# CHƯƠNG II: PHÂN TÍCH THIẾT KẾ

## Phân tích thiết kế khối xử lý trung tâm

Khối xử lý trung tâm có nhiệm vụ tiếp nhận và xử lý dữ liệu được truyền từ khối cảm biến. Với yêu cầu lựa chọn một vi điều khiển có hiệu năng cao, tích hợp nhiều công nghệ truyền thông như chỉ tiêu thiết kế , một trong những giải pháp tối ưu cho khối xử lý trung tâm của cả thiết bị cảm biến và thiết bị điều khiển trung tâm là ESP32 được phát triển bởi Espressif Systems. Vi điều khiển này có các tính năng cơ bản như sau:

Graphical user interface

Description automatically generated

Hình 1‑1: Sơ đồ khối vi điều khiển ESP32

* Là một vi điều khiển 32-bit lõi kép xung nhịp lên đến 240MHz.
* Tài nguyên lưu trữ lớn
* Tích hợp công nghệ Wifi tần số 2.4GHz ~ 2.5GHz
* Tích hợp công nghệ Bluetooth 5.0 kết nối tối đa 20 thiết bị
* Hỗ trợ nhiều chuẩn giao tiếp như UART, SPI, I2C
* Có nhiều chế độ quản lí năng lượng

Espressif đã thiết kế và sản xuất ra một số module để người dùng dễ dàng tiếp cận hơn với dòng chip ESP32. Các thành phần chính trên những module này bao gồm chip ESP32, bộ tạo dao động thạch anh, mạch ăngten, chỉ khác nhau về một số chức năng tùy từng phiên bản như số lượng chân GPIO, các thiết bị ngoại vi được thêm vào như: màn hình LCD, bảng cảm ứng, khe cắm thẻ SD, module máy ảnh….

A picture containing text, electronics, circuit

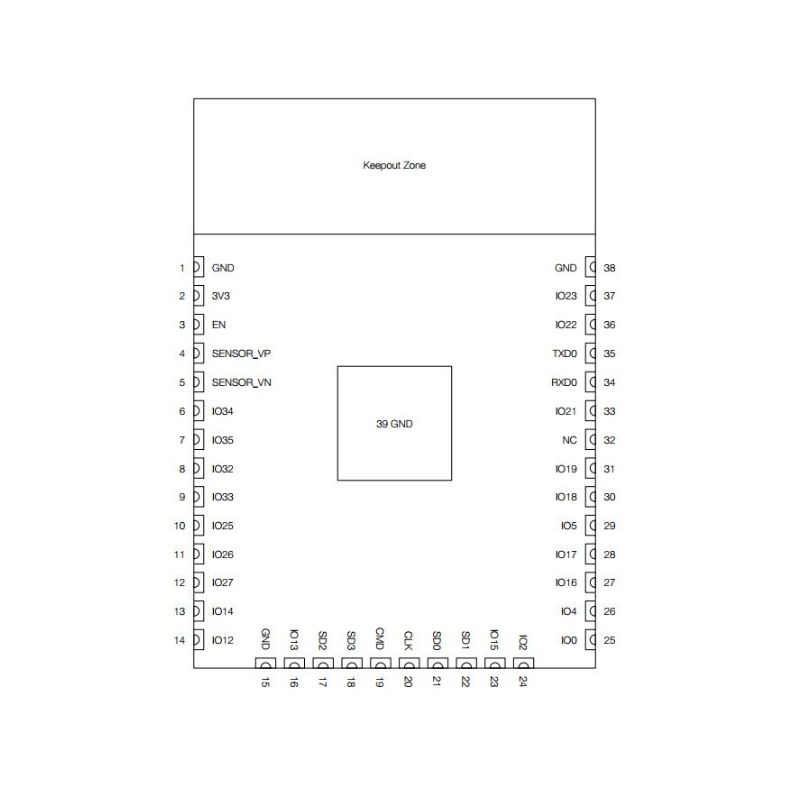
Description automatically generated

Hình 1‑2: Module ESP32-WOOM-32

Phiên bản ESP32-WOOM-32 là một module vi điều khiển (MCU) Wifi (Wireless Fidelity) – BT (Bluetooth) – BTE (Bluetooth Low Energy) phổ biến và mạnh mẽ phục vụ cho nhiều ứng dụng khác nhau

## Thiết bị ngoại vi trên Module ESP32-WOOM-32

GPIO (General Purpose Input/Output Interface) ESP32-WOOM-32 có tất cả 38 chân GPIO, mỗi chân được tích hợp nhiều chức năng khác nhau bằng cách lập trình thanh ghi.



Hình 2‑1: Sơ đồ chân GPIO của ESP32-WOOM-32

ADC (Analog to Digital Converter): tích hợp ADC 12bit và hổ trợ đo 18 kênh. Với thiết lập thích hợp, các ADC có thể được cấu hình để đo điện áp tối đa 18 chân.

UART (Universal Asynchronous Receiver Transmitter) : Tích hợp ba UART kí hiệu là UART1, UART2, UART3 với tốc độ lên đến 5Mbps, có thể truy cập bởi bộ điều khiển DMA hoặc trực tiếp bởi CPU.

I2C Interface (Inter-Integrated Circuit) Hai bộ giao tiếp I2C, có thể vận hành ở chế độ master hoặc slave phụ thuộc vào người sử dụng cấu hình, nó hổ trợ:

* Chế độ tiêu chuẩn (100Kbit/s).
* Chế độ nhanh (400Kbit/s).
* Tốc độ lên đến 5MHz.
* Chế độ kép.

SPI (Serial Peripheral Interface) ESP32 có ba SPI (SPI, HSPI, VSPI) ở chế độ master và slave. Tất cả SPI có thể kết nối đến Flash/SRAM và LCD.

PWM (Pulse Width Modulation) Bộ điều khiển độ rộng xung (PWM) được sử dụng để điều khiển động cơ kỹ thuật số và đèn thông minh. Bộ điều khiển bao gồm: định thời PWM, hệ thống PWM và các module chuyên dụng. Mỗi bộ định thời cung cấp thời gian ở dạng đồng bộ hoặc độc lập.

Timer: Có 4 Timer bên trong ESP32. Chúng đều là bộ định thời 64bit dựa trên bộ chia trước 16bit và bộ định thời đếm lên, xuống. Đặc trưng của Timer:

* Bộ chia 16bit từ 2 đến 65536.
* Timer 64bit.
* Cho phép cấu hình Timer đếm lên, đếm xuống.
* Cho phép dừng và tiếp tục bộ đếm thời gian.

Timer Watchdogs (Bộ định thời giám sát): ESP32 có 3 Timer Watchdogs bao gồm 2 module hẹn giờ (Main Watchdogs Timer) và một module RTC (RTC Watchdog Timer).

## Chuẩn giao tiếp I2C

I2C là một giao thức giao tiếp được phát triển bởi Philips Semiconductors để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một board mạch chỉ sử dụng hai đường truyền tín hiệu.

**a,** Bus vật lý I2C

Bus I2C (dây giao tiếp) chỉ gồm hai dây và được đặt tên là Serial Clock Line (SCL) và Serial Data Line (SDA). Dữ liệu được truyền đi được gửi qua dây SDA và được đồng bộ với tín hiệu đồng hồ(clock) từ SCL. Tất cả các thiết bị/ IC trên mạng I2C được kết nối với cùng đường SCL và SDA như sau:

Chart, box and whisker chart

Description automatically generated

Hình 3‑1: Bus vật lí I2C

Cả hai đường bus I2C (SDA, SCL) đều hoạt động như các bộ lái cực máng hở (open drain). Nó có nghĩa là bất kỳ thiết bị/ IC trên mạng I2C có thể lái SDA và SCL xuống mức thấp, nhưng không thể lái chúng lên mức cao. Vì vậy, một điện trở kéo lên (khoảng 1 kΩ đến 4,7 kΩ) được sử dụng cho mỗi đường bus, để giữ cho chúng ở mức cao (ở điện áp dương) theo mặc định.

**b,** Thiết bị chủ tớ Master và Slave trong I2C

Các thiết bị kết nối với bus I2C được phân loại hoặc là thiết bị Chủ (Master) hoặc là thiết bị Tớ (Slave). Giao tiếp I2C cho phép chế độ truyền nhận dữ liệu giữa nhiều thiết bị Master khác nhau với thiết bị Slave. Tuy nhiên, ở bất cứ thời điểm nào thì chỉ có duy nhất một thiết bị Master ở trang thái hoạt động trên bus I2C. Nó điều khiển đường tín hiệu đồng hồ SCL và quyết định hoạt động nào sẽ được thực hiện trên đường dữ liệu SDA.Tất cả các thiết bị đáp ứng các hướng dẫn từ thiết bị Master này đều là Slave.

Để phân biệt giữa nhiều thiết bị Slave được kết nối với cùng một bus I2C, mỗi thiết bị Slave được gán một địa chỉ vật lý 7-bit cố định. Khi một thiết bị Master muốn truyền dữ liệu đến hoặc nhận dữ liệu từ một thiết bị Slave, nó xác định địa chỉ thiết bị Slave cụ thể này trên đường SDA, phát hiện xem đường SDA đang ở trạng thái nào và sau đó tiến hành truyền dữ liệu. Nếu SDA ở mức 0, nghĩa là đang có 1 thiết bị Master khác đang có quyền điều khiển và phải chờ đến khi truyền xong. Ngược lại nếu SDA ở mức 1, nghĩa là đường truyền SDA đã an toàn và có sử dụng. Vì vậy, giao tiếp có hiệu quả diễn ra giữa thiết bị Master và một thiết bị Slave cụ thể. Tất cả các thiết bị Slave khác không phản hồi trừ khi địa chỉ của chúng được chỉ định bởi thiết bị Master trên dòng SDA.

Chart, diagram, box and whisker chart

Description automatically generated

Hình 3‑2: Giao tiếp master và slave trong giao thức I2C

### c, Các chế độ hoạt động của giao tiếp I2C

Giao tiếp I2C có hai chế độ hoạt động:

* Chế độ chuẩn (standard mode) với tốc độ 100 kBit/s.
* Chế độ tốc độ cao (fast mode) với tốc độ 400 kBit/s.

**d,** Cấu trúc khung bản tin giao thức I2C

Dữ liệu được truyền giữa thiết bị Master và các thiết bị Slave thông qua một đường dữ liệu SDA duy nhất, thông qua các chuỗi có cấu trúc gồm các số 0 và 1 (bit). Mỗi chuỗi số 0 và 1 được gọi là giao dịch (transaction) và dữ liệu trong mỗi giao dịch có cấu trúc như sau:

Diagram

Description automatically generated

Hình 3‑3: Cấu trúc khung bản tin giao thức I2C

**Điều kiện bắt đầu (stop condition):** Bất cứ khi nào một thiết bị chủ quyết định bắt đầu một giao dịch, nó sẽ chuyển mạch SDA từ mức điện áp cao xuống mức điện áp thấp trước khi đường SCL chuyển từ cao xuống thấp. Khi điều kiện bắt đầu được gửi bởi thiết bị Master, tất cả các thiết bị Slave đều hoạt động ngay cả khi chúng ở chế độ ngủ (sleep mode) và đợi bit địa chỉ.

**Khối địa chỉ:** Nó bao gồm 7-bit và được lấp đầy với địa chỉ của thiết bị Slave đến, từ đó thiết bị Master cần gửi/nhận dữ liệu. Tất cả các thiết bị Slave trên bus I2C so sánh các bit địa chỉ này với địa chỉ của chúng.

**Bit Read/Write:** Bit này xác định hướng truyền dữ liệu. Nếu thiết bị Master cần gửi dữ liệu đến thiết bị Slave, bit này được thiết lập là ‘0’. Nếu IC Master cần nhận dữ liệu từ thiết bị Slave, bit này được thiết lập là ‘1’.

**Bit ACK / NACK:** ACK /NACK là viết tắt của Acknowledged/Not Acknowledged. Nếu địa chỉ vật lý của bất kỳ thiết bị Slave nào trùng với địa chỉ được thiết bị Master phát, giá trị của bit này được set là ‘0’ bởi thiết bị Slave. Ngược lại, nó vẫn ở mức logic ‘1’ (mặc định).

**Điều kiện kết thúc (Stop condition):** Sau khi các khung dữ liệu cần thiết được truyền qua đường SDA, thiết bị Master chuyển đường SDA từ mức điện áp thấp sang mức điện áp cao trước khi đường SCL chuyển từ cao xuống thấp.

## Chuẩn giao tiếp SPI

SPI (Serial Peripheral Interface) là một giao diện ngoại vi nối tiếp được phát triển bởi hãng Motorola. Đây là chuẩn đồng bộ nối tiếp truyền dữ liệu ở chế độ full-duplex- song công (tại 1 thời điểm có thể xảy ra đồng thời quá trình truyền nhận).

### a, Bus vật lý SPI

Bus SPI (dây giao tiếp) chỉ gồm bốn dây và được đặt tên là SCK (Serial Clock), MISO(Master Input Slave Output), MOSI(Master Output Slave Input) và SS (Slave Select). Các thiết bị trên mạng SPI có mô hình được kết nối như sau:

Diagram

Description automatically generated

Hình 4‑1: Giao tiếp Master/ Slave trong SPI

* SCK (Serial Clock ): Thiết bị Master tạo xung tín hiệu SCK và cung cấp cho Slave. Xung này có chức năng giữ nhịp cho giao tiếp SPI. Mỗi nhịp trên chân SCK báo 1 bit dữ liệu đến hoặc đi giúp cho quá trình ít bị lỗi và tốc độ truyền cao hơn.
* MISO (Master Input Slave Output): Tín hiệu tạo bởi thiết bị Slave và nhận bởi thiết bị Master. Đường MISO phải được kết nối giữa thiết bị Master và Slave.
* MOSI (Master Output Slave Input): Tín hiệu tạo bởi thiết bị Master và nhận bởi thiết bị Slave. Đường MOSI phải được kết nối giữa thiết bị Master và Slave.
* SS (Slave Select): Chọn thiết bị Slave cụ thể để giao tiếp. Để chọn Slave giao tiếp thiết bị Master chủ động kéo đường SS tương ứng xuống mức 0 (Low). Chân này đôi khi còn được gọi là CS (Chip Select). Chân SS của vi điều khiển (Master) có thể được người dùng tạo bằng cách cấu hình 1 chân GPIO bất kỳ chế độ Output.

### b, Thiết bị chủ-tớ (Master/Slave) trong SPI

* 1 thiết bị Master và 1 thiết bị Slave: 1 thiết bị Master giao tiếp với 1 thiết bị Slave qua 4 dây nối.

Diagram

Description automatically generated

Hình 4‑2: Chế độ kết nối Master/Slave SPI(1/1)

* 1 thiết bị Master và nhiều thiết bị Slave (chế độ độc lập Independent): Ở chế độ này, mỗi thiết bị Slave kết nối với Master được quy định riêng bởi những chân SS khác nhau. Khi thiết bị Master muốn giao tiếp với Slave nào thì kéo chân SS tương ứng xuống mức 0, những chân SS còn lại giữ ở mức 1.

Diagram

Description automatically generated

Hình 4‑3: Chế độ kết nối độc lập Master/Slave SPI (Independent)

* 1 thiết bị Master và nhiều thiết bị Slave ( chế độ chuỗi – Daisy Chain): chúng ta có thể kết nối các thiết bị Slave theo kiểu dây chuyển như bên dưới mà chỉ cần 1 chân SS từ vi điều khiển. Chân MOSI của Slave này sẽ nối với MISO của Slave tiếp theo. Dữ liệu gửi từ vi điều khiển ( hay thiết bị Master), đi vào Slave 1 bằng đường MOSI. Sau đó lại đi ra từ chân MISO của Slave 1, gửi tới chân MOSI của Slave 2,…

Diagram

Description automatically generated

Hình 4‑4: Chế độ kết nối độc lập Master/Slave SPI (Daisy Chain)

### c, Cấu trúc khung bản tin giao thức SPI

Diagram

Description automatically generated

Hình 4‑5: Cấu trúc khung bản tin giao thức SPI

Mỗi chip Master hay Slave đều có một thanh ghi dữ liệu 8 bits. Quá trình truyền nhận giữa Master và Slave xảy ra đồng thời sau 8 chu kỳ đồng hồ, một byte dữ liệu được truyền theo cả 2 hướng.

**Điều kiện bắt đầu:** Quá trình trao đổi dữ liệu bắt đầu khi Master tạo 1 xung clock từ bộ tạo xung nhịp (Clock Generator) và kéo đường SS của Slave mà nó truyền dữ liệu xuống mức Thấp.

**Quá trình truyền:** Thiết bị Master bắt đầu việc trao đổi dữ liệu bằng cách truyền đi một Byte vào thanh ghi dịch của nó. Sau đó Byte dữ liệu sẽ được đưa sang Slave theo đường tín hiệu MOSI. Đồng thời, Thiết bị Slave sẽ truyền dữ liệu nằm trong thanh ghi dịch của chính nó ngược trở về Master thông qua đường tín hiệu MISO. Bằng cách này, dữ liệu của hai thanh ghi sẽ được trao đổi với nhau.

**Điều kiện kết thúc:** chân SS của các Slave ở mức 1, muốn giao tiếp với Slave nào thì ta chỉ việc kéo chân SS của Slave đó xuống mức 0.