

**BỘ GIÁO DỤC ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



# **ĐỒ ÁN TỐT NGHIỆP**

## **PHÁT TRIỂN CHƯƠNG TRÌNH QUÉT MÃ QR SỬ DỤNG ESP32-CAM VÀ ỨNG DỤNG**

**SINH VIÊN THỰC HIỆN : NGUYỄN ĐẠI NHÂN**

**MÃ SINH VIÊN : 1451020168**

**KHOA : CÔNG NGHỆ THÔNG TIN**

**HÀ NỘI - 2024**

**BỘ GIÁO DỤC ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**

-----



**NGUYỄN ĐẠI NHÂN**

**PHÁT TRIỂN CHƯƠNG TRÌNH QUÉT MÃ  
QR SỬ DỤNG ESP32-CAM VÀ ỨNG DỤNG**

**CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN**  
**MÃ SỐ : 74.80.201**

**NGƯỜI HƯỚNG DẪN: TS. NGUYỄN TÀI TUYÊN**

**HÀ NỘI - 2024**

## LỜI CAM ĐOAN

Em xin cam đoan đề tài “Phát triển chương trình quét mã QR sử dụng ESP32-CAM và ứng dụng” là đồ án tốt nghiệp của em được tiến hành một cách công khai và minh bạch, dựa trên sự cố gắng và nỗ lực của bản thân cùng sự giúp đỡ hướng dẫn tận tình của thầy hướng dẫn TS. Nguyễn Tài Tuyên.

Các số liệu nghiên cứu nêu trong đồ án đã được trích dẫn và nêu rõ trong mục Tài liệu tham khảo. Những kết quả nghiên cứu hoàn toàn trung thực, không sao chép hay sử dụng kết quả của bất kỳ công trình nào đã được công bố trước đây. Nếu có phát hiện sự sao chép, bản thân em xin chịu hoàn toàn trách nhiệm và kỷ luật từ phía nhà trường.

Ngày 16 tháng 06 năm 2024

Sinh viên ký tên

## LỜI CẢM ƠN

Để hoàn thành đồ án tốt nghiệp này, trước hết em chân thành cảm ơn các cá nhân và tổ chức đã tạo điều kiện hỗ trợ, giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài này. Trong suốt thời gian từ khi bắt đầu học tập đến nay, em luôn nhận được sự quan tâm, giúp đỡ từ quý thầy, cô giáo trong khoa Công Nghệ Thông Tin và các phòng ban - Đại Học Đại Nam đã luôn tận tình chỉ bảo, truyền đạt cho em những kiến thức quý báu trong suốt thời gian học tập và nghiên cứu tại trường.

Đặc biệt, em xin gửi đến thầy giáo TS. Nguyễn Tài Tuyên, người đã tận tình hướng dẫn, giúp đỡ để em có thể hoàn thành tốt báo cáo đồ án đúng tiến độ.

Do bản thân còn những hạn chế nhất định về chuyên môn và thời gian nên sẽ khó tránh khỏi những sai sót trong quá trình thực hiện. Em mong nhận được sự góp ý của các thầy/cô trong khoa và các bạn để bài báo cáo của em được hoàn thiện tốt hơn.

Em xin trân thành cảm ơn!

**DANH MỤC VIẾT TẮT**

<b>Từ viết tắt</b>	<b>Nghĩa tiếng Anh</b>	<b>Nghĩa tiếng Việt</b>
ESP32	Espressif Systems ESP32	Hệ thống ESP32 của Espressif
GPIO	General Purpose Input/Output	Đầu vào/ ra chung
IOT	Internet Of Things	Internet vạn vật
LCD	Liquid Crystal Display	Màn hình tinh thể lỏng
M2M	Machine to Machine	Máy tới máy
QR Code	Quick Response Code	Mã phản hồi nhanh
QR Code	Quick Response Code	Mã phản hồi nhanh
RFID	Radio Frequency Identification	Nhận dạng tần số vô tuyến
SMS	Short Message Service	Dịch vụ tin nhắn ngắn
SSMS	SQL Server Management Studio	Quản Lý Máy Chủ SQL
URL	Uniform Resource Locator	Địa chỉ web

## DANH MỤC HÌNH ẢNH

Hình 1.1 Cấu trúc của hệ thống IoT .....	9
Hình 1.2 Mã QR xét tuyển Trường Đại Nam .....	12
Hình 1.3 ESP32-CAM AI-Thinker .....	15
Hình 1.4 Sơ đồ chân ESP32-CAM.....	16
Hình 1.5 Ứng dụng của ESP32-CAM.....	17
Hình 1.7: Nhập link vào ô <i>Additional Board Manager URLs</i> .....	18
Hình 1.8: Mở Board Manager .....	18
Hình 1.9: Tải thư viện esp32.....	19
Hình 1.10: Hoàn thành cài đặt thư viện esp32 .....	19
Hình 1.11: Phần mềm Arduino IDE.....	20
Hình 2.1: Chương trình quét mã QR Code .....	31
Hình 2.2: Thiết kế giao diện Quản lý thực phẩm.....	35
Hình 2.3: Thiết kế giao diện Đăng nhập .....	35
Hình 2.4: Trang web Google Drive.....	37
Hình 2.5: Tạo trang Google sheet mới.....	38
Hình 2.6: Trang Google sheet mới.....	38
Hình 3.1 Khởi chạy chương trình quét mã QR trong Arduino .....	42
Hình 3.2: Phần mềm hiển thị thông tin mã QR.....	43
Hình 3.3: Nhập hàng .....	43
Hình 3.4: Sản phẩm tăng thêm.....	43
Hình 3.5: Tắt nhập hàng.....	44

Hình 3.6: Sản phẩm giảm đi.....	44
Hình 3.7: Mô hình tủ lạnh có sử dụng ESP32-CAM.....	45
Hình 3.8: Mô hình tủ lạnh bên trong.....	46

## **DANH MỤC BẢNG BIỂU**

Bảng 1: Bảng chi tiết hàng hóa. ....	30
--------------------------------------	----

## MỤC LỤC

LỜI CAM ĐOAN.....	i
LỜI CẢM ƠN .....	ii
DANH MỤC VIẾT TẮT .....	iii
DANH MỤC HÌNH ẢNH .....	iv
DANH MỤC BẢNG BIỂU .....	v
MỤC LỤC.....	vi
GIỚI THIỆU.....	1
1 Giới thiệu đồ án.....	1
a) Tên đồ án tốt nghiệp .....	1
b) Tổng quan đề tài .....	1
c) Lý do chọn đề tài .....	1
d) Mục tiêu đồ án .....	2
e) Phân tích đồ án.....	3
2 Đối tượng nghiên cứu .....	4
3 Phạm vi nghiên cứu.....	5
4 Bố cục đồ án.....	5
Chương 1: CỞ SỞ LÝ THUYẾT.....	6
1.1 Giới thiệu về IoT.....	6
1.1.1 Khái niệm IoT .....	6
1.1.2 Lịch sử phát triển .....	6
1.1.3 Cấu trúc của hệ thống IoT.....	8
1.1.4 Hoạt động của IoT .....	9
1.2 Giới thiệu về QR Code.....	11
1.2.1 Khái niệm Mã QR Code .....	11



1.2.2 Lịch sử phát triển .....	13
1.3 Giới thiệu về ESP32-CAM .....	14
1.3.1 ESP32-CAM .....	14
1.3.2 Thông số cơ bản.....	15
1.3.3 Ứng dụng của ESP32-CAM .....	16
1.3.4 Hướng dẫn cài đặt ESP32 với Arduino .....	17
1.4 Giới thiệu về Arduino IDE.....	19
1.4.1 Arduino IDE.....	19
1.4.2 Hoạt động của Arduino IDE .....	20
1.4.3 Ưu điểm của Arduino IDE.....	22
1.5 Giới thiệu SQL Server Management Studio .....	23
1.5.1 SQL Server Management Studio .....	23
1.5.2 Lịch sử phát triển .....	24
1.6 Giới thiệu về Visual Studio.....	25
1.6.1 Visual Studio.....	25
1.6.2 Tính năng của Visual Studio.....	27
1.6.3 Lịch sử phát triển .....	28
1.7 Tiểu kết.....	29
<b>Chương 2: THIẾT KẾ CHƯƠNG TRÌNH QUÉT MÃ QR SỬ DỤNG ESP32-CAM.....</b>	<b>30</b>
2.1 Chương trình của quét mã QR .....	30
2.1.1 Thiết kế cơ sở dữ liệu.....	30
2.1.2 Thiết kế chương trình quét mã QR .....	30
2.2 Lưu đồ giải thuật .....	32

2.3 Thiết kế phần mềm quét mã QR .....	34
2.3.1 Hoạt động của hệ thống. ....	34
2.3.2 Phần mềm quét mã QR .....	34
2.4 Chọn nền tảng quản lý-Google Sheet .....	35
2.4.1 Giới thiệu về Google Sheet.....	35
2.4.2 Hướng dẫn sử dụng Google Sheet .....	37
2.5 Tiểu kết.....	38
Chương 3: DEMO SẢN PHẨM SAU KHI HOÀN THÀNH .....	40
3.1 Cách chương trình thực thi .....	40
3.1.1 Khai báo chương trình .....	40
3.1.2 Hàm Void setup .....	40
3.1.3 Hàm Void loop.....	41
3.1.4 Hàm QRCodeReader .....	42
3.2 Demo chương trình .....	42
3.3 Mô hình thực tế .....	44
3.4 Tiểu kết.....	46
KẾT LUẬN VÀ KIẾN NGHỊ.....	47
TÀI LIỆU THAM KHẢO.....	48

## GIỚI THIỆU

### 1 Giới thiệu đồ án

#### a) Tên đồ án tốt nghiệp

“Phát triển chương trình quét mã QR sử dụng ESP32-CAM và ứng dụng”

#### b) Tổng quan đề tài

INTERNET – Phương tiện truyền thông (media network) để gửi (to send), nhận (to receive), và trao đổi (to communicate) thông tin.

Things – Vật (physical objects) được thiết kế để có những chức năng theo mục đích sử dụng (purpose).

Internet of things – kết nối mọi vật thông qua Internet để theo dõi (to monitor), sử dụng (operate) và điều khiển (control).

#### c) Lý do chọn đề tài

Việc em chọn đề tài "Phát triển chương trình quét mã QR sử dụng ESP32-CAM và ứng dụng" mang lại giá trị thực tiễn và giúp em phát triển kỹ năng và khám phá tiềm năng của công nghệ IoT và ứng dụng quét mã QR:

- Tính ứng dụng của Mã QR: Mã QR (Quick Response) đã trở thành một phương tiện thông tin phổ biến, được sử dụng trong nhiều lĩnh vực như quảng cáo, điều hướng trang web, thanh toán di động, v.v. Phát triển một chương trình quét mã QR có thể áp dụng rộng rãi trong thực tế, từ các cửa hàng bán lẻ đến các hệ thống quản lý kho hàng, giao thông vận tải, và nhiều lĩnh vực khác.
- Tính phổ biến của ESP32-CAM: ESP32-CAM là một mô-đun Wi-Fi có khả năng kết nối internet và tích hợp camera, rất phổ biến trong các ứng dụng IoT (Internet of Things). Việc sử dụng nó trong việc phát triển chương trình quét mã QR có thể mang lại tính linh hoạt và hiệu suất cao.
- Cơ hội học hỏi và phát triển kỹ năng: Phát triển một ứng dụng thực tế sử dụng ESP32-CAM yêu cầu kiến thức vững về vi điều khiển, vi xử lý, giao tiếp

mạng, và xử lý hình ảnh. Đây là một cơ hội tuyệt vời để học hỏi và phát triển kỹ năng kỹ thuật trong lĩnh vực IoT và lập trình nhúng.

- **Tìm tòi sáng tạo:** Phát triển một ứng dụng mới và hữu ích luôn là một thách thức. Việc tạo ra một hệ thống quét mã QR có thể kích thích sự sáng tạo và đem lại niềm vui khi thấy sản phẩm của mình hoạt động hiệu ích.
- **Tiềm năng ứng dụng:** Với sự phát triển của thị trường IoT và các ứng dụng liên quan, việc phát triển một chương trình quét mã QR có thể mở ra cơ hội kinh doanh trong tương lai..

#### **d) Mục tiêu đề án**

- **Tăng cường tính tiện lợi:** Mục đích chính là tạo ra một công cụ tiện lợi cho việc quét và trích xuất thông tin từ mã QR. Người dùng có thể dễ dàng sử dụng thiết bị ESP32-CAM để quét mã QR mà không cần phải sử dụng các thiết bị phức tạp hoặc phần mềm đặc biệt.
- **Tối ưu hóa quy trình làm việc:** Mục đích là cải thiện quy trình làm việc bằng cách sử dụng mã QR để truy cập thông tin nhanh chóng và chính xác. Điều này có thể áp dụng trong nhiều lĩnh vực như quản lý kho, bán lẻ, quảng cáo, v.v.
- **Đẩy mạnh sự linh hoạt và tính ứng dụng:** Mục đích là phát triển một giải pháp linh hoạt và có thể sử dụng trong nhiều ngữ cảnh khác nhau. Chương trình có thể được tùy chỉnh để đáp ứng các yêu cầu cụ thể của từng ứng dụng, từ việc hiển thị thông tin đến thực hiện các hành động phản ứng.
- **Tạo ra một sản phẩm hoặc dịch vụ mới:** Mục đích là phát triển một sản phẩm hoặc dịch vụ mới có thể mang lại giá trị cho người dùng cuối. Có thể dự định tích hợp chương trình vào các sản phẩm IoT hoặc cung cấp dịch vụ quét mã QR cho doanh nghiệp.
- **Nâng cao kỹ năng và kiến thức:** Mục đích cuối cùng là cung cấp cơ hội cho các nhà phát triển để thử nghiệm, học hỏi và nâng cao kỹ năng trong lĩnh vực IoT, xử lý hình ảnh và lập trình nhúng. Điều này có thể giúp họ phát triển sự

hiểu biết sâu hơn về công nghệ và có thể áp dụng kiến thức đó vào các dự án tương lai.

### **e) Phân tích đề án**

*Yêu cầu chức năng:*

- Yêu cầu chụp ảnh từ camera tích hợp trên ESP32-CAM.
- Xử lý hình ảnh để nhận diện và trích xuất mã QR.
- Giải mã mã QR để trích xuất thông tin.
- Hiển thị thông tin hoặc thực hiện các hành động phản ứng tương ứng với thông tin đã trích xuất.

*Yêu cầu phi chức năng:*

- Tích hợp một giao diện người dùng đơn giản để hiển thị kết quả hoặc thông báo lỗi.
- Tối ưu hóa hiệu suất và tốc độ xử lý để đảm bảo thời gian phản hồi nhanh chóng.
- Đảm bảo tính ổn định và độ tin cậy của hệ thống.

*Kiến trúc hệ thống:*

- Sử dụng ESP32-CAM làm nền tảng phần cứng chính.
- Phát triển chương trình nhúng sử dụng ngôn ngữ lập trình Arduino hoặc MicroPython để kiểm soát ESP32-CAM.
- Sử dụng thư viện xử lý hình ảnh như OpenCV hoặc các thư viện nhúng như TinyML để nhận diện và giải mã mã QR.
- Tích hợp giao diện người dùng đơn giản sử dụng màn hình hoặc giao tiếp không dây với thiết bị khác như điện thoại thông minh.

*Quản lý dự án và triển khai:*

- Phân chia dự án thành các phần nhỏ hơn có thể thực hiện độc lập.

- Sử dụng phương pháp phát triển phần mềm như Agile để quản lý và theo dõi tiến độ.
- Thực hiện thử nghiệm và kiểm tra đầy đủ trước khi triển khai sản phẩm.

#### *Thử nghiệm và đánh giá:*

- Tiến hành các bài kiểm tra chức năng và bài kiểm tra tính năng để đảm bảo rằng hệ thống hoạt động chính xác và đáng tin cậy.
- Thu thập phản hồi từ người dùng để cải thiện và điều chỉnh sản phẩm.

#### *Bảo trì và hỗ trợ:*

- Cung cấp tài liệu hướng dẫn sử dụng và hỗ trợ kỹ thuật cho người dùng cuối.
- Cập nhật và bảo trì định kỳ để đảm bảo tính ổn định và bảo mật của hệ thống.

#### *Tiềm năng mở rộng:*

- Xem xét các tính năng và ứng dụng mới có thể được tích hợp vào hệ thống trong tương lai, như tích hợp với các dịch vụ đám mây, tích hợp với các hệ thống quản lý, v.v.

## **2 Đối tượng nghiên cứu**

### **Chương trình quét mã QR:**

- Khả năng quét mã QR chính xác và nhanh chóng.
- Khả năng hoạt động trong môi trường có ánh sáng yếu.
- Khả năng quét mã QR từ xa.
- Khả năng quét mã QR bị méo hoặc mờ.

### **Ứng dụng:**

- Giao diện đơn giản và dễ sử dụng.
- Tốc độ xử lý nhanh chóng.
- Hỗ trợ nhiều định dạng mã QR.
- Tính bảo mật cao.

### **3 Phạm vi nghiên cứu**

*Phần cứng:*

- Module ESP32-CAM
- Camera OV2640
- Màn hình LCD (tùy chọn)
- Module cấp nguồn

*Phần mềm:*

- Phần mềm ESP-IDF
- Thư viện mã nguồn mở
- Ứng dụng di động Android/iOS

*Chức năng:*

- Quét mã QR từ camera ESP32-CAM.
- Giải mã thông tin trong mã QR.
- Gửi thông tin giải mã đến ứng dụng di động.
- Ứng dụng di động hiển thị thông tin được gửi từ ESP32-CAM.

### **4 Bố cục đồ án**

Chương 1: Cơ sở lý thuyết

Chương 2: Thiết kế chương trình quét mã QR sử dụng ESP32-Cam

Chương 3: Demo sản phẩm sau khi hoàn thành

## **Chương 1: CỞ SỞ LÝ THUYẾT**

Trong chương 1, em trình bày về cơ sở lý thuyết về Internet Of Thing, giới thiệu về mã QR và ESP32-CAM mà em sử dụng trong đồ án của mình và giới thiệu phần mềm mà em sử dụng để thiết kế chương trình quét mã QR như Arduino IDE, Visual Studio và SQL Server Management Studio.

### **1.1 Giới thiệu về IoT**

#### **1.1.1 Khái niệm IoT**

Internet of Things là một khái niệm thuật ngữ để chỉ các thiết bị vật lý phân cứng có kết nối Internet. Ngày nay các thiết bị IoT đang dần trở nên phổ biến tới từng góc ngách của đời sống, chúng thay thế các thiết bị điện tử truyền thống và cho phép giao tiếp dữ liệu thời gian thực. Vì vậy IoT đang làm cho cấu trúc của thế giới xung quanh chúng ta trở nên thông minh hơn và hiệu quả hơn. Các thiết bị IoT có khả năng thu thập dữ liệu dưới nhiều dạng như âm thanh, mùi, hình ảnh,... và có thể tương tác với con người thông qua hệ thống giao tiếp [1].

#### **1.1.2 Lịch sử phát triển**

- Năm 1982, Khái niệm về các thiết bị thông minh (khái niệm về IoT) được đưa ra khi áp dụng vào máy bán hàng tự động của Coca-Cola
- Năm 1991, Mark Weiser đã đưa ra viễn cảnh về IoT trong bài báo của mình, The computer of the 21<sup>st</sup> century.
- Từ năm 1993 đến 1997, một số công ty đã đưa ra các giải pháp của IoT như Microsoft at work hay Novell's NEST
- Năm 1999, Bill Joy giới thiệu sự kết nối trao đổi thiết bị tới thiết bị (device-to-device) trong dự án "Six webs" của mình tạo ra sự phát triển mới đối với IoT.
- Khái niệm và thuật ngữ Internet of Things lần đầu tiên xuất hiện trong bài nói chuyện của Peter Lewis vào tháng 9/1985 tại Washington. Theo Lewis,



Internet of Things là khi con người sử dụng các quy trình, công nghệ kết nối từ xa với những thiết bị, cảm biến để có thể theo dõi hoạt động và đánh giá xu hướng làm việc của các thiết bị này.

- Thuật ngữ Internet of things cũng được đưa ra 1 cách độc lập năm 1999 bởi Kevin Ashton để chỉ một công nghệ mới lúc đó, RFID (Radio Frequency Identification). Kevin Ashton tin tưởng rằng công nghệ RFID là một công cụ thiết yếu để các máy tính có thể quản lý được các thiết bị cá nhân.
- Theo Cisco system, nếu coi thời điểm “khai sinh” của IoT là thời điểm mà số thiết bị kết nối với nhau trên Internet nhiều hơn số người kết nối, thì đó là vào khoảng năm 2008, 2009. Tỷ lệ số thiết bị kết nối trên số người kết nối vào năm 2003 là 0,08 và vào năm 2010 là 1,84.

Bắt đầu từ hè năm 2010, IoT trở lên nổi tiếng từ vụ rò rỉ thông tin: Google Street View không chỉ làm những bức ảnh 360 độ mà còn lưu trữ một số lượng rất lớn dữ liệu của khách hàng sử dụng các mạng WiFi. Những chiến lược của Google bắt đầu bị xem xét vì không chỉ ảnh hưởng đến thế giới ảo mà ảnh hưởng đến thế giới thực.

Cũng trong năm 2010, Trung Quốc tuyên bố phát triển IoT là ưu tiên chiến lược trong kế hoạch phát triển 5 năm của họ (2010-2014)

Năm 2011, Gartner, một công ty nghiên cứu về thị trường, đưa IoT vào danh sách công nghệ mới xuất hiện cần phát triển đặc biệt

Năm 2012, Hội nghị khoa học lớn nhất của châu Âu là “Internet of things”. Các tạp chí nổi tiếng về công nghệ (Forbes, Fast Company, Wired) đều đề cập đến IoT như một hiện tượng mới.

Tháng 10/2013, IDC, công ty dữ liệu quốc tế, đưa ra báo cáo giá trị của IoT trên thị trường vào năm 2020 sẽ là 8,9 nghìn tỷ USD.

Hiện tại thuật ngữ Internet of things là một trong những thuật ngữ được sử dụng nhiều nhất trong thế giới vạn vật kết nối.

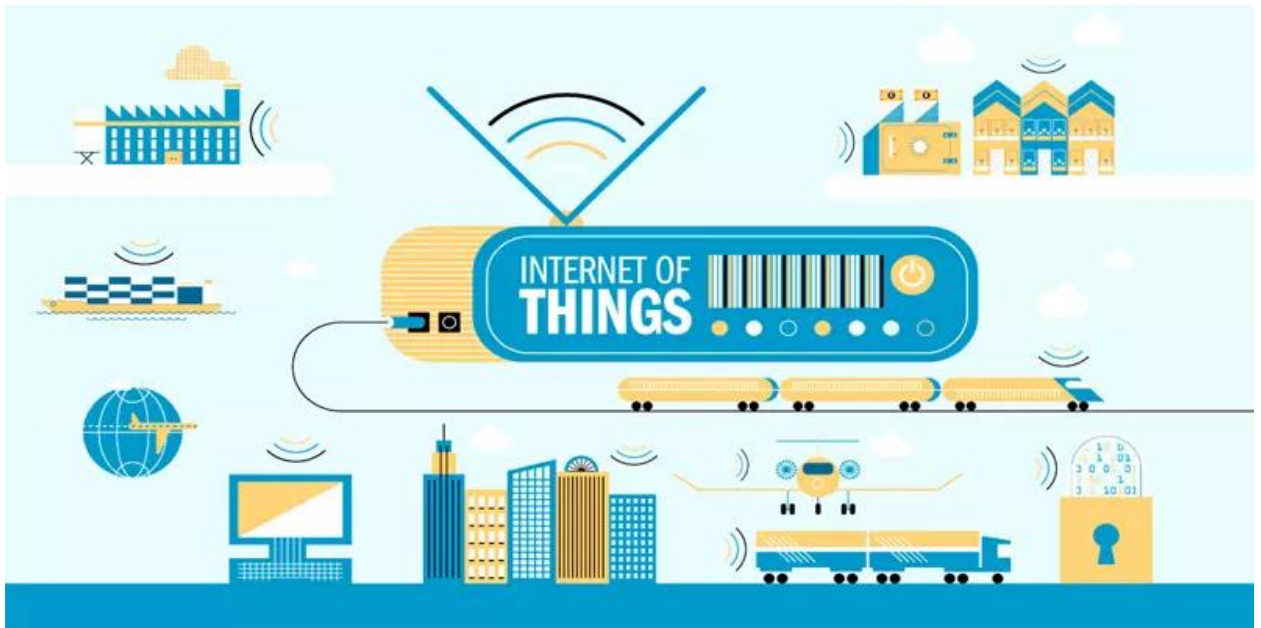
Một số thuật ngữ khác được đề xuất, gần giống với Internet of Things, nhưng có 1 số khác biệt là:

- M2M (Machine to machine) communication
- Web of things
- Industry 4.0
- Industrial Internet (of things)
- Pervasive computing
- Intelligent systems

### **1.1.3 Cấu trúc của hệ thống IoT**

Hệ thống IoT có thể được chia thành 5 thành phần chính, bao gồm:

- Thiết bị IoT (IoT devices): Đây là các thiết bị vật lý được kết nối với Internet và có khả năng thu thập, xử lý và truyền dữ liệu. Thiết bị IoT có thể là các thiết bị gia dụng, thiết bị công nghiệp, thiết bị y tế,...
- Kết nối (connectivity): Đây là thành phần cung cấp khả năng kết nối giữa các thiết bị IoT với nhau và với Internet. Kết nối IoT có thể sử dụng các công nghệ khác nhau, chẳng hạn như Wi-Fi, Bluetooth, Zigbee, LoRa,...
- Cơ sở hạ tầng (infrastructure): Đây là thành phần cung cấp nền tảng cho hoạt động của hệ thống IoT. Cơ sở hạ tầng IoT bao gồm các yếu tố như mạng lưới, trung tâm dữ liệu,...
- Dữ liệu (data): Đây là thành phần quan trọng nhất của hệ thống IoT. Dữ liệu được thu thập từ các thiết bị IoT được sử dụng để phân tích, ra quyết định và thực hiện các hành động.
- Ứng dụng (applications): Đây là thành phần sử dụng dữ liệu từ các thiết bị IoT để cung cấp các giá trị cho người dùng. Ứng dụng IoT có thể được sử dụng trong các lĩnh vực khác nhau, chẳng hạn như sản xuất, nông nghiệp, chăm sóc sức khỏe,....



Hình 1.1 Cấu trúc của hệ thống IoT

(Nguồn: IoT là gì và ứng dụng của IoT trong thực tiễn)

#### 1.1.4 Hoạt động của IoT

Tất cả các hệ thống IoT hoàn chỉnh đều giống nhau ở điểm chung là chúng thể hiện sự tích hợp của bốn thành phần riêng biệt:

- Cảm biến/thiết bị
- Kết nối
- Xử lý dữ liệu
- Giao diện người dùng

##### **Cảm biến/Thiết bị**

Cảm biến và thiết bị là một thành phần quan trọng và không thể thiếu trong hệ thống IoT. Chúng được sử dụng để thu thập dữ liệu từ môi trường.

Ở đây, chúng tôi đồng nhất cảm biến và thiết bị, vì nhiều cảm biến và thiết bị có thể được kết hợp với nhau hoặc cảm biến có thể là một phần của thiết bị.

Ví dụ: Điện thoại của chúng ta là một thiết bị có nhiều cảm biến (máy ảnh, gia tốc kế, GPS...), nhưng điện thoại không chỉ là một cảm biến vì nó còn có thể

thực hiện nhiều hành động khác nhau. Tuy nhiên, cho dù đó là một cảm biến độc lập hay một thiết bị đầy đủ, trong bước đầu tiên này, dữ liệu đang được thu thập từ môi trường bởi một thứ gì đó.

## **Kết nối**

Tiếp theo, dữ liệu đó được gửi đến đám mây, nhưng nó cần một cách để có thể đến được đám mây đó! Các cảm biến/thiết bị có thể được kết nối với đám mây thông qua nhiều phương thức như:

- Mạng di động
- Vệ tinh
- WiFi
- Bluetooth
- Mạng diện rộng công suất thấp (LPWAN)
- Cổng/bộ định tuyến
- Kết nối trực tiếp với Internet qua Ethernet.

Mỗi tùy chọn đều có sự khác biệt về mức tiêu thụ điện năng và phạm vi. Không có kết nối nào là tốt nhất, điều này phụ thuộc vào ứng dụng IoT cụ thể, nhưng tất cả chúng đều hoàn thành cùng một nhiệm vụ: đưa dữ liệu lên đám mây.

## **Xử lý dữ liệu**

Khi dữ liệu được đưa lên đám mây, phần mềm sẽ thực hiện một số xử lý trên đó. Các xử lý này có thể rất đơn giản, chẳng hạn như kiểm tra xem nhiệt độ có nằm trong phạm vi an toàn hay không. Hoặc chúng cũng có thể rất phức tạp, chẳng hạn như sử dụng thị giác máy tính trên video để xác định các đối tượng khả nghi.

Nhưng điều gì sẽ xảy ra khi nhiệt độ quá cao hoặc nếu có kẻ gian đột nhập vào tài sản? Đó chính là nhiệm vụ của giao diện người dùng mà chúng ta sẽ tìm hiểu ở phần tiếp theo.

## **Giao diện người dùng**

Sau cùng, những thông tin sau khi được xử lý hữu ích cho người dùng sẽ được thông báo theo nhiều cách khác nhau, có thể bằng tin nhắn, email, văn bản, thông báo,...

Người dùng có thể có một giao diện riêng, cho phép họ chủ động đăng ký trên hệ thống.

Ví dụ: người dùng có thể kiểm tra các nguồn cấp dữ liệu video trên các thuộc tính khác nhau, thông qua ứng dụng điện thoại hoặc trình duyệt web.

Tuy nhiên, tùy thuộc vào ứng dụng IoT, người dùng cũng có thể thực hiện các hành động gây ảnh hưởng đến hệ thống.

Ví dụ: người dùng có thể điều chỉnh nhiệt độ trong kho lạnh từ xa, thông qua một ứng dụng trên điện thoại.

Ngoài ra, một số hành động cũng có thể được thực hiện tự động hoá, ví dụ như:

- Thay vì đợi điều chỉnh nhiệt độ, hệ thống có thể tự động làm điều đó thông qua các quy tắc được xác định trước
- Thay vì chỉ gọi để cảnh báo về kẻ xâm nhập, hệ thống IoT còn có thể tự động thông báo cho các đội bảo mật hoặc các cơ quan chức năng có liên quan.

## **1.2 Giới thiệu về QR Code**

### **1.2.1 Khái niệm Mã QR Code**

QR Code (Quick response code) hay gọi là mã phản hồi nhanh, và một tên gọi khác là mã vạch ma trận. QR Code là dạng mã vạch hai chiều (2D) có thể được đọc bằng các thiết bị chuyên dùng để đọc mã vạch hoặc các dòng điện thoại smartphone hiện nay đều được trang bị các ứng dụng để đọc mã vạch này, đơn giản là ứng dụng camera của điện thoại hay là Zalo là phổ biến nhất.

Hình QR Code bao gồm các điểm đen và các ô vuông nằm trong một ô vuông lớn có kích thước khác nhau tùy thuộc vào từng loại sản phẩm. QR Code được thiết kế với mục đích thay thế các mã vạch truyền thống thông thường (các mã vạch dạng sọc có các dãy số). Mã

QR Code được đọc nhanh hơn, tiết kiệm thời gian và diện tích hơn so với các loại mã vạch truyền thống.

Các địa chỉ web (URL), địa chỉ email, tin nhắn SMS, thời gian của một sự kiện, thông tin, nội dung văn bản hay là thông tin định vị của một địa điểm nào đó có thể được chứa bởi mã QR. Tùy thuộc vào các mã QR Code khác nhau sẽ chứa những nội dung khác nhau và mỗi nội dung là một mã QR code.



Hình 1.2 Mã QR xét tuyển Trường Đại Nam

Trong Hình 1.2 ta có thể thấy đây là hình ảnh mã QR Code xét tuyển vào trường Đại học Đại Nam. Khi quét vào hình ảnh này thì nó sẽ cung cấp một địa chỉ web chứa thông tin đăng ký xét tuyển của trường.

Mã QR Code cũng tương tự như mã vạch truyền thống thông thường trên bao bì của sản phẩm khi mua hàng dưới dạng mã vạch. Điểm khác nhau giữa mã QR Code và mã vạch truyền thống là lượng dữ liệu chúng nắm giữ hay chia sẻ và cả về hình dạng của mã.

Loại mã vạch truyền thống chỉ có thể lưu giữ được 20 số chữ số còn trong đó loại mã QR hai chiều có thể lưu giữ thông tin lên tới hàng ngàn ký tự chữ số. Qua đó thấy được mã QR chứa nhiều thông tin, dễ dàng để tạo và đọc hơn so với mã vạch truyền thống thông thường.

Để tạo ra một mã QR Code hiện nay là một điều hết sức dễ dàng và tiện lợi. Như vậy chỉ cần lên google gõ từ khóa "QR code generator" là đã có thể tạo ra một mã QR Code riêng với công cụ miễn phí.

### 1.2.2 Lịch sử phát triển

Mã QR (Quick Response Code) được phát minh vào năm 1994 bởi Masahiro Hara, một kỹ sư của công ty Denso Wave thuộc Tập đoàn Denso Nhật Bản. Mục đích ban đầu của ông là tạo ra một loại mã vạch có thể được quét nhanh chóng và chính xác hơn so với mã vạch truyền thống, nhằm đáp ứng nhu cầu theo dõi chi tiết các bộ phận ô tô trong quá trình sản xuất.

Qua thời gian dài sáng tạo và phát triển mã QR đã trở thành một phần không thể thiếu trong đời sống hiện nay:

- 1994: Tháng 4, Masahiro Hara nộp đơn xin cấp bằng sáng chế cho mã QR. Tháng 8, mã QR chính thức được ra mắt tại thị trường Nhật Bản.
- 1997: Tiêu chuẩn mã QR được ISO phê chuẩn.
- 2000: Mã QR bắt đầu được sử dụng rộng rãi trong ngành công nghiệp ô tô.
- 2001: Denso Wave phát hành phiên bản mã QR màu đầu tiên.
- 2002: Mã QR được tích hợp vào các điện thoại thông minh đầu tiên.
- 2005: Mã QR trở nên phổ biến hơn với người tiêu dùng nhờ sự phát triển của điện thoại thông minh có camera.
- 2010: Mã QR được sử dụng trong nhiều ứng dụng khác nhau, bao gồm tiếp thị, thanh toán và giáo dục.
- 2011: Mã QR đóng vai trò quan trọng trong việc giải cứu các nạn nhân của thảm họa động đất và sóng thần Nhật Bản.
- 2014: Mã QR được sử dụng trong chiến dịch World Cup FIFA 2014.
- 2015: Mã QR được sử dụng rộng rãi trong các ứng dụng thanh toán di động như Alipay và WeChat Pay.
- 2017: Mã QR được sử dụng trong các ứng dụng truyền thông xã hội như Snapchat và Instagram.

- 2020: Mã QR được sử dụng rộng rãi trong đại dịch COVID-19 để theo dõi tiếp xúc và truy cập thông tin.

Hiện nay, việc sử dụng mã QR trong hầu hết mọi lĩnh vực của cuộc sống, từ tiếp thị và thanh toán đến giáo dục và giải trí. Mã QR là một công nghệ mạnh mẽ và linh hoạt có thể được sử dụng để lưu trữ và truy cập thông tin một cách nhanh chóng và dễ dàng

## **1.3 Giới thiệu về ESP32-CAM**

### **1.3.1 ESP32-CAM**

Mạch Thu Phát Wifi ESP32-CAM là mạch tích hợp với bộ xử lý chính là module ESP32 + Camera OV2640 được sử dụng trong các ứng dụng truyền hình ảnh, xử lý ảnh qua Wifi, Bluetooth hoặc các ứng dụng IoT.

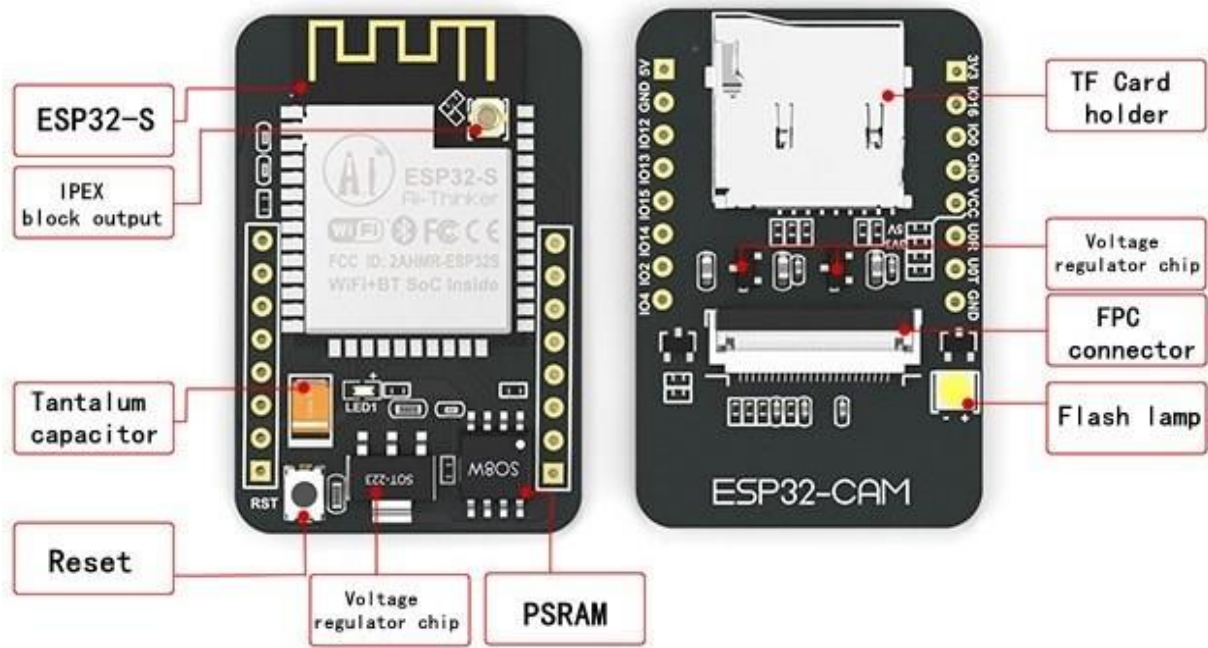
ESP32-CAM có một camera kích thước nhỏ, rất cạnh tranh trong ngành, giống như mô-đun chính, mô-đun này có thể được xử lý công việc độc lập, module có kích thước nhỏ gọn chỉ 40 x 27 x12 mm, dòng nghỉ chỉ 6mA.

ESP-32CAM có thể được sử dụng rộng rãi trong các ứng dụng IoT khác nhau, thích hợp cho thiết bị thông minh gia đình, điều khiển không dây công nghiệp, giám sát không dây kiểm soát, nhận dạng không dây QR, tín hiệu hệ thống định vị không dây... Nó là một giải pháp lý tưởng cho các ứng dụng IoT.

Mạch thu phát Wifi BLE ESP32 này là mạch chính hãng AI – Thinker có chất lượng độ ổn định và độ bền rất cao, sử dụng camera OV2640 chất lượng cao hình ảnh sắc nét, không nhiễu sọc, không xảy ra tình trạng treo khi hoạt động do sử dụng ic cấp nguồn chất lượng cao.

Mạch thu phát Wifi BLE ESP32-CAM Ai-Thinker này có thể sử dụng Arduino IDE để biên dịch và viết code, được hỗ trợ mạnh mẽ từ cộng đồng.





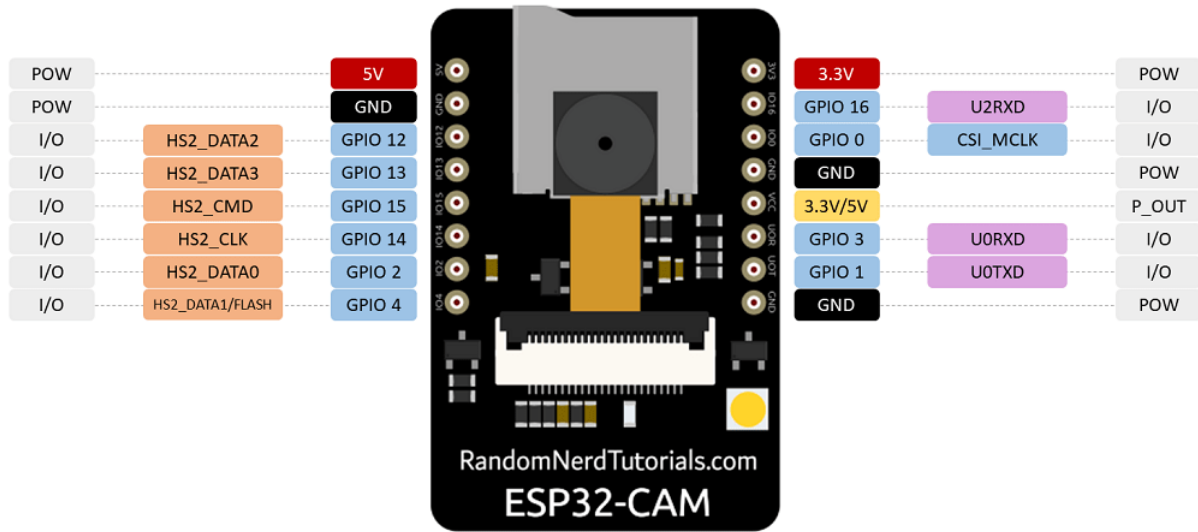
Hình 1.3 ESP32-CAM AI-Thinker

### 1.3.2 Thông số cơ bản

- IC chính: ESP32-S (AI-Thinker)
- Mô-đun Wi-Fi BT SoC 802.11 b/g/n/e/i
- CPU 32-bit công suất thấp, cũng có thể phục vụ bộ xử lý ứng dụng
- Tốc độ đồng hồ lên đến 160MHz, sức mạnh tính toán lên đến 600 DMIPS
- Tích hợp 520 KB SRAM, 4MPSRAM bên ngoài
- Dải tần số: 1421 ~ 2484 Mhz
- Bluetooth: 4.2 BR/EDR BLE
- Hỗ trợ UART / SPI / I2C / PWM / ADC / DAC
- Hỗ trợ máy ảnh OV2640 và OV7670, đèn flash tích hợp
- Hỗ trợ tải lên WiFi hình ảnh
- Hỗ trợ thẻ TF
- Hỗ trợ nhiều chế độ ngủ
- Nhúng Lwip và FreeRTOS
- Hỗ trợ chế độ hoạt động STA / AP / STA + AP
- Hỗ trợ cấu hình thông minh / công nghệ AirKiss

- Hỗ trợ nâng cấp cục bộ và từ xa cho công nối tiếp (FOTA)

### Sơ đồ chân:

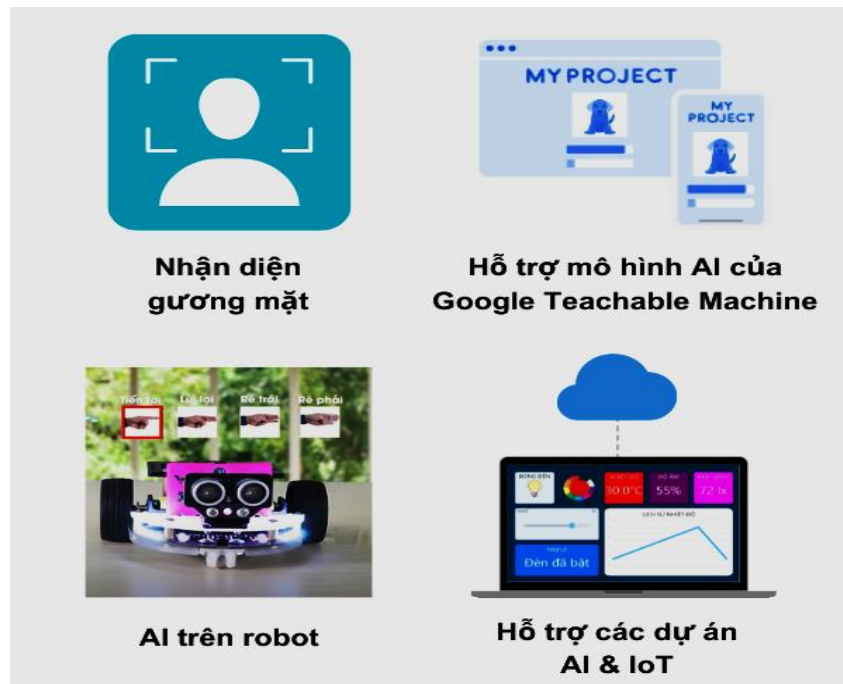


Hình 1.4 Sơ đồ chân ESP32-CAM

### 1.3.3 Ứng dụng của ESP32-CAM

ESP-32CAM có thể được sử dụng rộng rãi trong các ứng dụng IoT khác nhau, thích hợp cho thiết bị thông minh gia đình, điều khiển không dây công nghiệp, giám sát không dây kiểm soát, nhận dạng không dây QR, tín hiệu hệ thống định vị không dây... Nó là một giải pháp lý tưởng cho các ứng dụng IoT.

ESP32 Camera được dùng để stream video từ camera, xử lý các model AI, hỗ trợ chạy mô hình AI của Google Teachable Machine, dùng trong các ứng dụng nhận dạng hình ảnh, nhận dạng khuôn mặt và các mô hình AI người dùng tự xây dựng cho robot và các ứng dụng AI và IoT.



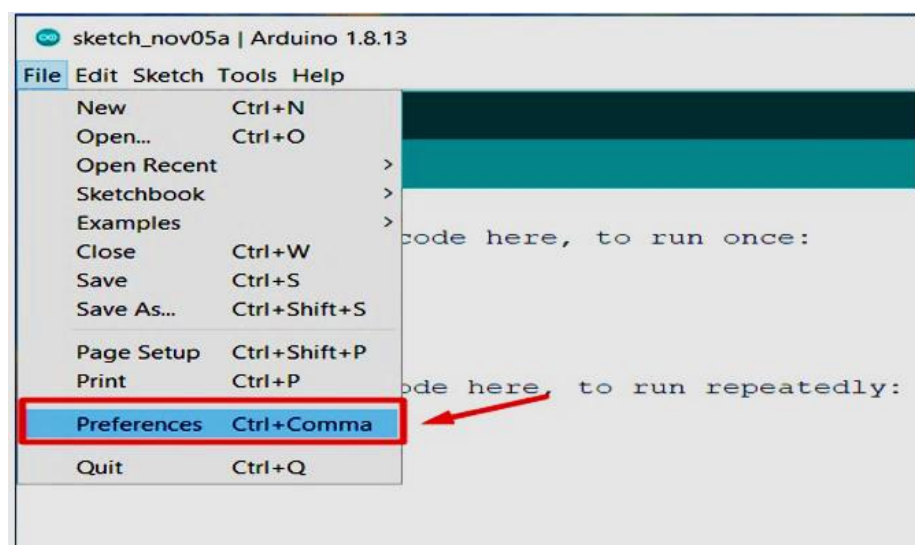
Hình 1.5 Ứng dụng của ESP32-CAM

#### 1.3.4 Hướng dẫn cài đặt ESP32 với Arduino

Để sử dụng module ESP32-CAM với phần mềm Arduino ta cần một module ESP32-CAM.

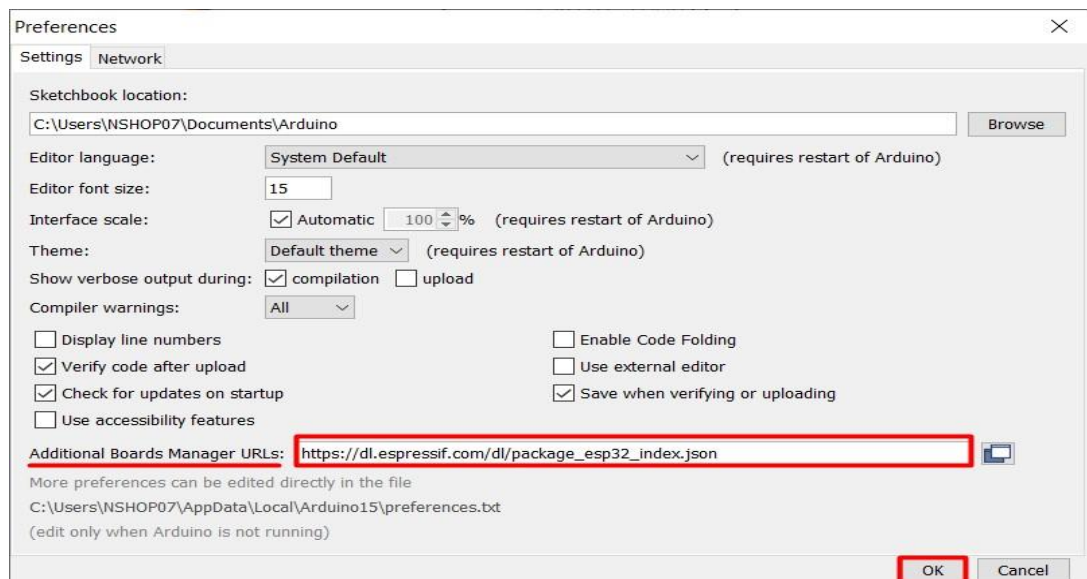
Để cài đặt board ESP32 trong Arduino IDE, cần làm theo các bước hướng dẫn sau:

Sau khi mở phần mềm Arduino, ấn vào File -> chọn Preferences.



Hình 1.6 Chọn File và chọn Preferences

Nhập “[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)” vào ô “Additional Board Manager URLs” như hình bên dưới, sau đó nhấn OK.

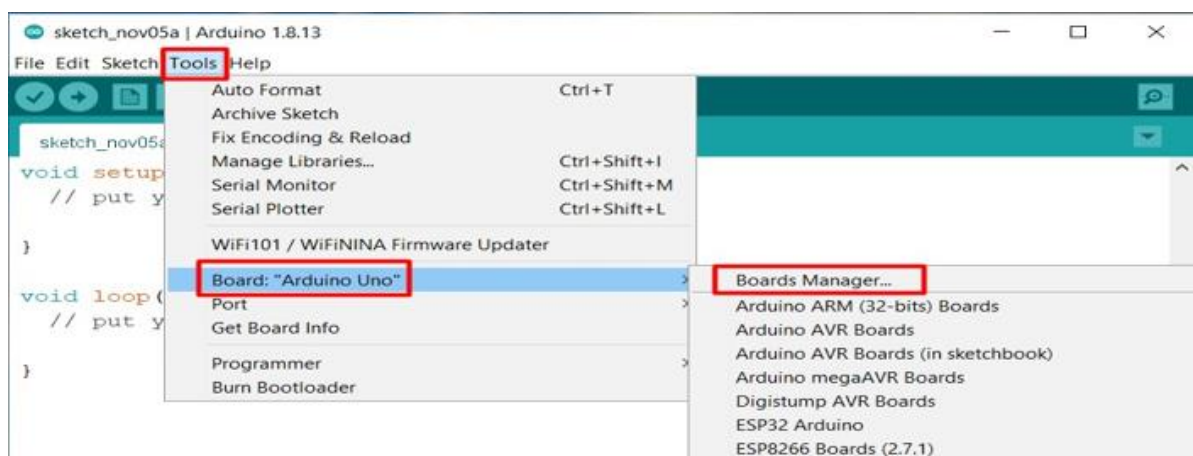


Hình 1.7: Nhập link vào ô *Additional Board Manager URLs*

Lưu ý: nếu chúng ta đã có URL board ESP8266, có thể phân tách các URL bằng dấu phẩy như sau:

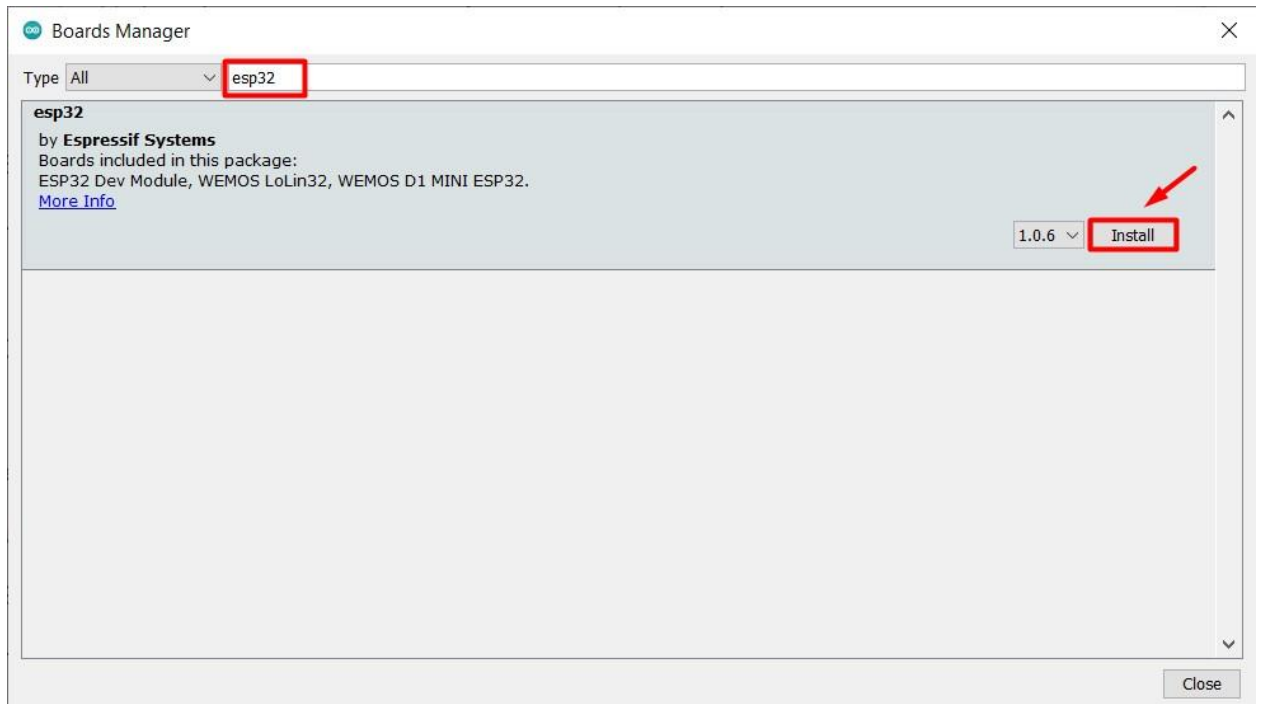
```
https://dl.espressif.com/dl/package_esp32_index.json,  
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

Mở Board Manager: Chọn Tools > Board > Boards Manager...



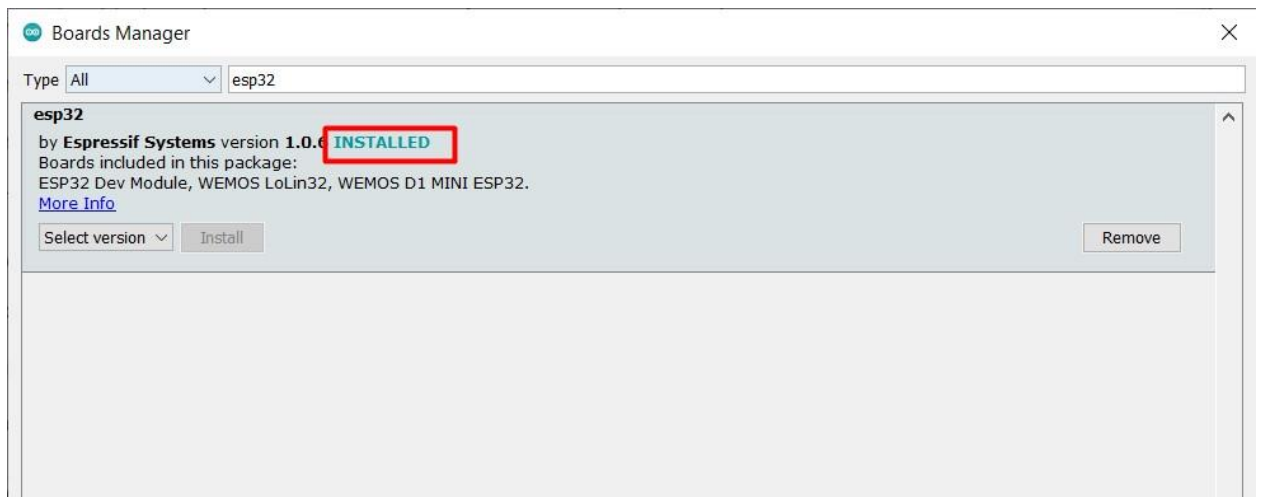
Hình 1.8: Mở Board Manager

Gõ esp32 vào ô tìm kiếm, sau đó nhấn Install để cài đặt “ESP32 by Espressif Systems”



Hình 1.9: Tải thư viện esp32

Sau khi báo Installed là đã cài đặt xong

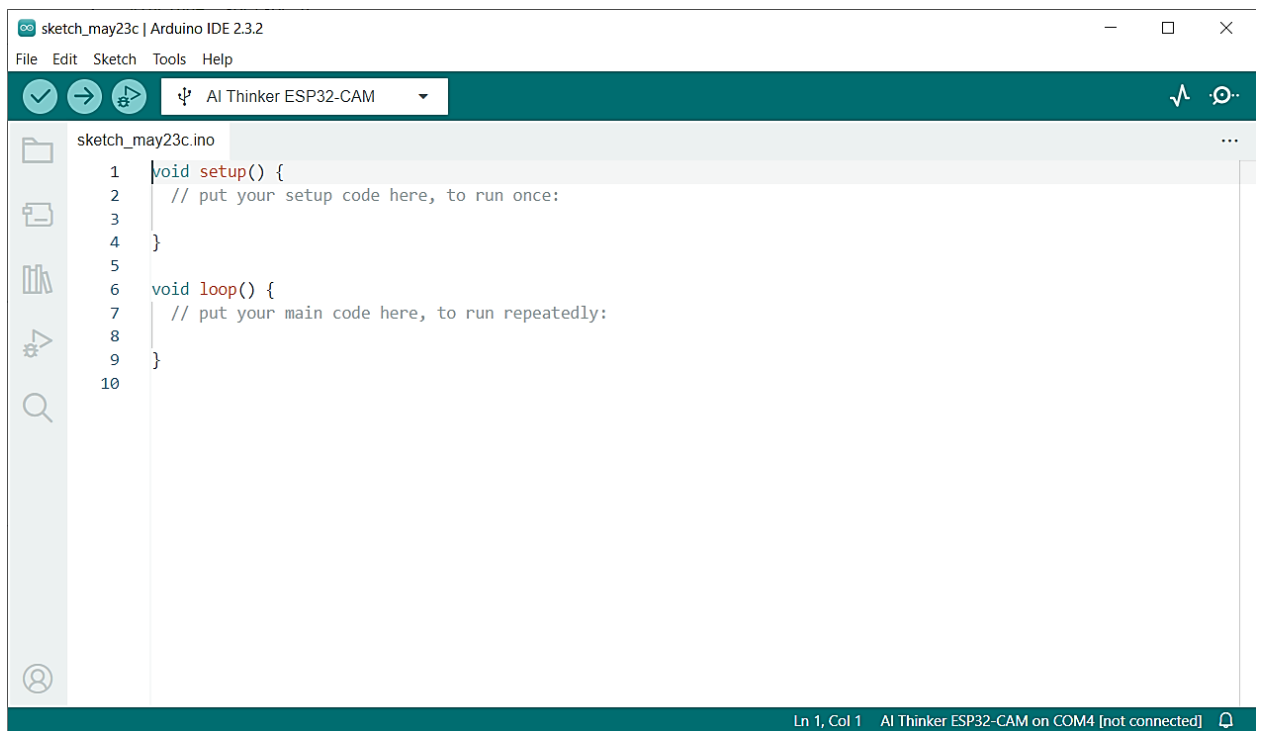


Hình 1.10: Hoàn thành cài đặt thư viện esp32

## 1.4 Giới thiệu về Arduino IDE

### 1.4.1 Arduino IDE

Arduino IDE là một phần mềm với một mã nguồn mở, được sử dụng chủ yếu để viết và biên dịch mã vào module Arduino [5]. Nó bao gồm phần cứng và phần mềm. Phần cứng chứa đến 300,000 board mạch được thiết kế sẵn với các cảm biến, linh kiện. Phần mềm giúp sử dụng các cảm biến, linh kiện ấy của Arduino một cách linh hoạt phù hợp với mục đích sử dụng.



Hình 1.11: Phần mềm Arduino IDE

Đây là một phần mềm Arduino chính thống, giúp biên dịch mã trở nên dễ dàng, ngay cả một người bình thường không có kiến thức kỹ thuật cũng có thể làm được.

#### 1.4.2 Hoạt động của Arduino IDE

Arduino IDE hoạt động bằng cách chuyển đổi mã được viết thành ngôn ngữ mà bo mạch Arduino có thể hiểu và thực thi. Để có thể sử dụng được phần mềm Arduino thì đầu tiên cần viết mã code chương trình, sau đó biên dịch mã, tải mã cho phần mềm rồi chạy chương trình vừa được tải.

Viết mã:

- Viết mã cho chương trình Arduino của mình trong cửa sổ chỉnh sửa mã của Arduino IDE. Ngôn ngữ lập trình được sử dụng là C/C++ nhưng được đơn giản hóa để dễ sử dụng hơn.
- Mã bao gồm các hướng dẫn cho bo mạch Arduino thực hiện, chẳng hạn như đọc dữ liệu từ cảm biến, điều khiển động cơ và hiển thị thông tin trên màn hình LCD.

Biên dịch mã:

- Khi nhấp vào nút "Biên dịch", Arduino IDE sẽ chuyển đổi mã thành mã máy mà bo mạch Arduino có thể hiểu.
- Quá trình biên dịch bao gồm kiểm tra cú pháp mã để tìm lỗi và sau đó chuyển đổi mã thành ngôn ngữ cấp thấp mà bo mạch Arduino có thể thực thi.

Tải mã:

- Khi nhấp vào nút "Tải", Arduino IDE sẽ tải mã đã biên dịch lên bo mạch Arduino.
- Việc tải mã được thực hiện thông qua cáp USB kết nối bo mạch Arduino với máy tính.

Chạy chương trình:

- Khi mã được tải lên bo mạch Arduino, nó sẽ tự động chạy.
- Bo mạch Arduino sẽ thực hiện các hướng dẫn trong mã, chẳng hạn như đọc dữ liệu từ cảm biến, điều khiển động cơ và hiển thị thông tin trên màn hình LCD.

Phần mềm Arduino IDE còn có nhiều tính năng giúp phát triển các ứng dụng Arduino:

- Hỗ trợ gỡ lỗi: Arduino IDE cung cấp các công cụ gỡ lỗi giúp xác định và sửa lỗi trong mã của mình.
- Giả lập: Arduino IDE có trình giả lập cho phép thử nghiệm mã của mình mà không cần kết nối với bo mạch Arduino thực.

- Quản lý thư viện: Arduino IDE cung cấp một trình quản lý thư viện giúp cài đặt và sử dụng các thư viện được viết sẵn cho các tác vụ phổ biến.
- Cộng đồng trực tuyến: Arduino có một cộng đồng trực tuyến lớn và tích cực, luôn sẵn sàng giúp đỡ và chia sẻ kiến thức.

Arduino IDE là một công cụ mạnh mẽ và linh hoạt giúp tạo ra các ứng dụng điện tử tương tác với Arduino. Nó hoạt động bằng cách chuyển đổi mã hành ngôn ngữ mà bo mạch Arduino có thể hiểu và thực thi.

### 1.4.3 Ưu điểm của Arduino IDE

Dễ sử dụng:

- Arduino IDE được thiết kế để dễ sử dụng cho người mới bắt đầu, ngay cả khi họ không có kiến thức chuyên sâu về lập trình hoặc điện tử.
- Giao diện đơn giản và trực quan với các tính năng được sắp xếp hợp lý.
- Sử dụng ngôn ngữ lập trình C/C++ được đơn giản hóa, dễ học và sử dụng hơn so với ngôn ngữ lập trình truyền thống.
- Cung cấp nhiều hướng dẫn, tài liệu và ví dụ tiện lợi cho việc sử dụng.

Linh hoạt:

- Hỗ trợ nhiều bo mạch Arduino khác nhau, giúp chọn bo mạch phù hợp với dự án của mình.
- Cung cấp nhiều thư viện được viết sẵn cho các tác vụ phổ biến, chẳng hạn như giao tiếp với cảm biến, điều khiển động cơ và hiển thị thông tin trên màn hình LCD.
- Cho phép mở rộng chức năng của Arduino bằng cách viết mã của riêng hoặc sử dụng các thư viện của bên thứ ba.

Miễn phí và mã nguồn mở:

- Arduino IDE là phần mềm miễn phí và mã nguồn mở, có thể tải xuống và sử dụng miễn phí mà không cần phải trả bất kỳ khoản phí nào.
- Cộng đồng Arduino lớn và tích cực luôn sẵn sàng giúp đỡ và chia sẻ kiến thức.



- Mã nguồn mở cho phép tùy chỉnh và sửa đổi phần mềm theo nhu cầu của mình.

Cộng đồng lớn:

- Arduino có một cộng đồng người dùng lớn và tích cực trên toàn thế giới.
- Dễ dàng tìm thấy sự trợ giúp và hỗ trợ trực tuyến từ các diễn đàn, nhóm và trang web cộng đồng.
- Cộng đồng luôn sẵn sàng chia sẻ kiến thức, kinh nghiệm và các dự án của họ.

Phù hợp cho nhiều đối tượng:

- Arduino IDE phù hợp cho nhiều đối tượng sử dụng, từ học sinh, sinh viên đến các nhà phát triển chuyên nghiệp, nghệ sĩ và nhà sáng tạo.
- Dễ sử dụng cho người mới bắt đầu nhưng cũng đủ mạnh mẽ cho các dự án phức tạp.
- Có thể sử dụng để tạo ra nhiều loại dự án khác nhau, từ đơn giản đến phức tạp.

Ngoài những lý do chính trên, Arduino IDE còn có nhiều ưu điểm khác như:

- Hỗ trợ nhiều hệ điều hành khác nhau (Windows, macOS, Linux).
- Cung cấp các công cụ gỡ lỗi để giúp xác định và sửa lỗi trong mã của mình.
- Cho phép mô phỏng chương trình của mình mà không cần kết nối với bo mạch Arduino thực.
- Cung cấp các bản cập nhật và cải tiến thường xuyên.

## **1.5 Giới thiệu SQL Server Management Studio**

### **1.5.1 SQL Server Management Studio**

Microsoft SQL Server Management Studio (SSMS) là một ứng dụng phần mềm do Microsoft phát triển, được sử dụng để định cấu hình, quản lý và quản lý tất cả các thành phần trong Microsoft SQL Server. Được ra mắt lần đầu tiên với Microsoft SQL Server 2005, nó là phiên bản kế thừa của Enterprise Manager trong

SQL 2000 trở về trước. Công cụ này bao gồm cả trình soạn thảo tập lệnh và công cụ đồ họa hoạt động với các đối tượng và tính năng của máy chủ.

Tính năng trung tâm của SSMS là Object Explorer, cho phép người dùng duyệt, chọn và hành động theo bất kỳ đối tượng nào trong máy chủ. Nó cũng cung cấp một phiên bản Express riêng biệt có thể tải xuống miễn phí; tuy nhiên các phiên bản SSMS gần đây hoàn toàn có khả năng kết nối và quản lý mọi phiên bản SQL Server Express. Microsoft cũng kết hợp khả năng tương thích ngược cho các phiên bản SQL Server cũ hơn, do đó cho phép phiên bản SSMS mới hơn kết nối với các phiên bản SQL Server cũ hơn. Nó cũng đi kèm với Microsoft SQL Server Express 2012 hoặc người dùng có thể tải xuống riêng.

Bắt đầu từ phiên bản 11, ứng dụng dựa trên Visual Studio 10 shell, sử dụng WPF cho giao diện người dùng. Các phiên bản 18 trở đi dựa trên Visual Studio 17 Isolate Shel .

Vào tháng 6 năm 2015, Microsoft đã công bố ý định phát hành các phiên bản SSMS trong tương lai độc lập với các bản phát hành công cụ cơ sở dữ liệu SQL Server.

Trong đồ án của bản thân, SSMS được em sử dụng để thiết kế database lưu trữ thông tin của mã QR Code sau khi ESP32-CAM quét đọc thông tin trong mã QR.

### **1.5.2 Lịch sử phát triển**

- SQL Server 6.5 (1995): Phiên bản đầu tiên của SSMS được phát hành cùng với SQL Server 6.5. Phiên bản này chỉ có một số tính năng cơ bản, chẳng hạn như khả năng kết nối với máy chủ, quản lý cơ sở dữ liệu và viết truy vấn SQL.
- SQL Server 7.0 (1999): SSMS 7.0 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý đối tượng database, giám sát hiệu suất máy chủ và cấu hình các tính năng bảo mật.
- SQL Server 2000 (2000): SSMS 2000 tiếp tục cải thiện các tính năng hiện có và bổ sung thêm nhiều tính năng mới, bao gồm khả năng sao lưu và khôi phục

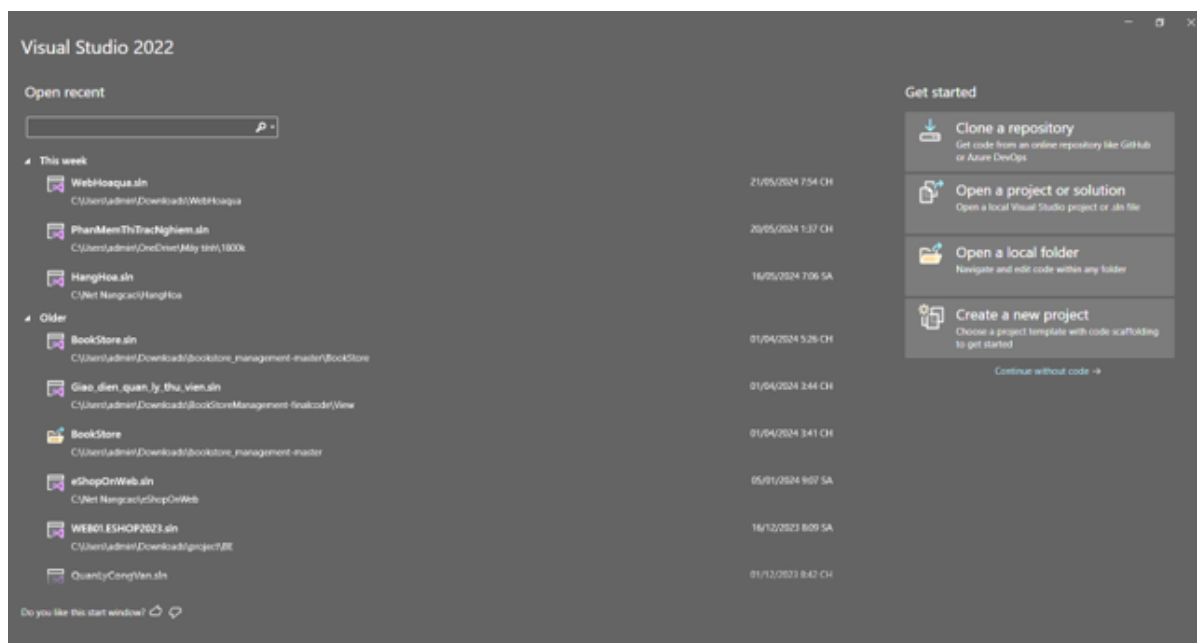
cơ sở dữ liệu, quản lý nhân viên bảo mật (DMO) và phát triển ứng dụng web với ASP.NET.

- SQL Server 2005 (2005): SSMS 2005 là một phiên bản thay đổi lớn, với giao diện người dùng được thiết kế lại hoàn toàn và nhiều tính năng mới, bao gồm khả năng quản lý dữ liệu XML, quản lý báo cáo và phân tích dữ liệu.
- SQL Server 2008 (2008): SSMS 2008 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Integration Services (SSIS), SQL Server Analysis Services (SSAS) và SQL Server Reporting Services (SSRS).
- SQL Server 2012 (2012): SSMS 2012 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Always On Availability Groups, PowerPivot và PolyBase.
- SQL Server 2014 (2014): SSMS 2014 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Managed Backup, SQL Server Mobile Reporting và Hekaton Analytics.
- SQL Server 2016 (2016): SSMS 2016 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Always On Failover Cluster Instances (FCI), SQL Server Data Discovery và Azure SQL Database.
- SQL Server 2017 (2017): SSMS 2017 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server on Linux, SQL Server Intelligent Query Processing (IQP) và SQL Server Machine Learning.
- SQL Server 2019 (2019): SSMS 2019 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Big Data Clusters, SQL Server Data Warehouse on Azure và SQL Server Intelligent Data Optimization (IDO).
- SQL Server 2022 (2022): SSMS 2022 bổ sung thêm nhiều tính năng mới, bao gồm khả năng quản lý SQL Server Always On Availability Groups for Azure SQL Database, SQL Server Intelligent Database Management (IDM) và SQL Server Data Governance.

## **1.6 Giới thiệu về Visual Studio**

### **1.6.1 Visual Studio**

Microsoft Visual Studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Microsoft Visual Studio còn được gọi là "Trình soạn thảo mã nhiều người sử dụng nhất thế giới", được dùng để lập trình C++ và C# là chính. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Store và Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý [4].



Hình 1.12: Visual Studio 2022

Visual Studio bao gồm một trình soạn thảo mã hỗ trợ IntelliSense cũng như cải tiến mã nguồn. Trình gỡ lỗi tích hợp hoạt động cả về trình gỡ lỗi mức độ mã nguồn và gỡ lỗi mức độ máy. Công cụ tích hợp khác bao gồm một mẫu thiết kế các hình thức xây dựng giao diện ứng dụng, thiết kế web, thiết kế lớp và thiết kế giản đồ cơ sở dữ liệu. Nó chấp nhận các plug-in nâng cao các chức năng ở hầu hết các cấp bao gồm thêm hỗ trợ cho các hệ thống quản lý phiên bản (như Subversion) và bổ sung thêm bộ công cụ mới như biên tập và thiết kế trực quan cho các miền ngôn ngữ cụ thể hoặc bộ công cụ dành cho các khía cạnh khác trong quy trình phát triển phần mềm.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI (thông qua Visual C++), VB.NET (thông qua Visual Basic.NET), C# (thông qua Visual C#) và F# (như của Visual Studio 2010). Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML/XHTML, JavaScript và CSS.

### 1.6.2 Tính năng của Visual Studio

Visual Studio cung cấp nhiều tính năng để giúp lập trình viên viết, gỡ lỗi và triển khai mã, bao gồm:

- Trình soạn thảo mã: Visual Studio có một trình soạn thảo mã mạnh mẽ với nhiều tính năng như tô sáng cú pháp, tự động hoàn thành mã, kiểm tra lỗi cú pháp và refactor code.
- Trình gỡ lỗi: Visual Studio có một trình gỡ lỗi tích hợp cho phép gỡ lỗi mã của mình từng dòng, đặt điểm ngắt và kiểm tra giá trị của các biến.
- Công cụ thiết kế: Visual Studio bao gồm các công cụ thiết kế cho tạo giao diện người dùng cho các ứng dụng Windows Forms, WPF và web.
- Kiểm soát phiên bản: Visual Studio tích hợp với các hệ thống kiểm soát phiên bản phổ biến như Git và Subversion, cho phép theo dõi thay đổi mã của mình và cộng tác với các lập trình viên khác.
- Triển khai: Visual Studio cung cấp các công cụ để giúp triển khai mã của mình lên máy chủ sản xuất.

Ngoài ra, Visual Studio còn có rất nhiều tính năng khác như:

- Trình duyệt đối tượng: Cho phép duyệt các lớp và không gian tên trong các thư viện .NET.
- Trình khám phá dữ liệu: Cho phép kết nối với cơ sở dữ liệu và xem và chỉnh sửa dữ liệu.
- Trình biên tập XML: Cho phép chỉnh sửa tệp XML.

- Công cụ quản lý dự án: Cho phép quản lý các dự án phát triển phần mềm của mình.

Visual Studio là một công cụ mạnh mẽ và linh hoạt được sử dụng bởi các lập trình viên ở mọi cấp độ kinh nghiệm. Nó là một lựa chọn phổ biến cho phát triển các ứng dụng Windows, web và di động.

Các lợi ích của việc sử dụng Visual Studio:

- Tăng năng suất: Visual Studio cung cấp nhiều tính năng giúp viết mã nhanh hơn và dễ dàng hơn.
- Ít lỗi hơn: Visual Studio có nhiều tính năng giúp viết mã không có lỗi hơn.
- Mã dễ bảo trì hơn: Visual Studio cung cấp các công cụ giúp viết mã dễ bảo trì hơn.
- Làm việc nhóm hiệu quả hơn: Visual Studio tích hợp với các hệ thống kiểm soát phiên bản và các công cụ cộng tác khác, giúp làm việc nhóm hiệu quả hơn.

### **1.6.3 Lịch sử phát triển**

Visual Studio đã phát triển qua rất nhiều năm. Visual Studio đã liên tục được cải tiến và cập nhật trong suốt lịch sử của nó từ năm 1997 đến nay. Qua từng đầy thời gian, Visual Studio đã và đang là một trong các công cụ hỗ trợ phổ biến cho lập trình viên ở mọi cấp độ kinh nghiệm và được sử dụng để phát triển nhiều loại ứng dụng khác nhau, bao gồm ứng dụng Windows, web, di động và game [4].

- 1997: Visual Studio 1.0 được phát hành cho Windows. Phiên bản này bao gồm các công cụ để phát triển các ứng dụng C++ và Visual Basic.
- 2000: Visual Studio 6.0 được phát hành, bao gồm hỗ trợ cho các ngôn ngữ mới như C# và Java.
- 2002: Visual Studio .NET được phát hành, là phiên bản đầu tiên dựa trên nền tảng .NET Framework.

- 2005: Visual Studio 2005 được phát hành, bao gồm nhiều tính năng mới như công cụ thiết kế Windows Forms được cải thiện và hỗ trợ cho Visual Basic Express.
- 2008: Visual Studio 2008 được phát hành, bao gồm hỗ trợ cho C++/CLI, F# và WPF.
- 2010: Visual Studio 2010 được phát hành, bao gồm nhiều tính năng mới như IntelliSense được cải thiện, hỗ trợ cho Windows 8 và Windows Phone 7.
- 2012: Visual Studio 2012 được phát hành, bao gồm hỗ trợ cho .NET Framework 4.5, Windows 8.1 và Visual Studio Tools for Office.
- 2013: Visual Studio 2013 được phát hành, bao gồm hỗ trợ cho .NET Framework 4.5.1, Git và Team Foundation Server 2013.
- 2015: Visual Studio 2015 được phát hành, bao gồm hỗ trợ cho .NET Framework 4.6, Universal Windows Apps và Visual Studio Code.
- 2017: Visual Studio 2017 được phát hành, bao gồm hỗ trợ cho .NET Framework 4.7, Visual Studio for Mac và Live Share.
- 2019: Visual Studio 2019 được phát hành, bao gồm hỗ trợ cho .NET Framework 4.8, C++20, Visual Studio IntelliCode và Visual Studio Live.
- 2022: Visual Studio 2022 được phát hành, bao gồm hỗ trợ cho .NET 6, C# 10, F# 6 và Visual Studio Build Tools.

## 1.7 Tiểu kết

Việc sử dụng ESP32-CAM cho phép chúng ta kết nối với internet và truy cập vào dữ liệu hình ảnh một cách dễ dàng. Khi kết hợp với mã QR code, chúng ta có thể tạo ra các giải pháp hiệu quả cho việc quản lý, theo dõi và giao tiếp thông tin.

Mã QR code cung cấp một cách tiện lợi và linh hoạt để chia sẻ dữ liệu, từ thông tin cá nhân đến địa chỉ URL và nhiều hơn nữa. Khi tích hợp vào các ứng dụng IoT, chúng ta có thể tận dụng sức mạnh của mã QR code để tạo ra các hệ thống tự động, từ việc đăng nhập vào mạng Wi-Fi cho đến theo dõi thông tin sản phẩm và dịch vụ.

## Chương 2: THIẾT KẾ CHƯƠNG TRÌNH QUÉT MÃ QR

### SỬ DỤNG ESP32-CAM

Đến với chương 2, em trình bày về chương trình quét mã QR. Đầu tiên, em giới thiệu về mã QR code. Sau đó, em nói về các công việc trong chương trình quét mã QR, phân tích hoạt động chương trình, vẽ lưu đồ thuật toán và thiết kế chương trình quét mã QR.

### 2.1 Chương trình của quét mã QR

#### 2.1.1 Thiết kế cơ sở dữ liệu.

Cơ sở dữ liệu được thiết kế bằng SQL Server Management Studio. Có bảng sau: ChiTiethangHoa(MaSP,TenSP,LoaiSP, SL)

Bảng 1: Bảng chi tiết hàng hóa.

Tên bảng	Mô tả bảng	Tên thuộc tính	Ý nghĩa	Kiểu dữ liệu
ChiTietHangHoa	Mô tả thông tin hàng hóa	MaSP	Mã sản phẩm	Nchar(10)
		TenSP	Tên sản phẩm	Nvarchar(Max)
		LoaiSP	Loại sản phẩm	Varchar(Max)
		SL	Số lượng của sản phẩm	Int

#### 2.1.2 Thiết kế chương trình quét mã QR

Cấu hình và khởi tạo:

- Khai báo GPIO: Đầu tiên, cần xác định các GPIO mà ESP32-CAM sử dụng để kết nối với camera và các linh kiện khác.
- Cấu hình camera: Thiết lập các cấu hình của camera như kích thước khung hình, định dạng pixel, và chất lượng JPEG.



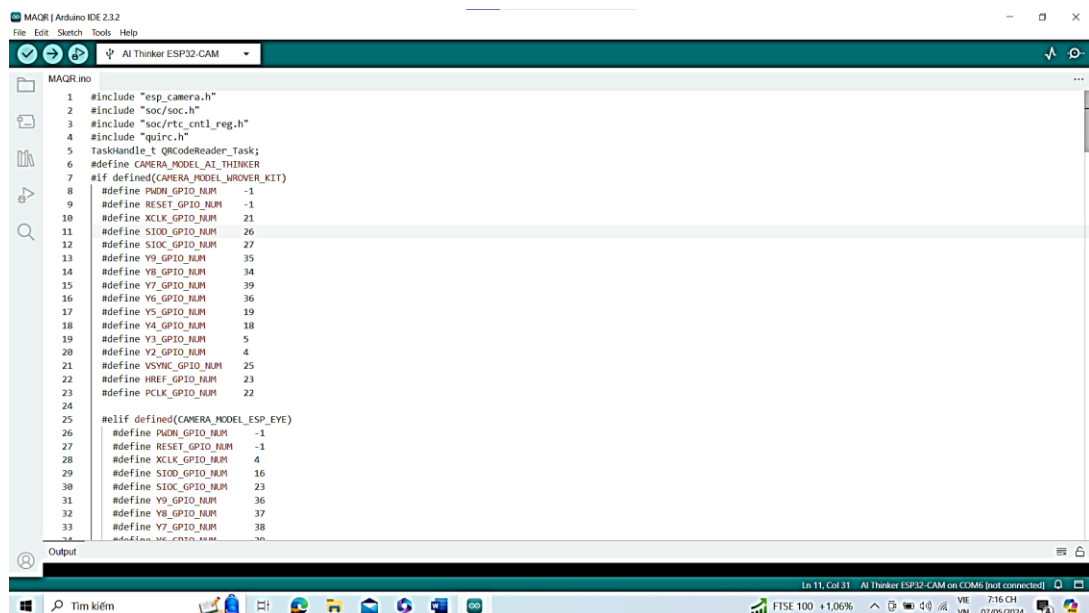
- Khởi tạo thư viện quét mã QR: Sử dụng thư viện phù hợp để quét và giải mã mã QR. Trong trường hợp này có thể sử dụng thư viện quirc.

Đọc và xử lý mã QR:

- Tạo task cho việc quét mã QR: Vì việc quét mã QR có thể mất thời gian, việc sử dụng một task riêng biệt sẽ giúp chương trình không bị chặn.
- Chụp ảnh từ camera: Sử dụng camera để chụp ảnh, sau đó truyền ảnh đó cho thư viện quét mã QR.
- Quét và giải mã: Sử dụng thư viện quét mã QR để quét ảnh từ camera và giải mã các mã QR được phát hiện.
- Xử lý kết quả: Xử lý nội dung của mã QR sau khi được giải mã. Có thể là hiển thị trên màn hình, lưu vào bộ nhớ hoặc truyền đi qua mạng.

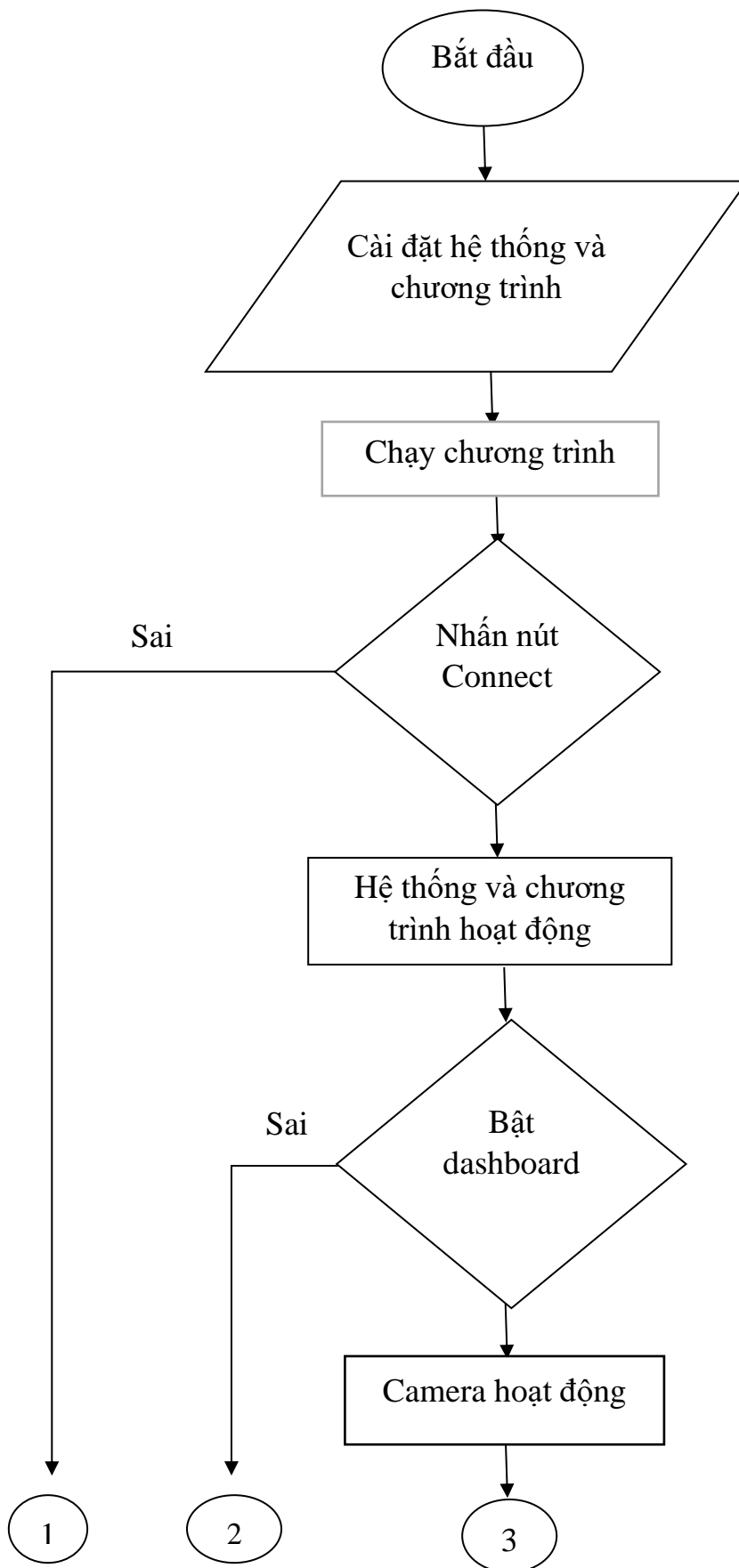
Luồng chính:

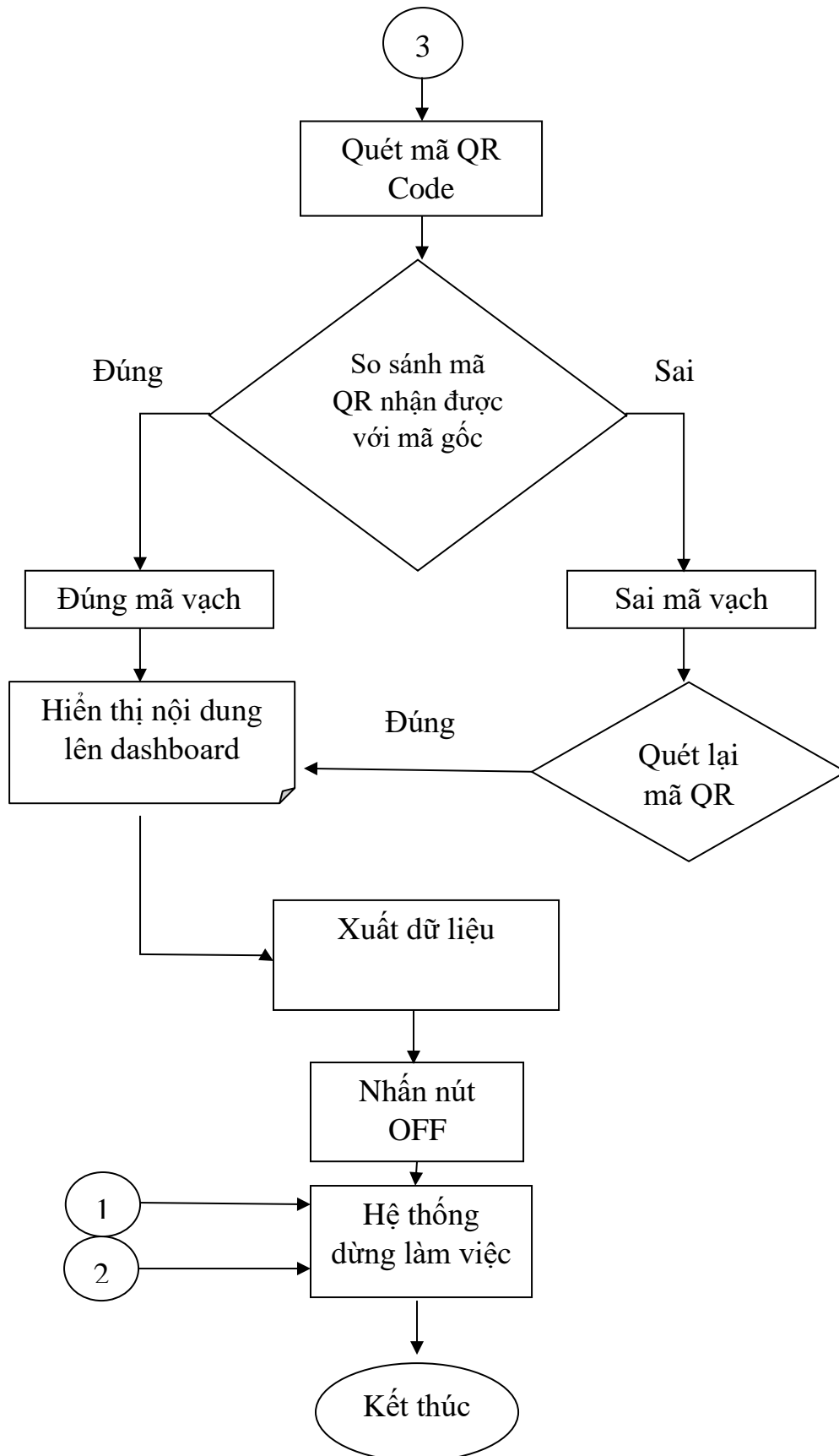
- Khởi tạo và quản lý các tác vụ: Trong hàm setup(), sẽ khởi tạo và cấu hình các thành phần cần thiết như camera và task quét mã QR.
- Chờ và xử lý kết quả: Trong hàm loop(), chương trình sẽ chờ và xử lý kết quả của các mã QR được quét.



Hình 2.1: Chương trình quét mã QR Code

## 2.2 Lưu đồ giải thuật





## **2.3 Thiết kế phần mềm quét mã QR**

### **2.3.1 Hoạt động của hệ thống.**

Sau đó ta cấp nguồn 5V cho khối camera ESP-32 thông qua module vào cổng USB của laptop. Đây cũng chính là nguồn cấp để cho khối camera wifi hoạt động ổn định.

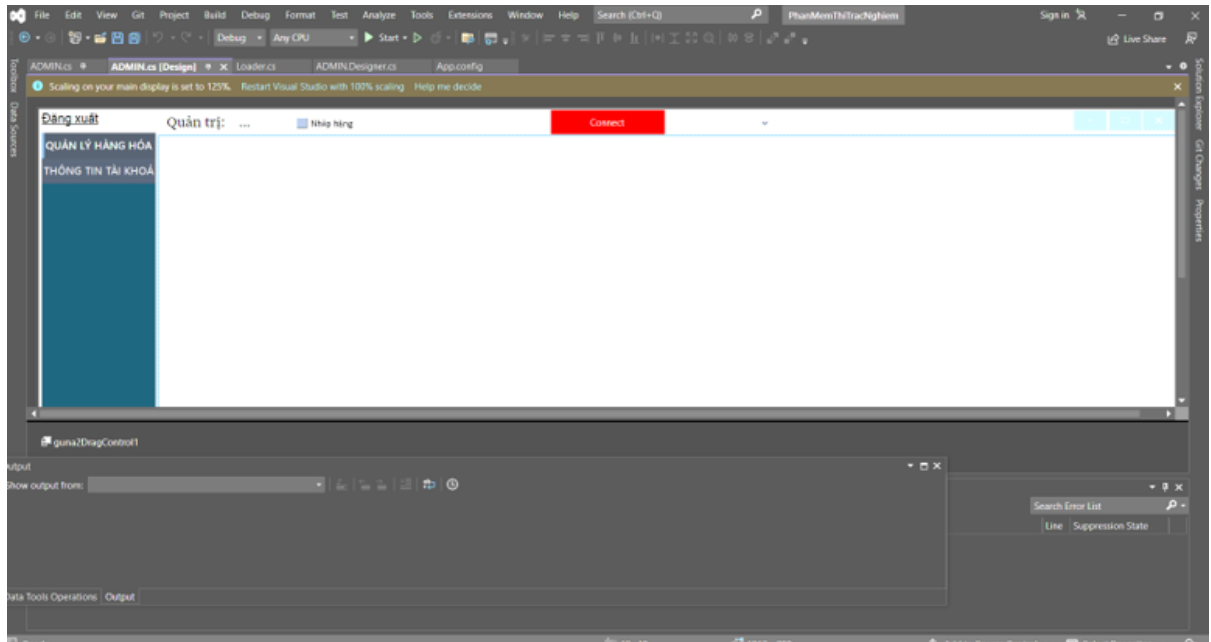
Mở phần mềm Arduino để cấp chương trình cho ESP32-Cam hoạt động. Nhấn nút RST trên ESP32-Cam để cam tiến hành chương trình quét mã QR đã được nạp vào ESP32-Cam bằng phần mềm Arduino.

Sau khi chương trình đã khởi động, ta mở giao diện chính dashboard của hệ thống. Nhấn nút “Connect” để khối camera có thể hoạt động được.

Khi sản phẩm được đưa đến gần màn hình camera, camera quét mã và gửi hình ảnh trực tiếp đến chương trình để đọc mã QR code. Mã QR code này sẽ được đối chiếu với mã QR code gốc của từng loại vật phẩm. Nếu đúng mã QR code thì sẽ hiện lên dashboard (bao gồm cả tên vật phẩm).

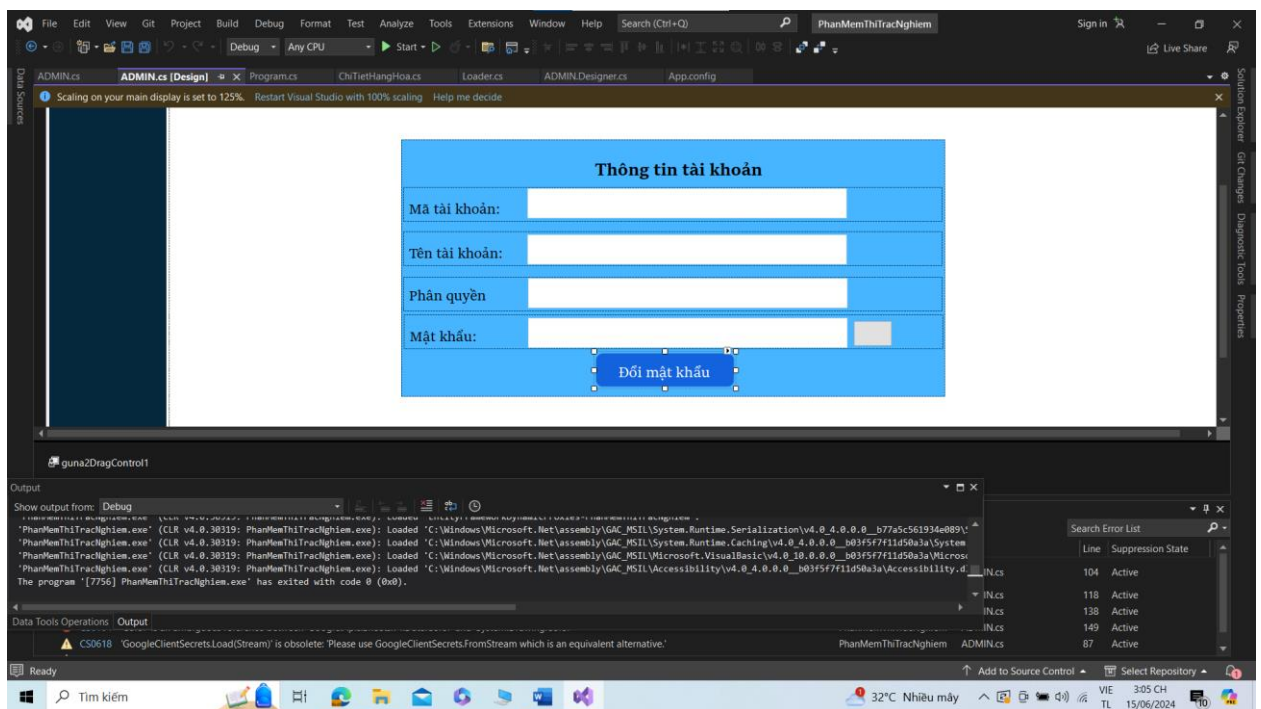
### **2.3.2 Phần mềm quét mã QR**

Thiết kế giao diện hiển thị thông tin sau khi quét và đọc thành công mã QRcode về các loại thực phẩm cần quản lý. Sau khi chạy phần mềm, các thông tin của thực phẩm khi đọc mã QR sẽ hiển thị trên gồm những thông tin như mã sản phẩm, tên, thuộc loại và số lượng của sản phẩm sau mỗi lần quét mã QR.



Hình 2.2: Thiết kế giao diện Quản lý thực phẩm

Ngoài ra, em còn thiết kế giao diện Đăng nhập để ứng dụng có thể được sử dụng bởi nhiều người dùng cùng 1 lúc



Hình 2.3: Thiết kế giao diện Đăng nhập

## 2.4 Chọn nền tảng quản lý-Google Sheet

### 2.4.1 Giới thiệu về Google Sheet

Google Sheets là một ứng dụng bảng tính trực tuyến được phát triển bởi Google, là một phần của bộ ứng dụng văn phòng Google Workspace (trước đây gọi là G Suite). Google Sheets cho phép người dùng tạo, chỉnh sửa và chia sẻ bảng tính trực tuyến trong thời gian thực với những người khác [6].

Google Sheets có sẵn dưới dạng một ứng dụng web, ứng dụng di động cho Android, iOS, Windows, BlackBerry và dưới dạng ứng dụng trên máy tính để bàn trên ChromeOS của Google. Ứng dụng này tương thích với các định dạng file Microsoft Excel. Ứng dụng cho phép người dùng tạo và chỉnh sửa các file trực tuyến trong khi cộng tác với những người dùng khác trong thời gian thực. Chỉnh sửa được theo dõi bởi người dùng với lịch sử sửa đổi trình bày các thay đổi. Vị trí của biên tập viên được tô sáng bằng màu sắc và con trỏ dành riêng cho biên tập viên và hệ thống quyền điều chỉnh những gì người dùng có thể làm. Các bản cập nhật đã giới thiệu các tính năng sử dụng máy học, bao gồm "Khám phá", đưa ra câu trả lời dựa trên các câu hỏi ngôn ngữ tự nhiên trong bảng tính.

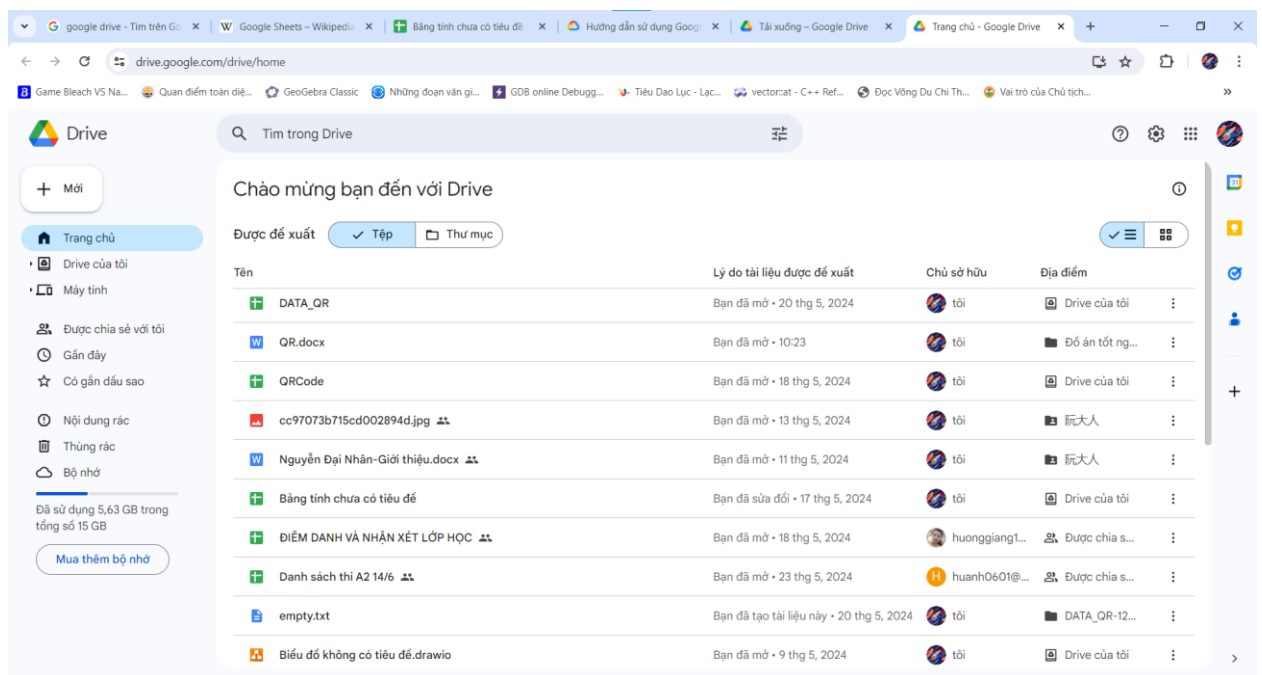
Tính năng chính của Google Sheets:

- Tạo và chỉnh sửa bảng tính: Google Sheets cung cấp đầy đủ các công cụ để tạo và chỉnh sửa bảng tính, bao gồm nhập dữ liệu, định dạng, hàm, biểu đồ, v.v.
- Cộng tác: Mọi người có thể cùng nhau làm việc trên cùng một bảng tính trong thời gian thực. Mọi thay đổi được theo dõi và lưu tự động.
- Lưu trữ đám mây: Bảng tính được lưu trữ an toàn trên Google Drive, vì vậy có thể truy cập từ mọi nơi có kết nối internet.
- Miễn phí: Google Sheets hoàn toàn miễn phí để sử dụng cho cá nhân và tổ chức phi lợi nhuận.
- Khả năng tương thích: Google Sheets tương thích với các định dạng file Microsoft Excel, vì vậy có thể dễ dàng mở và chỉnh sửa file Excel trong Google Sheets.

- Tính năng thông minh: Google Sheets sử dụng máy học để đề xuất các thông tin chi tiết, tự động hoàn thành công thức và tạo biểu đồ dựa trên dữ liệu của.
- Ứng dụng di động: Google Sheets có sẵn dưới dạng ứng dụng di động cho Android và iOS, vì vậy làm việc với bảng tính của mình khi đang di chuyển.

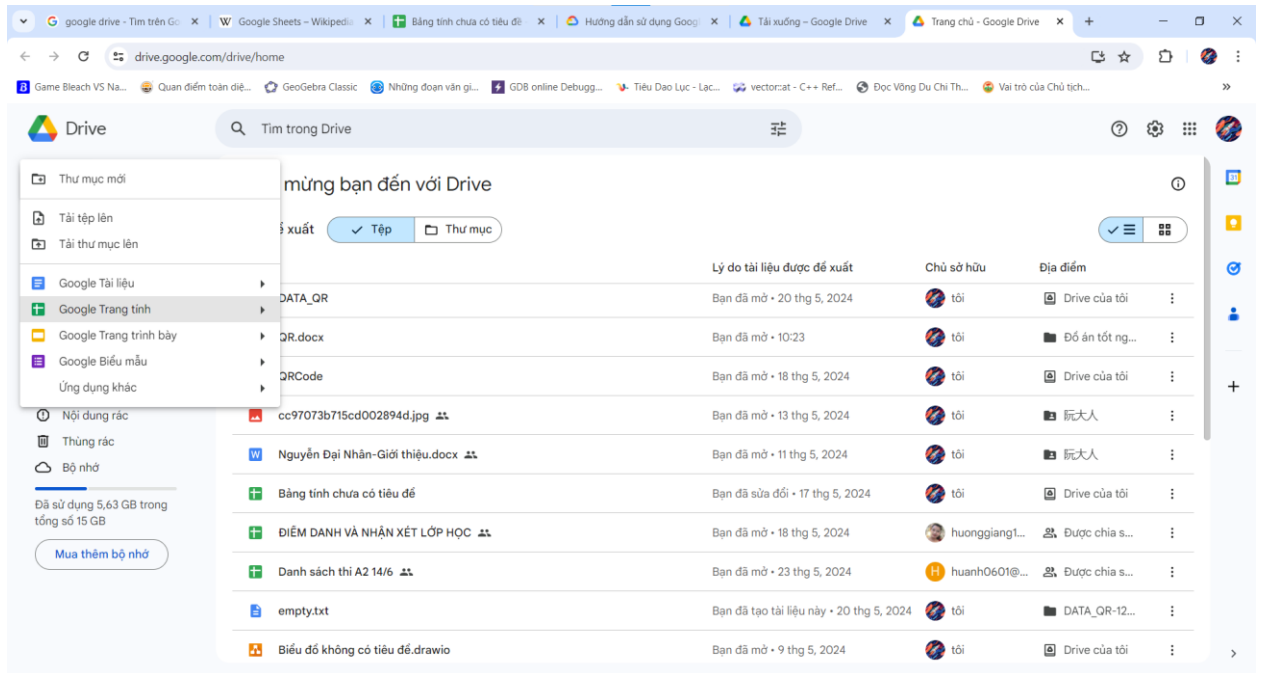
## 2.4.2 Hướng dẫn sử dụng Google Sheet

Để tạo một trang google sheet mới thì việc ta cần làm đầu tiên là đăng nhập vào google drive.



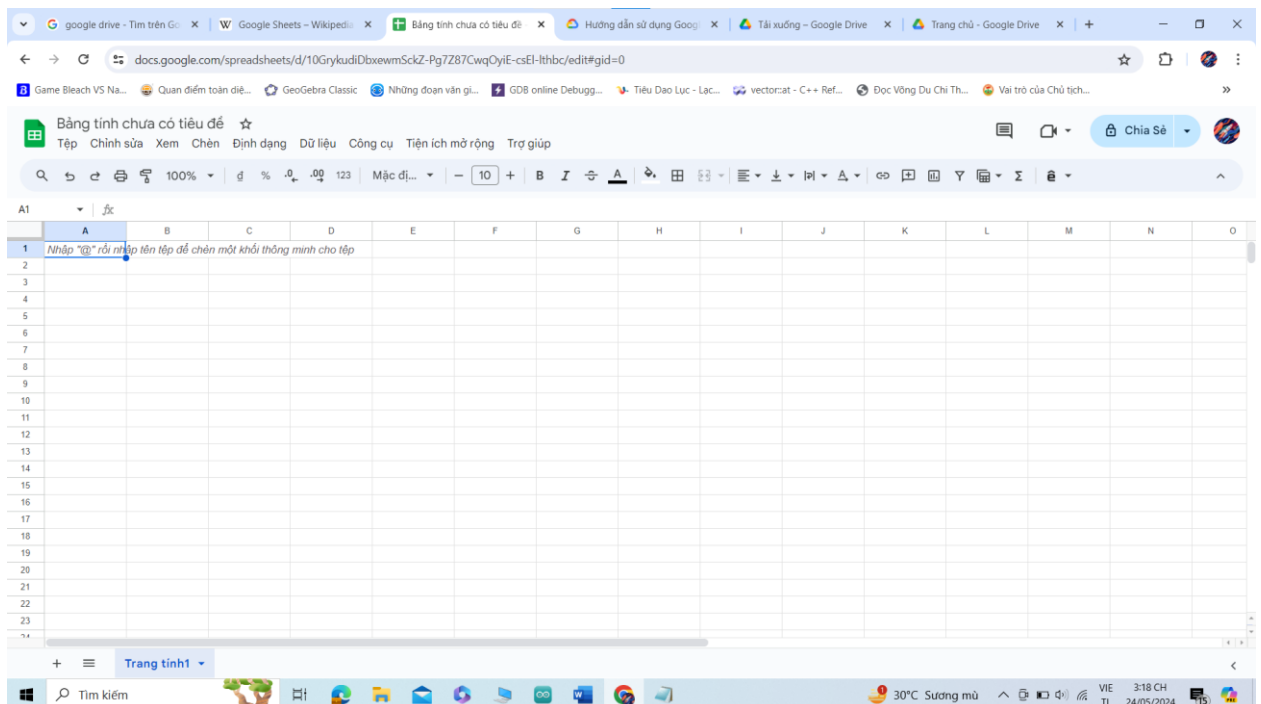
Hình 2.4: Trang web Google Drive

Sau khi vào được Google Drive, ta chọn “Mới” để ở góc trên bên trái của trang web và chọn Google trang tính để tạo một trang google sheet mới.



Hình 2.5: Tạo trang Google sheet mới

Ta tạo được một trang Google sheet mới.



Hình 2.6: Trang Google sheet mới

## 2.5 Tiểu kết

Trong chương 2, em đã nêu rõ việc thiết kế chương trình quét mã QR trên Arduino và phần mềm hiển thị thông tin được lưu trữ trong mã QR thành các thông



tin của sản phẩm cần quản lý. Em cần vẽ lưu đồ thuật toán của chương trình, cách hoạt động của chương trình quét mã QR.

Đến chương tiếp theo, em sẽ đưa ra chương trình hoàn thiện dựa trên các phần thiết kế trong chương 2, hoàn thiện code và chạy chương trình trong chương 3.

## **Chương 3: DEMO SẢN PHẨM SAU KHI HOÀN THÀNH**

Trong chương 3, em demo chạy chương trình quét mã QR code và giải thích các phần trong chương trình. Do vấn đề kiến thức và kỹ thuật, đồ án của em chỉ vừa hoàn thiện, có thể còn thiếu một số chức năng hay chưa đủ thực dụng.

### **3.1 Cách chương trình thực thi**

#### **3.1.1 Khai báo chương trình**

Đầu tiên là khai báo các thư viện ESP32, các chân kết nối, các giá trị đầu vào của biến.

Ví dụ:

```
#include "esp_camera.h"
```

```
#include "soc/soc.h"
```

```
#include "soc/rtc_cntl_reg.h"
```

```
#include "quirc.h"
```

Các thư viện trên bao gồm các tệp và thư viện được sử dụng trong dự án liên quan đến việc sử dụng máy ảnh với mô-đun ESP32. Đây là một phần của mã được sử dụng để giao tiếp với máy ảnh và xử lý hình ảnh trên nền tảng ESP32, một trong những mô-đun phổ biến cho IoT và các ứng dụng nhúng.

#### **3.1.2 Hàm Void setup**

Hàm void setup() là một hàm được gọi một lần duy nhất khi bắt đầu chạy chương trình. Hàm này được sử dụng để khởi tạo các thiết lập ban đầu cho Arduino trước khi nó bắt đầu thực hiện các hàm và chương trình chính. Hàm setup() không trả về giá trị và có kiểu dữ liệu là void.

Trong code của em, hàm setup() được sử dụng để cấu hình môi trường cho việc sử dụng máy ảnh và khởi động chương trình. Để hoàn thiện code hàm setup() em đã:

- `WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0)` code này em ghi giá trị 0 vào thanh ghi `RTC_CNTL_BROWN_OUT_REG` để tắt chức năng tự động khởi động lại của ESP32 khi phát hiện điện áp thấp.
- `Serial.begin(115200)` và `Serial.setDebugOutput(true)` ở đây em khởi động giao tiếp serial với tốc độ 115200 baud và kích hoạt đầu ra debug thông qua serial.
- Thiết lập cấu hình của camera thông qua biến `config`. Cấu hình này bao gồm các thông số như các GPIO được sử dụng cho việc truyền dữ liệu hình ảnh, tần số của clock, định dạng pixel, kích thước khung hình, chất lượng JPEG, và số framebuffer.
- `Esp_camera_init(&config)` lệnh này em khởi động camera với cấu hình được chỉ định trước đó. Nếu quá trình này thất bại, thiết bị sẽ khởi động lại.
- `Sensor_t *s = esp_camera_sensor_get()` và `s -> set_framesize (s, FRAMESIZE_QVGA)` em cho nhận thông tin về cảm biến camera và đặt kích thước khung hình là QVGA (320x240 pixels).
- `ledcAttachPin(4, 4)` và `ledcSetup(4, 5000, 8)` thì ở đây em thiết lập PWM cho đèn LED.
- `xTaskCreatePinnedToCore()` tạo một task mới để thực hiện việc đọc mã QR.
- Cuối cùng, hàm `setup()` in ra thông báo “Successfully” trên serial monitor và tạo một task mới được gọi là `QRCodeReader` để thực hiện việc đọc mã QR.

### 3.1.3 Hàm Void loop

Còn đối hàm void loop thì đây là hàm chạy liên tục sau khi hàm `setup()` đã hoàn thành. Hàm này chứa các lệnh và khối lệnh mà Arduino sẽ thực hiện lặp đi lặp lại trong suốt quá trình chạy chương trình.

Trong chương trình của em, hàm `loop()` không có nội dung cụ thể và chỉ có một độ trễ nhỏ là `delay(1)` để giữ cho vòng lặp chạy một cách nhẹ nhàng.

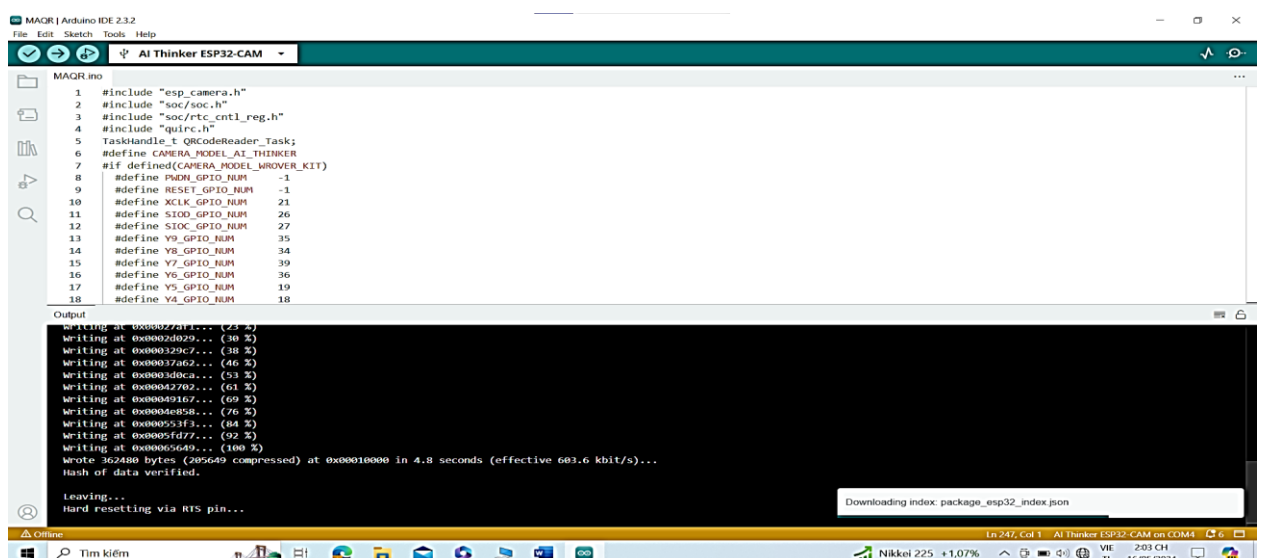
### 3.1.4 Hàm QRCodeReader

Hàm QRCodeReader() là một task độc lập chạy song song với hàm loop(). Nhiệm vụ của task này là đọc mã QR từ dữ liệu hình ảnh được chụp bởi camera. Trong hàm này sẽ chạy liên tục trong khi chương trình đang chạy, luôn kiểm tra và giải mã các mã QR từ hình ảnh mới được chụp. Để hoàn thiện hàm này em đã:

- Tạo một đối tượng quirc để xử lý mã QR.
- Chụp một khung hình từ camera bằng cách gọi esp\_camera\_fb\_get().
- Đưa dữ liệu hình ảnh vào quirc để xử lý mã QR.
- Kiểm tra xem có mã QR nào được tìm thấy không và thực hiện giải mã nếu có.
- Nếu giải mã thành công, dữ liệu của mã QR sẽ được in ra serial monitor thông qua hàm dumpData(&data).
- Cuối cùng, đối tượng khung hình và các tài nguyên được giải phóng.

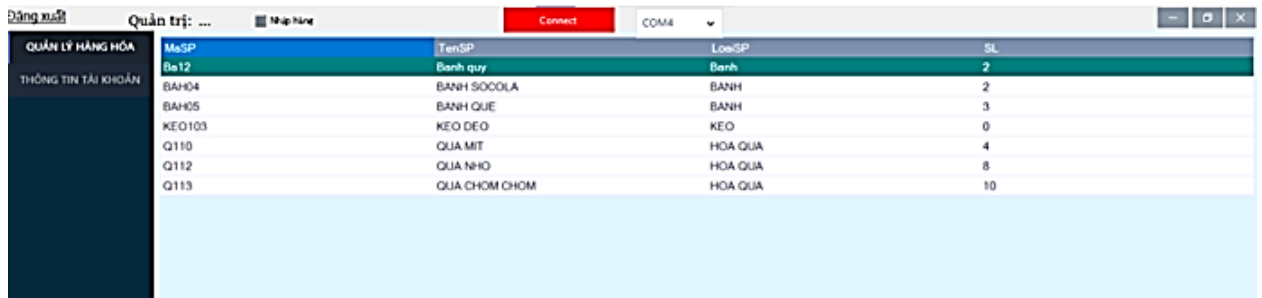
### 3.2 Demo chương trình

Khởi chạy chương trình quét mã QR để điều khiển ESP32-CAM bắt đầu hoạt động quét và đọc thông tin mã QR.



Hình 3.1 Khởi chạy chương trình quét mã QR trong Arduino

Sau đó, bắt đầu chạy phần mềm hiển thị mã QR code trong Visual Studio, nó sẽ hiển thị thông tin sản phẩm quét được bằng ESP32-CAM.

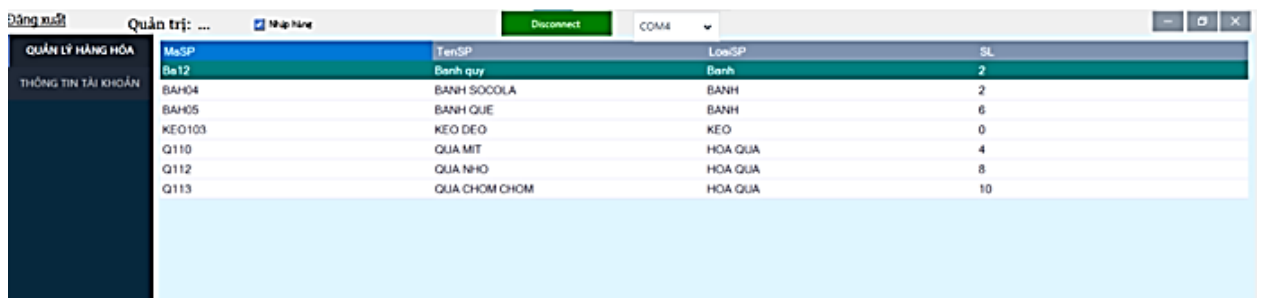


MaSP	TenSP	LoaiSP	SL
Ba12	Banh quy	Banh	2
BAH04	BANH SOCOLA	BANH	2
BAH05	BANH QUE	BANH	3
KEO103	KEO DEO	KEO	0
Q110	QUA MIT	HOA QUA	4
Q112	QUA NHO	HOA QUA	8
Q113	QUA CHOM CHOM	HOA QUA	10

Hình 3.2: Phần mềm hiển thị thông tin mã QR

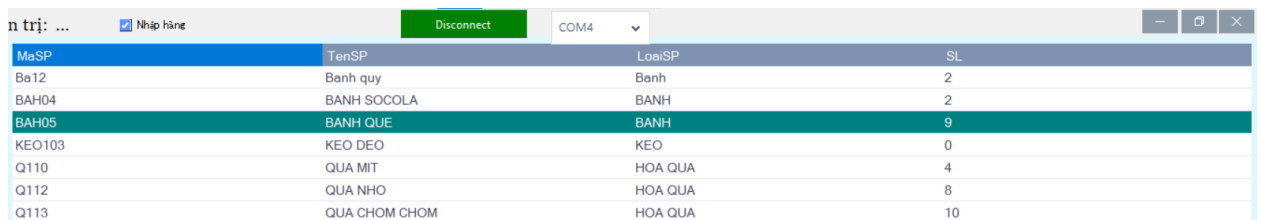
Nhấn nút “Connect” để kết nối với ESP32-CAM, chương trình bắt đầu hoạt động quét và đọc thông tin mã QR và thay đổi số lượng vật phẩm. Số lượng tăng thêm khi ta chọn nhập hàng và giảm đi khi tắt chọn nhập hàng.

Khi nhập hàng số lượng sản phẩm sẽ tăng lên( cộng 1).



MaSP	TenSP	LoaiSP	SL
Ba12	Banh quy	Banh	2
BAH04	BANH SOCOLA	BANH	2
BAH05	BANH QUE	BANH	6
KEO103	KEO DEO	KEO	0
Q110	QUA MIT	HOA QUA	4
Q112	QUA NHO	HOA QUA	8
Q113	QUA CHOM CHOM	HOA QUA	10

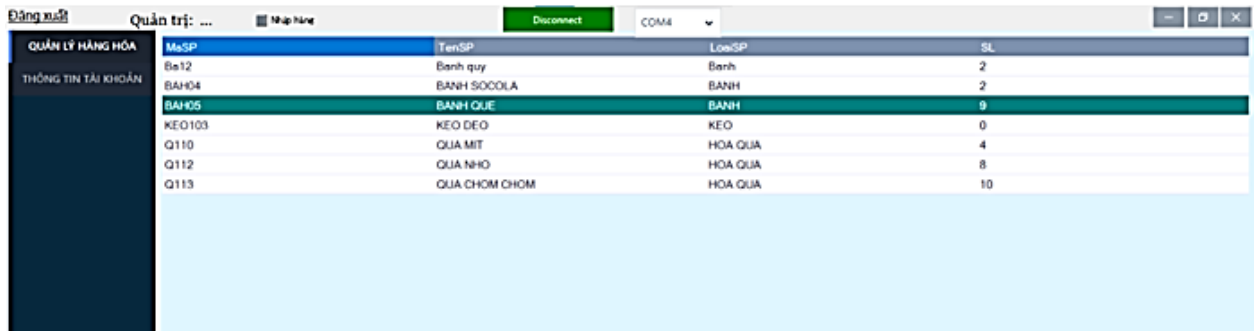
Hình 3.3: Nhập hàng



MaSP	TenSP	LoaiSP	SL
Ba12	Banh quy	Banh	2
BAH04	BANH SOCOLA	BANH	2
BAH05	BANH QUE	BANH	9
KEO103	KEO DEO	KEO	0
Q110	QUA MIT	HOA QUA	4
Q112	QUA NHO	HOA QUA	8
Q113	QUA CHOM CHOM	HOA QUA	10

Hình 3.4: Sản phẩm tăng thêm

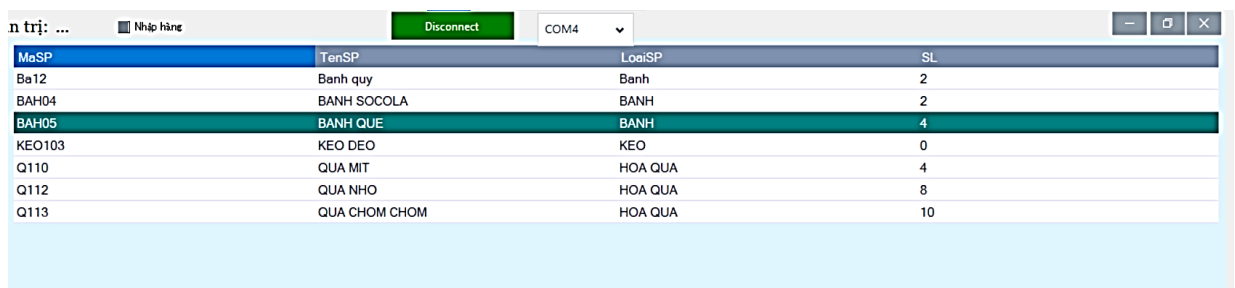
Khi tắt nhập hàng số lượng sản phẩm sẽ giảm đi( trừ 1)



MaSP	TenSP	LoaiSP	SL
Ba12	Banh quy	Banh	2
BAH04	BANH SOCOLA	BANH	2
<b>BAH05</b>	<b>BANH QUE</b>	<b>BANH</b>	<b>9</b>
KEO103	KEO DEO	KEO	0
Q110	QUA MIT	HOA QUA	4
Q112	QUA NHO	HOA QUA	8
Q113	QUA CHOM CHOM	HOA QUA	10

Hình 3.5: Tắt nhập hàng

Số lượng sản phẩm giảm đi.

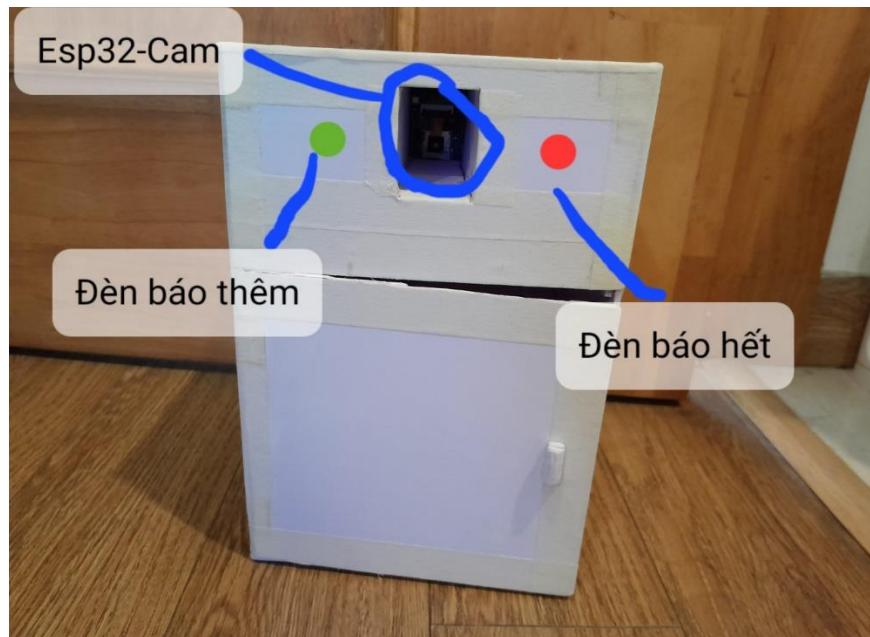


MaSP	TenSP	LoaiSP	SL
Ba12	Banh quy	Banh	2
BAH04	BANH SOCOLA	BANH	2
<b>BAH05</b>	<b>BANH QUE</b>	<b>BANH</b>	<b>4</b>
KEO103	KEO DEO	KEO	0
Q110	QUA MIT	HOA QUA	4
Q112	QUA NHO	HOA QUA	8
Q113	QUA CHOM CHOM	HOA QUA	10

Hình 3.6: Sản phẩm giảm đi

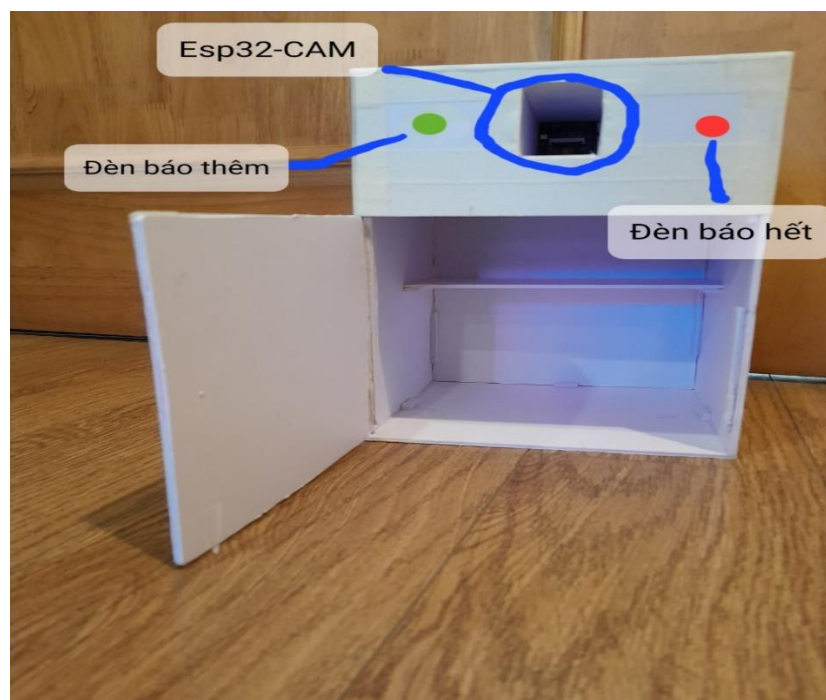
### 3.3 Mô hình thực tế

Mô hình tủ lạnh sử dụng tấm nhựa Fomex để lắp ghép. Mô hình có kích thước bao gồm chiều cao khoảng 23 cm, chiều dài và chiều rộng khoảng 15 cm. Bên trong tủ lạnh chia làm 2 ngăn trên dưới. Phần trên, em có lắp đặt ESP32-CAM để quét mã QR thực phẩm cho vào tủ. Đèn xanh sáng lên để thông báo một loại thực phẩm nào đó sắp hết, ta cần mua thêm. Đèn đỏ sáng lên để thông báo thực phẩm để quá lâu trong một thời gian dài, ta có thể lấy ra sử dụng hoặc đem bỏ nếu đã hỏng.



Hình 3.7: Mô hình tủ lạnh có sử dụng ESP32-CAM

Mô hình tủ lạnh ứng dụng chương trình quét mã QR. Trong mô hình, bản thân em đã sử dụng ESP32-CAM để thực hiện việc quét mã QR. Sau khi chạy chương trình ESP32-CAM được lắp bên trong mô hình sẽ chạy code quét mã QR thực phẩm. Sau đó, thông tin thực phẩm sẽ được chuyển lên phần mềm trên máy tính để hiển thị thông tin thực phẩm vừa quét được.



Hình 3.8: Mô hình tủ lạnh bên trong

### 3.4 Tiểu kết

Trong chương 3, em đã đưa ra các kết quả mà em đã đạt được trong đồ án của mình như chương trình quét mã QR Code chạy trên phần mềm Arduino IDE, phần mềm hiển thị các thông tin được lưu trữ trong QR Code. Em còn đưa ra mô hình tủ lạnh ứng dụng việc quét QR Code sử dụng ESP32-CAM để quét mã.



## KẾT LUẬN VÀ KIẾN NGHỊ

### 1. Kết luận

Mô hình tủ lạnh quản lý thực phẩm trong gia đình, có thể điều khiển được ở trên máy tính cá nhân. ESP32-CAM có thể đọc được thông tin được lưu trữ trong các mã QR. Chạy thành công phần mềm hiển thị và điều khiển xuất, nhập thông tin thực phẩm. Ta có thể biết và quản lý được số lượng và thông tin thực phẩm lấy ra và cho vào trong tủ lạnh của gia đình.

Ngoài những nội dung em đã đạt được thì vẫn còn những hạn chế thiếu sót từ con người là bản thân em. Đồ án của em vẫn chưa hoàn thành như chưa hoàn thiện hệ thống cảnh báo bằng đèn led. Chương trình vẫn chỉ chạy trên máy tính chưa hoàn thiện trên di động để có thể thông báo cho bản thân người dùng qua Google Sheet.

### 2. Kiến nghị

Trong tương lai, em mong muốn nhận được sự hỗ trợ, giới thiệu của thầy TS. Nguyễn Tài Tuyên, các thầy cô lãnh đạo khoa và lãnh đạo nhà trường tạo điều kiện giúp đỡ để em có hoàn thiện hơn và đưa kiến thức học được tại trường Đại học Đại Nam vào ứng dụng rộng rãi như máy quét mã QR trong các lĩnh vực khác nhau.

## TÀI LIỆU THAM KHẢO

### **Danh mục các tài liệu Tiếng Việt:**

- [1] TS. Hoàng Xuân Hiến, Giáo trình môn học phát triển ứng dụng IoT, Trường Đại học Đại Nam, 2021.
- [2] TS. Nguyễn Tài Tuyên, Bài giảng môn học phát triển ứng dụng IoT, Hà Nội, năm 2022.

### **Danh mục các Website tham khảo:**

- [1] <https://blynk.io/blynk-iot-low-code-software-platform>
- [2] <https://dlcorp.com.vn/iot-la-gi-ung-dung-cua-iot-trong-thuc-tien/>
- [3] <https://github.com>
- [4] <https://marketplace.visualstudio.com/>
- [5] <https://www.arduino.cc/>
- [6] <https://www.instructables.com/Google-Sheets-for-IOT/>