ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**KHOA VẬT LÝ-VẬT LÝ KỸ THUẬT**

**----------------🙞🙜----------------**

**BÀI TẬP MÔN HỌC**

**PHÁT TRIỂN ỨNG DỤNG INTERNET OF THINGS**

**Chuyên ngành: GIẢNG DẠY VẬT LÝ THỰC NGHIỆM**

**Khoá: K30**

**Họ & tên HV: CAO MINH KHÔI**

**MSHV: 20c39005**

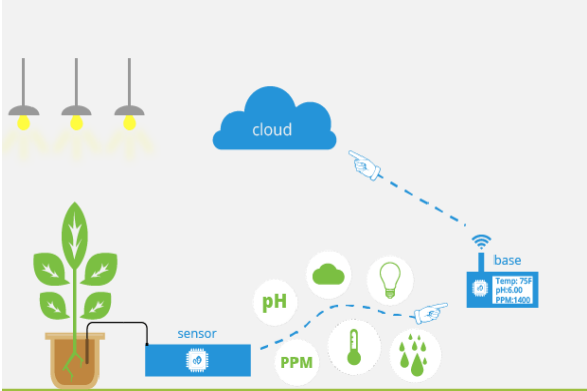
**-------------------------------------------**

**TP. HỒ CHÍ MINH - 07/2021**

**Mục lục**

**Bài tập 1:**

Câu 1: Tìm hiểu về hệ thống IoT ứng dụng trong nông nghiệp thông minh (có hình ảnh minh họa và giải thích các thành phần trong hệ thống).

Từ khi lần đầu được giới thiệu cách đây gần 20 năm, cho tới hiện nay các ứng dụng IOT là một trong những mảng công nghệ phát triển nhất trong cuộc cách mạng công nghiệp 4.0, nó xuất hiện và tác động tích cực tới từng ngành, từng lĩnh vực trong đó có ngành nông nghiệp.Ứng dụng IOT trong nông nghiệp góp phần tạo nên một môi trường sản xuất năng động, khoa học và giải phóng sức lao động, tăng năng suất, mang lại hiệu quả kinh tế cao, giúp nâng cao tính chuyên nghiệp và cải thiện bộ mặt cho cho nền nông nghiệp trong tương lai gần.

Minh họa về ứng dụng IOT trong nông nghiệp.

**Cấu trúc cơ bản của một hệ thống ứng dụng công nghệ IOT**

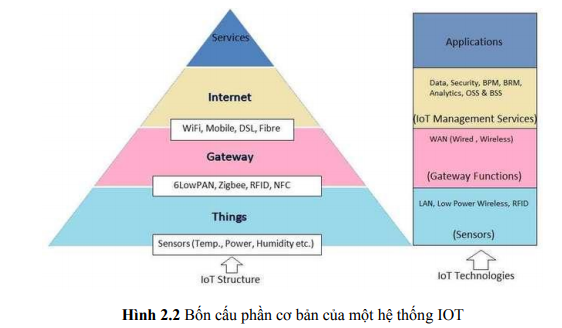
Kiến trúc của IOT gồm bốn thành phần cơ bản chính gồm: Vạn vật (Things), Trạm kết nối (Gateways), Hạ tầng mạng (Internet) và cuối cùng là lớp dịch vụ (Service).

**Vạn vật (Things):** Ngày nay có vô vàn vật dụng đang hiện hữu trong cuộc sống, ở trên các khu canh tác, ở trong nhà hoặc trên chính các thiết bị lưu động của người dùng. Giải pháp IoT giúp các thiết bị thông minh được sàng lọc, kết nối và quản lý dữ liệu của đối tượng nông nghiệp một cách cục bộ, còn các thiết bị chưa thông minh thì có thể kết nối được thông qua các trạm kết nối. Từ đó, các thiết bị, vật dụng sẽ có thể thực hiện nhiệm vụ của mình đối với đối tượng nông nghiệp cần quản lý.

**Trạm kết nối (Gateways):** Các trạm kết nối sẽ đóng vai trò là một vùng trung gian trực tiếp, cho phép các vật dụng có sẵn này kết nối với điện toán đám mây một cách bảo mật và dễ dàng quản lý. Gateways có thể là một thiết bị vật lý hoặc là một phần mềm được dùng để kết nối giữa Cloud (điện toán đám mây) và bộ điều khiển, các cảm biến, các thiết bị thông minh.

**Hạ tầng mạng (Internet):** Internet là một hệ thống toàn cầu của nhiều mạng IP được kết nối với nhau và liên kết với hệ thống máy tính. Cơ sở hạ tầng mạng này bao gồm thiết bị định tuyến, trạm kết nối, thiết bị tổng hợp, thiếp bị lặp và nhiều thiết bị khác có thể kiểm soát lưu lượng dữ liệu lưu thông và cũng được kết nối đến mạng lưới viễn thông và cáp - được triển khai bởi các nhà cung cấp dịch vụ.

**Lớp dịch vụ (Service):** Là các ứng dụng được các hãng công nghệ, hoặc thậm chí người dùng tạo ra để dễ dàng sử dụng các sản phẩm IOT một cách hiệu quả và tận dụng được hết giá trị của sản phẩm.

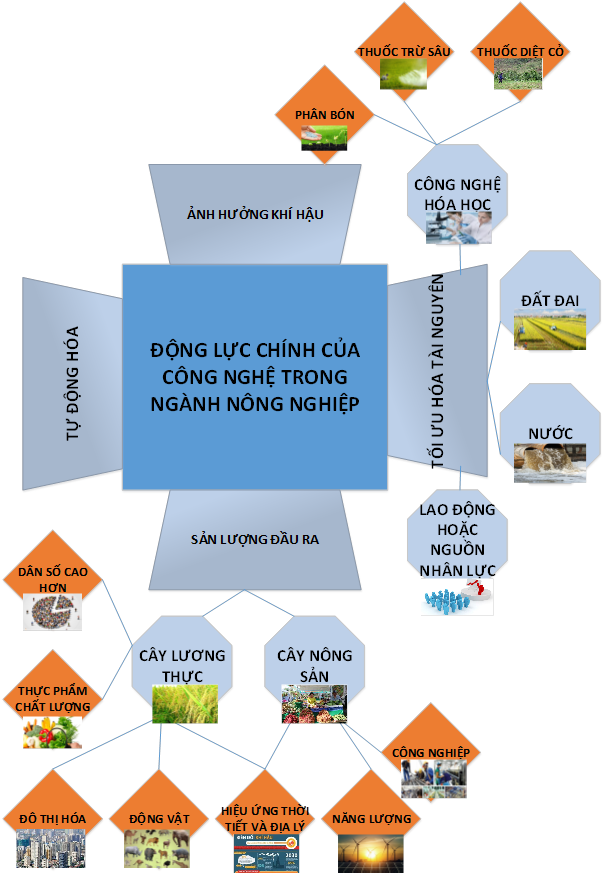


**Tác động của IoT đối với nông nghiệp thông minh**

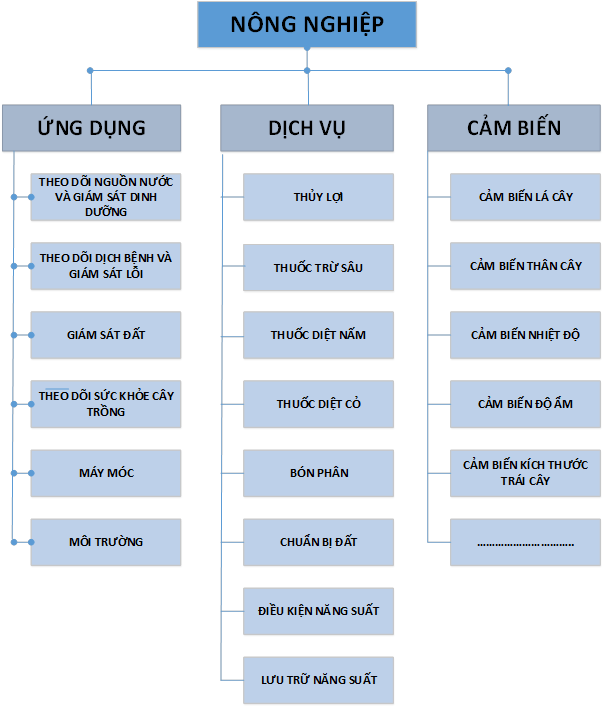
Theo số liệu dự báo, vào năm 2050, dân số thế giới dự kiến sẽ chạm ngưỡng 9.8 tỷ người, tăng khoảng 25% so với con số hiện tại. Xu hướng đô thị hóa sẽ tiếp tục tăng với tốc độ nhanh chóng, khoảng 70% dân số thế giới được dự đoán trở thành dân số thành thị đến năm 2050 (so với hiện nay là 49%). Hơn nữa, mức thu nhập sẽ tăng lên gấp đôi so với hiện tại, điều này thúc đẩy nhu cầu lương thực tăng cao, đặc biệt là ở các nước đang phát triển. Để nuôi sống một lượng dân số thành thị lớn này thì sản lượng lương thực phải tăng gấp đôi vào năm 2050.

Không chỉ đối với lương thực, mà lĩnh vực sản xuất cây trồng cũng trở nên quan trọng không kém đối với ngành công nghiệp. Các loại cây trồng như bông, cao su, đang đóng vai trò quan trọng trong nền kinh tế của nhiều quốc gia. Những nhu cầu này đang làm tăng thêm áp lực đối với các nguồn tài nguyên nông nghiệp vốn đã khan hiếm.

Do những hạn chế về mặt nhiệt độ, khí hậu, địa hình và chất lượng đất không đồng nhất nên khó định lượng sự khác biệt của các loại thực vật; các yếu tố về mô hình đất đai, khí hậu và mật độ dân số tăng nhanh trong quá trình đô thị hóa cũng đặt ra các mối đe dọa đối với sự sẵn có của đất canh tác. Đây là lý do tại sao nông nghiệp thông minh xuất hiện.

Nông nghiệp thông minh là một hệ thống nông nghiệp ứng dụng các công nghệ số để giám sát, điều khiển và chăm sóc cây trồng tự động, giúp tối đa hóa năng suất và chất lượng nông sản.Theo báo cáo của tổ chức Allied Market Research, Thị trường nông nghiệp thông minh toàn cầu sẽ đạt quy mô 48 tỷ USD vào năm 2025, với mức tăng trưởng trung bình 14,7% /năm.

IoT là công nghệ đóng vai trò quan trọng và bắt đầu tác động đến nhiều lĩnh vực và ngành công nghiệp, từ sản xuất, y tế, truyền thông, năng lượng cho đến ngành nông nghiệp. IoT bao gồm cơ sở hạ tầng truyền thông cơ bản được sử dụng để kết nối các đối tượng thông minh từ cảm biến, phương tiện, thiết bị di động đến việc thu thập dữ liệu từ xa dựa trên phân tích thông minh, giao tiếp người dùng và cách mạng hóa ngành nông nghiệp.

Bằng cách triển khai các công nghệ cảm biến và IoT trong thực tiễn nông nghiệp đã làm thay đổi mọi khía cạnh của phương pháp canh tác truyền thống. IoT giúp cải thiện các giải pháp về canh tác truyền thống như ứng phó với hạn hán, tối ưu hóa năng suất, tính phù hợp đất đai, tưới tiêu và kiểm soát dịch hại.

*Hệ thống phân cấp chung của các ứng dụng, dịch vụ và cảm biến có thể có cho nông nghiệp thông minh*

**1. Lấy mẫu và bản đồ đất**

Đất là “Dạ dày” của thực vật, việc lấy mẫu đất là bước đầu tiên để kiểm tra thông tin thực địa để đưa ra mục đích sử dụng đất ở các giai đoạn khác nhau (như mùa xuân, hạ, thu, đông). Mục tiêu của việc phân tích đất là xác định tình trạng dinh dưỡng của ruộng để thực hiện các biện pháp phù hợp khi phát hiện thấy thiếu dinh dưỡng.

Hiện nay, các nhà sản xuất đã cung cấp một loạt các bộ công cụ và cảm biến có thể hỗ trợ nông dân theo dõi chất lượng đất và dựa trên dữ liệu này để đề xuất các biện pháp khắc phục tình trạng suy thoái đất. Các bộ công cụ này cho phép theo dõi các đặc tính của đất, chẳng hạn như kết cấu, khả năng giữ nước và tỷ lệ hấp thụ để giúp giảm thiểu xói mòn, dày đặc, nhiễm mặn, axit hóa và ô nhiễm bằng cách tránh sử dụng quá nhiều phân bón. Labin-a-Box, một bộ công cụ kiểm tra đất do AgroCares phát triển có thể phân tích tới 100 mẫu mỗi ngày (hơn 22.000 mẫu dinh dưỡng mỗi năm) mà không cần đến bất kỳ phòng thí nghiệm nào.

Hạn hán là mối quan tâm lớn làm hạn chế năng suất cây trồng. Đặc biệt là ở các vùng nông thôn, viễn thám được sử dụng để thu thập dữ liệu về độ ẩm của đất, giúp phân tích tình trạng hạn hán nông nghiệp. Với mục đích này, vệ tinh Độ ẩm của đất và độ mặn của đại dương SMOS (Soil Moisture and Ocean Salinity) đã được phóng vào năm 2009 để cung cấp bản đồ độ ẩm của đất sau 1-2 ngày.

**2. Thủy lợi**

Khoảng 97% nước trên trái đấy là nước mặn được lưu giữ bởi đại dương và biển, chỉ còn 3% còn lại là nước ngọt (2/3 trong số lượng nước ngọt này bị đóng băng dưới sông băng, chỉ có 0.5% lượng nước ngọt chưa đóng băng ở trên mặt đất hoặc trong không khí, phần còn lại nằm dưới lòng đất). Như vậy, con người dựa vào 0.5% lượng nước ngọt này để sử dụng và duy trì hệ sinh thái. Điều đáng nói là chỉ riêng ngành nông nghiệp đã sử dụng đến 70% lượng nước ngọt, ở một số nước còn tăng lên 75%. Theo ước tính của Công ước Liên hợp quốc về chống sa mạc hóa UNCCD (UN Convention to Combat Desertification) năm 2013 cho thấy có 168 quốc gia bị ảnh hưởng bởi sa mạc hóa và đến năm 2030, gần một nửa dân số thế giới sẽ sống ở những khu vực thiếu nước cao. Vì vậy, việc bảo tồn nguồn nước hiện có dựa trên việc sử dụng hệ thống tưới tiêu hiệu quả.

Các phương pháp tưới tiêu truyền thống được sử dụng như nhỏ giọt và tưới phun không đều dẫn đến cây trồng bị thiếu nước, giảm chất dinh dưỡng trong đất, gây ra các bệnh vi sinh. Vì vậy, việc sử dụng các công nghệ kỹ thuật dựa trên IoT như: các cảm biến không dây, hệ thống giám sát và các ứng dụng phần mềm thông minh để thu thập, phân tích, kiểm tra độ ẩm đất và không khí sẽ tối ưu hóa được lượng nước, ước tính chính xác nhu cầu nước của cây trồng và khả năng giữ ẩm của đất tại từng thời điểm trong năm, cải thiện hiệu quả sử dụng nước.

**3. Phân bón**

Phân bón là một hóa chất tự nhiên cung cấp các chất dinh dưỡng quan trọng cho sự phát triển và khả năng sinh sản của cây trồng. Thực vật chủ yếu cần ba chất dinh dưỡng chính là:

- Nitơ (N) cho sự phát triển của lá;

- Lân (P) cho sự phát triển của rễ, hoa và quả;

- Kali (K) cho sự phát triển của thân cây và sự di chuyển của nước.

Bất kỳ loại chất dinh dưỡng nào hoặc bón phân không đúng cách đều có thể gây hại nghiêm trọng cho sức khỏe cây trồng. Quan trọng hơn, việc sử dụng quá nhiều phân bón không chỉ gây thiệt hại về tài chính mà còn tạo ra các tác động có hại cho đất và môi trường, làm suy giảm chất lượng đất, đầu độc nguồn nước ngầm và góp phần làm biến đổi khí hậu toàn cầu.

Bón phân trong nông nghiệp thông minh giúp ước tính chính xác liều lượng chất dinh dưỡng cần thiết, giảm thiểu tác động tiêu cực đối với môi trường. Bón phân đòi hỏi các phép đo mức độ dinh dưỡng của đất cụ thể dựa trên các yếu tố khác nhau, chẳng hạn như loại cây trồng, loại đất, khả năng hấp thụ của đất, năng suất sản phẩm, loại màu mỡ, tỷ lệ sử dụng và điều kiện thời tiết,... Các phương pháp bón phân dựa trên IoT mới giúp ước tính các mô hình không gian về nhu cầu của chất dinh dưỡng với độ chính xác cao hơn.

**4. Quản lý bệnh cây trồng**

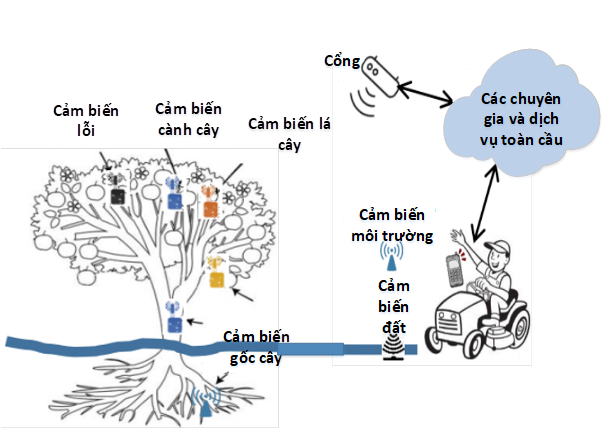
Nạn đói lớn, còn được gọi là Nạn đói khoai tây Ailen (Irish Potato Famine), đã làm cho một triệu người Ailen chết vào khoảng năm 1950, do mất mùa và giảm năng suất do bệnh. Thậm chí ngày nay, những người trồng ngô ở Hoa Kỳ và miền nam Canada đang phải đối mặt với thiệt hại kinh tế xấp xỉ một tỷ USD do bệnh cháy lá ngô ở miền Bắc. Tổ chức Nông lương FAO (Food and Agriculture Organization) ước tính rằng 20–40% sản lượng cây trồng toàn cầu bị mất hàng năm do sâu bệnh. Để kiểm soát thiệt hại, thuốc trừ sâu và các hóa chất nông nghiệp khác đã trở thành một thành phần quan trọng của ngành nông nghiệp trong thế kỷ trước. Người ta ước tính rằng, mỗi năm, chỉ riêng ở Hoa Kỳ có khoảng nửa triệu tấn thuốc trừ sâu được sử dụng, trong khi hơn hai triệu tấn được sử dụng trên toàn cầu. Hầu hết các loại thuốc trừ sâu đều có hại cho sức khỏe của con người và động vật, để lại tác động nghiêm trọng, thậm chí không thể phục hồi đối với môi trường, cuối cùng gây ô nhiễm đáng kể cho toàn bộ hệ sinh thái.

Các thiết bị thông minh dựa trên IoT như cảm biến không dây, robot và máy bay không người lái cho phép giảm đáng kể việc sử dụng thuốc trừ sâu bằng cách phát hiện chính xác sâu bệnh của cây trồng. So với quy trình kiểm soát dịch hại dựa trên truyền thống thì quản lý dịch hại dựa trên IoT cung cấp khả năng giám sát, mô hình hóa, dự báo dịch bệnh theo thời gian thực hiệu quả hơn. Các phương pháp tiếp cận nhận dịch bệnh và sâu bệnh tiên tiến dựa trên việc xử lý hình ảnh, trong đó các hình ảnh thô được thu thập bằng cách sử dụng cảm biến hiện trường, máy bay không người lái UAV (Unmanned Aerial Vehicle) hoặc vệ tinh viễn thám. Ví dụ: bẫy tự động dựa trên IoT có thể bắt, đếm và thậm chí xác định đặc điểm của các loại côn trùng, sau đó đẩy dữ liệu về sâu bệnh lên đám mây.

Các phương pháp tiếp cận như phun thuốc chính xác bằng xe và hóa chất VRT tự động được sử dụng theo phương pháp bón phân thông minh, cũng có thể được sử dụng để điều trị bệnh và các ứng dụng thuốc trừ sâu khác. Hơn nữa, sự tiến bộ của công nghệ robot mang đến những giải pháp mới. Khi trang bị cho một robot nông nghiệp với các thiết bị cảm biến đa hình ảnh và vòi phun chính xác, có thể xác định vị trí và phương pháp đối phó các vấn đề dịch hại chính xác hơn dưới sự điều khiển của hệ thống quản lý dịch bệnh IoT từ xa.

**5. Theo dõi, dự báo và thu hoạch**

Giám sát năng suất là cơ chế được sử dụng để phân tích các khía cạnh khác nhau tương ứng với năng suất nông nghiệp, như lưu lượng khối lượng hạt, độ ẩm và lượng hạt thu hoạch. Giám sát giúp đánh giá chính xác bằng cách ghi lại năng suất cây trồng và độ ẩm để ước tính cây trồng hoạt động tốt như thế nào và những việc gì cần làm tiếp theo. Giám sát năng suất được coi là một phần thiết yếu của canh tác chính xác không chỉ vào thời điểm thu hoạch mà cả chất lượng sản lượng trước đó. Chất lượng sản lượng phụ thuộc vào nhiều yếu tố, ví dụ: thụ phấn đủ chất lượng trong điều kiện môi trường thay đổi.

Dự báo mùa màng là một nghệ thuật để dự đoán năng suất và sản lượng (tấn / ha) trước khi vụ thu hoạch diễn ra. Dự báo này giúp người nông dân lập kế hoạch và ra quyết định về chất lượng sản phẩm để xác định thời điểm thu hoạch thích hợp. Việc giám sát này bao gồm các giai đoạn phát triển khác nhau. Dự đoán đúng thời điểm thu hoạch không chỉ giúp tối đa hóa sản lượng và chất lượng cây trồng mà còn tạo cơ hội điều chỉnh chiến lược quản lý. Để thu được lợi ích từ cây trồng, nông dân cần biết khi nào những cây trồng này được thu hoạch. Hình 4 thể hiện ảnh chụp nhanh của mạng khu vực nông trại (FAN) có thể hiển thị toàn bộ trang trại của người nông dân.

*Mạng khu vực trang trại dựa trên IoT FAN (Farm Area Network)*

Câu 2: Tìm hiểu về hệ thống IoT ứng dụng trong thành phố thông minh (có hình ảnh minh họa và giải thích các thành phần trong hệ thống).

Khi [Internet vạn vật](https://ictnews.vietnamnet.vn/internet-van-vat-tag24979.html)(IoT) bắt đầu thâm nhập vào các lĩnh vực của cuộc sống, chúng ta bắt đầu được chứng kiến sự trỗi dậy của ý tưởng thành phố thông minh.

Những thành phố kiểu này bắt đầu ứng dụng các công nghệ IoT như cảm biến và các bộ phát tín hiệu không dây để thu thập dữ liệu, qua đó quản lý các dịch vụ công, giao thông đô thị tốt hơn. Mục đích cuối cùng là biến các thành phố trở thành nơi đáng sống và an toàn.

Tuy vậy, công nghệ thành phố thông minh vẫn là một cái gì đó quá xa vời với những người ngoài ngành, hoặc có thể nó đang có những tác động nhất định đến một số mặt của đời sống mà nhiều người không nhận ra.

Dưới đây là 14 công nghệ nổi bật đã, đang và sẽ thay đổi bộ mặt của các siêu đô thị trên thế giới:



**Bảo mật công nghệ vận hành**

Công nghệ vận hành (Operational Technology) là một sự kết hợp giữa phần cứng lẫn phần mềm dùng để giám sát và điều khiển cơ sở hạ tầng, nhà máy lắp ráp trang thiết bị. Ngành này được ứng dụng rộng rãi trong công nghiệp sản xuất tự động hoặc bán tự động.

Tuy nhiên, với sự phát triển của mạng 5G cùng các thiết bị thông minh kết nối tốc độ cao, công nghệ vận hành đã không ngừng cải tiến với các giải pháp quản lý từ xa, máy học tự vận hành, phân tích dữ liệu thời gian thực.

Chính vì thế, bảo mật công nghệ vận hành nổi lên như một lĩnh vực quan trọng để bảo vệ thông tin, tài sản, thiết bị và toàn bộ những thứ liên quan đến nó bằng hệ thống siêu tường lửa và quản lý định danh.

**Giám sát giao thông**

Khi mạng 5G có mặt trên mọi thiết bị thông minh, bao gồm trên ôtô, camera ngoài đường, nó sẽ thúc đẩy sự phát triển không ngừng của hệ thống giám sát giao thông.

Những giám sát thời gian thực về tình hình giao thông sẽ liên tục được gửi đi giúp người tham gia giao thông quyết định chính xác tuyến đường nào nên đi và tuyến đường nào cần tránh.

**Mạng 5G**

Một công nghệ cốt lõi, kết nối vạn vật không thể không nhắc đến là 5G. Mạng di động thế hệ thứ 5 có khả năng kết nối hàng triệu thiết bị với tốc độ có thể lên tới 10 Gbps, nhanh gấp 100 lần tốc độ 4G hiện nay.

Kết nối tốc độ nhanh với độ trễ thấp cũng sẽ giúp hệ thống giao thông công cộng và xe tự hành vận hành trơn tru. Công nghệ thực tế ảo và đám mây cũng sẽ được hưởng lợi khi người dùng có thể trải nghiệm dịch vụ không gián đoạn ở bất kỳ đâu, trong bất kỳ hoàn cảnh nào.

**Cơ sở hạ tầng cho LPWAN**

Phục vụ hàng triệu người và quản lý cơ sở hạ tầng phức tạp đòi hỏi một kết nối liên tục, điều khó có thể đáp ứng trong bối cảnh đứt cáp quang biển triền miên như hiện nay.

Trong bối cảnh đó, mạng diện rộng không dây năng lượng thấp (LPWAN) và 5G nổi lên như một chiếc chìa khóa mở toang cánh cổng dẫn tới tương lai cho các thành phố thông minh. Ở đó, 5G là dòng chảy công nghệ đưa dữ liệu đến mọi người dùng còn LPWAN là suối nguồn khổng lồ để duy trì kết nối toàn thành phố.

**Giám sát chất lượng không khí**

Ô nhiễm môi trường ở các thành phố có mật độ giao thông cao và nhiều rác thải là vấn đề không mới, dù nó vẫn đang gây ra những mối nguy hiểm tiềm tàng đến sức khỏe của người dân.

Do đó, nhiều thành phố giờ đây đã có hệ thống đo đạc giám sát chất lượng không khí, nhằm đưa ra những cảnh báo sớm cho người dân. Với dữ liệu cục bộ, các cảm biến IoT có thể giúp xác định các nguồn ô nhiễm không khí từ đó kết hợp với AI để đưa ra các giải pháp thông minh như quy hoạch đô thị, phân luồng giao thông.

**Mô phỏng và mô hình hóa đám đông**

Mô phỏng và mô hình hóa đám đông là một công nghệ máy học rất mới mẻ, nhằm giúp các nhà hoạch định chính sách hiểu rõ hơn phản ứng xã hội về việc phân vùng, quản lý giao thông, và các mục tiêu phát triển kinh tế.

Đây mới chính là chiếc chìa khóa cốt lõi trong việc xây dựng thành phố thông minh, bắt đầu từ móng. Khám phá tiềm năng tương lai của số hóa qua mô phỏng trước được xem là cốt lõi của vận hành hoạt động đô thị thông minh.

**Công nghệ không tiếp xúc**

Bất cứ cái gì không tạo ra tiếp xúc giữa người với nhau đều được xem là công nghệ không chạm. Nhất là trong bối cảnh Covid-19 như hiện nay, những thứ như thanh toán quét mã không dùng thẻ, chẩn đoán từ xa sẽ giúp tiết kiệm tối đa thời gian và chi phí, cũng như ngăn ngừa sự lây lan của dịch bệnh này.

**Camera thông minh**

Camera thông minh là thứ nổi bật nhất ở công nghệ thành phố thông minh. Nhưng ở đây nó không chỉ đơn thuần là các camera giao thông mà là một tập hợp của tất cả các loại camera từ camera trước trên xe, camera nhà dân đến camera gắn trên người thi hành công vụ.

Tập hợp thông tin từ tất cả các camera này có thể giúp xác định tai nạn, ổ gà, các vụ cháy nổ và nhiều thứ khác nữa. Camera thông minh sẽ là một bước tiến dài cho các thành phố thông minh thông qua ứng dụng dữ liệu lớn, máy học và trí tuệ nhân tạo.

**Hệ thống bảo mật di động**

Công nghệ thành phố thông minh đồng nghĩa với ứng dụng phải an toàn và thuận tiện trên các thiết bị di động. Đây có thể là miếng mồi ngon cho các loại tội phạm công nghệ cao xâm nhập và tấn công vào thiết bị người dùng.

Vì thế, bảo mật di động cần phải được đặt ra và ưu tiên hàng đầu, tích hợp với hệ thống bảo mật tổng thể của thành phố thông minh.

**Cảnh báo biến động**

Dự báo thời tiết là một dạng của cảnh báo biến đổi. Nhưng các thành phố thông minh còn cần những thông tin cụ thể, chính xác và theo thời gian thực tốt hơn thế nữa.

Các cảnh báo này không chỉ hạn chế ở thời tiết mà còn phủ rộng sang cảnh báo ô nhiễm không khí cho bệnh nhân gặp vấn đề về hô hấp dựa trên hồ sơ bệnh án online, cảnh báo tai nạn phía trước, cảnh báo tiếp xúc F1, F2 hoặc bệnh nhân Covid-19 và các loại cảnh báo khác.

**Cảm biến và AI tiên tiến**

Sự kết hợp giữa trí tuệ nhân tạo và cảm biến sẽ giúp thành phố thông minh phản ứng tốt hơn trước những loại hình thời tiết dị thường và các vụ tấn công không gian mạng. Mô hình thành phố thông minh sẽ ngày càng phụ thuộc nhiều hơn vào dữ liệu của cảm biến thời gian thực và AI tiên tiến để dự đoán hoạt động trong tương lai và phản ứng với những bất thường.

**Nông nghiệp đô thị**

Thành phố thông minh cũng cần nông nghiệp đô thị do sự gia tăng dân số tại các thành phố này. Để giải quyết bài toán thiếu lương thực, cải thiện chất lượng cuộc sống và giảm chi phí vận chuyển, nông nghiệp đô thị đang là giải pháp cấp bách hàng đầu.

Hiện nay, Singapore đang là nước đi tiên phong trong việc ứng dụng nông nghiệp chiều đứng với các nhà trồng rau khép kín nhiều tầng để tự cung cấp lương thực tại chỗ trong khi giải quyết được bài toán quỹ đất hạn hẹp ở quốc đảo này.

**Drone tự động**

Việc sử dụng drone đang gia tăng nhanh chóng, nhất là trong bối cảnh hạn chế tiếp xúc vì Covid-19 như hiện nay. Amazon đã cung cấp dịch vụ giao hàng tận nhà thông qua drone, trong khi nông dân Mỹ bắt đầu sử dụng drone để làm nông nghiệp thông minh.

Không chỉ vì giãn cách xã hội, trong tương lai nhu cầu của drone sẽ còn tăng cao hơn nữa nếu nó có khả năng tự động hóa, giúp cung cấp nhu yếu phẩm ngay lập tức mà không cần sự hiện diện của con người.

**Giao thông thông minh**

Công nghệ giao thông thông minh sẽ giúp các thành phố phân luồng tốt hơn, tối ưu khoảng cách giữa các xe nhờ hệ thống đèn tín hiệu thông minh. Với tương lai của xe tự hành, các thiết bị thông minh kết nối với nhau có thể tạo ra giải pháp an toàn khi đi đường, tránh ùn tắc giao thông mà vẫn đảm bảo di chuyển với tốc độ cao.

Câu 3: Tìm hiểu về hệ thống IoT ứng dụng trong nhà thông minh (có hình ảnh minh họa và giải thích các thành phần trong hệ thống).

Trong cuộc sống con người luôn luôn đặt ra những câu hỏi, luôn có sự tò mò với những điều mới mẻ. Và chính sự tò mò ấy đôi khi lại đem lại những ý tưởng không ngờ. Ai trong số chúng ta từng nghĩ tới một cuộc sống nơi mà các thiết bị và máy móc điện tử có thể trở thành người bạn tốt nhất với con người. Giờ đây, điều đó đã dần trở thành hiện thực khi công nghệ **IoT** xuất hiện, dần ‘xâm chiếm” toàn bộ thế giới. [Ứng dụng của IoT](https://bkaii.com.vn/tin-tuc/167-61-y-tuong-iot-khoi-nghiep-gianh-cho-cac-ky-su-dien) gắn liền với cuộc sống của mỗi chúng ta nhất phải kể đến những mô hình nhà thông minh. Hôm nay BKAII sẽ cùng các bạn tìm hiểu về sự có mặt của IoT trong những smart home nhé!

**Smart home**có thể nói là ứng dụng của **IoT** gây được tiếng vang lớn trên toàn cầu và được tìm kiếm nhiều nhất trên Internet với ứng dụng **IoT**. Chỉ cần tưởng tượng khi bạn không có ở nhà mà vẫn có thể tắt các thiết bị trong nhà, mở khóa cho bạn bè vào nhà hay theo dõi các tài sản của bạn từ xa thông qua hệ thống camera, như vậy phần nào cho ta thấy được những lợi ích đáng kể của những ngôi nhà thông minh. Những ngôi nhà thông minh sẽ giúp chúng ta tiết kiệm được thời gian và những nỗ lực cũng như luôn tạo ra một sự yên tâm về độ an toàn.



**Smart home** đang mang lại những lợi ích đáng kể cho cuộc sống con người tuy nhiên để sở hữu được một ngôi nhà thông minh lại tốn rất nhiều chi phí. Hơn nữa các thiết bị được sử dụng trong nhà thông minh cũng có giá thành vô cùng đắt đỏ do việc cái tiến công nghệ, tính năng sản phẩm cũng như quá trình sản xuất khá tốn kém. Nếu vài năm trước đây người dùng còn e dè trước những chiếc smart phone hay giá thành của những chiếc điện thoại thông minh cũng không hề nhỏ thì giờ đây smart phone lại trở thành một thiết bị vô cùng quen thuộc và không thể thiếu với mỗi người và chi phí cho những chiếc điện thoại thông minh cũng được giảm rất nhiều phù hợp với từng đối tượng người dùng. **Smart home** trong tương lại không xa chắc chắc cũng sẽ trở nên quen thuộc như những chiếc smart phone bây giờ.

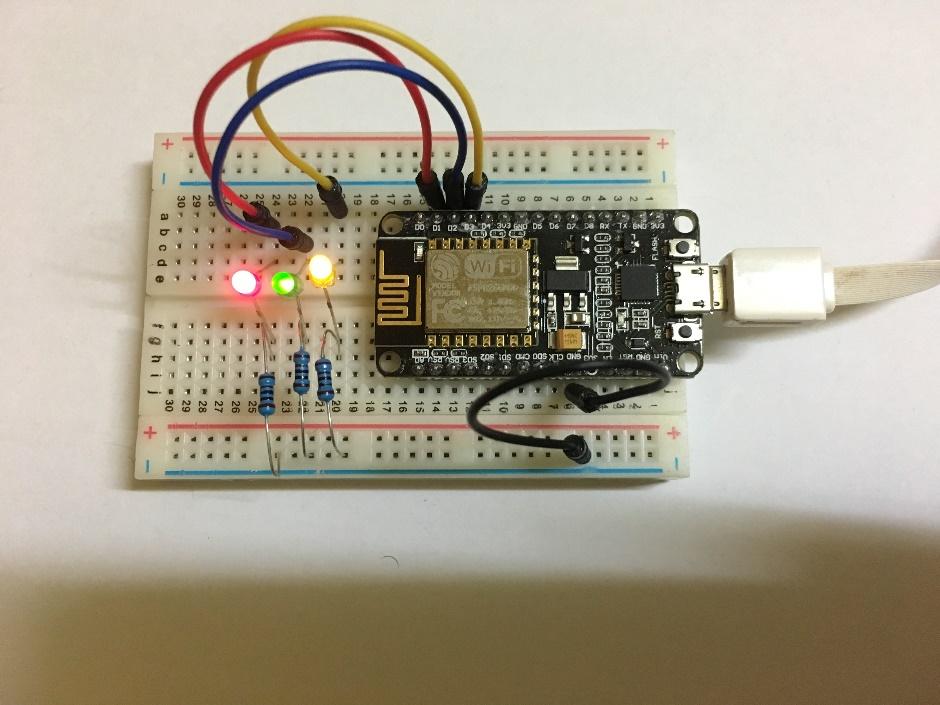
Vậy**IoT** được ứng dụng trong những ngôi nhà thông minh như thế nào. Dưới đây là một số ứng dụng phổ biến nhất khi áp dụng **IoT** trong **smart home**.

**Hệ thống chiếu sáng.**

Một hệ thống chiếu sang thông minh sẽ giúp bạn tiết kiệm được rất nhiều thời gian và công sức. Một hệ thống đền LED giúp cho bạn có thể tắt hay bật bất cứ chiếc đèn nào trong bất cứ căn phòng nào mà không cần di chuyển đến một bước chân, hay thậm chí khí bạn đi ra ngoài mà quên tắt đèn ta hoàn toàn có thể điều khiển các thiết bị chiếu sang này từ xa. Trong một ngôi nhà thông minh có những vùng sáng sẽ được cố định thời gian bật tắt, ví dụ như ngoài cổng hay sân vườn, bạn hoàn toàn có thể cài đặt chế độ hẹn giờ tự hoạt động cho các thiết bị mà không cần quan tâm đến chúng. Như vậy ta có thể thoải mái để thực hiện các công việc khác mà không lo sân vườn hay cổng ra vào không đủ ánh sáng vào buổi tối khi bạn quên không điều khiển chúng.

**Bài tập 2:**

Câu 1: Lắp mạch điện và viết chương trình điều khiển 3 đèn LED sáng tắt từ trái qua phải.



Code:

void setup() {

pinMode(D2, OUTPUT);

pinMode(D3, OUTPUT);

pinMode(D4, OUTPUT);

}

void loop() {

digitalWrite(D2, LOW);

delay(200);

digitalWrite(D2, HIGH);

delay(200);

digitalWrite(D3, LOW);

delay(200);

digitalWrite(D3, HIGH);

delay(200);

digitalWrite(D4, LOW);

delay(200);

digitalWrite(D4, HIGH);

delay(200);

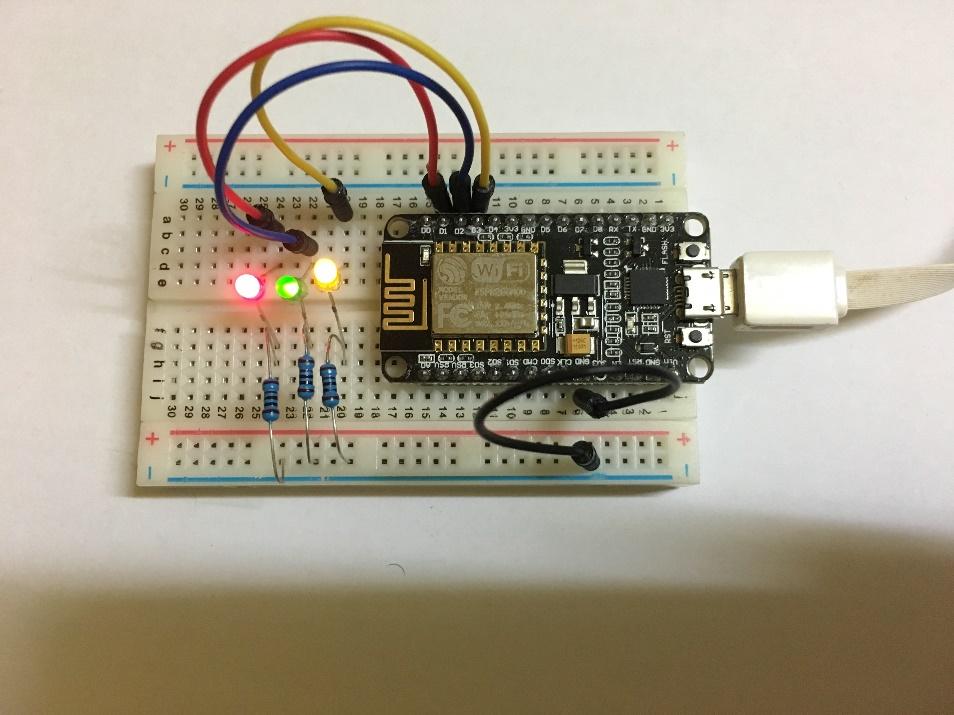
}

Câu 2: Lắp mạch điện (tương tự câu 1) và viết chương trình điều khiển đèn giao thông như sau:

- Đèn xanh: sáng 5 giây

- Đèn vàng: nhấp nháy luân phiên 3 lần mỗi lần 0,5 giây

- Đèn đỏ: sáng 5 giây



Code:

#define redled D2

#define yellowled D3

#define greenled D4

void setup() {

pinMode(redled, OUTPUT);

pinMode(yellowled, OUTPUT);

pinMode(greenled, OUTPUT);

}

void loop() {

digitalWrite(greenled, HIGH);

delay(5000); //Đèn xanh sáng 5s

digitalWrite(greenled, LOW);

for(int i=0;i<3;i++)

{

delay(500);//delay 0.5 giây

digitalWrite(yellowled, HIGH);

delay(500);//delay 0.5 giây

digitalWrite(yellowled, LOW);

}

delay(500);//delay 0.5 giây

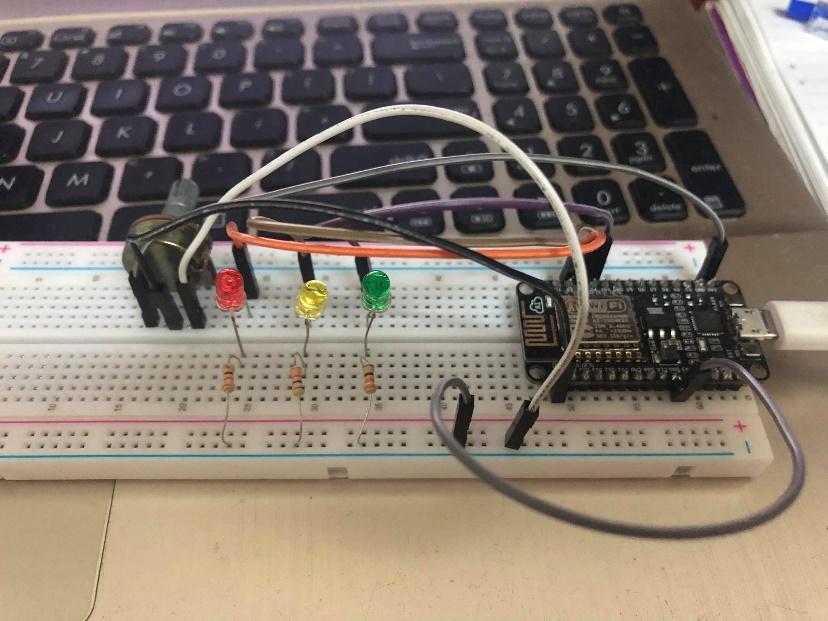
digitalWrite(redled, HIGH);

delay(5000);//Đèn đỏ sáng 5s

digitalWrite(redled, LOW);

}

Câu 3: Lắp mạch điện và viết chương trình điều khiển độ sáng của 3 led khi điều chỉnh biến trở (10 kΩ).



#define ledPin2 D2

#define ledPin3 D3

#define ledPin4 D4

#define analogPin A0

int analogVal;

void setup() {

// put your setup code here, to run once:

pinMode(ledPin2,OUTPUT);

pinMode(ledPin3,OUTPUT);

pinMode(ledPin4,OUTPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

analogVal=analogRead(analogPin);

Serial.println(analogVal);

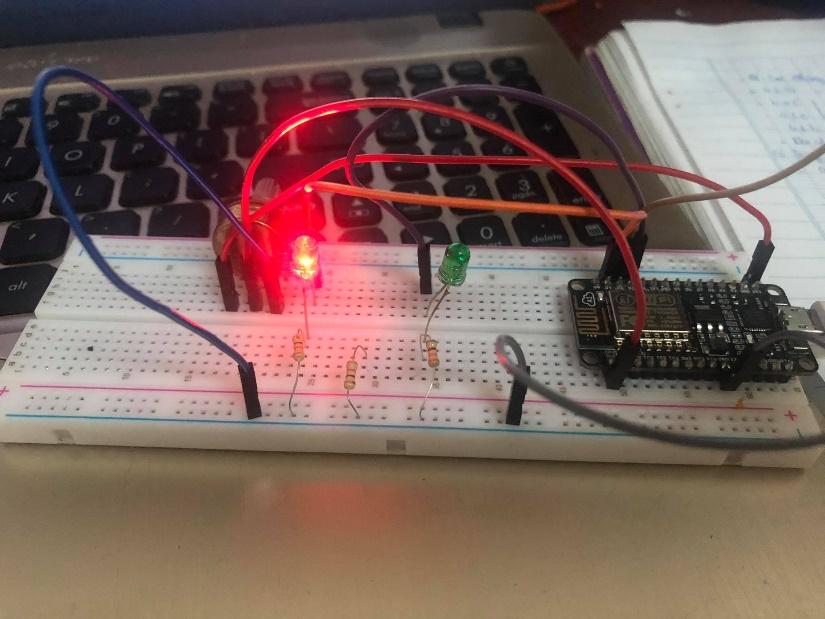
analogWrite(ledPin2,analogVal);

analogWrite(ledPin3,analogVal);

analogWrite(ledPin4,analogVal);

}

Câu 4: Lắp mạch điện và viết chương trình đọc giá trị biến trở (10 kΩ) chuyển đổi thành giá trị số 8 bit (0 -> 255). Kiểm tra giá trị chuyển đổi nếu lớn hơn 100 thì bật led đỏ sáng, ngược lại thì bật led xanh sáng.



#define ledPin2 D2

#define ledPin4 D4

#define analogPin A0

int analogVal,digitalVal;

void setup() {

// put your setup code here, to run once:

pinMode(ledPin2,OUTPUT);

pinMode(ledPin4,OUTPUT);

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

analogVal=analogRead(analogPin);

digitalVal=map(analogVal,0,1023,0,255);

Serial.println(digitalVal);

if(digitalVal>100){

digitalWrite(ledPin2,HIGH);

digitalWrite(ledPin4,LOW);

}

else {

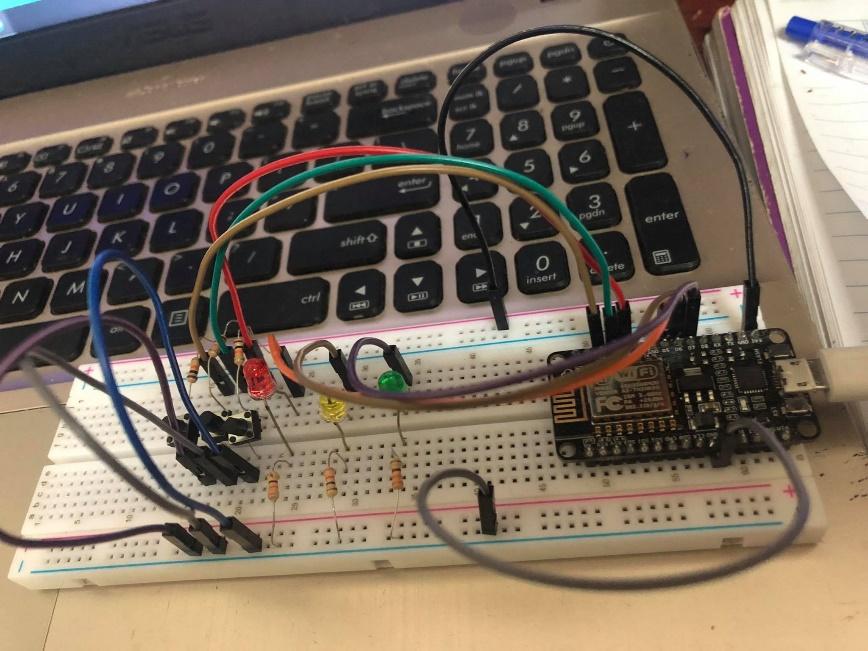
digitalWrite(ledPin2,LOW);

digitalWrite(ledPin4,HIGH);

}

}

Câu 5: Lắp mạch điện và viết chương điều khiển 3 nút nhấn để tắt mở 3 led tương ứng.



int ledPin1=D5;

int ledPin2=D6;

int ledPin3=D7;

int sw1=D2;

int sw2=D3;

int sw3=D4;

int switch1,switch2,switch3;

void setup() {

pinMode(ledPin1,OUTPUT);

pinMode(ledPin2,OUTPUT);

pinMode(ledPin3,OUTPUT);

pinMode(sw1,INPUT);

pinMode(sw2,INPUT);

pinMode(sw3,INPUT);

}

void loop() {

switch1=digitalRead(sw1);

if(switch1==HIGH){

digitalWrite(ledPin1,LOW);

}

else{ digitalWrite(ledPin1,HIGH);

}

switch2=digitalRead(sw2);

if(switch2==HIGH){

digitalWrite(ledPin2,LOW);

}

else{ digitalWrite(ledPin2,HIGH);

}

switch3=digitalRead(sw3);

if(switch3==HIGH){

digitalWrite(ledPin3,LOW);

}

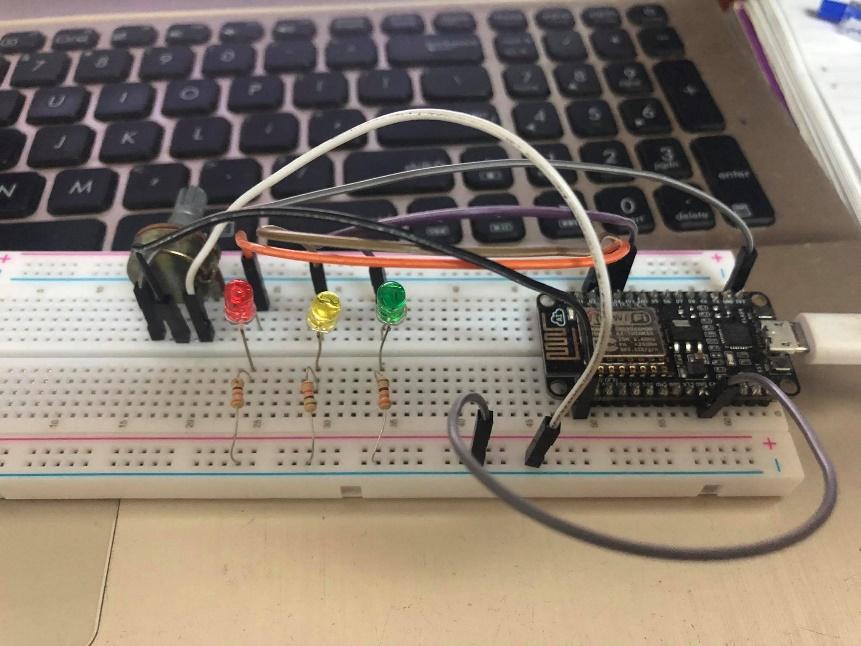
else{ digitalWrite(ledPin3,HIGH);

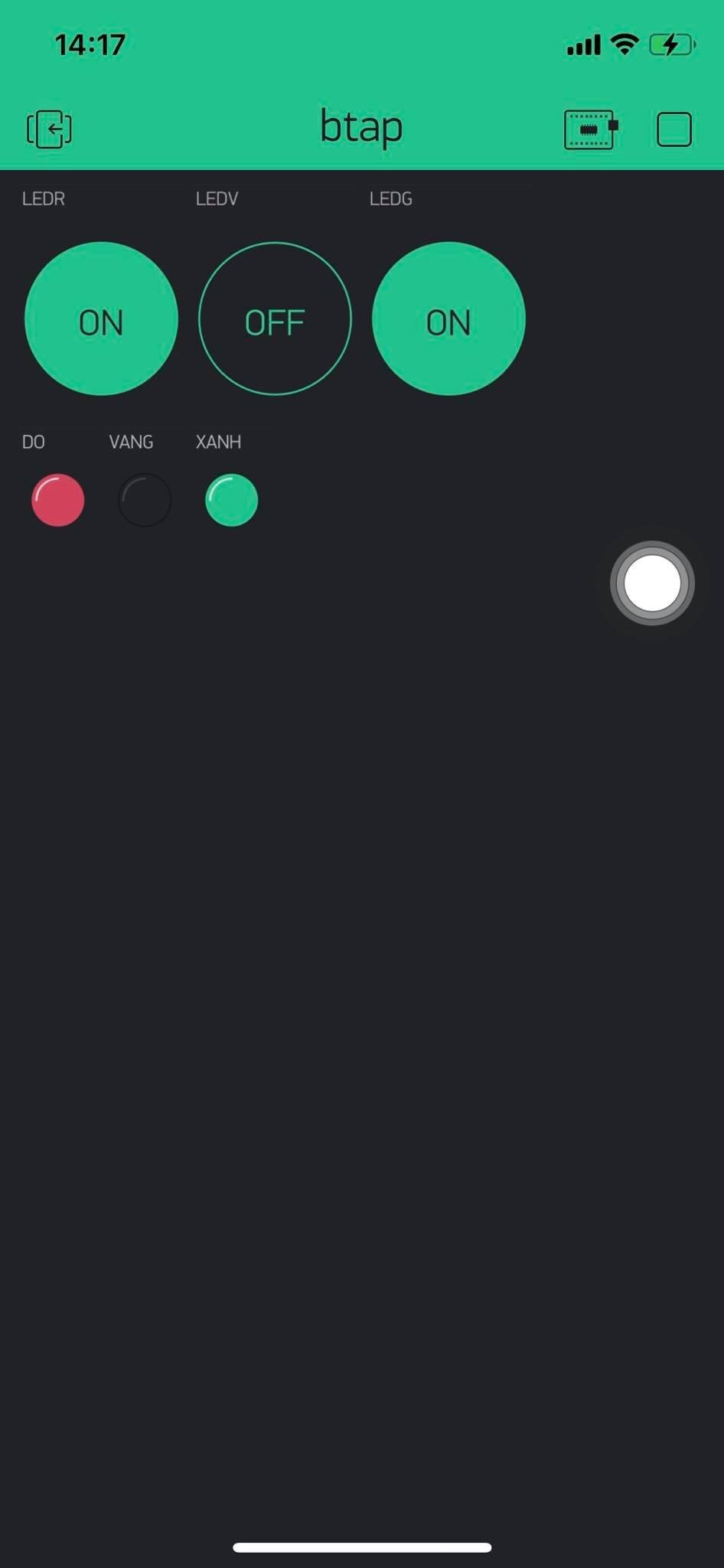
}

}

## Bài tập 3:

Câu 1: Khi nhấn 3 Button trên Blynk App sử dụng 3 Virtual Pin là V11, V12 và V13, thì sẽ điều khiển ON/OFF tương ứng 3 LED được kết nối với 3 chân D1, D2 và D3 trên NodeMCU ESP8266 và 3 LED trên Blynk sử dụng Virtual Pin V1, V2 và V3.





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "caominhvy123";

#define ledr D1

#define ledy D2

#define ledg D3

int value1,value2,value3;

WidgetLED appledr(V1);

WidgetLED appledy(V2);

WidgetLED appledg(V3);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledy,OUTPUT);

pinMode(ledg,OUTPUT);

}

BLYNK\_WRITE(V11) {

value1 = param.asInt();

if(value1 == 1){

digitalWrite(ledr,HIGH);

appledr.on();

}

else {

digitalWrite(ledr,LOW);

appledr.off();

}

}

BLYNK\_WRITE(V12) {

value2 = param.asInt();

if(value2 == 1){

digitalWrite(ledy,HIGH);

appledy.on();

}

else {

digitalWrite(ledy,LOW);

appledy.off();

}

}

BLYNK\_WRITE(V13) {

value3 = param.asInt();

if(value3 == 1){

digitalWrite(ledg,HIGH);

appledg.on();

}

else {

digitalWrite(ledg,LOW);

appledg.off();

}

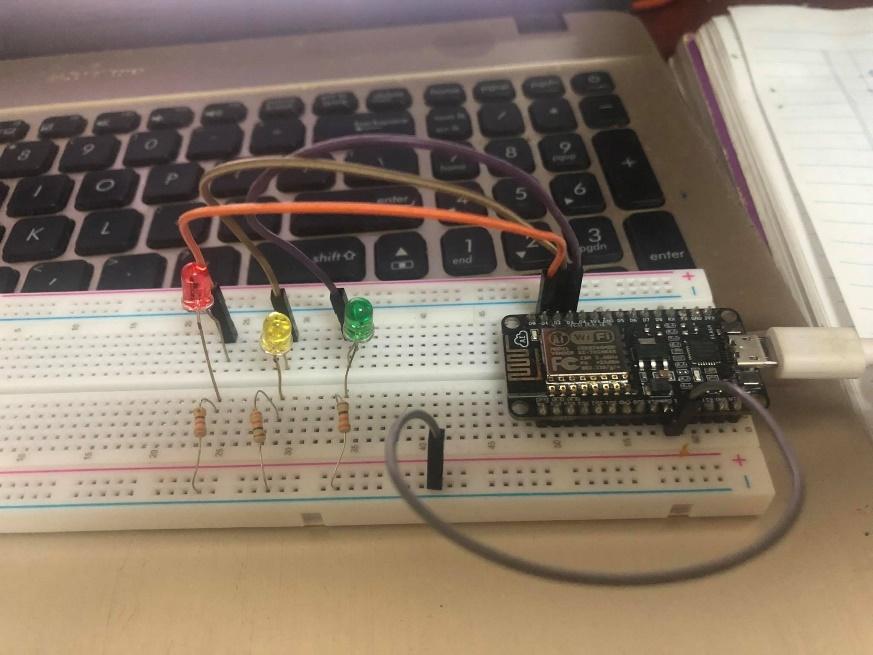
}

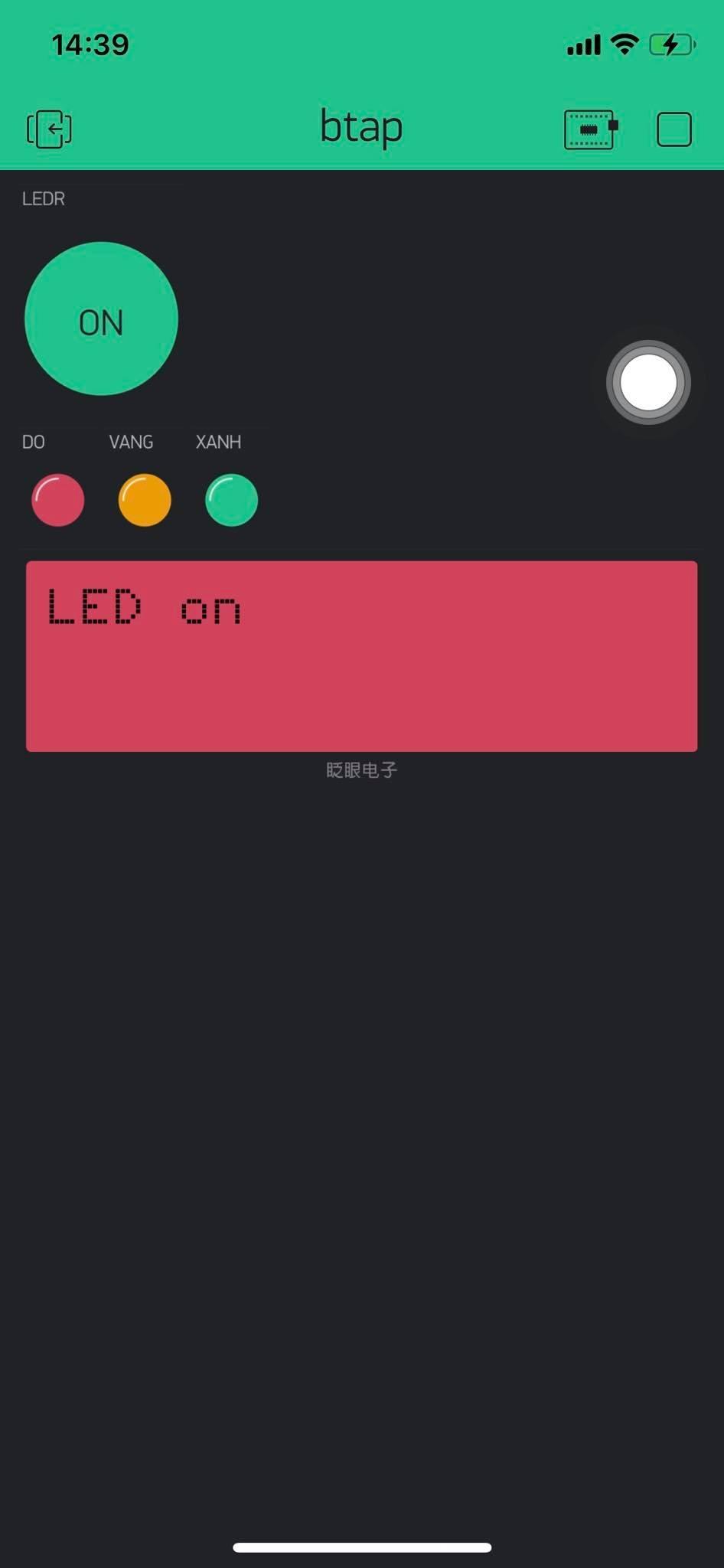
void loop() {

Blynk.run();

}

Câu 2: Khi nhấn Button trên Blynk App sử dụng Virtual Pin V11, thì sẽ điều khiển ON/OFF 3 led (được kết nối với Pin D1, D2, D3 trên trên NodeMCU ESP8266). Đồng thời điều khiển ON/OFF 3 Widget LED trên Blynk App sử dụng Virtual Pin V1, V2, V3 và hiển thị “LED ON” hoặc “LED OFF” trên LCD của Blynk App sử dụng Virtual Pin V4.





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "caominhvy123";

#define ledr D1

#define ledy D2

#define ledg D3

int value1,value2,value3;

WidgetLED appledr(V1);

WidgetLED appledy(V2);

WidgetLED appledg(V3);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledy,OUTPUT);

pinMode(ledg,OUTPUT);

}

BLYNK\_WRITE(V11) {

value1 = param.asInt();

if(value1 == 1){

digitalWrite(ledr,HIGH);

digitalWrite(ledy,HIGH);

digitalWrite(ledg,HIGH);

appledr.on();

appledy.on();

appledg.on();

Blynk.virtualWrite(V4,"LED on");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledy,LOW);

digitalWrite(ledg,LOW);

appledr.off();

appledy.off();

appledg.off();

Blynk.virtualWrite(V4,"LED off");

}

}

void loop() {

Blynk.run();

}

Câu 3: Đọc giá trị ADC của biến trở từ chân A0 của NodeMCU và hiển thị giá trị của biến trở lên Label Display (V5):

- Nếu giá trị biến trở >= 500 thì:

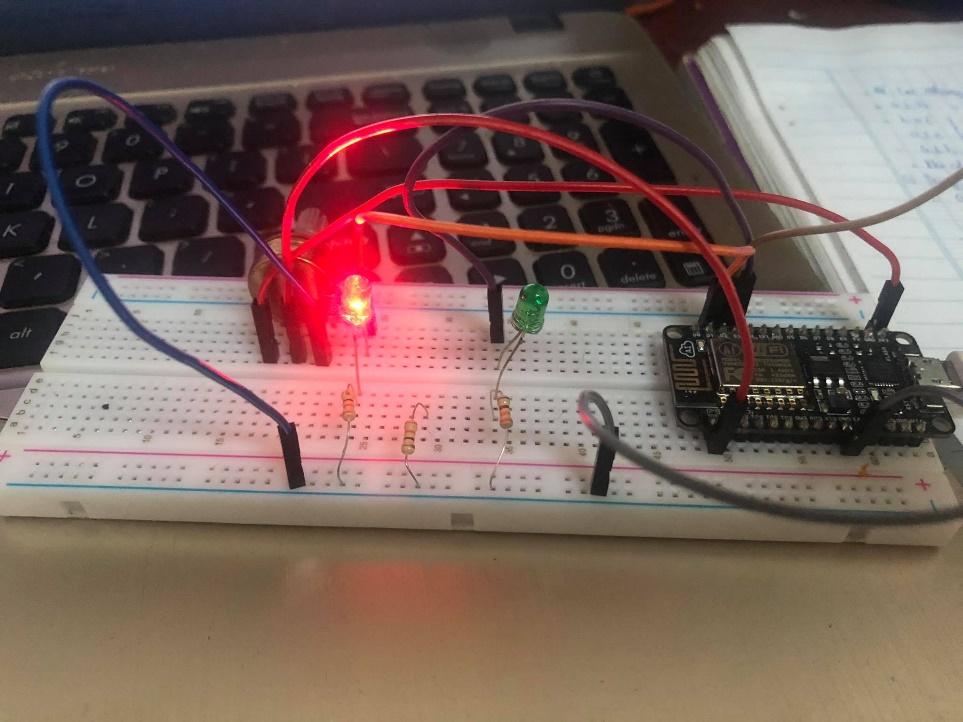
+ Trên board NodeMCU: bật LED đỏ, tắt LED xanh.

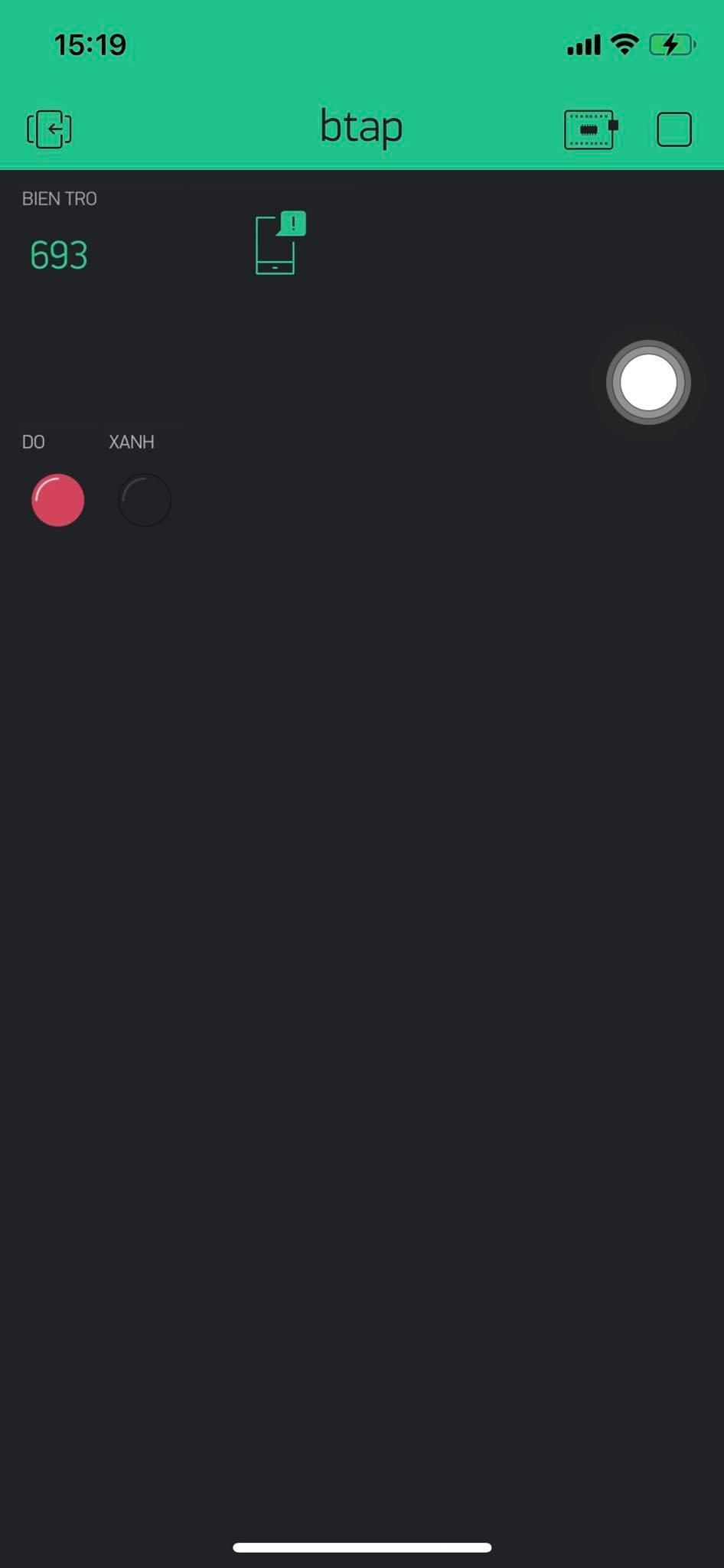
+ Trên Blynk App: bật LED đỏ, tắt LED xanh, đồng thời gửi cảnh báo đến người dùng thông qua Notification và email: “Vượt ngưỡng cho phép!”.

- Ngược lại thì:

+ Trên board NodeMCU: tắt LED đỏ, bật LED xanh.

+ Trên Blynk App: tắt LED đỏ, bật LED xanh.





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "caominhvy123";

#define ledr D1

#define ledg D3

#define analogPin A0

int value1,value2,value3;

int analogVal;

WidgetLED appledr(V1);

WidgetLED appledg(V3);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledg,OUTPUT);

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V5,analogVal);

if(analogVal>=500){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

appledr.on();

appledg.off();

Blynk.notify("Vuot nguong cho phep!");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,HIGH);

appledr.off();

appledg.on();

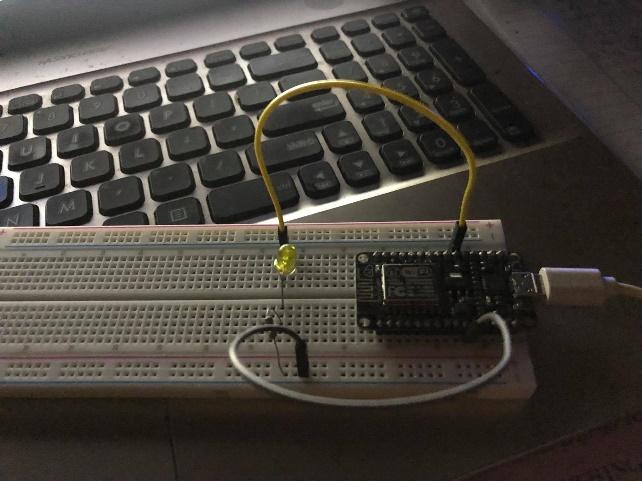
}

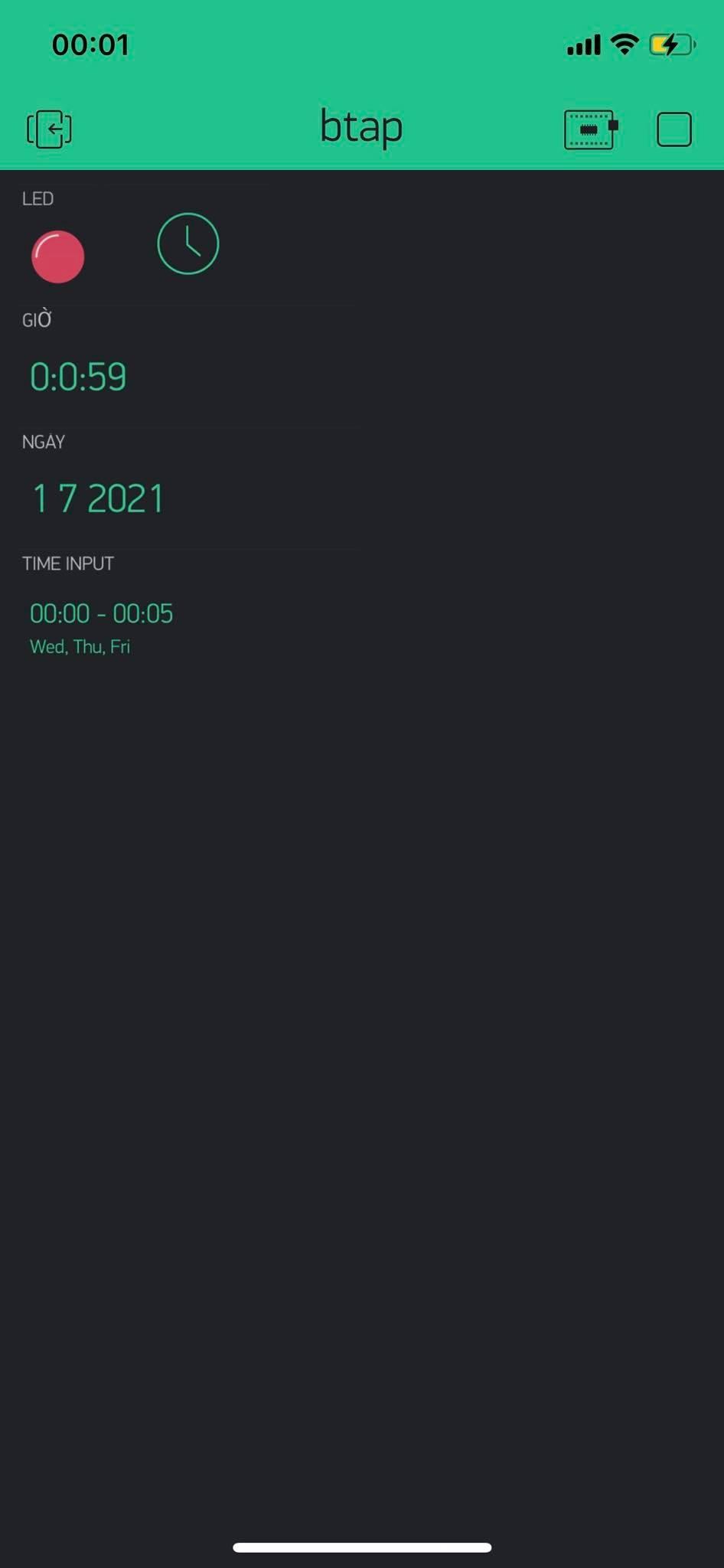
Blynk.run();

Timer.run();

}

Câu 4: Lấy ngày giờ sử dụng chức năng Real-time clock và hiển thị ngày giờ lên Blynk App (sử dụng Labeled Value) thông qua chân ảo V1 (hiển thị ngày) và V2 (hiển thị giờ). Định một khoảng thời gian 5 phút (từ lúc Start time đến lúc Stop time) sử dụng Time input (V5) để bật LED trên Blynk App (V6) và LED trên NodeMCU (D7).





#include <Time.h>

#include <TimeLib.h>

#include <Blynk.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <WidgetRTC.h>

#define BLYNK\_PRINT Serial

#define led D7

long nowtime,starttime,stoptime;

WidgetRTC rtc;

WidgetLED appled(V6);

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

void clockDisplay(){

String currentTime = String(hour()) + ":" + minute() + ":" + second();

String currentDate = String(day()) + " " + month() + " " + year();

Blynk.virtualWrite(V2, currentTime);

Blynk.virtualWrite(V1, currentDate);

}

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

rtc.begin();

timer.setInterval(1000L, clockDisplay);

pinMode(led,OUTPUT);

}

BLYNK\_WRITE(V5){

TimeInputParam t(param);

nowtime = ((hour()\*3600)+(minute()\*60)+second());//thời gian hiện tại

starttime = (t.getStartHour()\*3600)+(t.getStartMinute()\*60);//thời gian bắt đầu

stoptime = (t.getStopHour()\*3600)+(t.getStopMinute()\*60);//thời gian kết thúc

}

void loop() {

if(nowtime>=stoptime){

digitalWrite(led,LOW);

appled.off();

}

else if(nowtime>=starttime){

digitalWrite(led,HIGH);

appled.on();

}

Blynk.run();

timer.run();

}

Câu 5: Đọc giá trị biến trở từ chân A0 của NodeMCU và hiển thị giá trị biến trở lên Value Display và Gauge thông qua chân V3. Cho phép đặt ngưỡng sử dụng chức năng Numeric Input (V7).

- Nếu giá trị biến trở lớn hơn hoặc bằng giá trị ngưỡng thì:

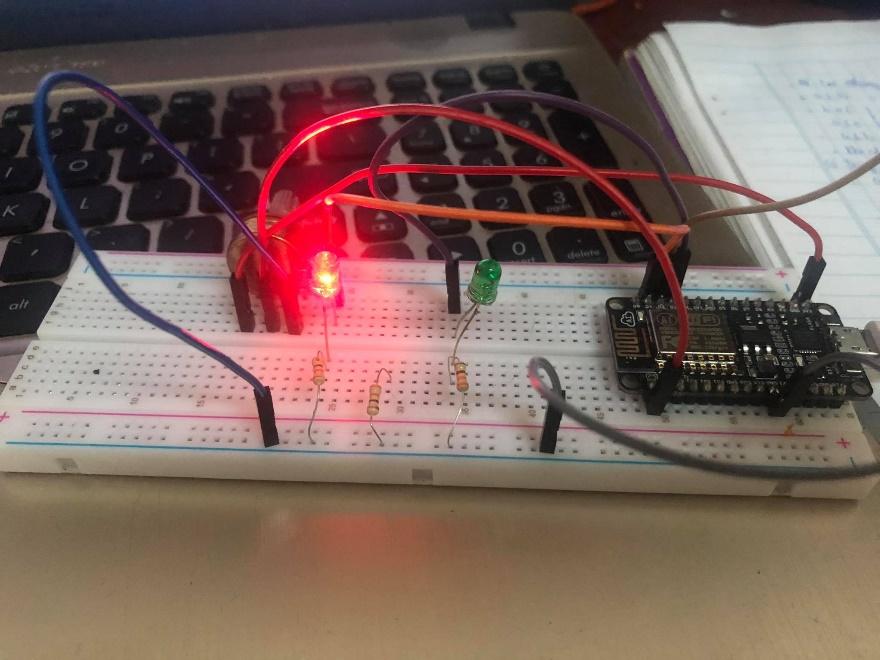
+ Trên board NodeMCU: bật LED đỏ, tắt LED xanh.

+ Trên Blynk App: bật LED đỏ, tắt LED xanh, hiển thị lên LCD “Vượt ngưỡng”, đồng thời gửi cảnh báo đến người dùng thông qua Notification và email: “Vượt ngưỡng cho phép!”.

- Ngược lại thì:

+ Trên board NodeMCU: tắt LED đỏ, bật LED xanh.

+ Trên Blynk App: tắt LED đỏ, bật LED xanh, hiển thị lên LCD “Bình thường”





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

#define ledr D1

#define ledg D3

#define analogPin A0

int Ref;

int analogVal;

WidgetLED appledr(V1);

WidgetLED appledg(V2);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledg,OUTPUT);

Blynk.syncVirtual(V7);

}

BLYNK\_WRITE(V7) { // Lấy giá trị ngưỡng

Ref=param.asInt();

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V3,analogVal);

if(analogVal>=Ref){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

appledr.on();

appledg.off();

Blynk.notify("Vuot nguong cho phep!");

Blynk.email("Canh bao","Vuot nguong");

Blynk.virtualWrite(V4,"Vuot nguong");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,HIGH);

appledr.off();

appledg.on();

Blynk.virtualWrite(V4,"binh thuong");

}

Blynk.run();

timer.run();

}

Câu 6: Đọc giá trị biến trở từ chân A0 của NodeMCU và hiển thị giá trị biến trở lên Value Display (V3). Cho phép đặt ngưỡng sử dụng chức năng Numeric Input (V7) và cho phép chọn mode ON/OFF sử dụng chức năng Styled Button (V5) để bật/tắt hệ thống cảnh báo.

* **Nếu mode ON**: bật hệ thống cảnh báo:

- Nếu giá trị biến trở lớn hơn hoặc bằng giá trị ngưỡng thì:

+ Trên board NodeMCU: bật LED đỏ, tắt LED xanh.

+ Trên Blynk App: bật LED đỏ, tắt LED xanh, hiển thị lên LCD -line1 “Vượt ngưỡng”, LCD-line 2 “Mode ON”, đồng thời gửi cảnh báo đến người dùng thông qua Notification “Vượt ngưỡng cho phép!”.

- Ngược lại thì:

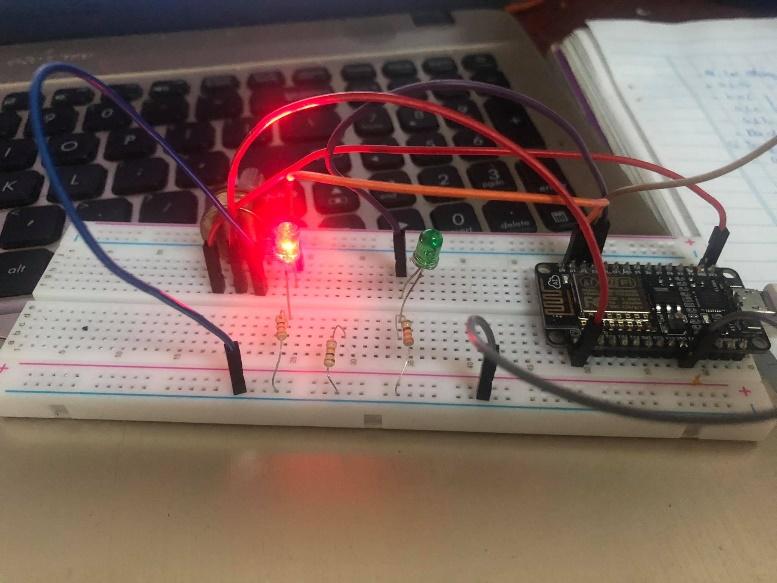
+ Trên board NodeMCU: tắt LED đỏ, bật LED xanh.

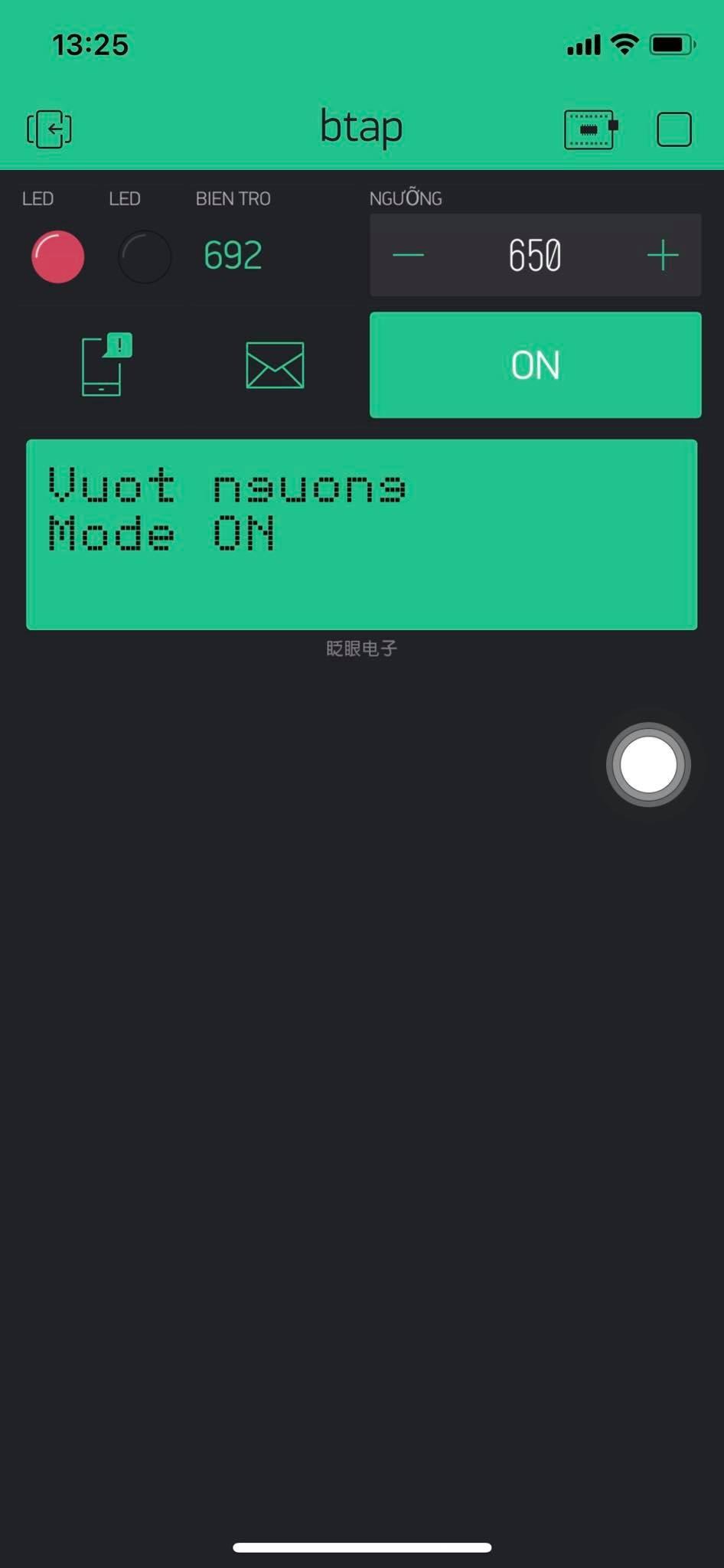
+ Trên Blynk App: tắt LED đỏ, bật LED xanh, hiển thị lên LCD-line1 “Bình thường”, LCD-line 2 “Mode ON”

* **Ngược lại (mode OFF)**: tắt hệ thống cảnh báo.

+ Trên board NodeMCU: tắt LED đỏ, tắt LED xanh.

+ Trên Blynk App: tắt LED đỏ, tắt LED xanh, hiển thị lên LCD-line1 “Tắt cảnh báo”, LCD-line 2 “Mode OFF”





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

#define ledr D1

#define ledg D3

#define analogPin A0

int Ref,value;

int analogVal;

WidgetLED appledr(V1);

WidgetLED appledg(V2);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledg,OUTPUT);

Blynk.syncVirtual(V7);

}

BLYNK\_WRITE(V5) { // điều chỉnh chế độ

value = param.asInt();

}

BLYNK\_WRITE(V7) { // Lấy giá trị ngưỡng

Ref=param.asInt();

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V3,analogVal);

if(value == 1){

if(analogVal>=Ref){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

appledr.on();

appledg.off();

Blynk.notify("Vuot nguong cho phep!");

Blynk.virtualWrite(V6,"Mode ON");

Blynk.virtualWrite(V4,"Vuot nguong");

}

else{

digitalWrite(ledr,LOW);

digitalWrite(ledg,HIGH);

appledr.off();

appledg.on();

Blynk.virtualWrite(V4,"binh thuong");

Blynk.virtualWrite(V6,"Mode ON");

}

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,LOW);

appledr.off();

appledg.off();

Blynk.virtualWrite(V4,"tat canh bao");

Blynk.virtualWrite(V6,"Mode OFF");

}

Blynk.run();

timer.run();

}

Câu 7: Mở rộng câu 6, cho phép đặt ngưỡng trên (V7) và ngưỡng dưới (V8) sử dụng chức năng Numeric Input.

* **Nếu mode ON**:

- Nếu giá trị biến trở cao hơn ngưỡng trên thì sẽ cảnh báo như sau:

+ Trên NodeMCU: LED đỏ sáng, LED xanh tắt, LED vàng tắt.

+ Trên Blynk App: LED đỏ sáng, LED xanh tắt, LED vàng tắt và gửi cảnh báo “VƯỢT NGƯỠNG TRÊN” thông qua chức năng Notification và hiển thị lên LCD dòng thông báo “BẬT CẢNH BÁO”.

- Nếu giá trị biến trở thấp hơn ngưỡng dưới thì sẽ cảnh báo như sau:

+ Trên NodeMCU: LED đỏ tắt, LED xanh tắt, LED vàng sáng.

+ Trên Blynk App: LED đỏ tắt, LED xanh tắt, LED vàng sáng và gửi cảnh báo “VƯỢT NGƯỠNG DƯỚI” thông qua chức năng Notification và hiển thị lên LCD dòng thông báo “BẬT CẢNH BÁO”.

- Nếu giá trị biến trở nằm trong ngưỡng cho phép thì:

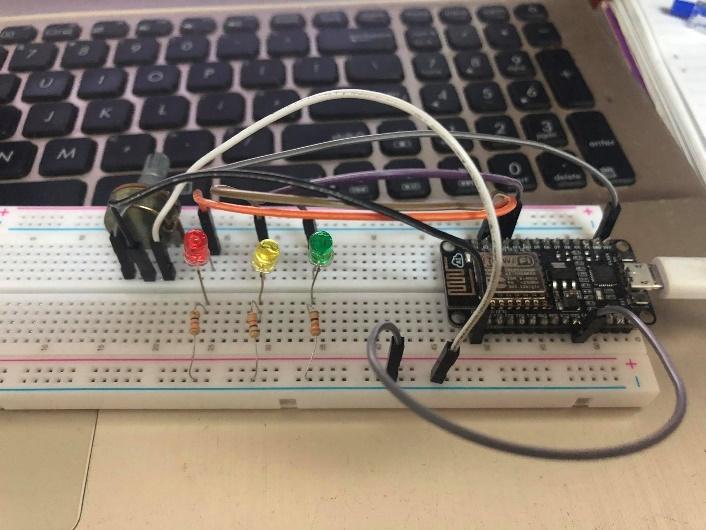
+ Trên NodeMCU: LED đỏ tắt, LED xanh sáng, LED vàng tắt.

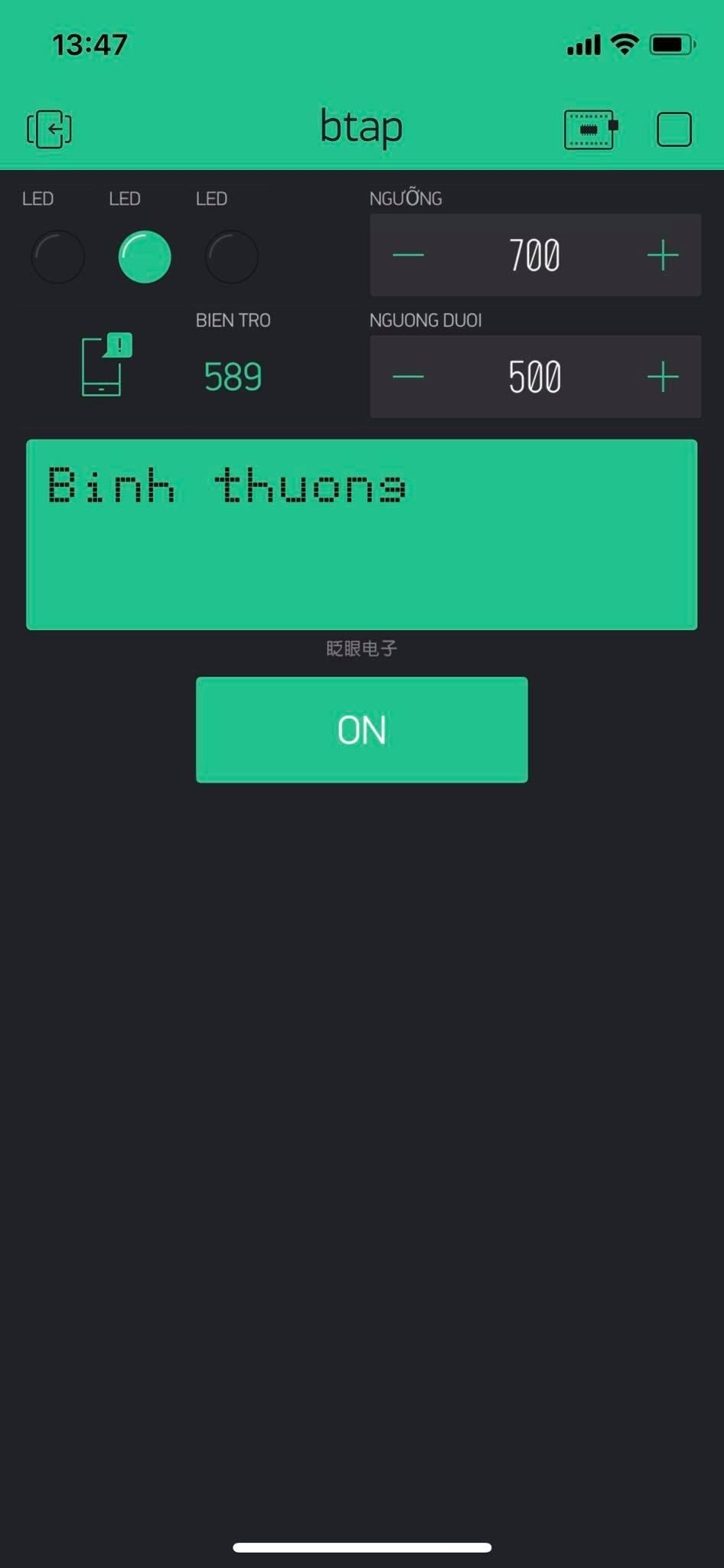
+ Trên Blynk App: LED đỏ tắt, LED xanh sáng, LED vàng tắt

* **Ngược lại (Mode OFF):** tắt hệ thống cảnh báo.

+ Trên NodeMCU: LED đỏ tắt, LED xanh tắt, LED vàng tắt.

+ Trên Blynk App: LED đỏ tắt, LED xanh tắt, LED vàng tắt và hiển thị lên LCD dòng thông báo “TẮT CẢNH BÁO”.





#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

#define ledr D1

#define ledy D2

#define ledg D3

#define analogPin A0

int Ref1,Ref2,value;

int analogVal;

WidgetLED appledr(V1);

WidgetLED appledg(V2);

WidgetLED appledy(V3);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledg,OUTPUT);

pinMode(ledy,OUTPUT);

Blynk.syncVirtual(V7);

Blynk.syncVirtual(V8);

}

BLYNK\_WRITE(V5) { // điều chỉnh chế độ

value = param.asInt();

}

BLYNK\_WRITE(V7) { // Lấy giá trị ngưỡng tren

Ref1=param.asInt();

}

BLYNK\_WRITE(V8) { // Lấy giá trị ngưỡng duoi

Ref2=param.asInt();

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V6,analogVal);

if(value == 1){

if(analogVal>=Ref1){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

digitalWrite(ledy,LOW);

appledr.on();

appledg.off();

appledy.off();

Blynk.notify("Vuot nguong tren!");

Blynk.virtualWrite(V4,"Bat canh bao");

}

else if(analogVal<Ref2){

digitalWrite(ledr,LOW);

digitalWrite(ledg,LOW);

digitalWrite(ledy,HIGH);

appledr.off();

appledg.off();

appledy.on();

Blynk.notify("Vuot nguong duoi!");

Blynk.virtualWrite(V4,"Bat canh bao");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,HIGH);

digitalWrite(ledy,LOW);

appledr.off();

appledg.on();

appledy.off();

Blynk.virtualWrite(V4,"Binh thuong");

}

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,LOW);

digitalWrite(ledy,LOW);

appledr.off();

appledg.off();

appledy.off();

Blynk.virtualWrite(V4,"Tat canh bao");

}

Blynk.run();

timer.run();

}

Câu 8: Đọc giá trị biến trở từ chân A0 của NodeMCU và hiển thị giá trị biến trở lên Value Display (V2). Cho phép đặt ngưỡng sử dụng chức năng Numeric Input (V7) và cho phép chọn mode Auto/Manual sử dụng chức năng Styled Button (V4).

* **Nếu mode Auto:** cho phép đặt ngưỡng độ ẩm đất như sau:

- Nếu giá trị biến trở lớn hơn hoặc bằng giá trị ngưỡng thì:

+ Trên board NodeMCU: bật LED đỏ, đóng relay.

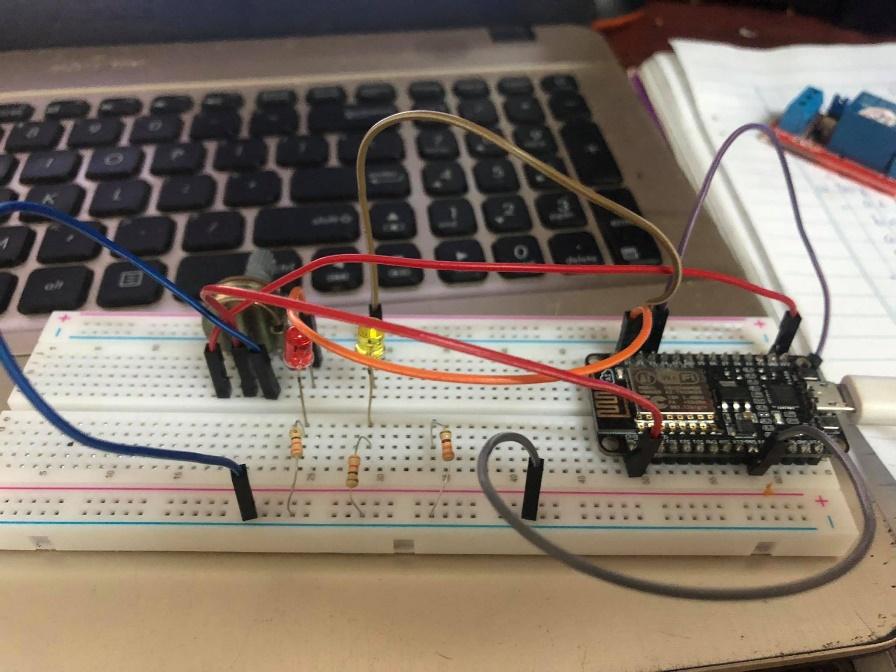
+ Trên Blynk App: bật LED đỏ, gửi cảnh báo “VƯỢT NGƯỠNG” qua chức năng Notification và hiển thị trên LCD thông báo “Mở máy bơm nước”.

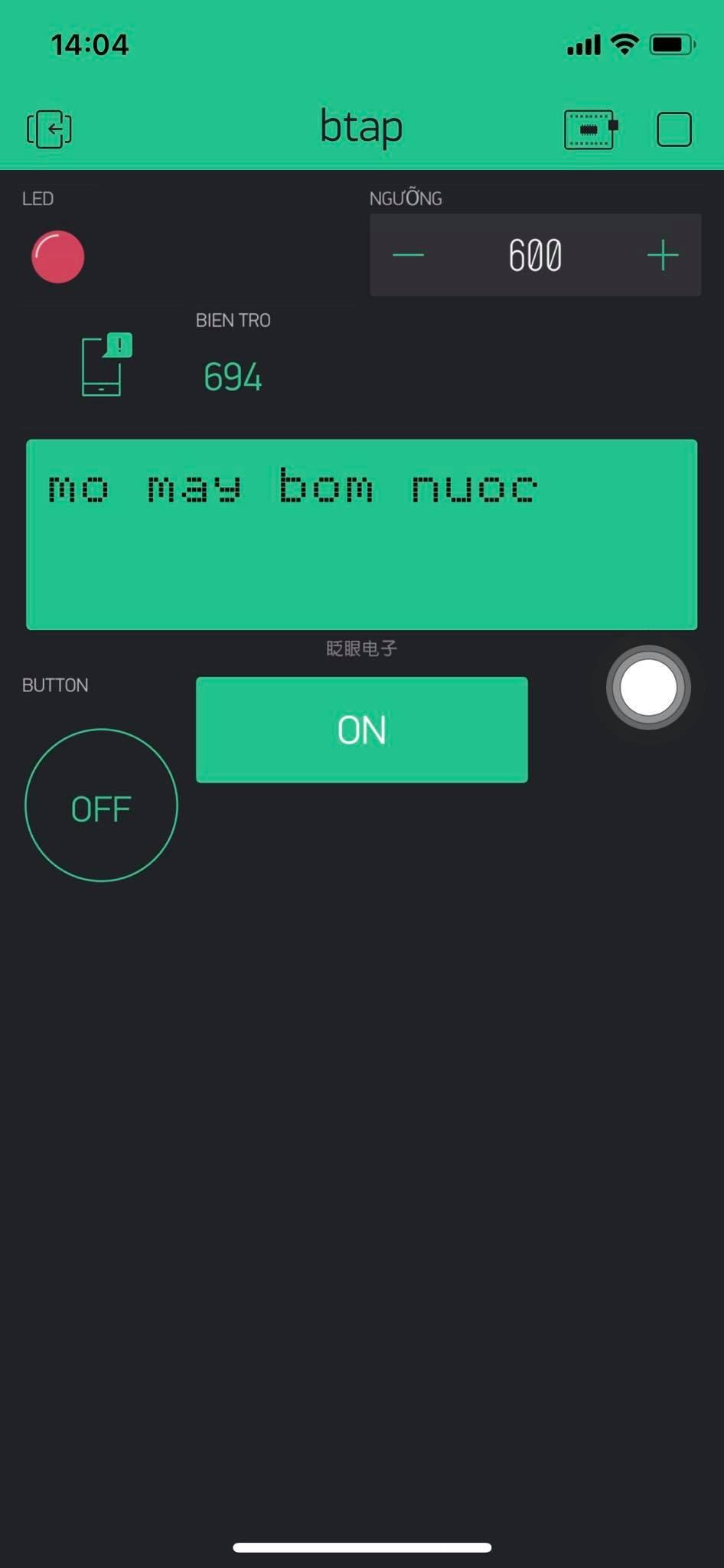
- Ngược lại thì:

+ Trên board NodeMCU: tắt LED đỏ, ngắt relay.

+ Trên Blynk App: tắt LED đỏ và hiển thị trên LCD thông báo “Tắt máy bơm nước”.

* **Ngược lại (Mode Manual):** cho phép điều khiển đóng/ngắt relay để tắt/mở máy bơm nước bằng nút nhấn trên Blynk App.





\*Chú ý: Led vàng tượng trưng cho relay.

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

#define ledr D1

#define relay D2

#define analogPin A0

int Ref1,button,value;

int analogVal;

WidgetLED appledr(V1);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(relay,OUTPUT);

Blynk.syncVirtual(V7);

}

BLYNK\_WRITE(V4) { // điều chỉnh chế độ

value = param.asInt();

}

BLYNK\_WRITE(V6) { // button

button = param.asInt();

}

BLYNK\_WRITE(V7) { // Lấy giá trị ngưỡng tren

Ref1=param.asInt();

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V2,analogVal);

if(value == 1){

if(analogVal>=Ref1){

digitalWrite(ledr,HIGH);

digitalWrite(relay,LOW);

appledr.on();

Blynk.notify("Vuot nguong!");

Blynk.virtualWrite(V5,"mo may bom nuoc");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(relay,LOW);

appledr.off();

Blynk.virtualWrite(V5,"tat may bom nuoc");

}

}

else {

if (button == 1){

digitalWrite(relay,HIGH);

}

else {

digitalWrite(relay,LOW);

}

}

Blynk.run();

timer.run();

}

Câu 9: Mở rộng câu 8, cho phép đặt ngưỡng trên (V7) và ngưỡng dưới (V8) sử dụng chức năng Numeric Input.

* **Nếu mode Auto**: cho phép đặt ngưỡng trên và ngưỡng dưới như sau:
* Nếu giá trị biến trở lớn hơn ngưỡng trên thì:

+ Trên NodeMCU: bật led đỏ, tắt led xanh, đóng relay 1, ngắt relay 2.

+ Trên Blynk App: bật led đỏ, tắt led xanh, gửi cảnh báo “VƯỢT NGƯỠNG TRÊN” thông qua chức năng Notification và hiển thị trên LCD thông báo “VƯỢT REF TRÊN”.

* Nếu độ ẩm đất thấp hơn ngưỡng dưới thì:

+ Trên NodeMCU: bật led đỏ, tắt led xanh, ngắt relay 1, đóng relay 2.

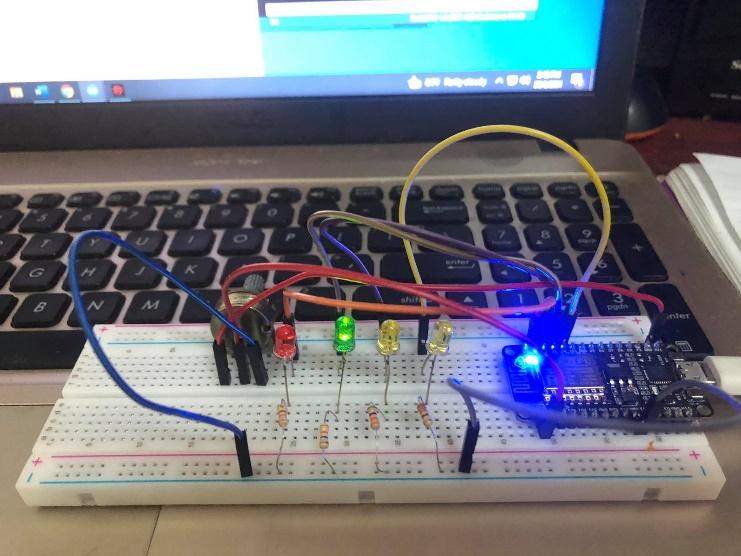
+ Trên Blynk App: bật led đỏ, tắt led xanh, gửi cảnh báo “VƯỢT NGƯỠNG DƯỚI” thông qua chức năng Notification và hiển thị trên LCD thông báo “VƯỢT REF DƯỚI”.

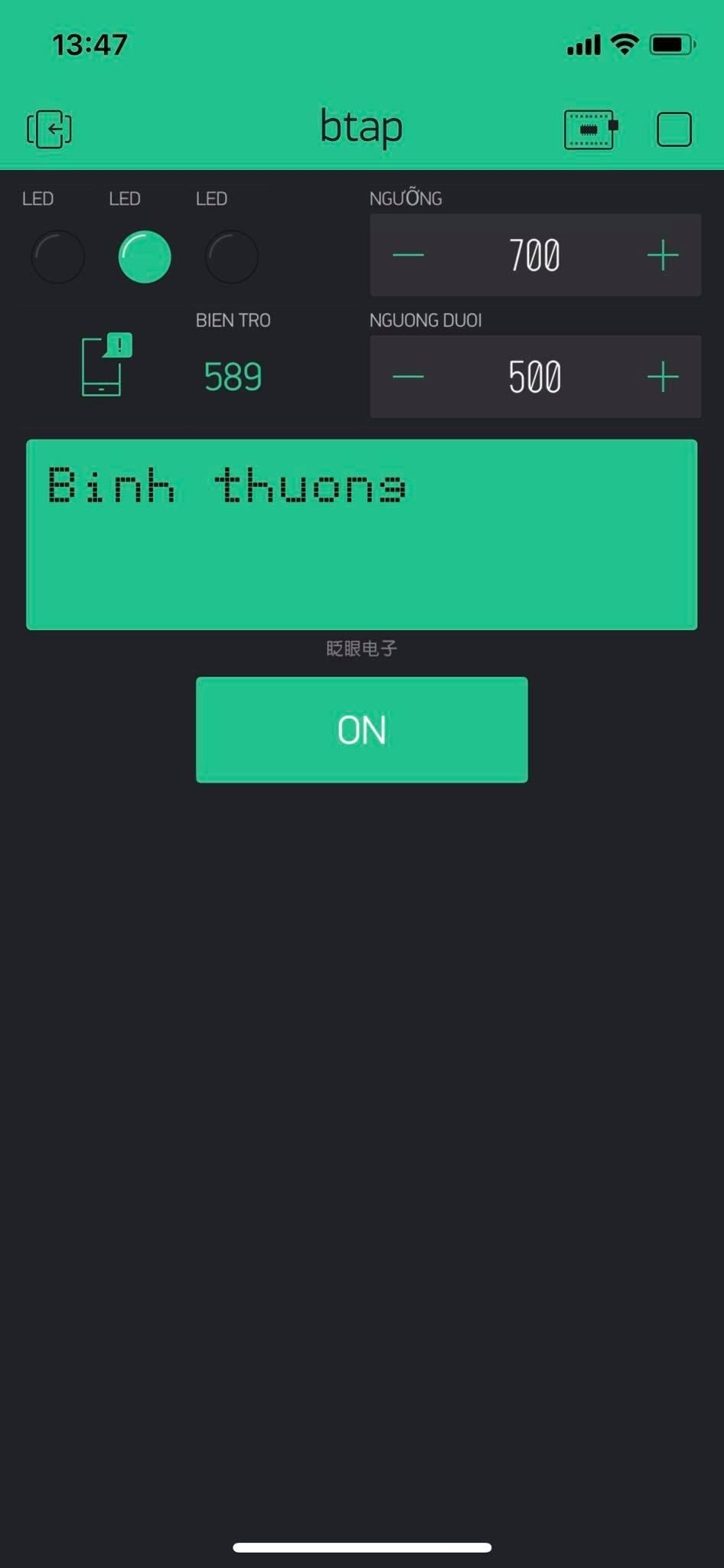
* Nếu độ ẩm đất nằm trong ngưỡng cho phép thì:

+ Trên NodeMCU: tắt led đỏ, bật led xanh, ngắt relay 1, ngắt relay 2.

+ Trên Blynk App: tắt led đỏ, bật led xanh và hiển thị trên LCD thông báo “BÌNH THƯỜNG”.

* Ngược lại (mode Manual): cho phép điều khiển đóng hoặc ngắt relay 1 và relay 2 bằng nút nhấn trên Blynk App.





\*Chú ý: 2 led vàng tượng trung cho 2 relay

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

BlynkTimer timer;

char auth[] = "GWMyNwYurOX9irzHRfVS4KHfbBad9J\_m";

char ssid[] = "FPT Cao Van Sa";

char pass[] = "11118888";

#define ledr D1

#define ledg D2

#define relay1 D3

#define relay2 D4

#define analogPin A0

int Ref1,Ref2,button,value;

int analogVal;

WidgetLED appledr(V1);

WidgetLED appledg(V2);

void setup() {

Serial.begin(9600);

Blynk.begin(auth, ssid, pass,"sv.bangthong.com",8080);

pinMode(ledr,OUTPUT);

pinMode(ledg,OUTPUT);

pinMode(relay1,OUTPUT);

pinMode(relay2,OUTPUT);

Blynk.syncVirtual(V7);

Blynk.syncVirtual(V8);

Blynk.syncVirtual(V5);

Blynk.syncVirtual(V9);

}

BLYNK\_WRITE(V5) { // điều chỉnh chế độ

value = param.asInt();

}

BLYNK\_WRITE(V7) { // Lấy giá trị ngưỡng tren

Ref1=param.asInt();

}

BLYNK\_WRITE(V8) { // Lấy giá trị ngưỡng duoi

Ref2=param.asInt();

}

BLYNK\_WRITE(V9) { // button

button = param.asInt();

}

void loop() {

analogVal=analogRead(analogPin);

Blynk.virtualWrite(V6,analogVal);

if(value == 1){

if(analogVal>=Ref1){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

digitalWrite(relay1,HIGH);

digitalWrite(relay2,LOW);

appledr.on();

appledg.off();

Blynk.notify("Vuot nguong tren!");

Blynk.virtualWrite(V4,"Vuot REF tren");

}

else if(analogVal<Ref2){

digitalWrite(ledr,HIGH);

digitalWrite(ledg,LOW);

digitalWrite(relay1,LOW);

digitalWrite(relay2,HIGH);

appledr.on();

appledg.off();

Blynk.notify("Vuot nguong duoi!");

Blynk.virtualWrite(V4,"Vuot REF duoi");

}

else {

digitalWrite(ledr,LOW);

digitalWrite(ledg,HIGH);

digitalWrite(relay1,LOW);

digitalWrite(relay2,LOW);

appledr.off();

appledg.on();

Blynk.virtualWrite(V4,"Binh thuong");

}

}

else {

if(button == 1){

digitalWrite(relay1,HIGH);

digitalWrite(relay2,HIGH);

}

else {

digitalWrite(relay1,LOW);

digitalWrite(relay2,LOW);

}

}

Blynk.run();

timer.run();

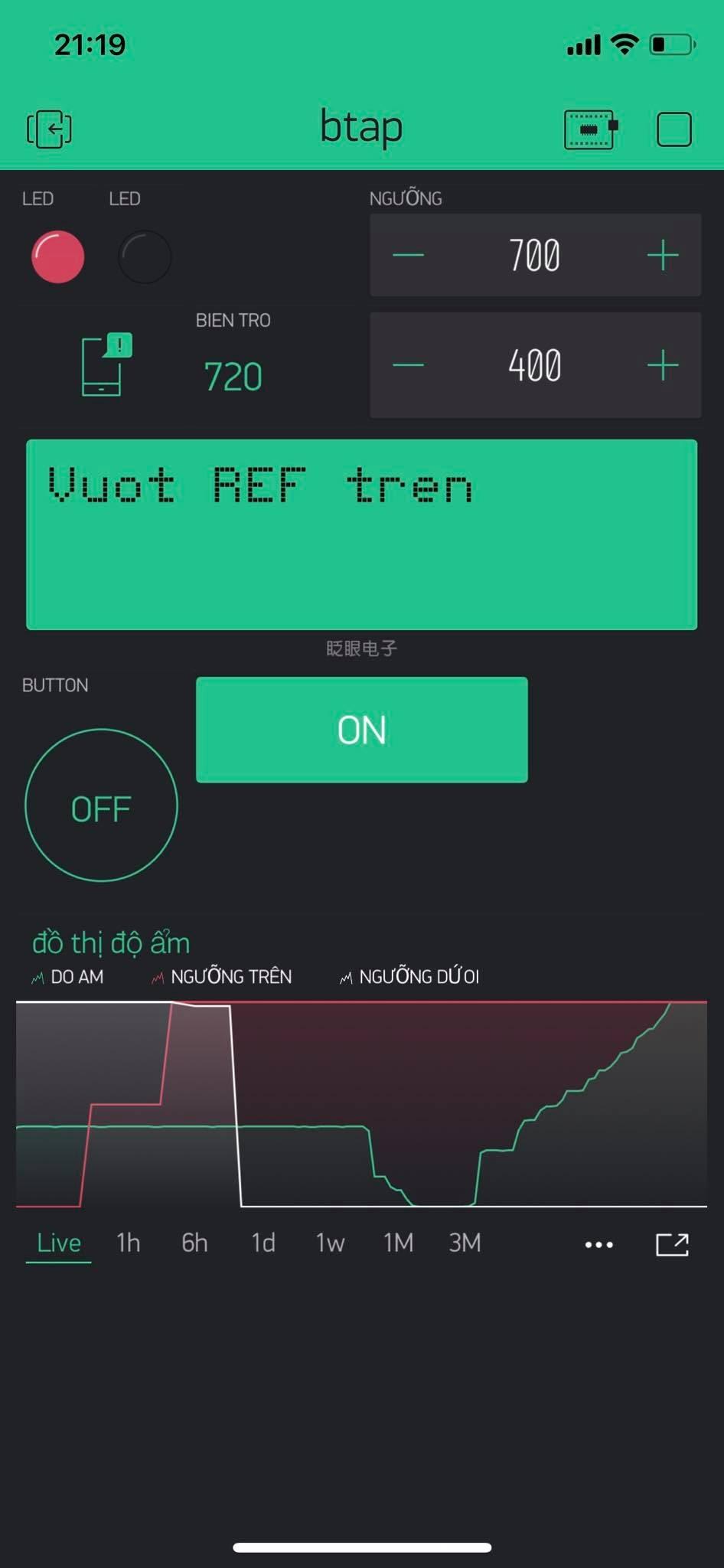
}

Câu 10: Mở rộng câu 9, bổ sung chức năng Supper chart để hiển thị giá trị biến trở và giá trị ngưỡng dưới dạng đồ thị. Đồng thời cho phép xem lại dữ liệu theo thời gian đã lưu: giờ, ngày, tuần, tháng và năm và xuất dữ liệu đã lưu thành file \*.CSV.

Blynk.virtualWrite(V10,String(analogVal)+"%");

Blynk.virtualWrite(V11,String(Ref1));

Blynk.virtualWrite(V12,String(Ref2));



Ghi chú: Mỗi câu trong bài tập cần trình bày gồm các phần sau:

1. Mạch điện đã lắp ráp trên test board
2. Chương trình điều khiển
3. Giao diện app điều khiển