

Report Task 2

- Dự tính thiết kế 4 phòng gồm (1 phòng bếp , 2 phòng ngủ, 1 ban công, cổng ra vào)
- Phòng bếp :
 - Dùng cảm biến DHT11 đo nhiệt độ , để báo cháy qua màn hình LCD và buzzer
 - Dùng Module Cảm Biến Khí Gas MQ-2 để thông báo qua màn hình LCD và buzzer
- Phòng ngủ:
 - Điều khiển đèn và quạt
 - Cảm biến siêu âm SRF04 để làm thùng rác thông minh(code có sẵn : <https://drive.google.com/file/d/1bEdmvtI346LCBCuOooIbkUyv5OR0Nhbv/vi ew>)
- Cổng ra vào:
 - Sử dụng nhập mật khẩu để mở cổng , hoặc nhận diện qua thẻ ID(Sử dụng ESP8266 kết hợp với RFID để đọc mã thẻ)
- Ban công:
 - Tự động đóng mở rèm che khi có mưa ,dùng cảm biến mưa

Phương thức giao tiếp :

- Phương thức giao tiếp giữa các phòng ...
- Link tham khảo các giao tiếp UART, IC2, SPI : Wikipedia
<https://www.seeedstudio.com/blog/2019/09/25/uart-vs-i2c-vs-spi-communication-protocols-and-uses/?fbclid=IwAR0upFGgz6XIjVnl6uogkC9TE6Qf8u9eZw-CeOPne2ian9JVDmIx7bfBtcA>
Link : https://www.rfwireless-world.com/Terminology/UART-vs-SPI-vs-I2C.html?fbclid=IwAR2h03VTHG4bqBq7OM9n3ouJ16yeztSJR_HVfcMEIzs1c_34040HpELtp24
-

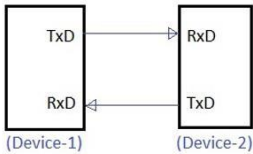
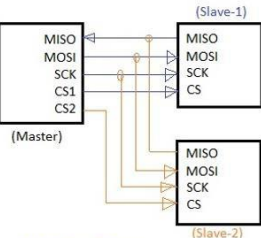
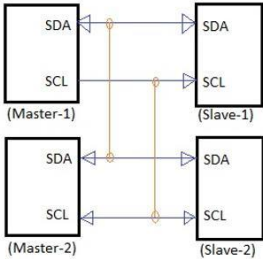
Về cơ bản:

- UART là một chuẩn nối tiếp không đồng bộ, còn SPI và I2C là 2 chuẩn nối tiếp đồng bộ, nên chúng có đường clock còn UART thì không.
- SPI hỗ trợ tốc độ truyền lớn nhất , tốc độ nhanh, và các kết nối của nó cũng là phức tạp nhất.

UART có thể truyền đi với khoảng cách khá ok (không cần thiết), trong khi SPI và I2C chỉ truyền trên bo mạch.

- UART là giao tiếp 1 vs 1, SPI và I2C hoạt động theo cơ chế master - slave. Với SPI thì có thể có nhiều slave, mỗi slave tốn một chân chọn chip. Trong khi I2C có thể có nhiều master và nhiều slave, do cơ chế định địa chỉ nên nó không cần tốn thêm dây như SPI nếu số slave tăng
- Giao tiếp I2C: truyền theo địa chỉ , Phổ biến rộng rãi , code mẫu nhiều

	UART	SPI	I2C
	<p>UART Interface Diagram</p>	<p>SPI Interface Diagram</p>	<p>I2C Interface Diagram</p>
Số I/O cần dùng	Thường là 2 I/O	Tối thiểu là 2 I/O	2 I/O
Số master, slave	Giao tiếp giữa một master, một slave	Giao tiếp có một master và nhiều slave	Giao tiếp có nhiều master, nhiều slave
Tốc độ truyền	Truyền dữ liệu không đồng bộ, có thể cài đặt theo mong muốn của người lập trình. Tốc độ truyền tối đa từ 320 kbps tới 460kbps	Mỗi module SPI có giới hạn tốc độ truyền khác nhau. Tốc độ truyền tối đa thường từ 10 Mbps tới 20 Mbps	Hỗ trợ truyền dữ liệu với tốc độ là: 100kbs, 400kbs, 3.4Mbps.
Ưu điểm	Giao tiếp đơn giản nhất, truyền tin xa, thích hợp cho các thiết bị ít connector, có chuẩn giao tiếp công nghiệp.	<ul style="list-style-type: none"> - Hỗ trợ giao tiếp song công hoặc bán song công. - Có thể giao tiếp với nhiều slave. - SPI sử dụng push – pull, tốc độ truyền cao. - Tiêu thụ năng lượng ít hơn I2C. 	<ul style="list-style-type: none"> - Giao tiếp đơn giản, sử dụng 2 dây để truyền data. - Dễ dàng thêm các slave do mỗi slave có địa chỉ riêng. - Linh hoạt, vai trò master và slave của các thiết bị có thể thay đổi cho nhau - Kiểm soát lỗi tốt hơn, mỗi khi 8 bits được truyền thì có bit \overline{ACK} báo liệu data có hợp lệ hay không
Nhược điểm	Chỉ thích hợp cho các giao tiếp giữa 2 thiết bị	<ul style="list-style-type: none"> - Khi có nhiều thiết bị slave, làm phức tạp trong kết nối phần cứng (chân CS tăng) - Không thay đổi được vai trò master, slave giữa các thiết bị 	<ul style="list-style-type: none"> - Bán song công - Sử dụng nhiều không gian trên mạch PCB (phải thêm các điện trở treo) - Tốc độ thấp hơn SPI

Đặc điểm	UART	SPI	I2C
<u>Sơ đồ kết nối</u>	<p>Kết nối đơn giản</p>  <p>UART Interface Diagram</p>	<p>Kết nối khá phức tạp</p>  <p>SPI Interface Diagram</p>	<p>Kết nối khá đơn giản</p>  <p>I2C Interface Diagram</p>
<u>Số lượng dây</u>	2 dây (1 dây truyền & 1 dây nhận dữ liệu)	4 dây Số lượng dây tăng khi số thiết bị tăng	2 dây
<u>Chế độ truyền</u>	Full duplex (Không phân biệt master – slave)	Full duplex (Một master & nhiều slave)	Half duplex (Nhiều master & nhiều slave)
<u>Tốc độ truyền</u>	Truyền không đồng bộ (Tốc độ tự đặt, tối đa khoảng 460kbps)	Truyền đồng bộ (Tốc độ khoảng 10Mbps đến 20Mbps)	Truyền đồng bộ (Hỗ trợ tốc độ 100kbps, 400kbps, 3.4Mbps, 1Mbps)
<u>Khoảng cách</u>	12m trên lý thuyết	Trên bo mạch	Trên bo mạch
<u>Số thiết bị</u>	2 thiết bị Giao tiếp 1 – 1	Số lượng hạn chế Giao tiếp bằng chân chọn chip	Lên đến 127 thiết bị Giao tiếp bằng địa chỉ

I2C

I2C (Inter-Integrated Circuit) là một giao thức giao tiếp được sử dụng để kết nối ESP8266 với các thiết bị khác. I2C sử dụng hai dây truyền thông, là dây SDA (Serial Data) và SCL (Serial Clock), để truyền dữ liệu giữa các thiết bị.

Thiết bị master sẽ giao tiếp vs slave qua địa chỉ

Module I2C (Inter-Integrated Circuit) hỗ trợ ESP8266 giảm số lượng dây dẫn cần thiết và làm đơn giản quá trình kết nối. Với module I2C, bạn có thể kết nối các thiết bị ngoại vi như cảm biến, màn hình hiển thị LCD ... với ESP8266 một cách dễ dàng

VD: Để kết nối ESP8266 với một thiết bị sử dụng giao thức I2C, một số step

Kết nối dây SDA và SCL

- Dây SDA sẽ được kết nối từ chân SDA trên ESP8266 đến chân SDA trên thiết bị khác.
- Dây SCL sẽ được kết nối từ chân SCL trên ESP8266 đến chân SCL trên thiết bị khác.

Cài đặt và cấu hình

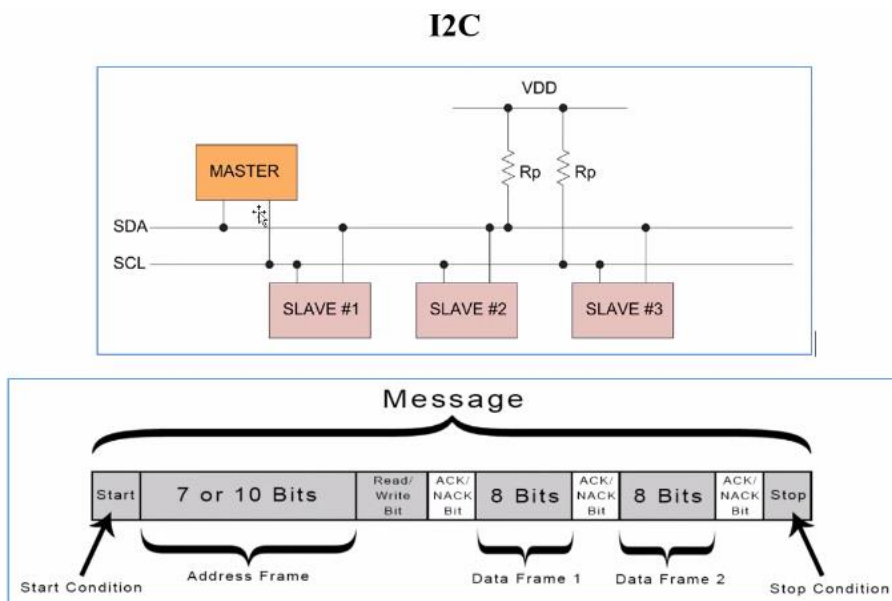
- Sử dụng một thư viện I2C phù hợp với ESP8266 để lập trình giao tiếp I2C.

- Import thư viện vào mã nguồn của bạn và sử dụng các hàm và phương thức tương ứng để truyền và nhận dữ liệu từ thiết bị khác.

Tùy từng thiết bị nhận, xử lý chúng khác nhau

Một số demo về i2C : esp 8266 với LCD

<https://lamchuongnghe.com/huong-dan-esp8266-4-i2c-va-cach-su-dung-voi-lcd16-2-ghiep-noi-voi-esp8266>



Trao đổi dữ liệu giữa 2 con esp8266 (các phòng vc nhau) :

Khởi tạo 1 board hoạt động như là WiFi Access Point, đồng thời khởi tạo 1 TCP Server. Board khác hoạt động như 1 WiFi client thông thường, kết nối vào mạng WiFi đã được tạo, và khởi động 1 TCP Client kết nối vào TCP Server kia.

1 ESP8266 sẽ khởi động SoftAP để tạo mạng WiFi, đồng thời khởi động 1 TCP Server để làm Server.

1 ESP8266 phải là client với chế độ WiFi Station và kết nối đến ESP8266 server đã tạo ở trên.

Sau mỗi giây, Board ESP8266 này sẽ gửi dữ liệu vào Board kia, board nhận được dữ liệu sẽ in ra cổng Serial và gửi ngược lại.

