedToCore(
        Task1code, /* Task function. */
        "Task1", /* name of task. */
        10000, /* Stack size of task */
        NULL, /* parameter of the task */
        1, /* priority of the task */
        &Task1, /* Task handle to keep track of created task */
        0); /* pin task to core 0 */
    delay(50);
    xTaskCreatePinnedToCore(
        Task2code, /* Task function. */
```

```
        "Task2",      /* name of task. */
        10000,        /* Stack size of task */
        NULL,         /* parameter of the task */
        1,            /* priority of the task */
        &Task2,       /* Task handle to keep track of created task */
        1);           /* pin task to core 1 */

    delay(50);
}

void loop() {
    value = digitalRead(14);

    delay(100);
}

void LCD(){
    // set cursor to first column, first row
    lcd.setCursor(3, 0);
    lcd.print("WARNING!!!");
    delay(200);
    lcd.clear();
    lcd.setCursor(3,1);
    lcd.print("WARNING!!!");
    delay(200);
    lcd.clear();
}

void Task1code( void * pvParameters ){
    Serial.print("Task1 running on core ");
    Serial.println(xPortGetCoreID());
    for(;;){
        if (value ==0){
            Serial.println("Khong Rung");
            state1=0;
            digitalWrite(buzzer, state1);
            // delay(0.00001);
        }
        else if (value == 1){
            Serial.println("Rung");
            for (int i=0;i<20;i++){
                state1=1;
                digitalWrite(buzzer,state1);

                delay(100);
                LCD();
                if(state1==0){
                    return;
                    return;
                }
            }
        }
    }
}

void Task2code( void * pvParameters ){
    Serial.print("Task2 running on core ");
    Serial.println(xPortGetCoreID());
    for(;;){
        if (state1==1) {
            state2=~state2;
            delay(100);
            digitalWrite(led2, state2);
        }
        else if (state1==0){
```

```
state2=0;  
digitalWrite(led2, state2);  
}  
}  
}
```

6. Thi công sản phẩm



Hình 5.1 Mô hình sản phẩm khi bật nguồn.



Hình 5.2 Mô hình sản phẩm khi phát ra cảnh báo.

7. Kết luận

7.1 Kết quả và đánh giá

- Sau khi nạp chương trình vào Board, lắp ráp và đóng gói, mạch hoạt động đúng với yêu cầu đã đặt ra. Khi nhấn nút bật nguồn thiết bị được cấp nguồn và sẵn sàng nhận dữ liệu để bật chế độ cảnh báo. Dữ liệu về rung chấn gửi thông tin chính xác cho vi xử lý. Nếu có rung chấn (mức độ 4 theo thang đo MKS trở lên) hệ thống sẽ phát ra cảnh báo led sáng tắt liên tục, buzzer phát ra âm thanh to rõ để chú ý đồng thời hiển thị WARNING!!! trên màn hình LCD.
- So với mục tiêu ban đầu, nhóm đã hoàn thành được 90% so với mục tiêu ban đầu đề ra. Sau đây là một số ưu, nhược điểm của hệ thống:

- **Ưu điểm:**

- + Đã giải quyết được các vấn đề đặt ra của đề tài.
- + Tốc độ đáp ứng nhanh.
- + Hệ thống tiêu thụ điện năng thấp.
- + Sử dụng nguồn điện pin dễ dàng thay thế, nâng cấp.
- + Dễ dàng quan sát, có thể sử dụng ngay sau khi xem hướng dẫn.
- + Tối ưu chi phí.
- + Khi phát hiện được rung chấn hệ thống sẽ phát âm thanh, led cảnh báo ngay lập tức cũng như màn hình LCD sẽ thông báo WARNING!!! liên tục.

- **Nhược điểm**

- + Chưa tối ưu được đường dây điện trong hệ thống.
- + Hệ thống không thể cảm nhận rung chấn khi ở khoảng cách quá xa.

7.2 Hướng phát triển đề tài

Hệ thống cần được hoàn thiện hơn về những yếu tố phần cứng cũng như tối ưu hóa code để hệ thống hoạt động chính xác và hiệu quả hơn trong thời gian dài. Sau đây là một số hướng phát triển thêm về đề tài mà nhóm em đặt ra để hoàn thiện sản phẩm một cách tốt nhất:

- + Nâng cấp thêm module cảnh báo qua các thiết bị di động: smartphone, ipad, PC...
- + Thiết kế mạch PCB tích hợp tất cả các module trong hệ thống.
- + Mở rộng thêm các module cảnh báo khác: cảm biến khí gas, cảm biến nhiệt độ, độ ẩm, cảm biến con quay hồi chuyển...
- + Thiết kế Web, App dễ dàng giao tiếp với người dùng.
- + Mở rộng nguồn điện để hệ thống duy trì lâu hơn.

7.3 Kết luận:

- Mục tiêu của hệ thống cảnh báo cảm nhận rung chấn, rung lắc cảnh báo thiên tai động đất là cảnh báo ngay lập tức cho người dùng mỗi khi có rung chấn mạnh gây nguy hiểm, giúp người dùng có thể nắm bắt được thông tin nhanh nhất khi xảy ra thiên tai động đất, giảm thiểu tổn thất về con người và tài sản.
- Sau quá trình nghiên cứu, thiết kế và thực hiện mạch nhóm chúng em đã hoàn thành đề tài " HỆ THỐNG CẢNH BÁO CẢM NHẬN RUNG CHẤN, RUNG LẮC CẢNH BÁO THIÊN TAI ĐỘNG ĐẤT ". Dù đã gặp nhiều khó khăn trong quá trình thực hiện từ việc nghiên cứu đề tài đến thực hiện thi công mạch thực tế. Nhưng từ đó nhóm em đã rút ra được kinh nghiệm và cho ra được sản phẩm đúng với yêu cầu mà nhóm đặt ra cũng như hiểu hơn về bộ môn Hệ thống nhúng.
- Sản phẩm dù đã thực hiện được hầu hết các yêu cầu đã đặt ra nhưng vẫn còn những sai sót, những điểm cần hoàn thiện, tối ưu hơn.
- Nhóm chúng em xin chân thành cảm ơn sự giúp đỡ, các ý kiến góp ý từ thầy để giúp nhóm hoàn thiện đề tài hoàn thiện.

8. Phụ lục

8.1 Bảng phân công

STT	Công việc	Phân công	Mô tả cụ thể	Nguồn	Điều kiện tiên quyết
1	Lên ý tưởng thực hiện đề tài. Xây dựng mô hình trên lý thuyết	Cả nhóm	- Xây dựng các chức năng của hệ thống và những yêu cầu đặt ra trên lý thuyết.		Không
2	Đặc tả hệ thống và tìm hiểu thông số các linh kiện	Cả nhóm	Nêu được yêu cầu của đề tài và tìm thông số các linh kiện	Tham khảo datasheet các linh kiện	Tiến trình 1
3	Thiết kế kiến trúc hệ thống	Cả nhóm	Xây dựng được sơ đồ khối của hệ thống	Tham khảo các mô hình minh họa	Tiến trình 2
4	Thiết kế sơ đồ nguyên lý và lắp mạch	Nguyễn Trần Thành Trung	Xây dựng hệ thống hoàn chỉnh đạt được các yêu cầu đề ra	Cảm biến MQ2 ESP 8266 Buzzer 5V Adapter 5V	Tiến trình 3
5	Thử nghiệm hệ thống	Vũ Quốc Việt	Kiểm tra các trường hợp, kiểm tra lỗi và khắc phục	Hệ thống đã thiết kế và lắp đặt	Tiến trình 4
6	Đóng gói	Nguyễn Trần Thành Trung	Đóng gói mô hình hoàn chỉnh	Các hệ thống hoàn chỉnh đã xây dựng và thử nghiệm	Tiến trình 5
7	Viết báo cáo	Cả Nhóm	Tổng hợp và viết báo cáo	Mô hình hệ thống hoàn chỉnh	Tiến trình 6

Bảng 8.1 Bảng phân công

8.2 Danh mục vật tư

ID	Parts/Components	Amount	Price per Unit	Total
1	Kit RF thu phát Wifi BLE ESP32 NodeMCU LuaNode32	1	120.000	120.000
2	Mạch giảm áp Buck DC-DC 5V 2.1A	1	17.000	17.000
3	Đế pin 9V	1	1.200	1.200
4	Pin 9V	1	7.000	7.000
5	Bread board	1	25.000	25.000
6	LCD 1602 tích hợp sẵn module chuyển đổi I2C	1	80.000	80.000
7	Cảm biến rung SW420	1	7.000	7.000
8	Dây bus	30	400	12.000
9	Buzzer 3V	1	6.000	6.000
10	Led 5mm	10	350	3.500
11	Nút nhấn nhả DS-318 12MM	1	8.000	8.000
12	Nút nhấn giữ DS-428	1	4.000	4.000
13	Tấm mica	2	18.000	36.000

Bảng 8.2 Danh mục vật tư

8.3 Biểu đồ hoạt động

Công việc	Ngày bắt đầu	Ngày kết thúc	Số ngày	Tuần 16							Tuần 17						
				2	3	4	5	6	7	CN	2	3	4	5	6	7	CN
Lên ý tưởng thực hiện đề tài.	5/12/2022	6/12/2022	2	■	■												
Xây dựng mô hình trên lý thuyết	7/12/2022	8/12/2022	2		■	■											
Đặc tả hệ thống và tìm hiểu thông số các linh kiện	9/12/2022	10/12/2022	2				■	■									
Thiết kế kiến trúc hệ thống	11/12/2022	12/12/2022	2					■	■								
Thiết kế sơ đồ nguyên lý và lắp mạch	13/12/2022	14/12/2022	2							■	■						
Thử nghiệm hệ thống	15/12/2022	15/12/2022	1										■				
Đóng gói	15/12/2022	15/12/2022	1														
Viết báo cáo	16/12/2022	17/12/2022	2											■	■		

Bảng 8.3 Biểu đồ hoạt động

8.4 Hướng dẫn sử dụng

Nhấn nút nguồn ON và đó là những điều bạn cần làm.

Khi có các dữ liệu rung chấn các bạn sẽ nhận được thông báo ngay lập tức từ thiết bị.

Hệ thống có sử dụng module giảm áp 5V, các bạn có thể nâng cấp nguồn thêm trong khoảng 5V - 24VDC)