- Đồ án 01 = Đồ án điện tử (Tập trung lập trình điện tử)

- Đồ án 02 = Đồ án chuyên ngành (AI, vi mạch, IoT, điện tử chuyên sâu,...)

**\*Bước 01: - Tìm vấn đề thực tế (ý tưởng) cần giải quyết/nâng cấp.**

**- Song song đó, là tìm kiếm/mua/chạy thử nghiệm các module điện tử để đánh giá “độ khả thi” của đề tài.**

**- Đồng thời, nhận góp ý từ Thầy để có góc nhìn rõ ràng hơn.**

(Lĩnh vực bất kì: Xã hội, giáo dục, trường lớp, con người, giao thông,...)

**\*Bước 02**: Chốt đề tài & bắt đầu triển khai theo nội dung bên dưới:

**ĐẶC TẢ HỆ THỐNG (Project Specification)**

|  |  |
| --- | --- |
| **Sinh viên 01** | **Sinh viên 02** |
| Họ tên: Đào Ngọc Minh Huy | Họ tên: Phan Nhật Minh |
| MSSV: 20119045 | MSSV: 20119033 |
| **Năng lực hiện tại** (tự đánh giá)  *Khả năng lập trình (C, khác), đam mê kĩ thuật mảng nào (lập trình điện tử, web, vi mạch, AI,...), đã lập trình VĐK gì (nhập môn, khá, giỏi), đã làm project gì (mô tả tính năng), số ngày/buổi còn rãnh trong tuần,...* | |
| * Khả năng lập trình C khá * Đam mê mảng lập trình điện tử * Những vi điều khiển đã lập trình: * ESP32/8266(nhập môn):Giao tiếp với một số thiết bị ngoại vi như DHT11, MQ2, LCD,… * 8051(khá): Lập trình ngắt, timer, các thiết bị ngoại vi: button, LCD,… * Atmega128(nhập môn):lập trình ngắt, timer, ADC, PWM,... * Số buổi rãnh: T5, T6, chiều T7 | * Khả năng lập trình C khá * Đam mê mảng lập trình điện tử * Những vi điều khiển đã lập trình: * ESP32/8266(nhập môn):Giao tiếp với một số thiết bị ngoại vi như DHT11, MQ2, LCD,… * 8051(khá): Lập trình ngắt, timer, các thiết bị ngoại vi: button, LCD,… * Atmega128(nhập môn):lập trình ngắt, timer, ADC, PWM,... * Số buổi rãnh: T5, T6, chiều T7 |
| **Vai trò công việc cụ thể** (1 người làm 1 việc riêng) – Soạn dự kiến đến tuần 18 | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | STT | Nội dung | TG dự kiến | KQ (onTime,Late, Outstanding) | Ai? | | 1 | Tìm hiểu, định ra chức năng hệ thống, chọn và mua sắm linh kiện. | Tuần 5 (27/2/2023 – 3/3/2023) | onTime | M.Huy | | 2 | Kiểm tra hoạt động từng module trên Test Board với Arduino UNO. | Tuần 5 (3/3 – 5/3) | Outstanding | M.Huy | | 3 | Lập trình WEB | Tuần 5 & Tuần 6 | onTime | N.Minh | | 4 | Viết báo cáo mục 1 đến mục 6 | Tuần 6 (6/3 – 9/3) | onTime | M.Huy | | 5 | Vẽ sơ đồ khối hệ thống + Schematic | Tuần 6 (6/3 – 10/3) | onTime | N.Minh | | 6 | Lập trình kiểm tra kết nối các module kết hợp với nhau. | Tuần 6 (10/3 – 12/3)  Tuần 7 (13/3 – 16/3) |  | M.Huy | | 7 | Dựa vào mạch Schematic bắt đầu thực hiện nghiên cứu vẽ mạch PCB | Tuần 7 (13/3 – 19/3)  Tuần 8 (20/3 – 26/3) |  | N.Minh | | 8 | Vẽ lưu đồ giải thuật các chức năng của hệ thống (Nhiều lưu đồ). | Tuần 7 (16/3 – 19/3)  Tuần 8 (20/3 – 23/3) |  | M.Huy | | 9 | Viết chương trình các chức khối chức năng theo lưu đồ giải thuật đã vẽ | Tuần 8 (23/3 – 26/3)  Tuần 9 (27/3 – 2/4)  Tuần 10 (3/4 – 9/4) |  | M.Huy | | 10 | Tạo thư viện (file .c và .h) các khối chức năng của hệ thống | Tuần 11 (10/4 – 13/4) |  | M.Huy | | 11 | Viết báo các mục phần lập trình điện tử | Tuần 7 – Tuần 11 |  | M.Huy | | 12 | Vẽ PCB | Tuần 11 (10/4 – 16/4)  Tuần 12 (17/4 – 23/4) |  | N.Minh | | 13 | Thi công PCB | Tuần 13 (24/4 – 30/4) |  | N.Minh | | 14 | Thực nghiệm chức năng hệ thống | Tuần 14 (1/5 – 7/5) |  | N.Minh và M.Huy | | 15 | Thực hiện viết báo cáo phần thi công mạch. | Tuần 15 (8/5 – 14/5) |  | N.Minh | | 16 | Hoàn chỉnh toàn bộ báo cáo | Tuần 16 (15/5 – 18/5) |  | Cả 2 | | 17 | Đóng gói sản phẩm | Tuần (22/5 – 27/5) |  | Cả 2 | | |

|  |
| --- |
| 1. Tên Project + Lý do chọn đề tài + Nơi áp dụng |
| * Tên: HỆ THỐNG GIÁM SÁT VÀ CẢNH BÁO THÂN NHIỆT TÍCH HỢP ĐIỂM DANH * Lý do: Vì tình hình dịch bệnh có nhiều diễn biến, để đảm bảo sức khỏe học sinh, sinh viên hoặc công nhân viên, sớm sàng lọc những người có nguy cơ mắc bệnh cũng như có thể quản lý số lượng người tham gia vào việc học tập và công tác của nhà trường và công ty. * Nơi áp dụng: Trường học, công ty (văn phòng), phòng nghiên cứu. |
| 2. Mô tả ngắn gọn (3-5 dòng) |
| \*(Liệt kê: 1 số version/option của sản phẩm tùy tình huống thực tế)   * Hệ thống quét vân tay nhận dạng đối tượng cần đo được lưu trước trong bộ nhớ hệ thống, sử dụng cảm biến vật cản và cảm biến đo nhiệt độ hồng ngoại để đo nhiệt độ cơ thể. Nếu nhiệt độ cao hơn 38 độ thì sẽ có cảnh báo phát ra. Hệ thống sử dụng pin có kết nối wifi đưa dữ liệu lên firebase và web. |
| 3. Đặc tả hệ thống (Quan trọng trước) |
| * Quét và lưu trữ thông tin của dấu vân tay (Tên và ID). * Xóa dấu vân tay. * Quét dấu vân tay để hiện thị thông tin lên màn hình OLED. * Chuyển đổi giữa các chế độ hoạt động của cảm biến quét vân tay bằng nút bấm. * Phát hiện đối tượng bằng cảm biến vật thể YS-29 để kích hoạt việc đo nhiệt độ. * Đo nhiệt độ bằng cảm biến nhiệt độ bằng hồng ngoại MLX90614. * Màn hình OLED để hiện thị các thông tin bao gồm: Dấu vân tay, nhiệt độ. * Buzzer để cảnh báo khi nhiệt độ đối tượng được đo cao bất thường. * Laser module để nhận biết hệ thống đang có hoạt động hay không. * Nguồn pin với mạch đo dung lượng pin để người giám sát có thể giám sát được hệ thống có đang được cấp điện để hoạt động hay không. * Module Wifi ESP8266 để hệ thống kết nối với Internet dùng để gửi dữ liệu về cho người giám sát. |
| 4. Sơ đồ khối |
|  |
| 5. Chọn linh kiện |
| (\*Mỗi chức năng: chọn 2-3 linh kiện. Chọn “phù hợp nhất” kèm lý do.)   * Vi điều khiển: 8051, ESP32 (8266), ATmega328P (Arduino UNO R3), … Chọn ATmega328P (Arduino UNO R3). Lý do: Được sử dụng rộng rãi, dễ tiếp cận, có tốc độ xử lý cao hơn, nhiều tính năng bổ sung và được hỗ trợ bởi nhiều công cụ phát triển phần mềm (Arduino IDE, MPLAB, …). * Cảm biến vân tay: R307, R308, R305, … Chọn R308. Lý do: Vị trí lắp đặt hệ thống không có quá nhiều đối tượng nên chỉ cần sử dụng R308 (lưu trữ được 500 dấu vân tay) để quét vân tay. * Cảm biến vật thể: Cảm biến vật cản hồng ngoại, Cảm biến YS-29, … Chọn cảm biến cảm biến YS-29. Lý do nhỏ gọn, dễ lắp đặt, khoảng cách đo xa (1.2m – có thể tùy chỉnh khoảng cách đo). * Cảm biến nhiệt độ hồng ngoại: Cảm biến nhiệt không tiếp xúc: MLAB01105, Max30205MTA I2C, GY096 (MLX90614), … Chọn GY-096 (MLX90614) vì lý do tối ưu hóa chi phí mà vẫn được yêu cầu về khoảng cách đo, độ chính xác (± 0.5), chế độ tiêu thụ điện áp và tốc độ đo. * Màn hình hiển thị: OLED, LCD, … Chọn màn hình OLED do hiển thị một cách trực quan thông tin. * Module Wifi: ESP8266, ESP32. Chọn module ESP8266. |
| 6. Sơ đồ nguyên lý (Schematic) |
|  |
| 7. Cấu trúc dữ liệu |
| * Hệ thống cần lưu những thông số/giá trị là: ID (dấu vân tay), nhiệt độ cơ thể. * Lựa chọn cấu trúc để lưu trữ dữ liệu là Mảng. * Có lưu lại HISTORY, sử dụng thẻ nhớ SD card để làm nơi lưu trữ dữ liệu và dùng dữ liệu được lưu trữ lại để “backup” lại dữ liệu khi mất điện. * Dữ liệu dấu vân tay thì được lưu trữ tại bộ nhớ Flash của cảm biến vân tay, vì tính chất sử dụng của sản phẩm nên bộ nhớ chứa được dữ liệu của 500 dấu vân tay. |
| 7. Các lưu đồ giải thuật |
| (Nhiều lưu đồ. Mỗi lưu đồ thể hiện 1 chức năng(1 hàm xử lý). Các lưu đồ có liên kết với nhau)  (Link web vẽ lưu đồ online: [here](https://app.diagrams.net/) )   * Module chức năng nhận biết chuyển động để đo nhiệt độ và Module chức năng cảm biến vân tay * Hiển thị thời gian thực: |
| * Lưu dữ liệu và hiển thị OLED: |

|  |  |
| --- | --- |
| SƠ ĐỒ TỔNG: | |
| 8. Cấu trúc file chương trình | |
| \*Ví dụ:  - main.c  - display.c (display.h)  - Fingerprint.c (Fingerprint.h) |  |
| 9. Các hàm xử lý (Chạy thực nghiệm với module rời) | |
| MLX\_sensor và YS-29 (ir\_sensor):  readTemp(); | @brief (Mô tả chức năng)   * Hàm đọc giá trị tín hiệu số thay đổi khi có vật thể. * Hàm đọc giá trị nhiệt độ cơ thể và môi trường   @param (tham số truyền vào dạng tham chiếu)  @retval (kết quả trả về)   * Hàm trả về giá trị số nguyên để nhận biết có người. * Hàm trả về 2 giá trị cảm biến nhiệt độ (kiểu số thực). |
| OLED (display):   * print\_text\_1x(const char \*str, int x, int y); * print\_int\_1x(int num, int x, int y); * print\_float\_1x(float num, int x, int y); * print\_uint8t\_1x(uint8\_t num, int x, int y); * print\_text\_2x(const char \*str, int x, int y); * print\_int\_2x(int num, int x, int y); * print\_float\_2x(float num, int x, int y); * print\_uint8t\_2x(uint8\_t num, int x, int y); * print\_text1x(const char \*str); * print\_int1x(int num); * print\_float1x(float num); * print\_uint8t1x(uint8\_t num); * print\_text2x(const char \*str); * print\_int2x(int num); * print\_float2x(float num); * print\_uint8t2x(uint8\_t num); | @brief (Mô tả chức năng)   * Hàm hiển thị giá trị ra màn hình bao gồm ký tự, biến.   @param (tham số truyền vào dạng tham chiếu)   * x,y: Tọa độ bắt dầu hiện ký tự * String data (lable): Dữ liệu hiện ra dạng ký tự. * float (int) data: dữ liệu dạng số (biến giá trị).   @retval (kết quả trả về)   * Hàm trả về giá trị số nguyên, ký tự và số thực. |
| R308 (Fingerprint):   * uint8\_t readnumber(); | @brief (Mô tả chức năng)   * Hàm đọc giá trị ID được nhập vào.   @param (tham số truyền vào dạng tham chiếu)   * uint8\_t num: Giá trị được dùng để gán cho ký tự được nhập vào từ bàn phím   @retval (kết quả trả về)   * Hàm trả về giá trị số nguyên. |
| R308 (Fingerprint):   * uint8\_t getFingerprintEnroll(); | @brief (Mô tả chức năng)   * Nhập dữ liệu vân tay vào bộ nhớ Flash.   @param (tham số truyền vào dạng tham chiếu)   * id: Giá trị id của vân tay   @retval (kết quả trả về) |
| R308 (Fingerprint):   * uint8\_t getFingerprintID(); | @brief (Mô tả chức năng)   * Hàm quét vân tay và so sánh dữ liệu vân tay với vân tay được quét có khớp với nhau hay không.   @param (tham số truyền vào dạng tham chiếu)  @retval (kết quả trả về)   * Hàm trả về giá trị số nguyên của ID nếu dấu vân tay trùng. |
| R308 (Fingerprint):   * uint8\_t deleteFingerprint(); | @brief (Mô tả chức năng)   * Hàm xóa dấu vân tay.   @param (tham số truyền vào dạng tham chiếu)   * int id: Giá trị id vân tay cần xóa.   @retval (kết quả trả về) |
| R308 (Fingerprint):   * fingerprintSetup(); | @brief (Mô tả chức năng)   * Hàm thiết lập hệ thống cho cảm biến (khai báo chân, …).   @param (tham số truyền vào dạng tham chiếu)  @retval (kết quả trả về) |
| R308 (Fingerprint):   * fingerprintLoop(); | @brief (Mô tả chức năng)   * Hàm thiết lập chế độ hoạt động của cảm biến.   @param (tham số truyền vào dạng tham chiếu)   * int mode: Chế độ hoạt động của cảm biến.   @retval (kết quả trả về) |

|  |
| --- |
| 10. Chương trình chính main(){...} |
| (Theo lưu đồ, kết hợp các hàm xử lý) (Chạy thực nghiệm hoàn chỉnh trên Testboard) => Kết quả MC: Hình ảnh, video demo. |
| 11. Thiết kế mạch in PCB |
| (Bảng vẽ)  A picture containing screenshot, text, diagram, line  Description automatically generated |
| 12. Thi công mạch in PCB |
| (Hình ảnh Sản phẩm mạch in chưa hàn)  (Hình ảnh Sản phẩm mạch in đã hàn) |
| 13. Thử nghiệm tính năng |
| |  |  |  | | --- | --- | --- | | Thao tác | Kết quả | Thời gian đáp ứng (trung bình) | | Bật 5 lần đèn 01 | Đạt 100% | 1-3s/lần |   . |
| 14. Viết cuốn báo cáo |
| \*Theo mẫu. |
| 15. Link nộp bài (google drive – nhớ share quyền) |
| (Thư mục rõ ràng: source (web, ESP32, firebase.json file), proj\_spec.docx, file schematic/PCB, video demo(youtu) |

\*Quy trình làm project

(1) Phân tích – (2) Code với Testboard – (3) Vẽ schematic/PCB & thi công – (4) Test & sửa.

\*Lựa chọn dòng VĐK

- Arduino, ESP: SV chưa master Arduino. (Level 01)

- 8051, AVR, PIC, ARM: SV đã khá Arduino. (Level 02)

- SV làm đồ án 01,02: Làm với Arduino/ESP trước – đủ giỏi/nhanh (Thầy đánh giá trong quá trình làm) thì port sang các dòng VĐK (chuyên sâu lập trình nhúng).

- SV làm đồ án 01,02: Vẽ mạch Schematic (dạng đế cắm module)/thi công PCB – đủ giỏi/nhanh (Thầy đánh giá trong quá trình làm) thì vẽ lại toàn bộ schematic/thi công PCB cho toàn bộ linh kiện rời. Vẽ thêm Hộp 2D, 3D, đóng hộp cho đẹp.

\*Quy định:

* Báo cáo/thảo luận với Thầy 1 lần/tuần. (Vắng 2-3 lần = fail)
* Nhóm: nên có 2 bạn (ngang xíu). Mỗi bạn làm 1 mảng khác nhau.
* Kiến thức về C phải khá. (Ôn lại)