

TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO

SINH HOẠT HỌC THUẬT

**MÔ PHÒNG TRUYỀN DỮ LIỆU TỪ MẠCH IOT ẢO
TRONG TINKERCAD LÊN MÁY CHỦ WEB THINGSPEAK
SỬ DỤNG ESP8266**

Người báo cáo: GVC.TS. Nguyễn Duy Huy

Đơn vị: Bộ môn Khoa học máy tính

HÀ NỘI – 6/2024

MỤC LỤC

MỤC LỤC.....	i
DANH MỤC HÌNH VẼ.....	iii
DANH MỤC BẢNG BIỂU.....	iv
DANH MỤC CHỮ VIẾT TẮT.....	v
MỞ ĐẦU.....	1
Tính cấp thiết:.....	1
Mục tiêu:.....	1
Phạm vi:.....	1
Nội dung:.....	1
Phương pháp:.....	1
CHƯƠNG I: TỔNG QUAN VỀ INTERNET OF THINGS.....	2
I.1. Khái niệm.....	2
I.2. Đặc tính.....	2
I.3. Yêu cầu của một hệ thống IoT.....	2
I.3. Kiến trúc.....	3
I.4. Ứng dụng.....	5
I.4.1. Thành phố thông minh – Smart City.....	5
I.4.2. Môi trường thông minh – Smart Environment.....	6
I.4.3. Nước thông minh – Smart Water.....	6
I.4.4. Đo lường thông minh – Smart Metering.....	6
I.4.5. Bảo mật và tình trạng khẩn cấp – Security and Emergencies.....	6
I.4.6. Hệ thống bán lẻ - Retail.....	6
I.4.7. Điều khiển trong công nghiệp – Industrial Control.....	7
I.4.8. Trang trại gia súc thông minh – Smart Animal Farming.....	7
I.4.9. Tự động nhà và domotic.....	7
I.4.10. Chăm sóc sức khỏe – eHealth.....	7
CHƯƠNG 2: TỔNG QUAN VỀ MÔI TRƯỜNG PHÁT TRIỂN.....	9

II.1. Phần mềm tinkercad	9
II.2. Nền tảng thingspeak	11
CHƯƠNG III: MÔ PHỎNG HỆ THỐNG INTERNET OF THINGS VỚI TINKERCAD VÀ THINGSPEAK	16
III.1. Mô hình truyền dữ liệu	16
III.2. Thiết kế mạch trong tinkercad.....	17
III.3. Thiết kế ứng dụng web trên nền tảng thingspeak.....	18
III.4. Lập trình nhúng trong tinkercad.....	20
III.5. Mô phỏng mạch trong tinkercad và hiển thị kết quả trên ứng dụng web thingspeak	22
III.6. Đánh giá	23
KẾT LUẬN	25
TÀI LIỆU THAM KHẢO.....	26

DANH MỤC HÌNH VẼ

Hình I.1 - Các khối chức năng của hệ thống IoT.....	4
Hình I.2 - Các giai đoạn của kiến trúc IoT.....	4
Hình II.1 - Giao diện chính phần mềm tinkercad (tinkercad.com).....	9
Hình II.2 – Giao diện chính website tinkercad.com.....	9
Hình II.3 – Chọn phương pháp đăng nhập tinkercad.....	10
Hình II.4 – Giao diện lựa chọn mạch có sẵn/mới trong tinkercad.....	10
Hình II.5 – Giao diện thiết kế mạch IoT ảo trong tinkercad.....	11
Hình II.6 – Giao diện chính website thingspeak.com.....	12
Hình II.7 – Giao diện đăng nhập trên thingspeak.com.....	12
Hình II.8 – Giao diện chính sau khi đăng nhập trên thingspeak.....	13
Hình II.9 – Giao diện thiết kế/cài đặt kênh mới trên thingspeak.....	13
Hình II.10 – Giao diện IoT dashboard sau khi thiết kế xong.....	14
Hình II.11 – API Keys.....	15
Hình III.1 – Mô hình kết nối tinkercad với thingspeak.....	16
Hình III.2 – Mạch thiết kế đề xuất trong tinkercad.....	17
Hình III.3 – Thiết lập kênh trên thingspeak.....	19
Hình III.4 – Tạo Write API Key.....	20
Hình III.5 – Source code lập trình cho mạch.....	21
Hình III.6 – Mô phỏng mạch trong tinkercad.....	23
Hình III.7 – Kết quả mô phỏng mạch trên thingspeak.....	23

DANH MỤC BẢNG BIỂU

Bảng III.1 - So sánh hệ thống IoT với thiết bị ảo và thật.....	24
--	----

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa	Ghi chú
API	Application Programming Interface	Giao diện lập trình ứng dụng
BLE	Bluetooth Low Energy	Bluetooth năng lượng thấp
COM	Serial port	Cổng COM hay Cổng nối tiếp
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
ID	Identification	Nhận dạng
IDE	Integrated Development Environment	Môi trường tích hợp phát triển
IoT	Internet of Things	Internet vạn vật
LED	Light-emitting diode	Đi-ốt phát sáng
NFC	Near-Field Communications	Công nghệ kết nối không dây tầm ngắn (4cm)
PCB	Printed circuit board	Bộ mạch in
RFID	Radio Frequency Identification	Nhận dạng đối tượng bằng sóng vô tuyến
TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
UDP	User Datagram Protocol	Giao thức thư tín người dùng

MỞ ĐẦU

Tính cấp thiết:

Học phần **Phát triển ứng dụng IoT (7080517)** là học phần thuộc nhóm Cơ sở ngành do Bộ môn Khoa học máy tính quản lý. Để minh hoạt trực quan và giúp sinh viên nắm vững kiến thức đã học cần có các thiết bị phần cứng để thực hành mô phỏng. Tuy nhiên việc mua sắm các thiết bị phần cứng còn nhiều hạn chế do tốn kém kinh phí, thiết bị có thể nhanh hỏng (không sử dụng lâu dài được). Do vậy việc nghiên cứu ứng dụng các phần mềm hỗ trợ để thiết kế mạch mô phỏng (thiết bị ảo), lập trình nhúng, triển khai web server, webapp hoặc mobile app là hết sức cần thiết để giúp sinh viên củng cố kiến thức đã học. Việc mô phỏng với thiết bị ảo sẽ giúp tiết kiệm chi phí mua thiết bị, sinh viên có thể tự thực hành mọi lúc mọi nơi, học nhóm, tự học v.v.

Mục tiêu:

Mô phỏng thành công hệ thống IoT cơ bản trên máy tính với phần mềm Tinkercad (tinkercad.com) và nền tảng Thingspeak (thingspeak.com) phục vụ cho thực hành học phần **Phát triển ứng dụng IoT**.

Phạm vi:

Mô phỏng được hệ thống IoT cơ bản trong phần mềm tinkercad kết hợp với nền tảng thingspeak. Các thiết bị được triển khai trong phần mềm tinkercad, lập trình nhúng trong tinkercad, kết quả mô phỏng được gửi lên máy chủ thingspeak và hiển thị trên ứng dụng web của thingspeak.

Nội dung:

Tập trung nghiên cứu một số nội dung cơ bản sau:

- Thiết kế mạch mô phỏng trong phần mềm tinkercad
- Lập trình nhúng trong tinkercad
- Tạo tài khoản trên thingspeak
- Thiết kế và cấu hình kênh (channel) trên thingspeak
- Mô phỏng ứng dụng và đánh giá

Phương pháp:

- Nghiên cứu lý thuyết: Về thiết kế mạch trong tinkercad, lập trình nhúng trong tinkercad, thiết kế webapp trên nền tảng thingspeak.

- Thực nghiệm: Thiết kế, xây dựng, mô phỏng hệ thống IoT thí điểm trên các phần mềm/nền tảng đề xuất.

CHƯƠNG I: TỔNG QUAN VỀ INTERNET OF THINGS

I.1. Khái niệm

Theo Wikipedia: “Mạng lưới vạn vật kết nối Internet (*Internet of Things, IoT*) là mạng lưới thiết bị kết nối Internet là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.”

Hay hiểu một cách đơn giản: IoT là tất cả các thiết bị có thể kết nối với nhau. Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại. Các thiết bị có thể là điện thoại thông minh, máy pha cafe, máy giặt, tai nghe, bóng đèn, và nhiều thiết bị khác [1, 2].

I.2. Đặc tính

Theo [2], đặc tính của IoT bao gồm:

- ❖ **Tính kết nối liên thông (interconnectivity):** với IoT, bất cứ điều gì cũng có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.
- ❖ **Những dịch vụ liên quan đến “Things”:** hệ thống IoT có khả năng cung cấp các dịch vụ liên quan đến “Things”, chẳng hạn như bảo vệ sự riêng tư và nhất quán giữa Physical Thing và Virtual Thing. Để cung cấp được dịch vụ này, cả công nghệ phần cứng và công nghệ thông tin(phần mềm) sẽ phải thay đổi.
- ❖ **Tính không đồng nhất:** Các thiết bị trong IoT là không đồng nhất vì nó có phần cứng khác nhau, và network khác nhau. Các thiết bị giữa các network có thể tương tác với nhau nhờ vào sự liên kết của các network.
- ❖ **Thay đổi linh hoạt:** Status của các thiết bị tự động thay đổi, ví dụ, ngủ và thức dậy, kết nối hoặc bị ngắt, vị trí thiết bị đã thay đổi, và tốc độ đã thay đổi... Hơn nữa, số lượng thiết bị có thể tự động thay đổi.
- ❖ **Quy mô lớn:** Sẽ có một số lượng rất lớn các thiết bị được quản lý và giao tiếp với nhau. Số lượng này lớn hơn nhiều so với số lượng máy tính kết nối Internet hiện nay. Số lượng các thông tin được truyền bởi thiết bị sẽ lớn hơn nhiều so với được truyền bởi con người.

I.3. Yêu cầu của một hệ thống IoT

Một hệ thống IoT phải thoả mãn các yêu cầu sau [2]:

- ✓ **Kết nối dựa trên sự nhận diện:** Nghĩa là các “Things” phải có ID riêng biệt. Hệ thống IoT cần hỗ trợ các kết nối giữa các “Things”, và kết nối được thiết lập dựa trên định danh (ID) của Things.

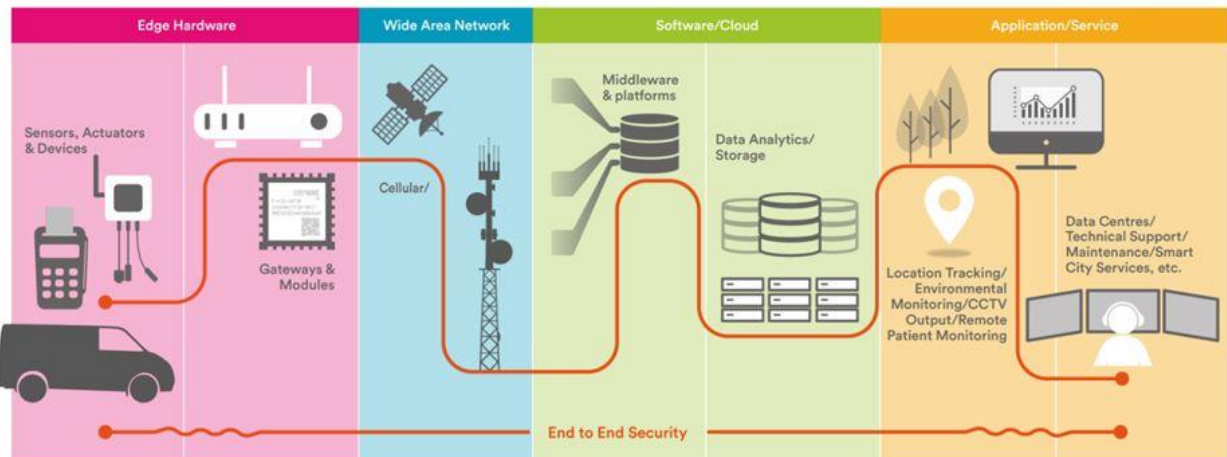
- ✓ **Khả năng cộng tác:** hệ thống IoT khả năng tương tác qua lại giữa các network và Things.
- ✓ **Khả năng tự quản của network:** Bao gồm tự quản lý, tự cấu hình, tự chữa bệnh, tự tối ưu hóa và tự có cơ chế bảo vệ. Điều này cần thiết để network có thể thích ứng với các domains ứng dụng khác nhau, môi trường truyền thông khác nhau, và nhiều loại thiết bị khác nhau.
- ✓ **Dịch vụ thỏa thuận:** dịch vụ này để có thể được cung cấp bằng cách thu thập, giao tiếp và xử lý tự động các dữ liệu giữa các “Things” dựa trên các quy tắc(rules) được thiết lập bởi người vận hành hoặc tùy chỉnh bởi các người dùng.
- ✓ **Các khả năng dựa vào vị trí(location-based capabilities):** Thông tin liên lạc và các dịch vụ liên quan đến một cái gì đó sẽ phụ thuộc vào thông tin vị trí của Things và người sử dụng. Hệ thống IoT có thể biết và theo dõi vị trí một cách tự động. Các dịch vụ dựa trên vị trí có thể bị hạn chế bởi luật pháp hay quy định, và phải tuân thủ các yêu cầu an ninh.
- ✓ **Bảo mật:** Trong IoT, nhiều “Things” được kết nối với nhau. Chình điều này làm tăng mối nguy trong bảo mật, chẳng hạn như bí mật thông tin bị tiết lộ, xác thực sai, hay dữ liệu bị thay đổi hay làm giả.
- ✓ **Bảo vệ tính riêng tư:** tất cả các “Things” đều có chủ sở hữu và người sử dụng của nó. Dữ liệu thu thập được từ các “Things” có thể chứa thông tin cá nhân liên quan chủ sở hữu hoặc người sử dụng nó. Các hệ thống IoT cần bảo vệ sự riêng tư trong quá trình truyền dữ liệu, tập hợp, lưu trữ, khai thác và xử lý. Bảo vệ sự riêng tư không nên thiết lập một rào cản đối với xác thực nguồn dữ liệu.
- ✓ **Plug and play:** các Things phải được plug-and-play một cách dễ dàng và tiện dụng.
- ✓ **Khả năng quản lý:** hệ thống IoT cần phải hỗ trợ tính năng quản lý các “Things” để đảm bảo network hoạt động bình thường. Ứng dụng IoT thường làm việc tự động mà không cần sự tham gia người, nhưng toàn bộ quá trình hoạt động của họ nên được quản lý bởi các bên liên quan.

I.3. Kiến trúc

Sự phát triển của IoT không phải là sự phát triển của một công nghệ riêng lẻ nào mà là sự tổng hợp, thúc đẩy cải tiến không ngừng của hàng loạt các lĩnh vực công nghệ nền tảng khác nhau, trong đó gồm 4 lĩnh vực công nghệ chính luôn gắn liền với sự phát triển xu thế IoT bao gồm: phần cứng, truyền thông kết nối, phần mềm và ứng dụng.

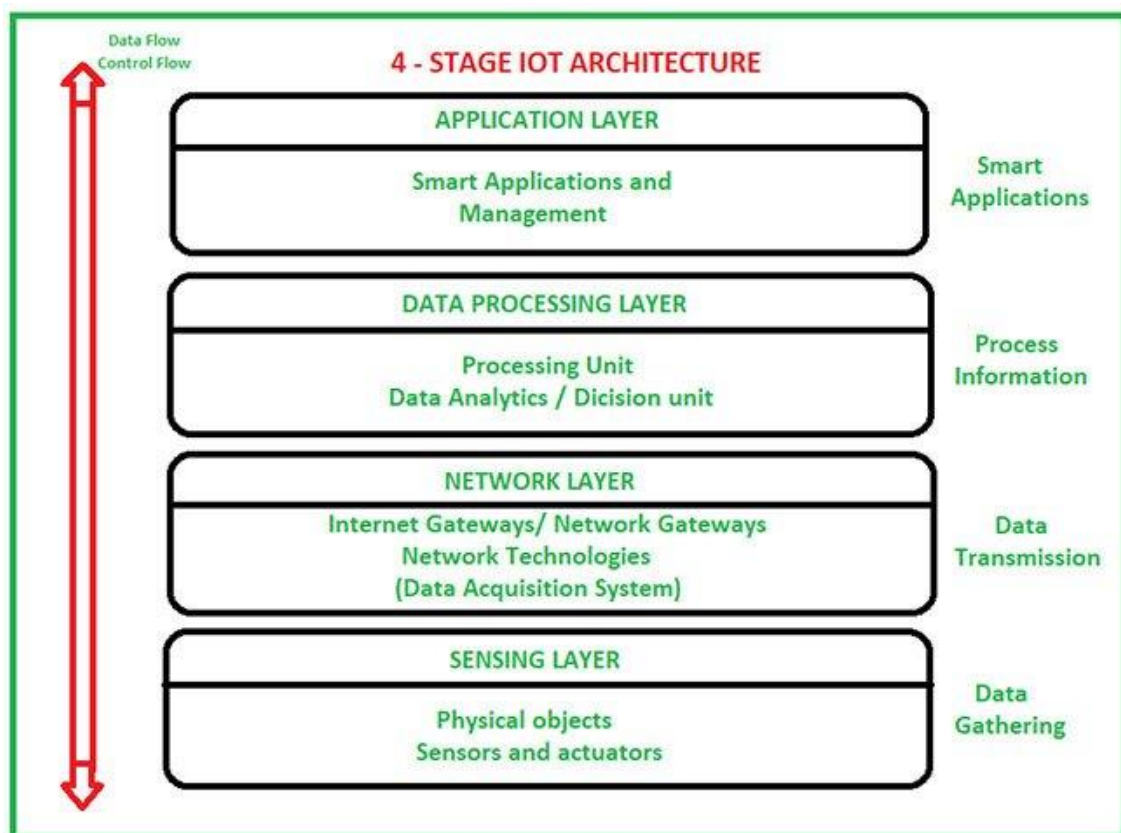
Để triển khai thực tế, đòi hỏi định nghĩa một kiến trúc tổng quát IoT theo các lớp của mô hình lý thuyết OSI để có thể ứng dụng chung trong nhiều lĩnh vực, ngữ cảnh. Sau nhiều năm phát triển và từ các nghiên cứu chỉ ra rằng kiến trúc IoT có thể chia thành 3 lớp, 4 lớp hoặc thậm chí là 5 lớp tùy theo tính chi tiết của kiến trúc, nhưng nhìn chung mô hình 4 lớp phổ biến và được ứng dụng nhiều nhất gồm các lớp: tri giác (perceptual), mạng (network),

hỗ trợ (support) và ứng dụng (application), bên cạnh đó yếu tố bảo mật đi kèm mỗi lớp cũng cực kỳ quan trọng [3].



Hình 1.1 - Các khối chức năng của hệ thống IoT

Công nghệ IoT có rất nhiều ứng dụng và việc sử dụng IoT đang phát triển nhanh hơn. Tùy thuộc vào các lĩnh vực ứng dụng khác nhau của IoT, nó sẽ hoạt động tương ứng như đã được thiết kế hay phát triển. Nhưng IoT không có một kiến trúc làm việc xác định tiêu chuẩn được tuân thủ nghiêm ngặt trên toàn thế giới. Kiến trúc của IoT phụ thuộc vào chức năng và việc triển khai của nó trong các lĩnh vực khác nhau. Tuy nhiên, có một quy trình cơ bản dựa trên đó IoT được xây dựng, gồm 3 giai đoạn (tầng) như sau [4]:



Hình 1.2 - Các giai đoạn của kiến trúc IoT

Trong đó:

- ✚ **Lớp cảm biến (SENSING LAYER):** Cảm biến, thiết bị truyền động, các thiết bị có trong lớp cảm biến này. Các bộ phận cảm biến hoặc bộ truyền động này nhận dữ liệu (Thông số vật lý/ môi trường), xử lý dữ liệu và phát dữ liệu qua mạng.
- ✚ **Lớp mạng (NETWORK LAYER):** Các cổng Internet (mạng), hệ thống thu nhập dữ liệu (Data Acquisition System -DAS) xuất hiện trong lớp này. DAS thực hiện chức năng tổng hợp và chuyển đổi dữ liệu (Thu thập dữ liệu và tổng hợp dữ liệu sau đó chuyển đổi dữ liệu analog của cảm biến sang dữ liệu digital,...). Các cổng nâng cao chủ yếu mở ra kết nối giữa mạng cảm biến và Internet cũng thực hiện nhiều chức năng cơ bản, như bảo vệ chống phần mềm độc hại và lọc một số lần ra quyết định dựa trên dữ liệu đã nhập và các dịch vụ quản lý dữ liệu.
- ✚ **Lớp xử lý dữ liệu (DATA PROCESSING LAYER):** Đây là đơn vị xử lý của hệ sinh thái IoT. Tại đây, dữ liệu được phân tích và xử lý trước khi gửi đến trung tâm dữ liệu, nơi dữ liệu được truy cập bởi các ứng dụng phần mềm thường được gọi là ứng dụng kinh doanh. Đây là nơi dữ liệu được theo dõi và quản lý. Các hành động khác cũng được chuẩn bị tại lớp này.
- ✚ **Lớp ứng dụng (APPLICATION LAYER):** Đây là lớp cuối cùng trong 4 giai đoạn của kiến trúc IoT. Trung tâm dữ liệu hoặc đám mây (Cloud) là giai đoạn quản lý, nơi dữ liệu được quản lý và sử dụng bởi các ứng dụng người dùng cuối như nông nghiệp, chăm sóc sức khỏe, hàng không vũ trụ, nông nghiệp, quốc phòng và những ứng dụng khác.

I.4. Ứng dụng

Các ứng dụng của Internet of Things [1]:

I.4.1. Thành phố thông minh – Smart City

- *Công viên thông minh:* giám sát không gian đỗ xe của thành phố.
- *Kiểm tra xây dựng:* giám sát các rung động và các điều kiện vật chất trong các tòa nhà, cầu và công trình lịch sử.
- *Bản đồ tiếng ồn thành phố:* giám sát âm thanh trong các phạm vi quán bar và các khu trung tâm theo thời gian thực.
- *Tắc nghẽn giao thông:* giám sát các phương tiện và mức độ người đi bộ để tối ưu việc lái xe và đi lại.
- *Chiếu sáng thông minh:* chiếu sáng thông minh và tương ứng với thời tiết trong hệ thống đèn đường.
- *Quản lý chất thải:* phát hiện mức độ rác thải trong các container để tối ưu đường đi thu gom rác.
- *Hệ thống vận tải thông minh:* các tuyến đường và cao tốc thông minh với các thông điệp cảnh báo và các điều chỉnh theo điều kiện thời tiết và các sự kiện không mong muốn như tai nạn, tắc đường.

1.4.2. Môi trường thông minh – Smart Environment

- *Phát hiện cháy rừng*: giám sát khí gas đốt cháy và các điều kiện cảnh báo cháy rừng để đưa ra vùng cảnh báo.
- *Ô nhiễm không khí*: điều khiển khí CO₂ thải ra từ các nhà máy, các khí gây ô nhiễm từ phương tiện và khí độc trong các nông trại.
- *Phòng ngừa lũ quét và lở đất*: giám sát độ ẩm đất, các rung chấn và mật độ đất để phát hiện các mối nguy hiểm theo điều kiện đất.
- *Phát hiện sớm động đất*: phân bố giám sát ở các vùng đặc biệt có rung chấn.

1.4.3. Nước thông minh – Smart Water

- *Chất lượng nước*: nghiên cứu về sự thích hợp của nước trên các sông, vùng biển đối với hệ động vật và tiêu chuẩn nước để sử dụng.
- *Rò rỉ nước*: phát hiện chất lỏng bên ngoài các kết và các biến động áp suất bên trong các đường ống.
- *Lũ trên các con sông*: giám sát biến động mực nước trên các sông, đập và nguồn nước.

1.4.4. Đo lường thông minh – Smart Metering

- *Mạng lưới thông minh*: giám sát và quản lý tình hình tiêu thụ năng lượng.
- *Tính hình các bể chứa*: giám sát mức nước, dầu, khí ga trong các kết và bể chứa.
- *Cài đặt hệ thống quan điện*: giám sát và tối ưu hiệu quả của các thiết bị năng lượng mặt trời.
- *Lưu lượng nước*: đo lường áp suất nước trong các hệ thống dẫn nước.
- *Tính toán trữ lượng hàng*: đo lường mức độ còn và khối lượng hàng hóa.

1.4.5. Bảo mật và tình trạng khẩn cấp – Security and Emergencies

- *Kiểm soát vùng hạn chế truy nhập*: điều khiển truy nhập tới các vùng hạn chế và phát hiện người không phận sự.
- *Sự có mặt của chất lỏng*: phát hiện chất lỏng trong các trung tâm dữ liệu, kho dữ liệu và vị trí xây dựng nhạy cảm để chống sự hỏng hóc và ăn mòn.
- *Mức bức xạ*: phân phối đo lường phạm vi bức xạ xung quanh các khu năng lượng hạt nhân để cảnh báo rò rỉ kịp thời.
- *Các khí nguy hiểm và nổ khí*: phát hiện mức độ khí gas và rò rỉ trong các môi trường công nghiệp, xung quanh các nhà máy hóa chất và trong các mỏ.

1.4.6. Hệ thống bán lẻ - Retail

- *Thanh toán NFC*: quá trình chi trả dựa trên vị trí hoặc khoảng thời gian hoạt động để chuyên chở công cộng, thể thao, các công viên chủ đề...
- *Điều khiển băng chuyền*: giám sát các điều kiện lưu trữ trong các dây chuyền và theo dõi sản phẩm cho mục đích giám sát.

- *Các ứng dụng mua sắm thông minh*: đưa ra lời khuyên tại các thời điểm bán theo thói quen khách hàng, sở thích, các thành phần gây phản cảm với người mua hoặc hạn sử dụng.
- *Quản lý sản phẩm thông minh*: điều khiển sự luân phiên các sản phẩm trong giá và kho để tự động các quá trình bổ sung.

1.4.7. Điều khiển trong công nghiệp – Industrial Control

- *Các ứng dụng M2M*: kiểm soát tài sản và các máy tự chẩn đoán.
- *Chất lượng không khí trong nhà*: giám giá khi độc và mức độ khí oxi trong các thiết bị hóa học để đảm bảo cho công nhân và hàng hóa an toàn.
- *Giám sát nhiệt độ*: kiểm soát nhiệt độ trong công nghiệp và tử y học với hàng hóa nhạy cảm.
- *Sự có mặt của khí Ozone*: kiểm soát mức độ khí Ozone trong khi làm khô các sản phẩm thịt trong các nhà máy thực phẩm.
- *Định vị trong nhà*: vị trí các tài sản trong nhà bằng cách sử dụng các thẻ tích cự (ZigBee) và bộ động (RFID/NFC).
- *Các phương tiện tự chẩn đoán*: thu thập thông tin từ CanBus để gửi các cảnh báo thời gian thực để nhanh chóng đưa lời khuyên cho lái xe.

1.4.8. Trang trại gia súc thông minh – Smart Animal Farming

- *Chăm sóc*: kiểm soát các điều kiện phát triển của con non trong trang trại gia súc để đảm bảo sống sót và khỏe mạnh.
- *Theo dõi gia súc*: vị trí và định danh của các gia súc trên đồng cỏ và phạm vi chăn thả.
- *Mức độ khí độc*: nghiên cứu thông hơi và chất lượng không khí trong các nông trại và phát hiện khí có hại.

1.4.9. Tự động nhà và domotic

- *Sử dụng năng lượng và nước*: hỗ trợ giám sát tiêu dùng nước và năng lượng để cho lời khuyên làm thế nào để tiết kiệm chi phí và tài nguyên.
- *Các thiết bị điều khiển từ xa*: bật tắt bằng thiết bị điều khiển từ xa để tránh tai nạn và tiết kiệm năng lượng.
- *Các hệ thống phát hiện xâm nhập*: phát hiện của sổ và cửa ra vào và các xâm phạm để chống đột nhập.
- *Bảo quản hàng hóa và nghệ thuật*: giám sát các điều kiện bên trong bảo tàng và nơi lưu trữ nghệ thuật.

1.4.10. Chăm sóc sức khỏe – eHealth

- *Phát hiện ngã*: giúp đỡ người già hoặc người tàn tật sống độc lập.
- *Các tủ thuốc*: giám sát các điều kiện bên trong tủ lạnh lưu trữ vaccine, thuốc, và các yếu tố hữu cơ.

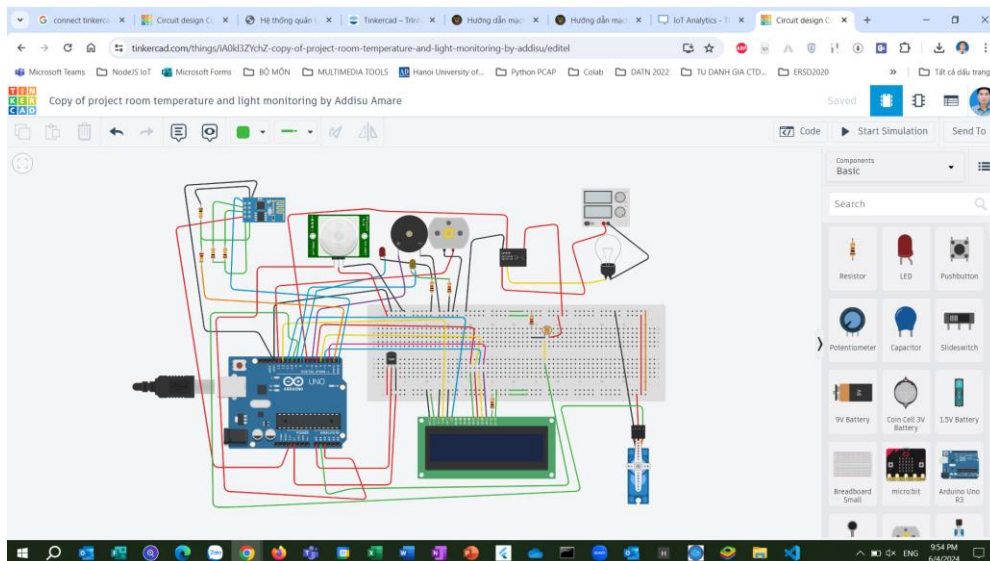
- *Chăm sóc vận động viên:* giám sát các dấu hiệu trong trung tâm chăm sóc.
- *Giám sát bệnh nhân:* giám sát điều kiện của các bệnh nhân trong bệnh viện và tại nhà họ.
- *Bức xạ tử ngoại:* đo lường lượng tia mặt trời UV để cảnh báo mọi người không phơi nắng trong những giờ nhất định.

CHƯƠNG 2: TỔNG QUAN VỀ MÔI TRƯỜNG PHÁT TRIỂN

II.1. Phần mềm tinkercad

Khái niệm:

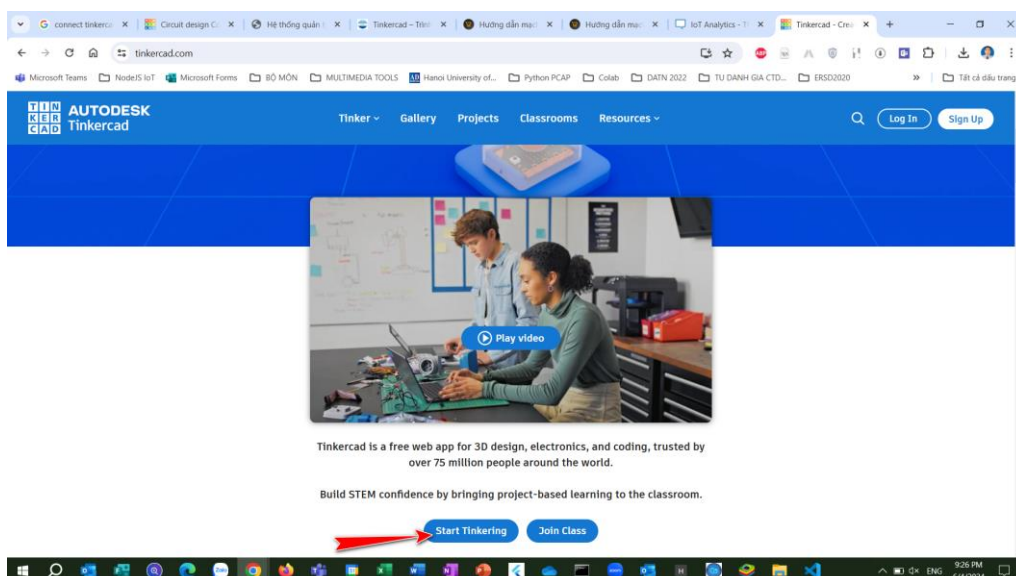
Theo [5], tinkercad là một trình giả lập Arduino + Circuit đầy đủ chức năng chạy trên trình duyệt web. Tinkercad cung cấp “chế độ xem bảng mạch” với đồ họa trông giống như các bộ phận mạch vật lý mà bạn sẽ sử dụng trong đời thực, đây là một cách tiếp cận thân thiện với người mới bắt đầu hơn nhiều. Khả năng mô phỏng Arduino và chạy mã trực tiếp trong trình duyệt web là một ưu điểm nổi bật.



Hình II.1 - Giao diện chính phần mềm tinkercad (tinkercad.com)

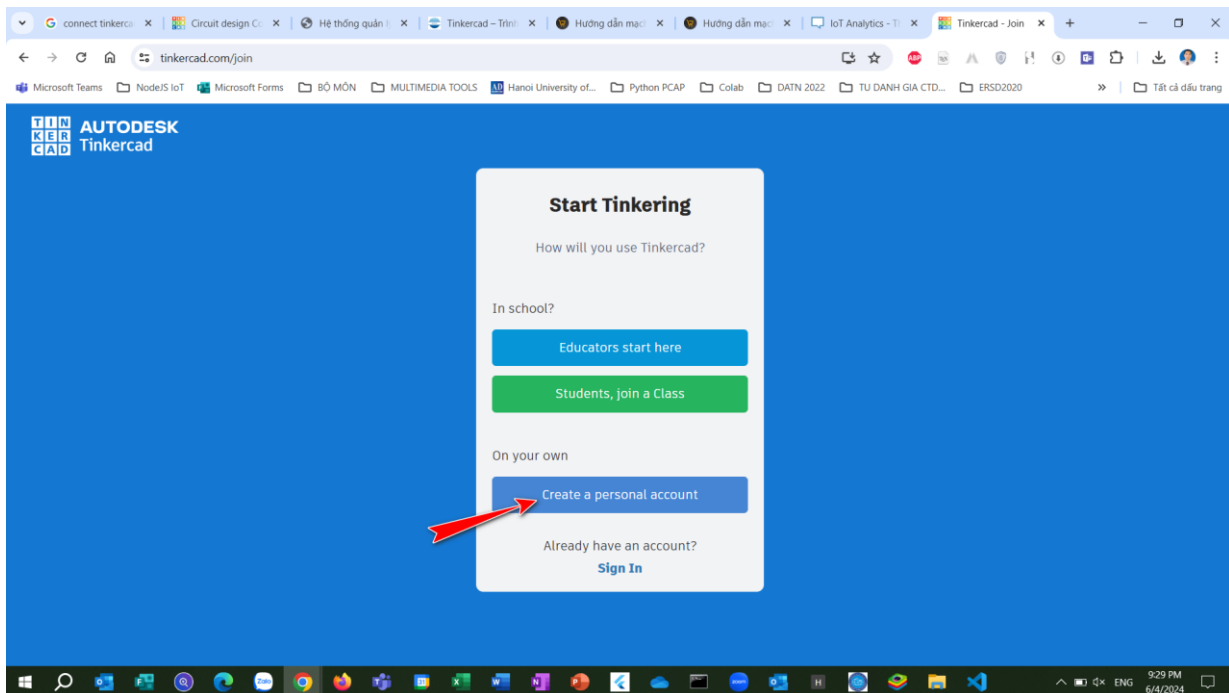
Các bước thực hiện:

- **Bước 1:** Truy cập website tinkercad.com ⇒ chọn Start Tinkering



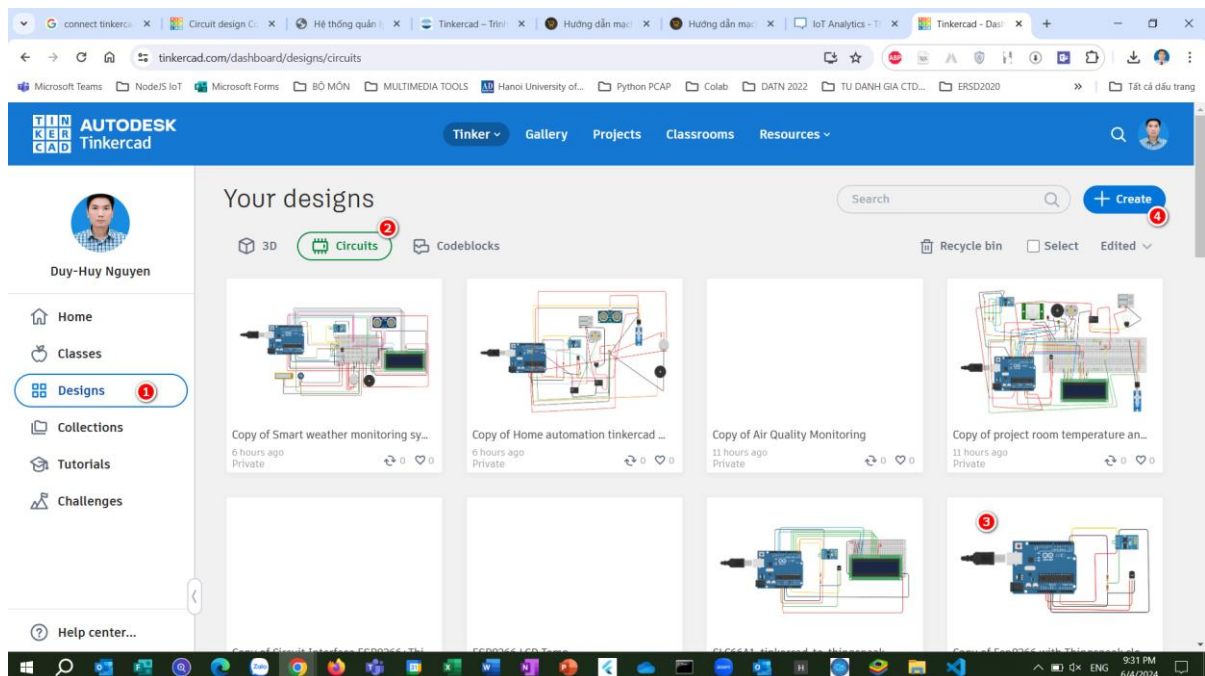
Hình II.2 – Giao diện chính website tinkercad.com

- **Bước 2:** Tạo tài khoản và đăng nhập \Rightarrow Chọn Create a personal account (có thể đăng nhập qua tài khoản Gmail)



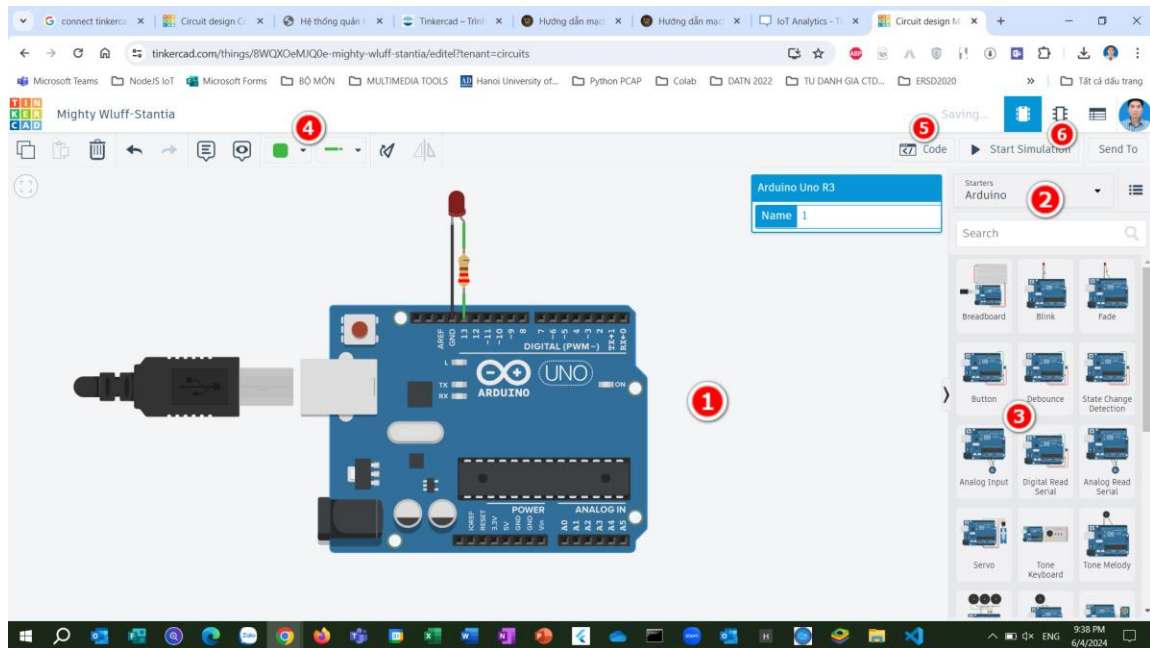
Hình II.3 – Chọn phương pháp đăng nhập tinkercad

- **Bước 3:** Trên màn hình giao diện chính chọn (1) – Designs \Rightarrow (2) – Circuits \Rightarrow (3) – Chọn mạch có sẵn hoặc chọn (4) – Create để tạo mạch mới (chọn Circuits)



Hình II.4 – Giao diện lựa chọn mạch có sẵn/mới trong tinkercad

Nếu chọn (4) sẽ xuất hiện giao diện sau:



Hình II.5 – Giao diện thiết kế mạch IoT ảo trong tinkercad

Trong đó:

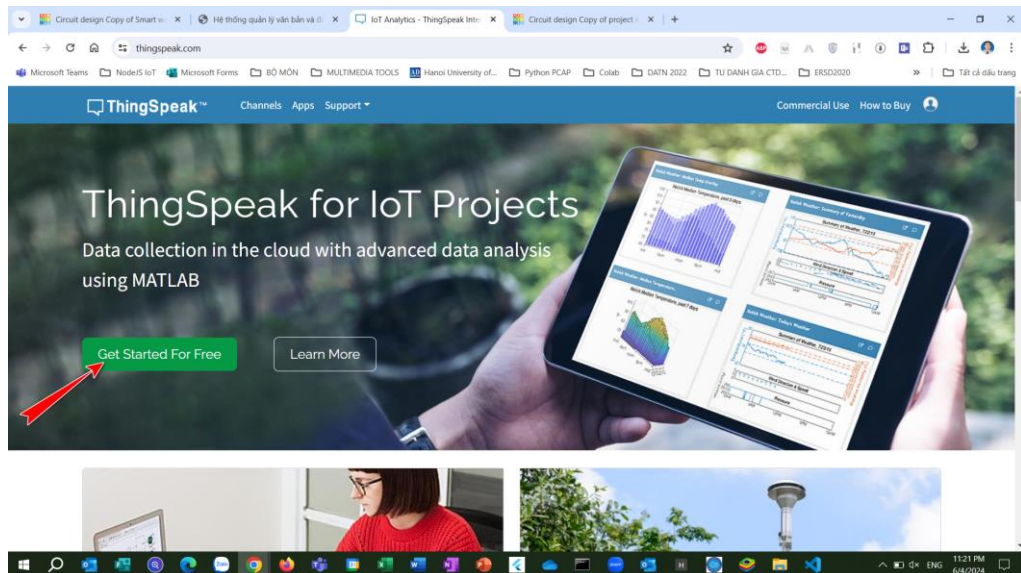
- (1): Khu vực vẽ mạch
- (2): Nhóm mạch/link kiện
- (3): Chọn mạch/linh kiện mẫu: Chọn Arduino hoặc Basic
- (4): Công cụ hiệu chỉnh mạch
- (5): Mở cửa sổ code để lập trình cho mạch
- (6): Mô phỏng mạch

II.2. Nền tảng thingspeak

Theo [7], [8] tổng quan về thingspeak có thể biểu diễn như sau:

ThingSpeak là một IoT Platform cho phép bạn gửi dữ liệu cảm biến (sensor) tới đám mây (cloud). Bạn cũng có thể phân tích và trực quan hóa dữ liệu sử dụng công cụ MATLAB hoặc các phần mềm khác. Dịch vụ ThingSpeak được vận hành bởi MathWorks. Để sử dụng ThingSpeak, bạn phải tạo tài khoản MathWorks. Hiện tại ThingSpeak đang miễn phí cho các dự án nhỏ không mang tính chất thương mại.

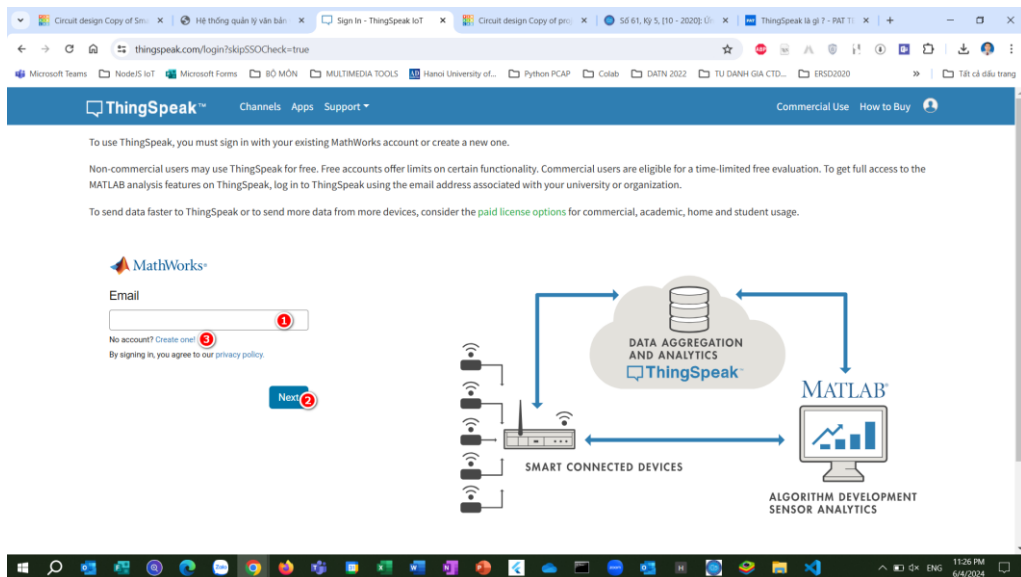
ThingSpeak bao gồm một Web Service (REST API) giúp bạn có thể thu thập, lưu trữ dữ liệu cảm biến trên cloud và phát triển các ứng dụng IOT. ThingSpeak làm việc được với Arduino, Raspberry Pi, và MATLAB. Tuy nhiên nó cũng có thể ghép nối với bất cứ phần mềm nào bằng việc sử dụng REST API và HTTP.



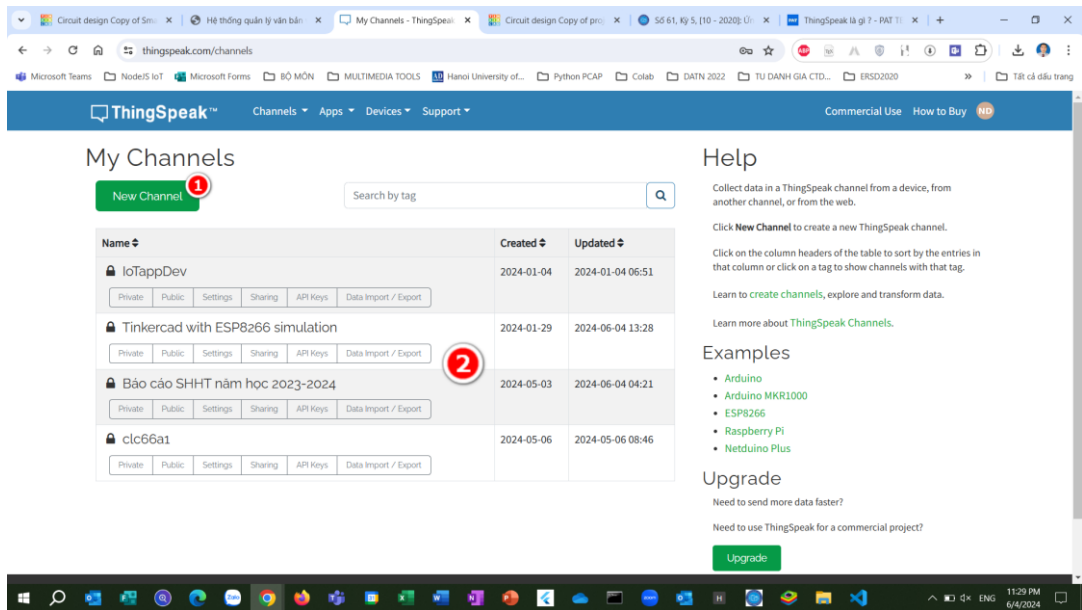
Hình II.6 – Giao diện chính website thingspeak.com

Các bước thiết kế webapp với thingspeak:

- **Bước 1:** Truy cập website thingspeak.com ⇒ chọn Get Started For Free
- **Bước 2:** Nhập email (1), nhấn (2) để tiếp tục. Nếu chưa có tài khoản nhấn (3) và làm theo hướng dẫn.



Hình II.7 – Giao diện đăng nhập trên thingspeak.com



Hình II.8 – Giao diện chính sau khi đăng nhập trên thingspeak

Trên giao diện chính sau khi đăng nhập:

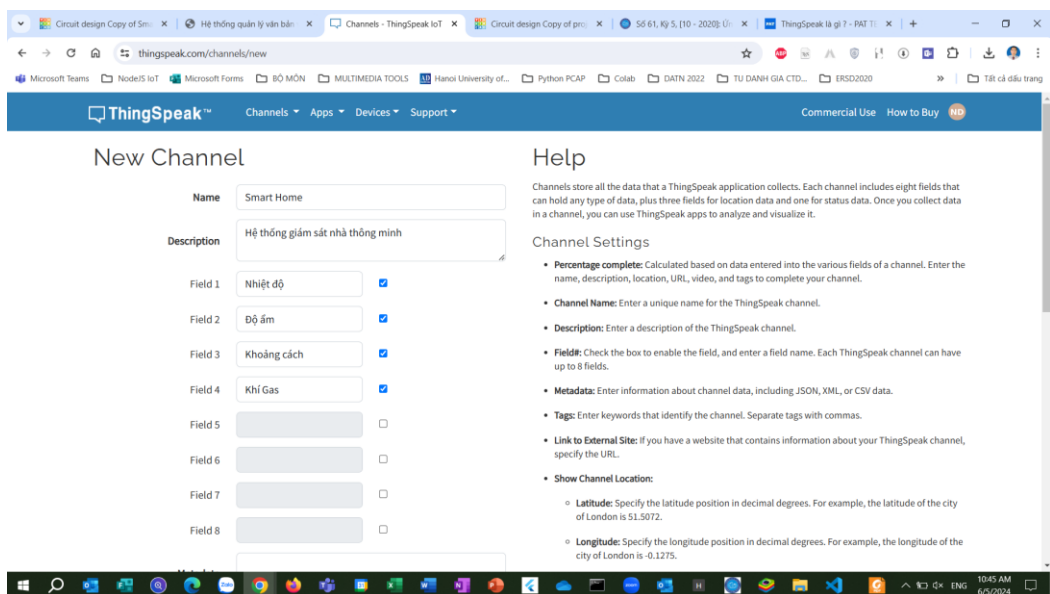
(1): Tạo kênh mới

(2): Hiện thị thông tin tất cả các kênh đã tạo (tài khoản free chỉ tạo được tối đa 4 kênh)

➤ **Tạo kênh mới:**

✓ **Bước 1:** Chọn New Channel (1) ⇒ Nhập thông tin cho kênh

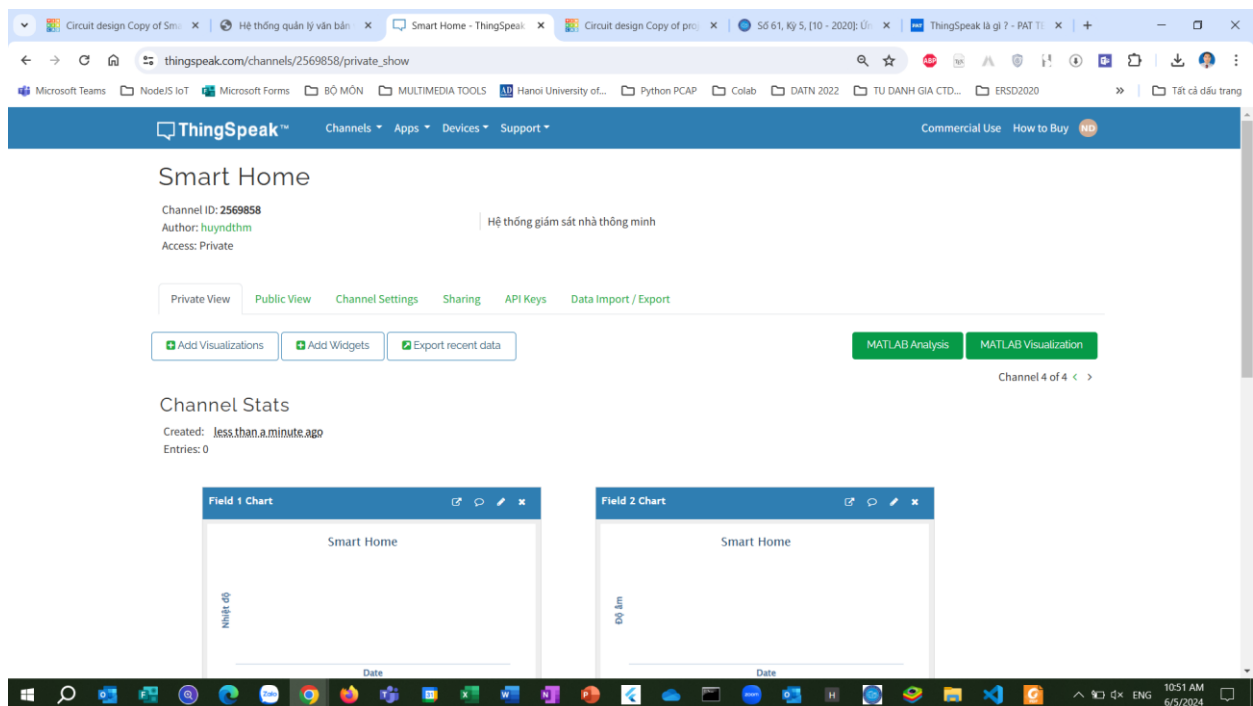
- **Name:** tên kênh
- **Description:** Mô tả về kênh
- **Field 1 – 8:** Các trường sẽ nhận dữ liệu từ các cảm biến/khối thao tác/thiết bị (tích chọn)



Hình II.9 – Giao diện thiết kế/cài đặt kênh mới trên thingspeak

Lưu ý: Tài khoản free chỉ hỗ trợ tối đa 8 fields

Các thông tin còn lại có thể không cần nhập (có gợi ý bên cạnh), xuống cuối trang chọn **Save Channel**. Kết quả:



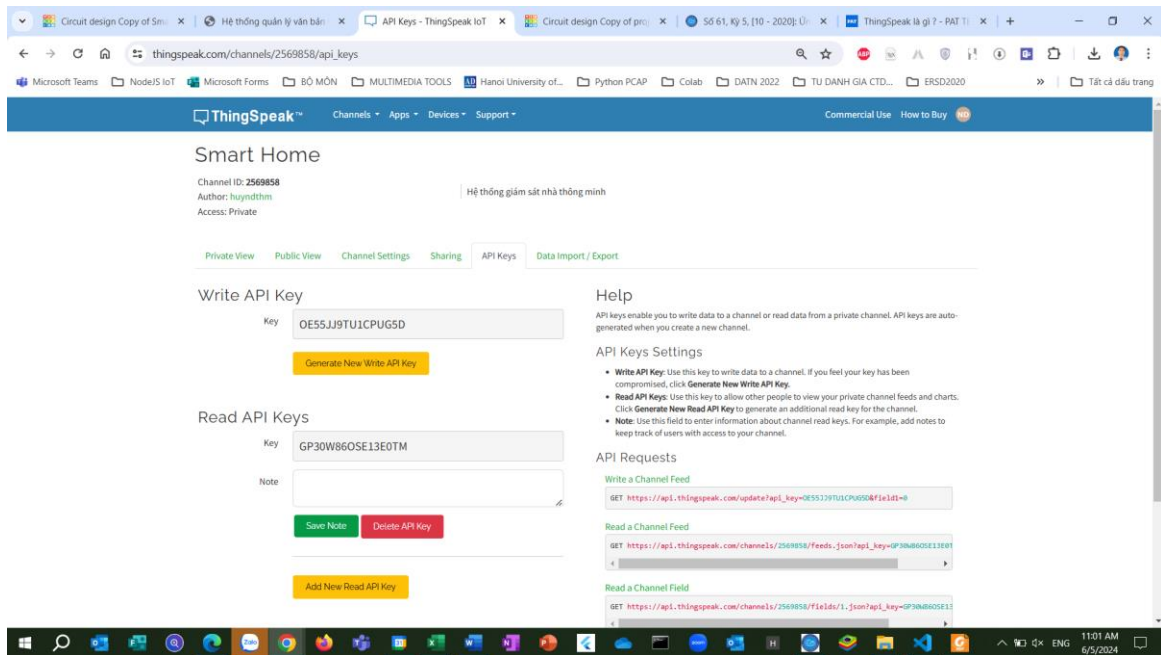
Hình II.10 – Giao diện IoT dashboard sau khi thiết kế xong

Một số thông tin chính:

- **Private View:** Phải đăng nhập tài khoản mới xem được thông tin
- **Public View:** Không cần đăng nhập tài khoản vẫn xem được thông tin
- **Channel Settings:** Cài đặt/Chỉnh sửa kênh
- **Sharing:** Chế độ chia sẻ kênh
- **API Keys:** Quyền ghi dữ liệu vào kênh hoặc đọc dữ liệu từ kênh
- **Data Import/Export:** Nhập xuất dữ liệu cho kênh
- **Add Visualizations:** Thêm biểu đồ trực quan hóa dữ liệu
- **Add Widgets:** Thêm kiểu biểu đồ khác hiển thị dữ liệu cho kênh
- **Export recent data:** Xuất các field dữ liệu ra file: JSON/XML/CSV
- **Field 1 Chart, Filed 2 Chart:** Biểu đồ hiển thị dữ liệu thu thập được. Có thể tùy chỉnh bằng cách chọn biểu tượng cái bút trên thanh tiêu đề.

Để kết nối mạch từ tinkercad đến thingspeak, API keys đóng vai trò rất quan trọng, gồm 2 loại:

- ✓ **Write API Key:** Key để ghi dữ liệu vào thingspeak
- ✓ **Read API Keys:** Key để đọc dữ liệu từ thingspeak



Hình II.11 – API Keys

Để ghi dữ liệu từ tinkercad vào thingspeak thì cần Write AP Key. Key này được thêm vào trong code cho mạch trên tinkercad.

Nếu muốn tạo key mới thì nhấn nút **Generate New Write API Key**

CHƯƠNG III: MÔ PHỎNG HỆ THỐNG INTERNET OF THINGS VỚI TINKERCAD VÀ THINGSPEAK

III.1. Mô hình truyền dữ liệu

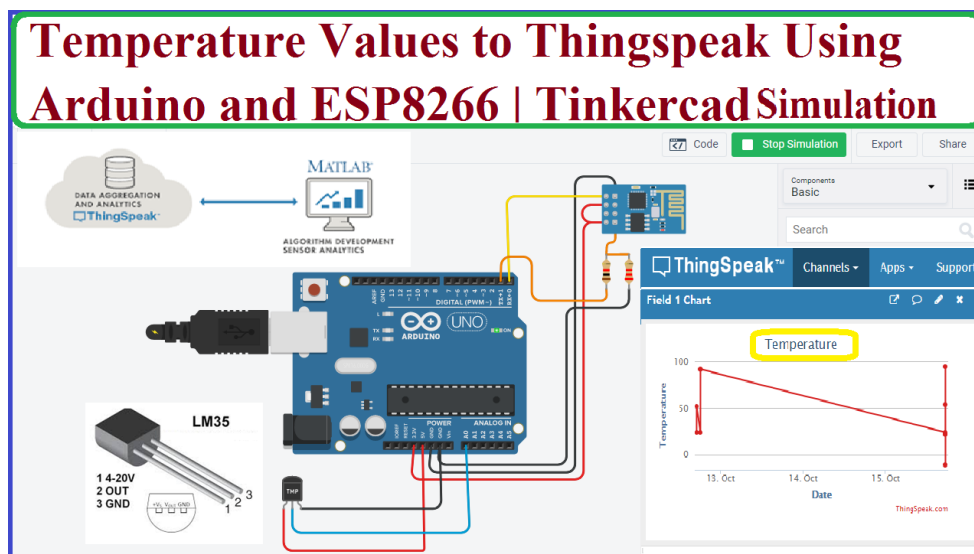
Trong chương này sẽ trình bày các bước thiết kế, xây dựng, lập trình và mô phỏng hệ thống IoT đơn giản trên máy tính với thiết bị phần cứng ảo. Hệ thống demo được xây dựng dựa trên ý tưởng:

- Hiển thị dữ liệu thu thập được từ các cảm biến (trong mạch ảo tinkercad) trên thingspeak.
- Hệ thống có một số chức năng xử lý tự động
- Hệ thống mô phỏng một số chức năng trong nhà thông minh (Smart Home)

Yêu cầu: Máy tính có kết nối Internet, đã đăng ký và đăng nhập vào các tài khoản cá nhân trên website tinkercad.com và thingspeak.com.

Để xây dựng được hệ thống trên cần thực hiện theo các bước sau:

- **Bước 1:** Thiết kế mạch trong tinkercad (tinkercad.com)
- **Bước 2:** Xây dựng webapp trên nền tảng thingspeak (thingspeak.com)
- **Bước 3:** Lập trình nhúng cho mạch trong tinkercad (tinkercad.com)
- **Bước 4:** Chạy mạch mô phỏng mạch trong tinkercad, quan sát kết quả trên thingspeak.



Hình III.1 – Mô hình kết nối tinkercad với thingspeak

Trong mô hình trên có thể chia thành các khối:

- (1) Khối cảm biến + cơ cấu chấp hành + thiết bị: Thu thập dữ liệu hoặc điều khiển thiết bị
- (2) Board Arduino (UNO): Nhận tín hiệu từ cảm biến, xử lý và đẩy lên web server hoặc gửi tín hiệu điều khiển cho cơ cấu chấp hành.
- (3) Module ESP8266: Truyền/nhận tín hiệu lên/từ internet qua wifi

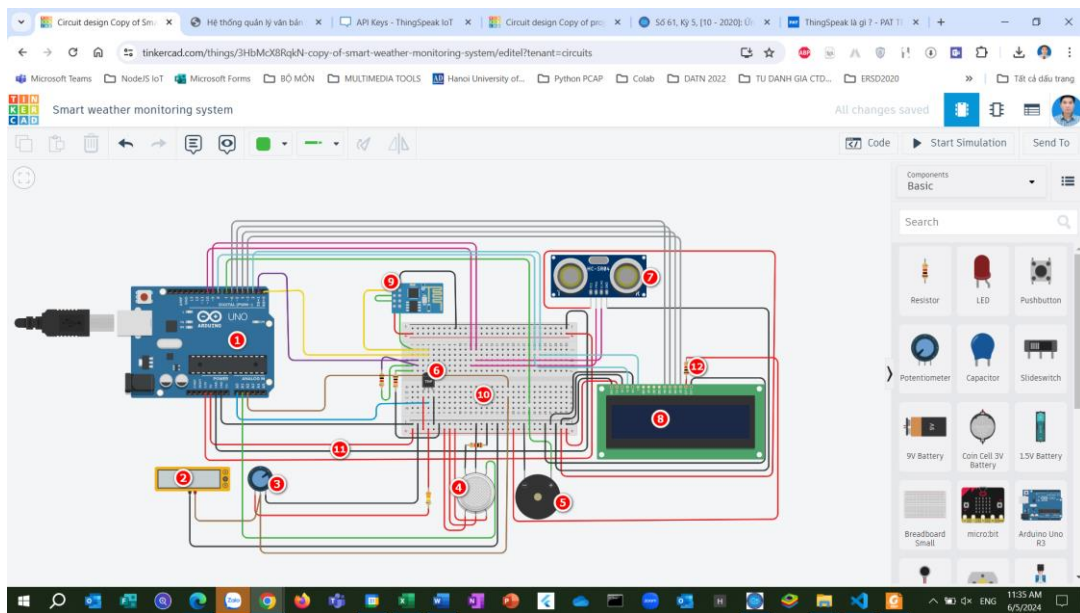
(4) Thingspeak platform: Tạo Realtime Database, IoT dashboard...

III.2. Thiết kế mạch trong tinkercad

Trong mục này sẽ trình bày **Bước 1**. Để thực hiện được cần có một số phần mềm sau:

- ✓ **Tinkercad:** Truy cập website tinkercad.com, đăng nhập với tài khoản đã đăng ký ⇒ chọn Designs ⇒ Chọn Circuits ⇒ Vẽ mạch (đã được mô tả ở những phần trên).

Mạch mô phỏng trong báo cáo này sau khi vẽ được thể hiện như dưới đây:



Hình III.2 – Mạch thiết kế đề xuất trong tinkercad

Các thành phần chính gồm:

- ✓ (1) **Board Arduino UNO R3:** Thiết bị IoT chính, giúp kết nối các thiết bị, nhận tín hiệu từ cảm biến, gửi tín hiệu điều khiển đến cơ cấu chấp hành, gửi dữ liệu cho module ESP8266 để đẩy lên web server thingspeak.
- ✓ (2) **Multimeter:** máy đo điện áp
- ✓ (3) **Potentiometer:** máy đo điện trở
- ✓ (4) **Gas Sensor:** cảm biến khí gas
- ✓ (5) **Piezo:** loa/chuông báo động
- ✓ (6) **Temperature Sensor [TMP36]:** cảm biến đo nhiệt độ
- ✓ (7) **Ultrasonic Distance Sensor (HC-SR04):** cảm biến siêu âm đo khoảng cách
- ✓ (8) **LCD 16x2:** màn hình LCD
- ✓ (9) **Wifi Module (ESP8266):** module wifi ESP8266
- ✓ (10) **Breadboard small:** bảng mạch mở rộng
- ✓ (11) Hệ thống dây dẫn (wire)
- ✓ (12) **Resistor:** điện trở

Nguyên lý hoạt động của hệ thống: Các cảm biến thu thập dữ liệu và gửi cho board Arduino, board sẽ tính toán và gửi kết quả ra màn hình LCD, đồng thời gửi lên thingspeak

qua module ESP8266. Nếu kết quả thu thập được từ cảm biến vượt quá giới hạn cho phép, tín hiệu cảnh báo sẽ được gửi đến loa (piezo).

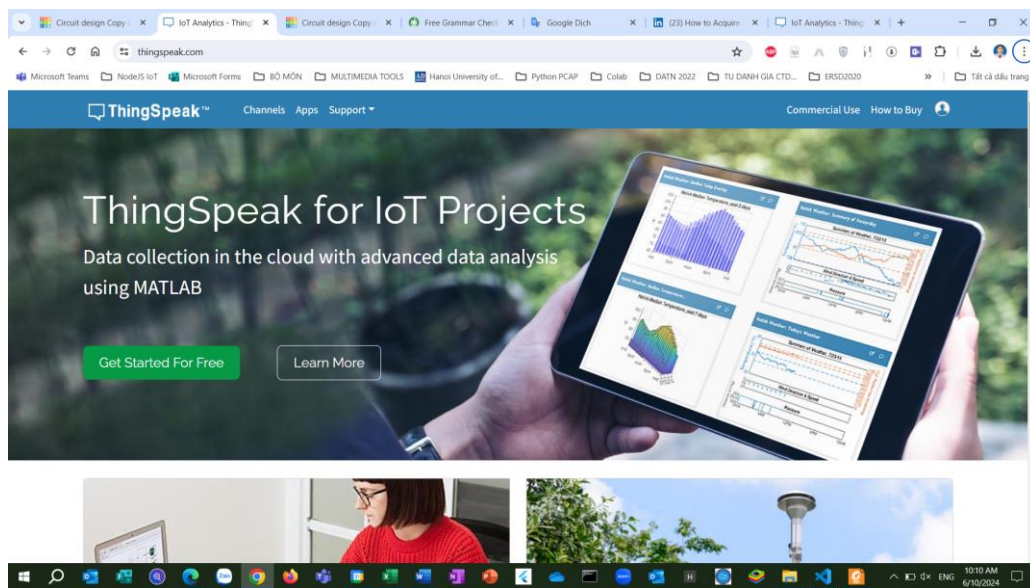
III.3. Thiết kế ứng dụng web trên nền tảng thingspeak

Mục này trình bày nội dung **Bước 2** - Xây dựng ứng dụng web để hiển thị kết quả thu thập được từ mạch trong Proteus lên webapp. Để làm được việc đó cần thực hiện như sau:

- **B1:** Truy cập website <https://thingspeak.com/>, đăng nhập hoặc đăng ký tài khoản (nếu chưa có).

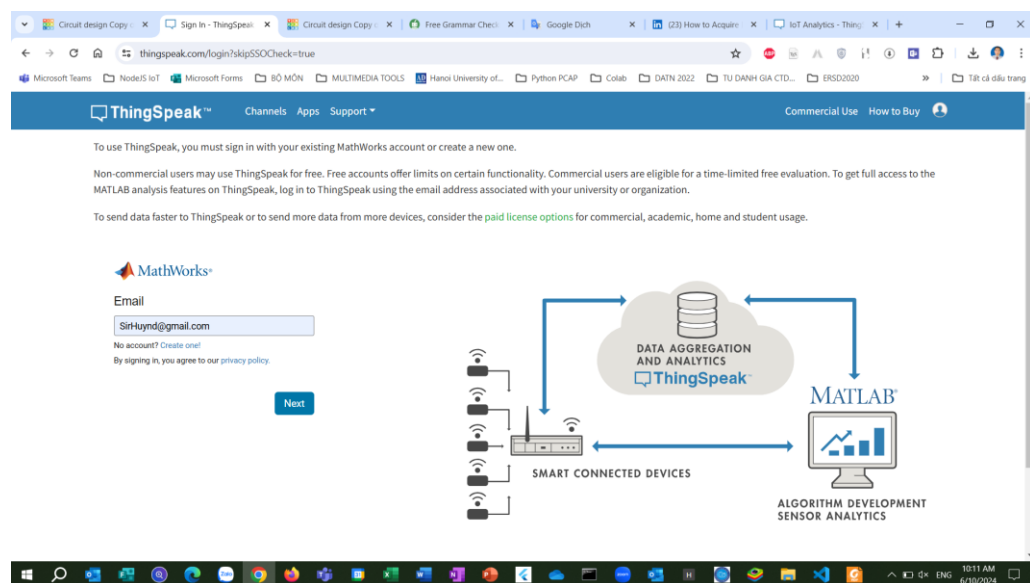
- **B2:** Thiết kế webapp (Web Dashboard)

Sau khi đăng nhập có giao diện chính như sau:

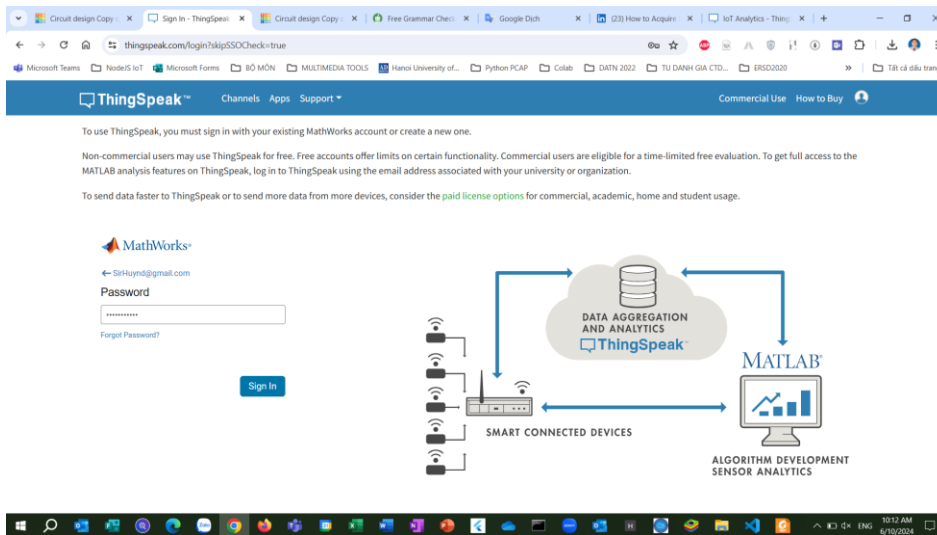


⇒ Nhấn chọn **Get Started For Free.**

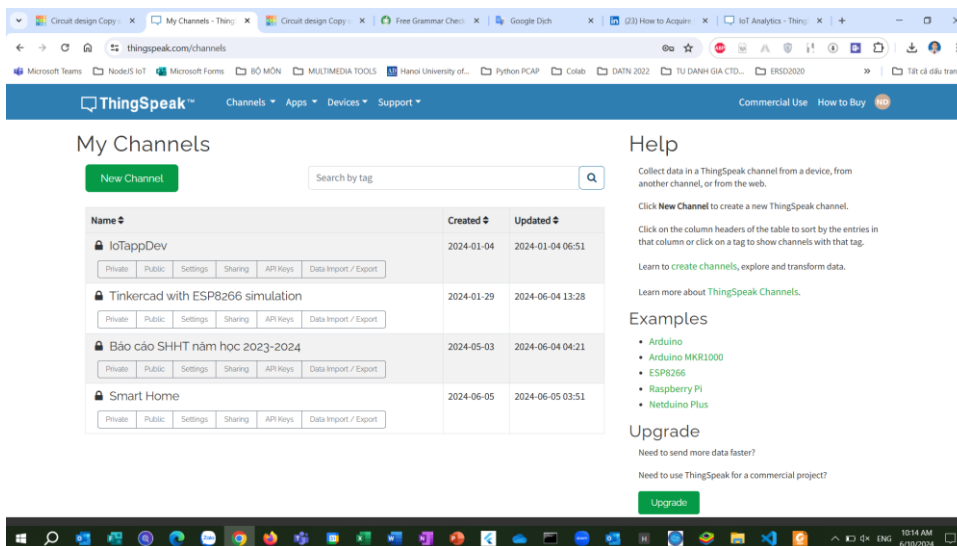
Nhập email ⇒ nhấn **Next:**



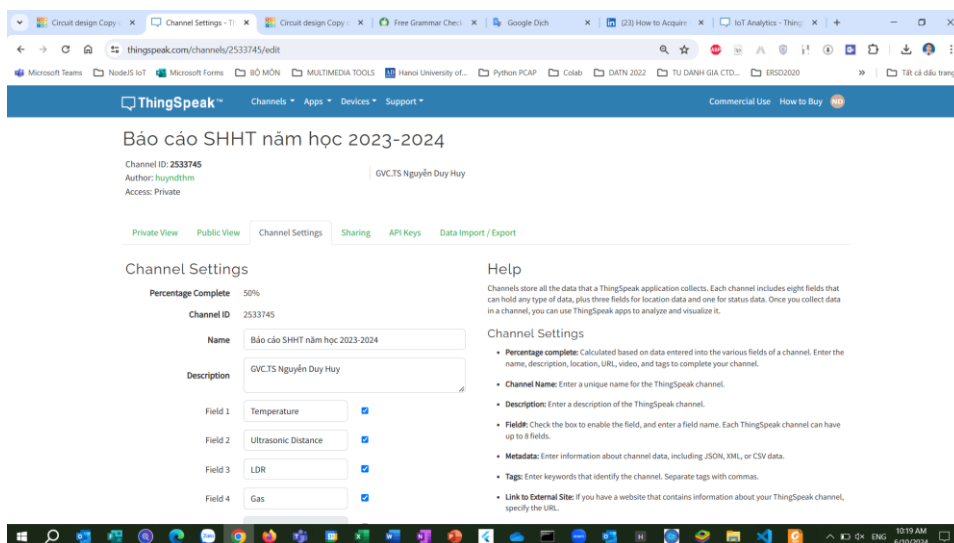
Nhập mật khẩu ⇒ **Sign In**



Xuất hiện giao diện sau:



(1) Nhấn **New Channel** ⇒ Tạo kênh mới (tài khoản free chỉ tạo được tối đa 4 kênh)



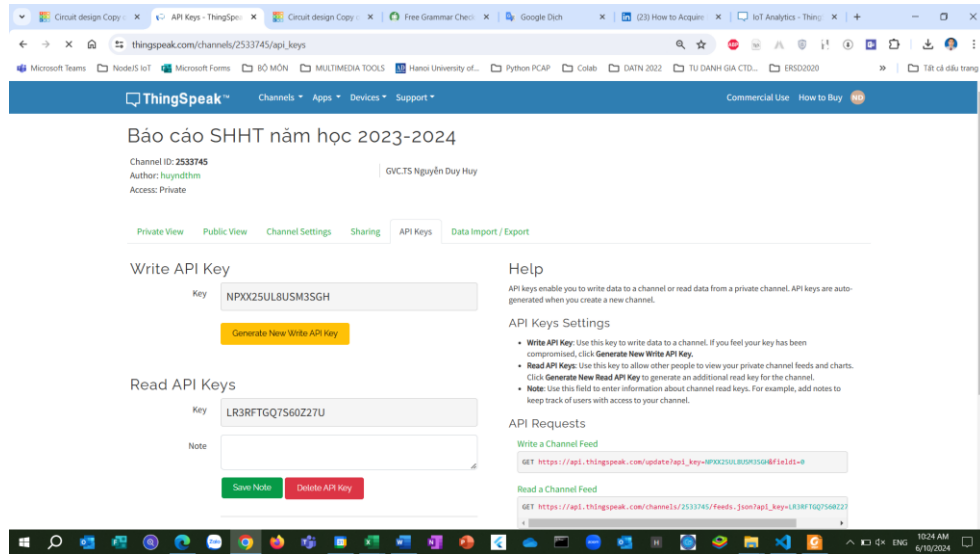
Hình III.3 – Thiết lập kênh trên thingspeak

(2) Thiếp lập kênh ⇒ chọn tab **Channel Settings**

- **Name:** Tên kênh
- **Description:** Mô tả về kênh (tùy chọn)
- **Field 1:** Temperature (Nhiệt độ) ⇒ tích chọn
- **Field 2:** Ultrasonic Distance (Khoảng cách sóng siêu âm) ⇒ tích chọn
- **Field 3:** LDR (Cảm biến ánh sáng) ⇒ tích chọn
- **Field 4:** Gas (cảm biến khí gas) ⇒ tích chọn

(3) Chọn **Save Channel**

Tiếp đến chọn **API Keys:**



Hình III.4 – Tạo Write API Key

Nhớ sao chép **Write API Key**, những cài đặt khác có thể để ở chế độ mặc định. Đến đây kết thúc tạo Web Dashboard cho hệ thống.

III.4. Lập trình nhúng trong tinkercad

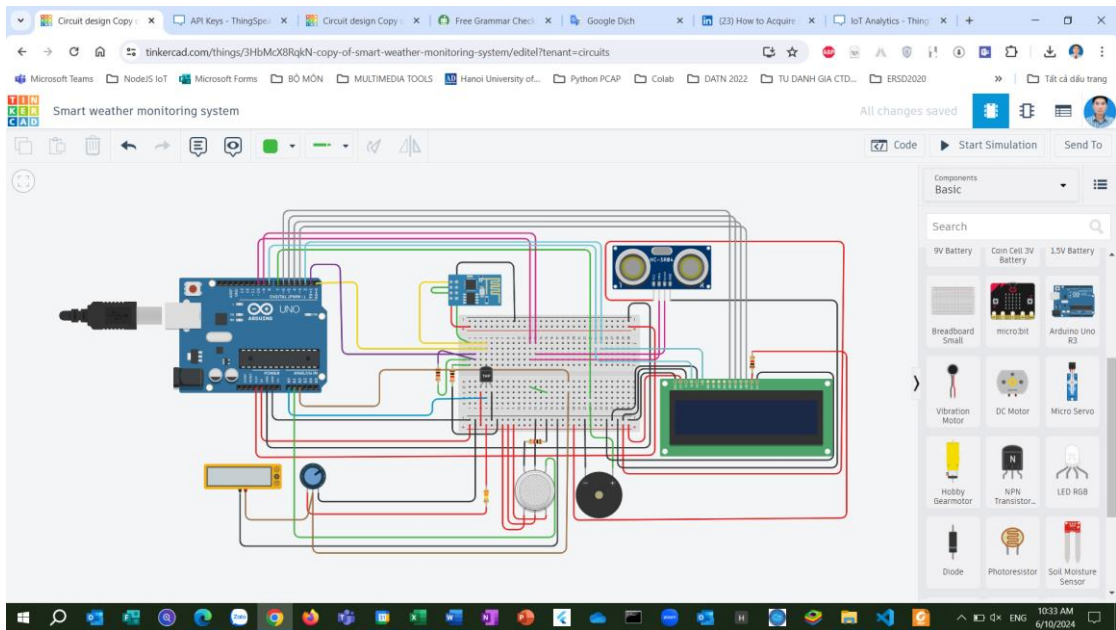
Phần này thực hiện **Bước 3**, mục tiêu chính là lập trình nhúng cho board Arduino UNO R3. Một số nhiệm vụ chính của board này:

- + Nhận dữ liệu từ các cảm biến
- + Tính toán giá trị thu thập được từ cảm biến
- + Gửi kết quả ra LCD
- + Gửi tín hiệu cảnh báo ra piezo
- + Gửi dữ liệu lên Thingspeak qua module wifi ESP8266

Yêu cầu:

- + Máy tính có kết nối Internet
- + Có tài khoản trên tinkercad

Lập trình: Chương trình được viết bằng ngôn ngữ Arduino trong tinkercad như sau:



Nhấn nút Code (góc trên phải)

```

1 //Smart weather reporting system
2 #include <liquidCrystal.h>
3
4 //temperature sensor variables
5 float temp_vout;
6 float temp;
7 float voltage;
8
9 //gas sensor variables
10 int gas_sensor_port = A1;
11 int gas_sensor_value = 0;
12
13 //rainfall measurement variables
14 float rain;
15 const int triggerPin = 10;
16 const int echoPin = 9;
17 long duration;
18
19 //wind speed measurement variables
20 float V_wind = 0;
21 float Windspeed(float);
22 int Windspeedint;
23 //LCD
24 LiquidCrystal lcd(8, 2, 6, 5, 4, 3); //Parameters: (rs, enable, d4, d5, d6, d7)
25
26 //WiFi module variables
27 String ssid = "Simulator WiFi"; // SSID to connect to
28 String password = ""; // virtual wifi has no password
29 String host = "api.thingspeak.com"; // Open Weather Map API
30 const int httpPort = 80;
31 String url = "/update?api_key=NPXX25U18USM3RGH&field1="; //ThingSpeak Channel API Key
32
33
34 //setting up the wifi module
35 void actupESP8266(void)
36 {
37     // Start our ESP8266 Serial Communication
38     Serial.begin(115200); // baud rate
39     Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266
40     delay(10); // Wait a little for the ESP to respond
41
42     if (Serial.find("OK"))
43         Serial.println("ESP8266 OK!!!"); // Connect to Simulator WiFi
44     Serial.println("AT+CWJAP=\"\" + ssid + "\",\"\" + password + \"\"); //AT+CWJAP - list near
45     delay(10); // Wait a little for the ESP to respond
46
47     if (Serial.find("OK"))
48         Serial.println("Connected to WiFi!!!"); // Open TCP connection to the host:
49
50     //ESP8266 connects to the server as a TCP client.
51     Serial.println("AT+CWJAPSTART=\"TCP\",\"\" + host + "\",\"\" + httpPort);
52     delay(50); // Wait a little for the ESP to respond
53
54     if (Serial.find("OK"))
55         Serial.println("ESP8266 Connected to server!!!");
56 }
57
58 //Sends data to Thingspeak
59 void send_data(void)
60 {
61     //for temperature sensor
62     temp_vout = analogRead(A0);
63     voltage = temp_vout * 0.0048828125; //convert analog value between 0 to 1023 with 5000mV
64     temp = (voltage - 0.5) * 100.0;
65
66     // Construct HTTP call
67     String httpPacket = "GET " + url + String(temp) + "&field2=" + String(gas_sensor_value)
68     int length = httpPacket.length();
69     // Send our message length
70     Serial.print("AT+CIPSEND=");
71     Serial.println(length);
72     delay(10); // wait a little for the ESP to respond if (!Serial.find(">")) return
73     // Send our http request
74     Serial.print(httpPacket);
75     delay(10); // wait a little for the ESP to respond
76     if (Serial.find("SEND OK\r\n"));
77         Serial.println("ESP8266 sends data to the server");
78 }
79
80 void setup()
81 {
82     pinMode(A1, INPUT); //gas sensor analog input
83     pinMode(7, OUTPUT); //gas sensor digital output
84     pinMode(A0, INPUT); //temperature sensor analog input.
85     pinMode(A2, INPUT); //potentiometer analog input
86     setupESP8266();
87     lcd.begin(16,2);
88     pinMode(triggerPin, OUTPUT);
89     pinMode(echoPin, INPUT);
90 }
91
92 void loop()
93 {
94     //for temperature sensor
95     //temp_vout = analogRead(A0);
96     //voltage = temp_vout * 0.0048828125; //convert analog value between 0 to 1023 with 5000
97     //temp = (voltage - 0.5) * 100.0;
98     Serial.println("Current temperature: " + String(temp));
99     //-----
100     //for gas sensor
101     gas_sensor_value = analogRead(gas_sensor_port);
102     Serial.println("Gas sensor value: " + String(gas_sensor_value));
103     if (gas_sensor_value > 200)
104     {
105         tone(7,523,1000);
106     }
107     //-----
108     //for rainfall measurement
109     digitalWrite(triggerPin, LOW);
110     delayMicroseconds(2);
111     // Sets the trigger pin to HIGH state for 10 microseconds
112     digitalWrite(triggerPin, HIGH);
113     delayMicroseconds(10);
114     digitalWrite(triggerPin, LOW);
115     // Reads the echo pin, and returns the sound wave travel time in microseconds
116     duration = pulseIn(echoPin, HIGH);
117     rain = 0.01/23 * duration;
118     delay(10); // Delay a little bit to improve simulation performance
119
120     lcd.setCursor(0,0); // Sets the location at which subsequent text written to the LCD will
121     lcd.print("Rainfall: "); // Prints string "Rainfall" on the LCD
122     lcd.print(rain); // prints the distance value from the sensor
123     Serial.println("Rainfall: " + String(rain));
124     delay(10);
125
126     //-----
127     //for wind speed measurement
128     float V_wind = analogRead(A2) * (5.0 / 1023.0);
129     // Voltage converted to MPH
130     Windspeedint = (V_wind - 0.4) * 10 * 2.025 + 2.237; // For LCD screen output
131     Windspeedfloat = (V_wind - 0.4) * 10 * 2.025 + 2.237; // For Serial monitor output
132     //wind speed LCD output
133     lcd.setCursor(0,1); // adjust cursor
134     lcd.print("Wind speed");
135     lcd.print(" MPH");
136     if (V_wind < 0.4)
137     {
138         lcd.print("0");
139     }
140     else
141     {
142         lcd.print(Windspeedint);
143     }
144     lcd.print(" MPH");
145     //wind speed serial monitor output
146     Serial.print("Wind Speed: ");
147     if (Windspeedfloat <= 0)
148     {
149         Serial.print("0.0");
150     }
151     else{
152         Serial.print(Windspeedfloat); // Output Wind speed value
153         Serial.println(" MPH");
154         delay(100);
155     }
156     Serial.print("Anemometer Voltage: ");
157     if (V_wind > 2){
158         Serial.println("Out of range!");
159     }
160     else if (V_wind < 0.4)
161     {
162         Serial.println("Out of range!");
163     }
164     else{
165         Serial.print(V_wind);
166         Serial.println(" V");
167     }
168     //send data to thingspeak
169     send_data();
170     delay(1000); // delay changed for faster analytics
171 }

```

Hình III.5 – Source code lập trình cho mạch

Một số module quan trọng:

* Module kết nối mạch IoT ảo và thingspeak:

```
//WIFI module variables
String ssid      = "Simulator Wifi"; // SSID to connect to
String password = ""; //virtual wifi has no password
String host      = "api.thingspeak.com"; // Open Weather Map API
const int httpPort = 80;
String url       = "/update?api_key=NPXX25UL8USM3SGH&field1="; //ThingSpeak Channel API Key
```

* Module cài đặt ESP8266:

```
//setting up the wifi module
void setupESP8266(void)
{
    // Start our ESP8266 Serial Communication
    Serial.begin(115200); // Baud rate
    Serial.println("AT"); // Serial connection on Tx / Rx port to ESP8266
    delay(10); // Wait a little for the ESP to respond

    if (Serial.find("OK"))
        Serial.println("ESP8266 OK!!!"); // Connect to Simulator Wifi

    Serial.println("AT+CWLAP=\"" + ssid + "\",\"" + password + "\""); //AT+CWLAP - list nearby
    delay(10); // Wait a little for the ESP to respond

    if (Serial.find("OK"))
        Serial.println("Connected to WiFi!!!"); // Open TCP connection to the host:

    //ESP8266 connects to the server as a TCP client.

    Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\",\" + httpPort);
    delay(50); // Wait a little for the ESP to respond

    if (Serial.find("OK"))
        Serial.println("ESP8266 Connected to server!!!");
}
}
```

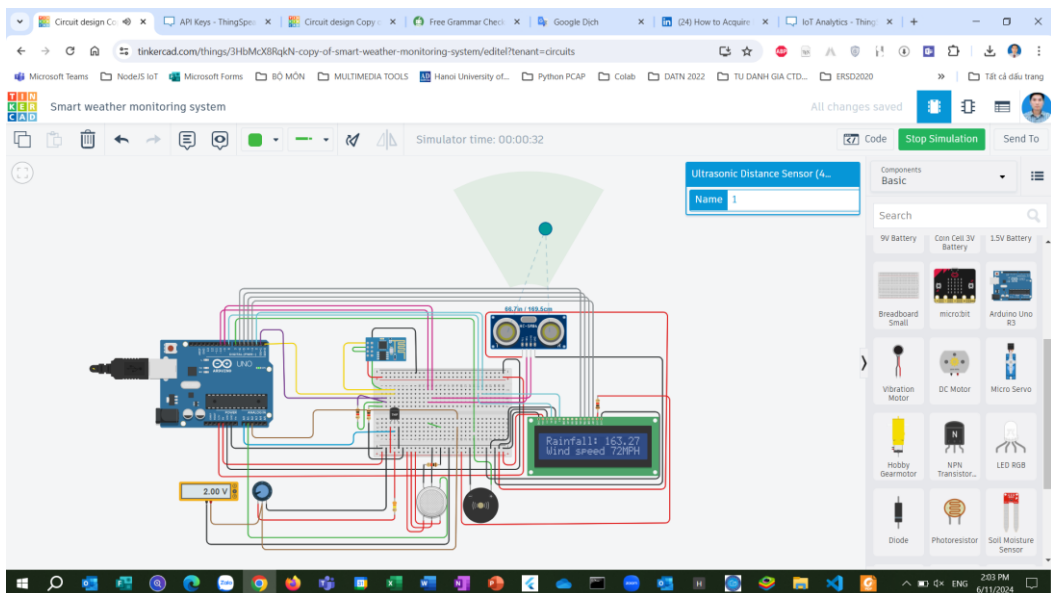
* Module gửi dữ liệu lên thingspeak:

```
//Sends data to Thingspeak
void send_data(void)
{
    // Construct HTTP call
    String httpPacket = "GET " + url + String(temp) + "&field2=" + String(gas_sensor_value) + "&field3=" + String(rain) +
        "&field4=" + String(Windspeedfloat) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
    int length = httpPacket.length();
    // Send our message length
    Serial.print("AT+CIPSEND=");
    Serial.println(length);
    delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;
    // Send our http request
    Serial.print(httpPacket);
    delay(10); // Wait a little for the ESP to respond
    if (Serial.find("SEND OK\r\n"))
        Serial.println("ESP8266 sends data to the server");
}
}
```

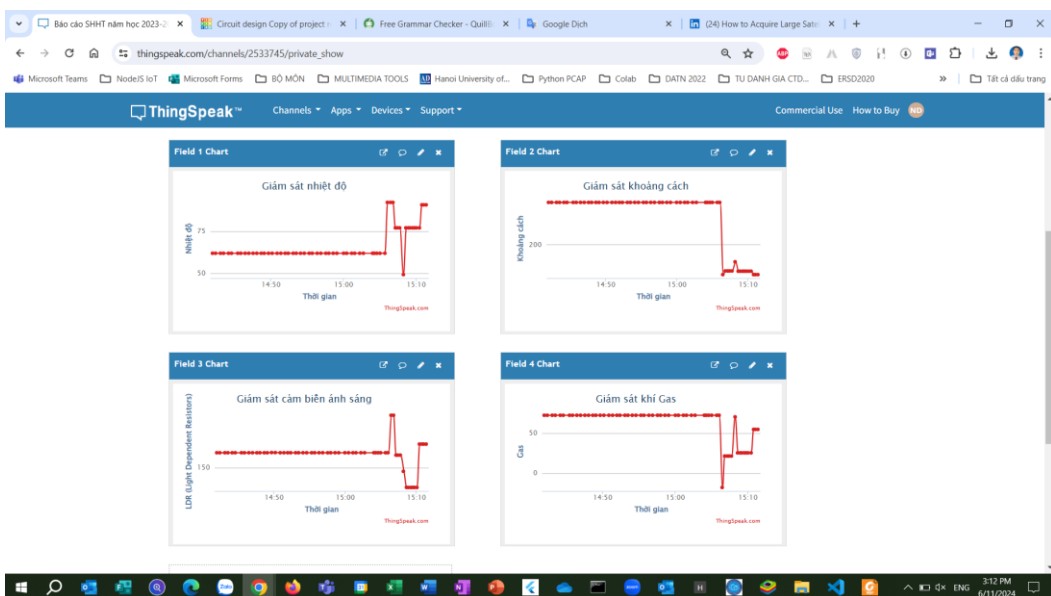
III.5. Mô phỏng mạch trong tinkercad và hiển thị kết quả trên ứng dụng web thingspeak

Sau khi lập trình xong trong tinkercad thì tiến hành mô phỏng mạch. Để mô phỏng nhấn nút **Start Simulation** ở góc trên phải.

Khi mạch hoạt động, quan sát kết quả hiển thị trên RTDB firebase. Để thay đổi giá trị mô phỏng, kích chọn cảm biến và điều chỉnh giá trị, quan sát sự thay đổi trên RTDB firebase cũng như trên màn hình LCD.



Hình III.6 – Mô phỏng mạch trong tinkercad



Hình III.7 – Kết quả mô phỏng mạch trên thingspeak

III.6. Đánh giá

Khi chạy mô phỏng, hệ thống hoạt động ổn định. Khi điều chỉnh kết quả trên các cảm biến nhiệt độ, điện áp, khí gas, khoảng cách, giá trị được cập nhật lên firebase khá nhanh và chính xác. Hạn chế duy nhất của mô phỏng với thiết bị ảo là phải điều chỉnh thủ công giá trị của các cảm biến.

Với việc mô phỏng thành công một số dịch vụ quan trọng của nhà thông minh mở ra cơ hội mô phỏng cho những dịch vụ khác của nhà thông minh cũng như các lĩnh vực khác. Mỗi sinh viên có thể tự mô phỏng hệ thống IoT mà không cần phải mua thiết bị thật. Việc này không đòi hỏi tốn kém chi phí mua thiết bị nên tất cả sinh viên đều có cơ hội mô phỏng các dự án IoT. Để thực hiện các dự án phức tạp hoặc tăng cường kỹ năng làm việc nhóm thì có thể thực hiện theo nhóm thông qua chia sẻ mạch cho các thành viên trong nhóm để trực tiếp điều chỉnh mạch, code...

Bảng III.1 - So sánh hệ thống IoT với thiết bị ảo và thật

TT	Tiêu chí	Hệ thống IoT với thiết bị ảo	Hệ thống IoT với thiết bị thật
1	Thời gian triển khai	Nhanh	Chậm
2	Kinh phí	Không	Tốn kém
3	Kết nối thiết bị thu thập dữ liệu	x	x
4	Kết nối thiết bị điều khiển	x	x
5	Kết nối Internet	x	x
6	Hiển thị kết quả lên webapp, mobile app	x	x
7	Hỗ trợ làm việc cá nhân, nhóm	x	x
8	Triển khai mọi lúc, mọi nơi	x	
9	Phù hợp với học trực tuyến	x	
10	Phù hợp với học trực tiếp	x	x
11	Khả năng áp dụng thực tế	Thấp	Cao
12	Triển khai cho nhiều sinh viên thực hành cùng lúc	Tốt	Hạn chế

KẾT LUẬN

Việc mô phỏng thành công hệ thống IoT cơ bản bằng thiết bị ảo mở ra cơ hội thực hành triển khai hệ thống trong phòng lab trước khi triển khai ngoài thực tế. Hệ thống mô phỏng từ thu thập dữ liệu đến gửi lên web server, hiển thị trên webapp/mobile app còn cho phép điều khiển thiết bị trong mạch. Việc mô phỏng bằng thiết bị ảo giúp người học có thể mô phỏng ở mọi lúc, mọi nơi, có thể thực hiện theo nhóm hoặc cá nhân, đặc biệt không tốn kém chi phí mua thiết bị thật. Đề tài thực sự có ý nghĩa với sinh viên ngành Công nghệ thông tin khi học học phần **“Phát triển ứng dụng IoT”**. Kết quả của đề tài là tài liệu tham khảo hữu ích giúp sinh viên có thể mô phỏng các hệ thống IoT cơ bản bằng các thiết bị ảo, giúp tiết kiệm chi phí và có thể thực hiện mọi lúc, mọi nơi.

Trong thời gian tới, tác giả sẽ tiếp tục nghiên cứu triển khai xây dựng mô phỏng hệ thống IoT ảo trên nền tảng di động. Ngoài ra sẽ nghiên cứu các phần mềm, công cụ mới như nền tảng wokwi, RTDB firebase, NodeJS giúp mô phỏng, phát triển webapp cho các hệ thống IoT với nhiều thiết bị đa dạng hơn nhằm hỗ trợ mô phỏng nhiều loại hệ thống IoT trong thực tế.

TÀI LIỆU THAM KHẢO

- [1]. *Tổng quan về Internet of Things (IoTs)*. Online: <https://tek4.vn/tong-quan-ve-internet-of-things-iots>. Truy cập ngày 03/6/2024.
- [2]. *Tìm hiểu Internet of Things công nghệ tương lai*. Online: <https://tltvietnam.vn/tim-hieu-internet-of-things-cong-nghe-tuong-lai.html>. Truy cập ngày 03/6/2024.
- [3]. Tapit. *Internet of Things: Định nghĩa và kiến trúc*. Online: <https://tapit.vn/internet-things-dinh-nghia-va-kien-truc/>. Truy cập ngày 03/6/2024.
- [4]. *IoT là gì? Kiến trúc hệ thống mạng IoT*. <https://sieuthimang.vn/tin-tuc/IoT-la-gi-Kien-truc-he-thong-mang-IoT.html>. Truy cập ngày 03/6/2024.
- [5]. *Tinkercad – Trình giả lập Arduino và bảng mạch*. Online: <https://www.vniteach.com/2023/01/16/tinkercad-trinh-gia-lap-arduino-va-bang-mach/>. Truy cập ngày 04/6/2024.
- [6]. <https://www.tinkercad.com/>. Truy cập ngày 04/6/2024.
- [7]. <https://thingspeak.com/>. Truy cập ngày 04/6/2024.
- [8]. Đặng Văn Chí, Nguyễn Đức Khoát, Đào Hiếu và Nguyễn Thế Lực, 2020. *Ứng dụng Matlab trong Thingspeak Server để xây dựng hệ thống đo lường và phân tích dữ liệu từ xa bằng công nghệ IoT Gateway*, Tạp chí Khoa học kỹ thuật Mở - Địa chất, số 61, kỳ 5, tr. 88-95.