

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN ĐIỆN**



## **BÁO CÁO ĐỒ ÁN 1**

**ĐỀ TÀI:**

### **IOT DEVICE - THIẾT KẾ THIẾT BỊ ĐO CƯỜNG ĐỘ ÁNH SÁNG**

*Sinh viên thực hiện:*

- Lê Công Tráng 20164195
- Trần Tiến Quân 20163391
- Vũ Thanh Bình 20160378

*Giảng viên hướng dẫn:* PGS.TS Nguyễn Quốc Cường

Hà Nội, 20/5/2019

# Contents

---

Lời nói đầu.....	2
Lời cảm ơn.....	2
Chương 1. Mục đích đề tài .....	3
1.1 Giới thiệu tổng quan.....	3
1.2 Chức năng .....	3
Chương 2. Tổng quan hệ thống .....	4
2.1 Sơ đồ khối - Block diagram.....	4
2.2 Lưu đồ thuật toán - Flow Chart .....	5
Chương 3. Thiết kế phần cứng .....	5
3.1 Lựa chọn phần cứng .....	5
3.2 Phân tích datasheet .....	5
3.2.1 Cảm biến ánh sáng BH1750.....	5
3.2.2 Modul Bluetooth .....	6
3.2.3 Màn hình hiển thị LCD 16X2 .....	7
3.3.4 Vi xử lý AT89S52.....	7
3.3 Chế độ sử dụng.....	8
3.3.1 Cảm biến ánh sáng BH1750.....	8
3.3.2 Modul Bluetooth HC05.....	10
3.3.3 LCD 16x2.....	10
3.3.4 Vi xử lý AT89S52.....	11
3.4 Sơ đồ mạch nguyên lý .....	11
Chương 4. Thiết kế phần mềm .....	13
4.1 Thiết kế chương trình cho vi điều khiển.....	13
4.2 Xây dựng chương trình trên máy tính .....	15
Chương 5. Triển khai thử nghiệm đề tài.....	17
5.1 Mô hình thử nghiệm .....	17
5.2 Đánh giá đề tài.....	18
Lời kết .....	18
Tài liệu tham khảo .....	18

# IOT-Device: Thiết bị đo cường độ ánh sáng

## Lời nói đầu

---

Nhắc tới chiếu sáng rất nhiều người trong chúng ta thường lập tức nghĩ ngay tới việc tìm đến công tắc và bật đèn lên. Nhưng ẩn sâu trong đó có rất nhiều các tiêu chuẩn quan trọng để xây dựng một hệ thống chiếu sáng như: độ rọi, chỉ số màu, màu sắc ánh sáng,.. Việc thiết kế một hệ thống chiếu sáng đạt chuẩn rất phức tạp, phải trải qua nhiều quá trình nghiên cứu đo đạc cầu kì. Sự thật không phải hệ thống chiếu sáng trong nhà, khu công nghiệp nào cũng đạt tiêu chuẩn. Không phải mô hình chiếu sáng nào cũng có thể đáp ứng được các tiêu chuẩn chiếu sáng.

Ngày nay với sự phát triển của khoa học công nghệ, các hệ thống chiếu sáng đang dần được tự động hóa và dần trở nên thông minh. Để phục vụ thiết kế các hệ thống chiếu sáng trong công nghiệp, sử dụng trong các dự án Internet of Things, Smart Home. Đề tài “Thiết kế thiết bị đo cường độ ánh sáng” được hình thành và phát triển.

## Lời cảm ơn

---

Để thiết kế và xây dựng đề tài này chúng tôi đã trải qua rất nhiều quá trình nghiên cứu và học tập. Những khó khăn trong việc tìm hiểu phần cứng, xây dựng chương trình cho các module, triển khai hệ thống thử nghiệm... Trong những giai đoạn này chúng tôi được sự giúp đỡ rất nhiều từ các Thầy/Cô, bạn bè, cộng đồng internet. Đặc biệt chúng tôi xin gửi lời cảm ơn tới thầy giáo PGS.TS **Nguyễn Quốc Cường** đã hướng dẫn và đồng hành cùng trong suốt quá trình xây dựng đề tài.

Xin cảm ơn!.

# Chương 1. Mục đích đề tài

## 1.1 Giới thiệu tổng quan

Chiếu sáng có thể chiếm tới 10-38% tổng hóa đơn thanh toán tiền điện trong bất cứ thành phố nào trên thế giới. Điều đáng quan tâm nhất trong công nghệ cho lĩnh vực này hiện tại là tự động hóa, tiêu thụ năng lượng và giá thành hợp lý. Tự động hóa có xu hướng giảm sức người với sự giúp đỡ của trí tuệ nhân tạo. Tiết kiệm năng lượng là điều đáng bàn nhất vì nó là nguồn năng lượng có hạn do nhiều lý do khác nhau.

Việc chiếu sáng thực tế đời sống hiện nay vẫn còn đang thủ công. Tức là phần đa trong gia đình, văn phòng, nhà máy... Hệ thống chiếu sáng vẫn đang còn rất đơn giản và bằng tay. Dù là hoạt động gì cũng chỉ bật tắt không cần biết cường độ sáng có phù hợp hay không.

Độ sáng ảnh hưởng rất lớn tới thị giác của con người. Cao quá sẽ gây hại cho mắt và lãng phí năng lượng. Thấp quá không nhưng ảnh hưởng tới thị giác mà còn làm giảm năng suất làm việc. Độ rọi tiêu chuẩn cho việc đọc sách là 500-550 lux, nghiên cứu 700-750lux..., Đối với các công việc mang tính chất tỉ mỉ, chi tiết như chế tác mỹ ký, chi tiết đồng hồ... độ rọi tối thiểu 1000 lux. Do tính chất này chúng ta cần một hệ thống có chiếu sáng theo các công việc cụ thể với các giá trị độ rọi đã đưa ra. Hoặc có thể tùy chỉnh theo ý muốn. Để phát triển hệ thống này cần một thiết bị đo đặc cường độ ánh sáng. Độ tin cậy và chính xác của thiết bị quyết định phần lớn chất lượng của hệ thống.

Vậy đích chính của đồ án là xây dựng một thiết bị đo cường độ ánh sáng. Dải đo lớn phục vụ đa số mục đích chiếu sáng. Giao diện sử dụng đơn giản có thể kết nối với điện thoại đặc biệt là máy tính. Có thể gửi dữ liệu giá trị độ sáng phục vụ người dùng và các hệ thống cần sử dụng cường độ sáng.

## 1.2 Chức năng

Nhằm giải quyết vấn đề đo cường độ ánh sáng và khả năng tích hợp với các module cảm biến khác ( Nhiệt độ, độ ẩm,..). Đề tài sẽ thiết kế theo phương châm ***“Do one thing and do it well”***. Hệ thống sẽ chỉ đo cường độ ánh sáng hiển thị kết quả đo trực tiếp lên LCD và gửi giá trị cường độ ánh sáng đến máy tính nhúng, điện thoại hoặc các thiết bị master khác.



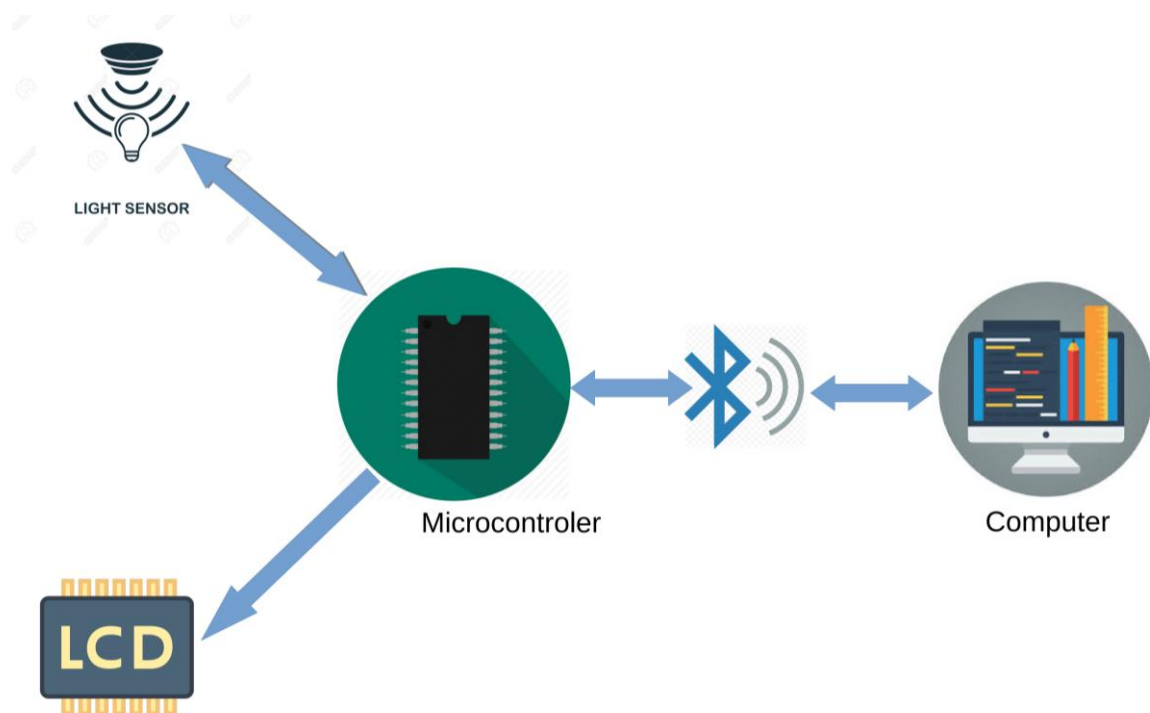
Chỉ khi có thông điệp yêu cầu đo từ thiết bị master thì khối cảm biến sẽ tiến hành đo và gửi giá trị đo được cho master đồng thời hiển thị ra LCD. Nhờ việc cấp quyền đo cho khối điều khiển như thế hệ thống đo có thể hoạt động linh hoạt theo thiết bị master. Có thể đo liên tục

giá trị cường độ ánh sáng hoặc đo một lần hoàn toàn theo yêu cầu của người sử dụng dựa trên ứng dụng đã được lập trình trên khối điều khiển.

## Chương 2. Tổng quan hệ thống

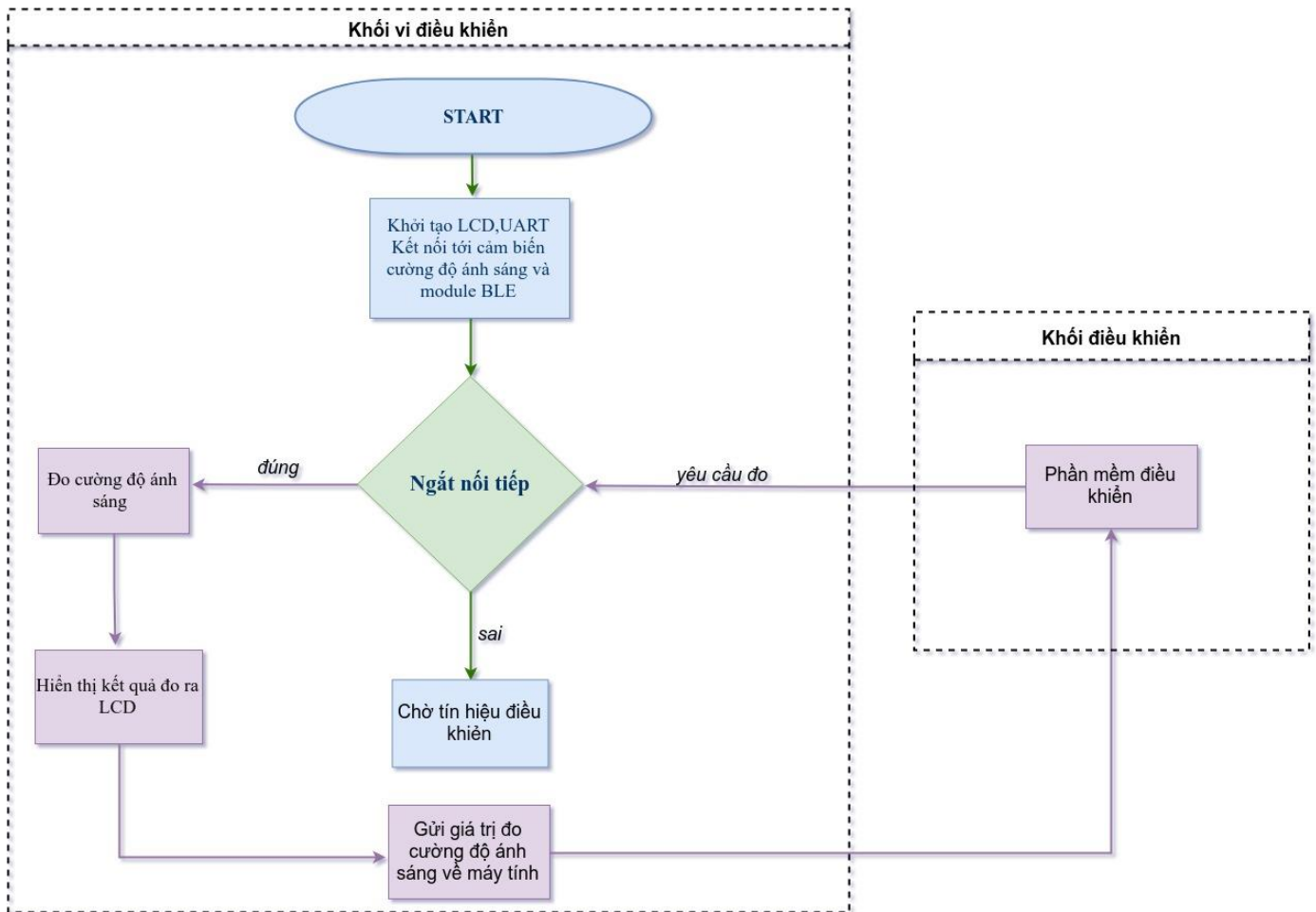
### 2.1 Sơ đồ khối - Block diagram

Dựa trên những chức năng và mục đích của đề tài. Hệ thống sẽ bao gồm **cảm biến ánh sáng** để dùng để đo cường độ ánh sáng. Để nhận giá trị đo được từ cảm biến, sử dụng một **vi điều khiển** đơn giản nhằm mục đích tiền xử lí. Nhận giá trị đo được từ cảm biến ánh sáng từ đó tính toán giá trị cường độ ánh sáng thực.



Để tiện trong quá trình thiết kế xây dựng thử nghiệm đề tài. Cần thiết một **màn hình LCD** để hiển thị giá trị cường độ ánh sáng. Do đó vi điều khiển sẽ gửi giá trị đo được và giá trị thực (tức là sau khi đã tính toán ra màn hình) để kiểm tra thuật toán sai số xây dựng trên vi điều khiển. Trong quá trình thiết kế giá trị thực trên LCD sẽ được so sánh với giá trị nhận được trên **máy tính** để xem trong quá trình truyền thông có bị sai khác hay không. Khi triển khai hệ thống thực tế có thể bỏ qua màn hình LCD khi đã hoàn thiện. Để gửi giá trị cường độ ánh sáng từ vi điều khiển về máy tính sẽ dùng Zigbee, Wifi.... Trong quá trình thử nghiệm để tiện trong việc xây dựng và tìm phần cứng, sẽ sử dụng **Bluetooth** để truyền thông. Như vậy vi điều khiển và máy tính nhúng sẽ giao tiếp với nhau thông qua BLE. Trong thực tế trên hệ thống IoT như Smart home cần dùng nhiều thiết bị đo lường như: cảm biến nhiệt độ, độ ẩm, âm thanh, cảm biến ánh sáng,...Sẽ phải dùng máy tính nhúng có cấu hình khá mạnh như Raspberry Pi 3 B+, Qualcomm Snapdragon, BeagleBone Black .. Phần đa đều sử dụng hệ điều hành nhúng Embedded Linux. Do đó để tương thích với các hệ thống trên trong quá trình thử nghiệm sẽ dùng máy tính sử dụng hệ điều hành Linux.

## 2.2 Lưu đồ thuật toán - Flow Chart



## Chương 3. Thiết kế phần cứng

### 3.1 Lựa chọn phần cứng

- Cảm biến ánh sáng : BH1750
- Màn hình hiển thị : LCD16x2
- Modul Bluetooth: HC-05
- Vi xử lý(VXL) : AT89S52

### 3.2 Phân tích datasheet

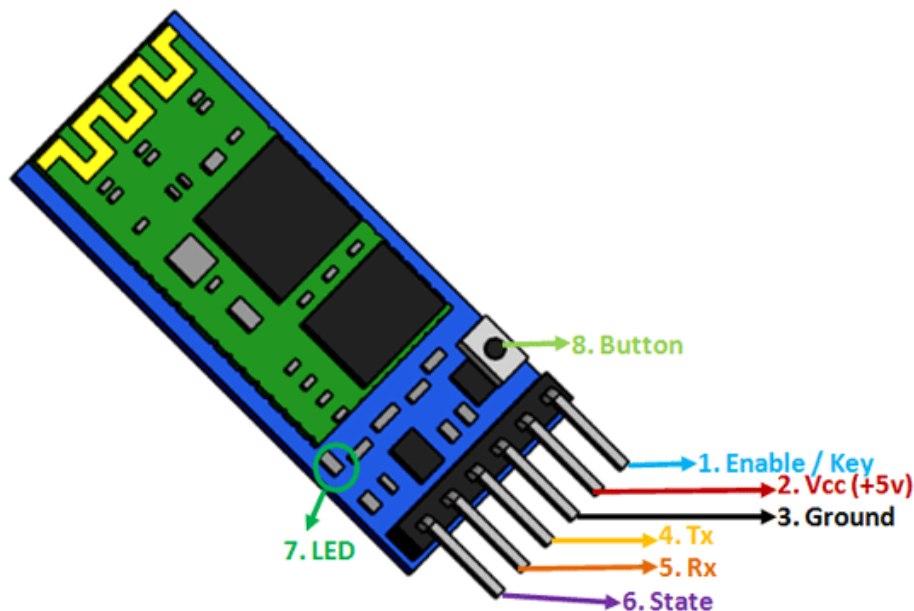
#### 3.2.1 Cảm biến ánh sáng BH1750

- Cảm biến tích hợp ADC và bộ tiền xử lý nên giá trị đầu ra là giá trị trực tiếp cường độ ánh sáng lux.
- Sử dụng chuẩn giao tiếp chuẩn I2C nên dễ dàng giao tiếp với VXL.
- Khoảng đo rộng từ 1-65535 lux.
- Tắt nguồn khi dòng thấp .



- Tần số quét 50Hz/60Hz Lọc ánh sáng nhiều tốt.
- Độ biến thiên đo lường nhỏ ( $\pm 20\%$ ) .
- Ảnh hưởng bởi tia hồng ngoại là rất nhỏ .
- Nguồn cung cấp 2,4-3,6 (V) phù hợp với điện áp của các thành phần khác.

### 3.2.2 Modul Bluetooth

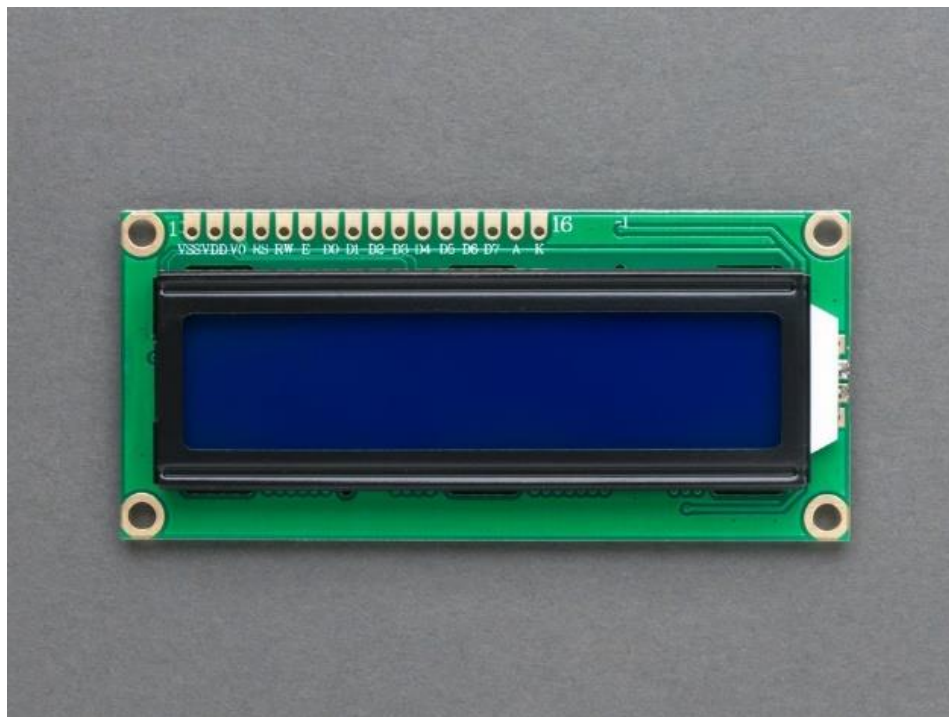


- HC05 là một module có chức năng truyền (nhận) không dây hai chiều (song công hoàn toàn).
- Module có thể sử dụng để liên lạc giữa hai bộ vi điều khiển như arduino hoặc giao tiếp với bất kỳ thiết bị nào có chức năng Blue-tooth như điện thoại hoặc máy tính xách tay



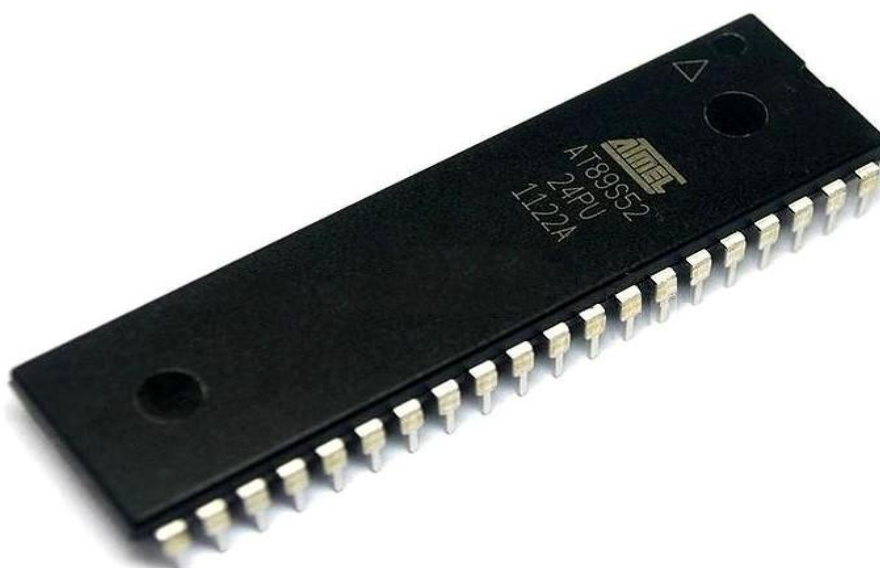
- Điện áp hoạt động từ 3-5,5(V) phù hợp với các chân của các thành phần khác của mạch
- Giao tiếp bằng phương thức UART đặc biệt có thể vừa là Slave vừa là Master không như HC-06 chỉ có chế độ Slave, phù hợp với quá trình gửi và nhận thông tin
- Baudrate UART có thể chọn được: 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

### 3.2.3 Màn hình hiển thị LCD 16X2



- Nguồn cung cấp 3V hoặc 5V phù hợp với nguồn cung cấp cho các thành phần khác của mạch
- Có chế độ hiển thị cả số, chữ và kí tự
- Chế độ hiển thị 2 dòng phù hợp với mục đích
- Có thể giao tiếp 4 bit giúp giảm nối chân, mạch đơn giản hơn

### 3.3.4 Vi xử lý AT89S52





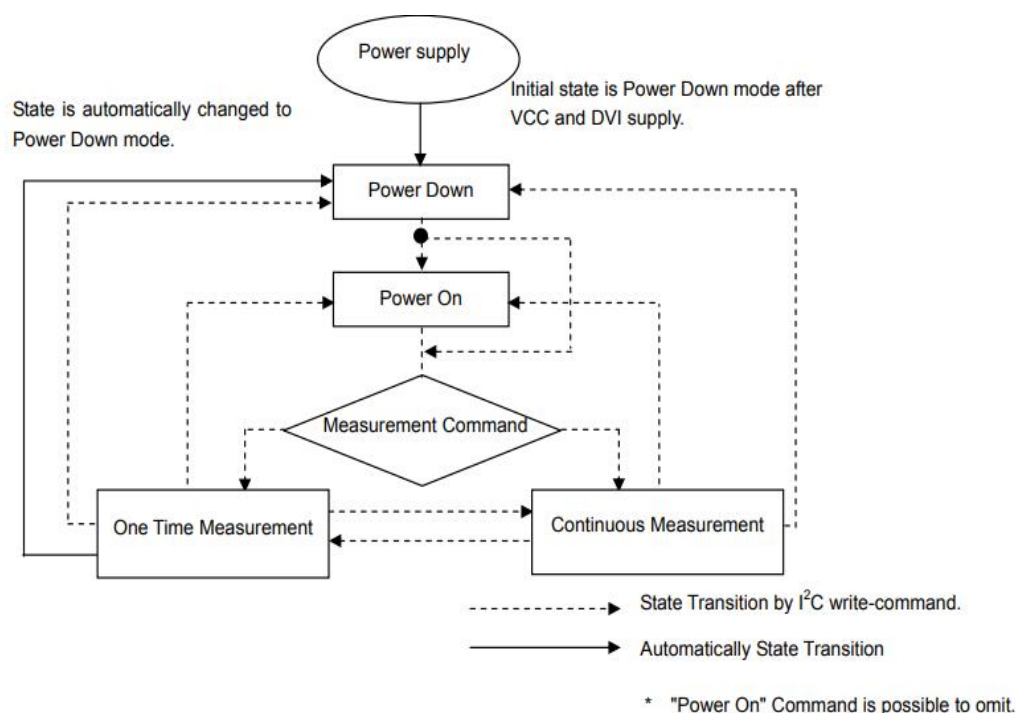
- Là VXL 8 bit họ MCS-51 khá quen thuộc với chúng em
- Nguồn cung cấp là 4V-5,5V cùng nguồn với các thành phần khác
- Có 8k Bộ nhớ flash trong, có 32 I/O pin đủ để kết nối với các thành phần của mạch
- Có kênh kết nối UART nên có thể kết nối với HC05
- Tuy không hỗ trợ chuẩn I2C nhưng có thể lập trình, việc đó giúp chúng em hiểu rõ cách hoạt động của giao tiếp
- Có thể nạp lại bộ nhớ chương trình khi đang được cắm vào socket, không bất tiện như 89Cxx

### 3.3 Chế độ sử dụng

### 3.3.1 Cảm biến ánh sáng BH1750

- Có 2 chế độ địa chỉ :

Chân số	Tên	Chức năng
1	VCC	Chân cung cấp nguồn 2,4-3,6 V DC
2	SCL	Chân clock
3	SDA	Chân data
4	ADD	Chân địa chỉ
5	GND	Chân đất

$$\text{ADDR} = \text{'H'} \text{ ( ADDR } \geq 0.7V_{CC} \text{ )} \rightarrow \text{"1011100"}$$
$$\text{ADDR} = \text{'L'} \text{ ( ADDR } \leq 0.3\text{VCC )} \rightarrow \text{"0100011"}$$


Sau khi cấp nguồn thì trạng thái của cảm biến là Power Down

Muốn nạp lệnh cho Cảm biến ta phải đưa nó về trạng thái Power on (có thể bỏ qua bước này)

Sau đó mình sẽ nạp lệnh đo cho cảm biến, có 2 chế độ đo là:

- Đo 1 lần
- Đo liên tục
- Với chế độ đo 1 lần

Có 3 chế độ đo là:

- One times H-Resolution Mode
- One times H-Resolution Mode2
- One times H-Resolution

Ta có bảng so sánh sau:

	H-Resolution Mode	H-Resolution Mode2	H-Resolution Mode
Lệnh	00100000	00100010	00100011
Độ phân giải ban đầu(1x)	1	0.5	4
Thời gian đo	120	120	16

Sau khi quá trình đo được thực hiện xong thì trạng thái của nó tự động quay lại chế độ Power Down

- Chế độ đo liên tục

Có 3 chế độ đo:

- Continously H-Resolution Mode
- Continously H-Resolution Mode2
- Continously H-Resolution

Ta có bảng so sánh

	H-Resolution Mode	H-Resolution Mode2	H-Resolution Mode
Lệnh	00010000	00010001	00010011
Độ phân giải ban đầu(1x)	1	0.5	4
Thời gian đo	120	120	16

*Kết quả đo:* Giả sử kết quả đo được 8 bit cao là 10010000 và 8 bit thấp là 00010001 thì giá trị của cường độ sáng là:

$$(2^{15}+2^{12}+2^4+2^0)/1.2= 30734 \text{ lux}$$

### 3.3.2 Modul Bluetooth HC05

Số thứ tự	Tên	Chức năng
1	State	Chân test
2	RX	Đây là hai chân UART để giao tiếp module hoạt động ở mức logic 3.3V
3	TX	
4	GND	Nối với chân nguồn GND
5	VCC	chân này có thể cấp nguồn từ 3.6V đến 6V
6	EN	Chân này để chọn chế độ hoạt động AT Mode hoặc Data Mode

Hoạt động ở 2 chế độ command và thông thường

- Chế độ command:

Đầu tiên cấp nguồn và đồng thời đưa chân EN lên mức cao

- Ngắt nguồn và ấn giữ nút trên modul
- Cấp lại nguồn và thả nút bấm
- Chế độ thông thường
- Cấp nguồn và chân EN nối đất, hoạt động ở chế độ mặc định

Có thể là slave hoặc master:

- chế độ slave: cần thiết lập kết nối giữa smartphone, laptop,.. với modul với mật khẩu được cấu hình cho modul
- chế độ master: Modul sẽ tự tìm kiếm và kết nối với các Bluetooth khác.

### 3.3.3 LCD 16x2

Nguồn cung cấp là 4,7 – 5,5 V DC

Hoạt động ở chế độ 4 bit hoặc 8 bit

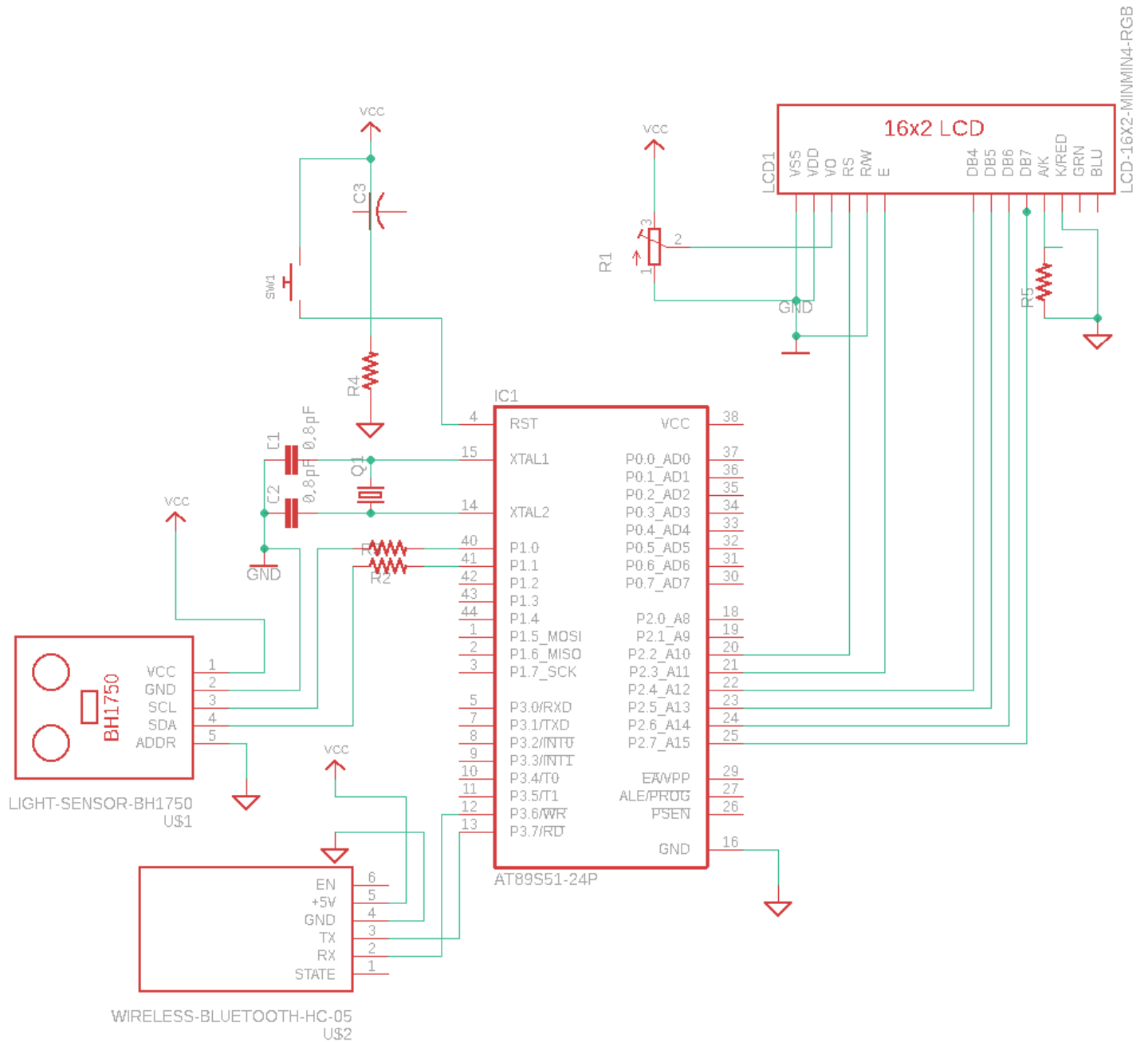
Chân số	Tên	Chức năng
1	VSS	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển

3	VEE	Chân này dùng để điều chỉnh độ tương phản của LC
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD
5	RW	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0-DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào(chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp
7-14	D0-7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này : + Chế độ 8 bit : Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit : Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7

### 3.3.4 Vi xử lý AT89S52

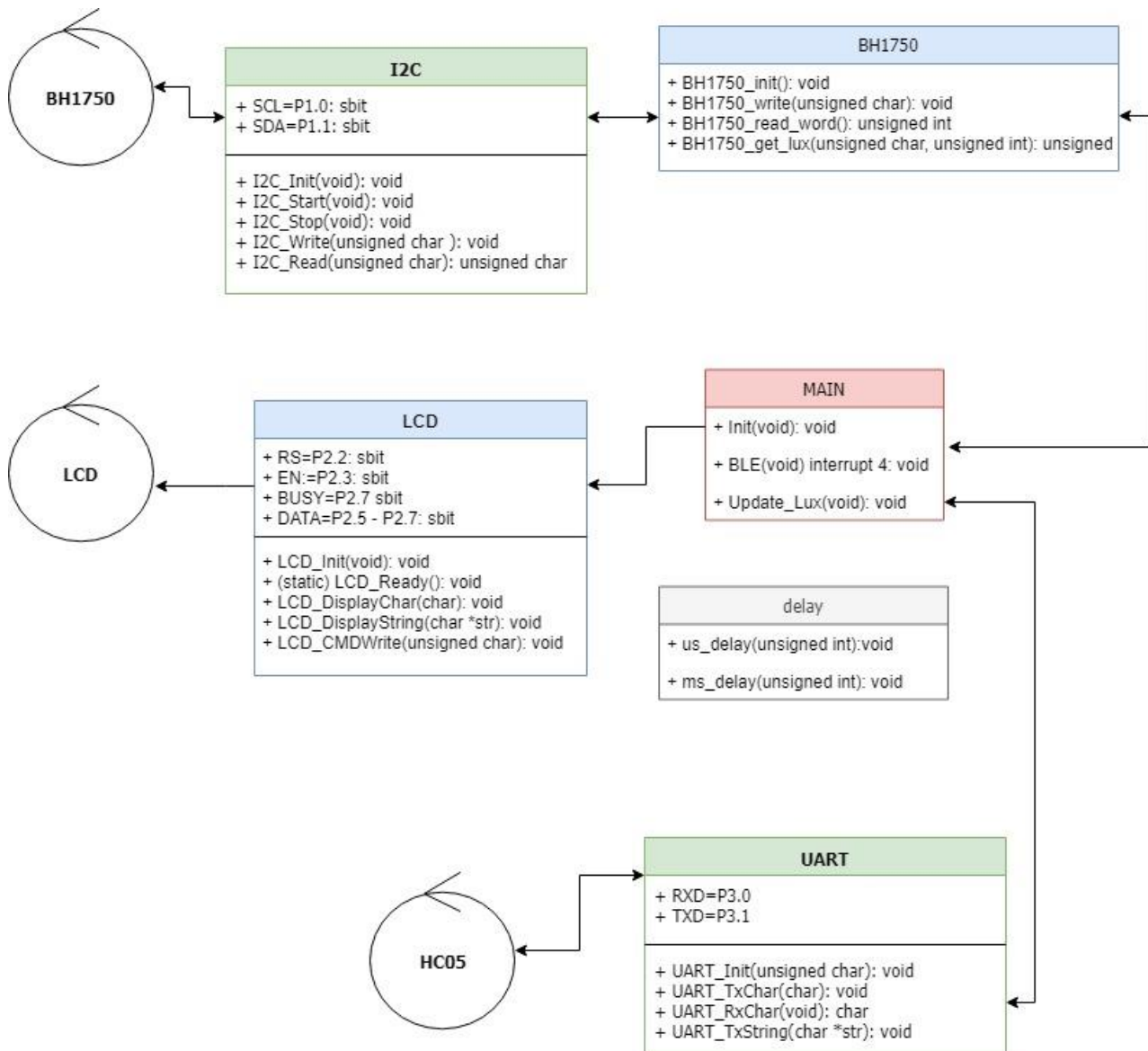
Xem lưu đồ thuật toán

## 3.4 Sơ đồ mạch nguyên lý



# Chương 4. Thiết kế phần mềm

## 4.1 Thiết kế chương trình cho vi điều khiển



Các bước xây dựng và hoạt động của khối vi xử lý

Khởi tạo:

1. Xây dựng các thư viện file.h và file.c tương ứng của nó để thuận tiện cho việc viết chương trình.
2. Tạo ngắt truyền thông nối tiếp bằng bluetooth để có thể trao đổi dữ liệu với máy tính.
3. Khởi tạo LCD, giao thức I2C ở 2 chân P1.0(SCL) và P1.1(SDA), cảm biến ánh sáng BH1750 và truyền thông nối tiếp UART



4. Khởi động chương trình được lập trình trên máy tính có khả năng giao tiếp với vi xử lý bằng giao thức bluetooth có nhiệm vụ nhận dữ liệu mà cảm biến đo được và hiển thị trên màn hình máy tính.

Chương trình chính:

1. Cảm biến ánh sáng được đặt ở chế độ đo liên tục. Mỗi lần đo kết quả sẽ được gửi về vi xử lý thông qua giao thức I2C. Trong giao thức I2C, vi xử lý đóng vai trò là Master, BH1750 đóng vai trò là Slave với địa chỉ là 1011100 (nếu chân ADDR  $\geq 0,7VCC$ )

ST	0100011	0	Ack	00010000	Ack	SP
----	---------	---	-----	----------	-----	----

hoặc 0100011 (nếu chân ADDR  $\leq 0,3VCC$ ). Trong đồ án này chân ADDR nối đất nên địa chỉ Slave của BH1750 là 0100011.

Sơ đồ gửi mode đo liên tục với độ phân giải 1 lux từ vi xử lý tới Bh1750

ST	0100011	1	Ack	High Byte [ 15:8 ]	Ack
----	---------	---	-----	--------------------	-----

Low Byte [ 7:0 ]	Ack	SP
------------------	-----	----

Sơ đồ gửi tín hiệu để đọc dữ liệu trong thanh ghi của BH1750

2. Khi đã có dữ liệu về kết quả đo từ BH1750, dữ liệu này được lưu trong bộ nhớ 16 byte dưới dạng string. Sau đó dữ liệu này gửi tới để hiện lên LCD với mục đích hiển thị và so sánh với dữ liệu được truyền lên máy tính.
3. Sau đó dữ liệu được truyền tới phần mềm trên máy tính thông qua việc giao tiếp bằng giao thức bluetooth. Dữ liệu này sẽ hiển thị lên màn hình máy tính

Các bước cứ lặp lại như trên. Màn hình máy tính sẽ liên tục hiện lên các kết quả mà cảm biến đo được.

## 4.2 Xây dựng chương trình trên máy tính

Chương trình trên máy tính bao gồm hai chức năng cơ bản:

1. Kết nối với module HC05 thông qua truyền thông Bluetooth
2. Hiện thị giá trị cường độ ánh sáng trên màn hình Terminal

Trong quá trình thử nghiệm đề tài có thể bỏ qua thiết kế giao diện cho chương trình. Chương trình sẽ chạy trên Terminal của máy tính chạy hệ điều hành linux.

Để kết nối với HC05 cần phải lập trình Bluetooth. Trước tiên phải cài đặt thư viện phát triển Bluetooth BlueZ (BlueZ development libraries). Sau đó cần phải chọn một giao thức truyền (transport protocol) Bluetooth. Có 3 giao thức phổ biến hay được sử dụng như: RFCOMM, L2CAP with infinite retransmit, L2CAP (0-1279 ms retransmit)...

Requirement	Internet	Bluetooth
Reliable, streams-based	TCP	RFCOMM
Reliable, datagram	TCP	RFCOMM or L2CAP with infinite retransmit
Best-effort, datagram	UDP	L2CAP (0-1279 ms retransmit)

Giao thức RFCOMM cung cấp dịch vụ đảm bảo và tin cậy gần giống như TCP. Mặc dù thông số kỹ thuật nói rõ rằng nó được thiết kế để mô phỏng các cổng nối tiếp RS-232 (để giúp các nhà sản xuất dễ dàng thêm Bluetooth vào các thiết bị có cổng nối tiếp của họ), nhưng khá đơn giản để sử dụng nó trong nhiều tình huống giống như TCP. Vì tính đơn giản và độ tin cậy cao, sẽ dùng giao thức RFCOMM để truyền dữ liệu.

Để tìm ra cách giao tiếp với một máy từ xa, một khi đã biết địa chỉ số và giao thức truyền tải, là chọn số cổng. Hầu như tất cả các giao thức truyền tải Internet được sử dụng phổ biến đều được thiết kế với khái niệm số cổng, do đó nhiều ứng dụng trên cùng một máy chủ có thể đồng thời sử dụng giao thức truyền tải. Bluetooth cũng không ngoại lệ, nhưng sử dụng thuật ngữ hơi khác nhau.

Trong L2CAP, các cổng được gọi là bộ ghép kênh dịch vụ giao thức và có thể nhận các giá

protocol	terminology	reserved/well-known ports	dynamically assigned ports
TCP	port	1-1024	1025-65535
UDP	port	1-1024	1025-65535
RFCOMM	channel	none	1-30
L2CAP	PSM	odd numbered 1-4095	odd numbered 4097 - 32765

trị số lẻ trong khoảng từ 1 đến 32767. Trong RFCOMM, các kênh 1-30 có sẵn để sử dụng. Bộ ghép kênh dịch vụ giao thức (Protocol Service Multiplexer) và kênh điều phục vụ cùng một mục đích chính xác như các cổng thực hiện trong TCP / IP. L2CAP, không giống như RFCOMM, có một loạt các số cổng dành riêng (1-1023) không được sử dụng cho các ứng dụng và giao thức tùy chỉnh.

Lập trình chương trình Bluetooth sử dụng Python – PyBlueZ. Python là một ngôn ngữ hướng đối tượng linh hoạt và mạnh mẽ, cung cấp cú pháp rõ ràng cùng với tích hợp quản lý bộ nhớ để lập trình viên có thể tập trung vào thuật toán trong tay mà không phải lo lắng về rò rỉ bộ nhớ hay các dấu ngoặc. Mặc dù Python có một thư viện tiêu chuẩn lớn và toàn diện, hỗ trợ Bluetooth chưa phải là một phần của phân phối chuẩn.. PyBlueZ là một mô-đun mở rộng Python được viết bằng C, cung cấp quyền truy cập vào tài nguyên Bluetooth của hệ thống theo cách thức mô-đun hướng đối tượng. Nó được viết cho Windows XP (Microsoft Bluetooth stack) và GNU / Linux (BlueZ stack).

Dưới đây là đoạn mã sử dụng để truyền thông Bluetooth giữa Linux và HC05 sử dụng module BlueZ. Mục đích gửi liên tục mã điều khiển để nhận giá trị cường độ ánh sáng từ khối cảm biến.

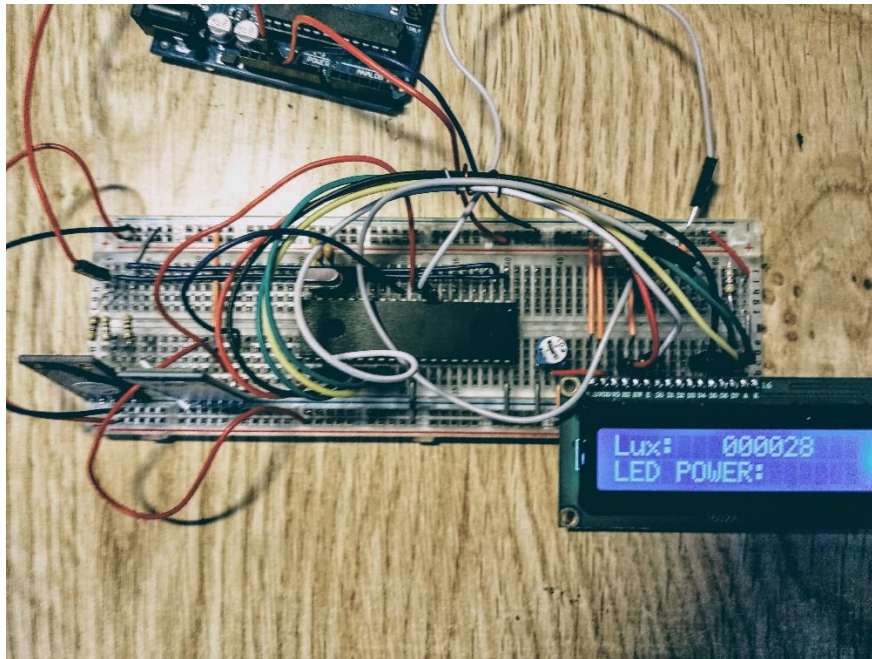
```
1  #!/usr/bin/env python3
2
3  import bluetooth
4  import time
5
6  #BLE Communication
7  bd_addr = "00:21:13:05:E1:45"
8  port = 1
9  sock=bluetooth.BluetoothSocket(bluetooth.RFCOMM )
10 sock.connect((bd_addr,port))
11
12 # Display Lux
13 while True:
14     sock.send("e")
15     time.sleep(.200)
16     lux=str(sock.recv(10))
17     time.sleep(.300)
18     print("Lux value is: " + lux[2:9])
```

Cụ thể đoạn mã này sẽ liên kết cổng Bluetooth với cổng nối tiếp trên Linux. Địa chỉ MAC của HC05: 00:21:13:05:E1:45 sẽ được nhận dạng và kết nối. Và sẽ được kéo về cổng nối tiếp RFCOMM1. Thông qua socket API sử dụng các hàm để gửi và nhận dữ liệu.

# Chương 5. Triển khai thử nghiệm đề tài

## 5.1 Mô hình thử nghiệm

Từ sơ đồ mạch nguyên lý chúng ta dễ dàng xây dựng mạch test. Hình dưới là mạch test đề tài sau khi đã hoàn thành công đoạn thiết kế phần cứng và phần mềm. Độ sáng được update hiển thị liên tục ra LCD và gửi liên tục gửi giá trị cường độ ánh sáng về máy tính. Việc đo đạc liên tục này do script python đã lập trình sẵn. Do đó có thể thay đổi và đo theo ý muốn.



Giá trị cường độ ánh sáng được hiển thị trên Terminal của máy tính. Giá trị này có thể nhận dạng và xử lý được. Khi gửi thông qua Bluetooth giá trị cường độ ánh sáng được gửi theo từng ký tự và mỗi giá trị cường độ tạo thành một chuỗi string được lên khung sẵn. Chúng ta có thể dễ dàng chuyển chuỗi này thành các định dạng toán học để tính toán và xử lý khi cần thiết.

```
import bluetooth
import time

Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000016
Lux value is: 000040
Lux value is: 000464
Lux value is: 000588
Lux value is: 000628
Lux value is: 000540
Lux value is: 000492
Lux value is: 000572
Lux value is: 000584
Lux value is: 000580
Lux value is: 000560
Lux value is: 000512
```

## 5.2 Đánh giá đề tài

Khối cảm biến sử dụng vi điều khiển do đó có thể kết hợp với các cảm biến khác như nhiệt độ, độ ẩm,... Tạo thành một chuỗi cảm biến phục vụ cho các hệ thống smarthome.... Giá trị cường độ ánh sáng được xử lý bằng script python nên dễ dàng kết nối với internet, cloud... Một ngôn ngữ rất hay được sử dụng để lập trình server, socket... Đặc biệt là cloud MQTT một trong những cloud được sử dụng phổ biến trong các dự án Internet of things cũng hỗ trợ và có module MQTT Python.

## Lời kết

---

Một lần nữa chúng tôi xin cảm ơn tất cả những người đã đồng hành cùng trong quá trình phát triển và xây dựng đề tài.

## Tài liệu tham khảo

---

1. [http://jsjyl.chd.edu.cn/The\\_8051\\_Microcontroller\\_and\\_Embedded\\_Systems\\_Using\\_Assembly\\_and\\_C.pdf](http://jsjyl.chd.edu.cn/The_8051_Microcontroller_and_Embedded_Systems_Using_Assembly_and_C.pdf)
2. <http://www.electronicaestudio.com/docs/istd016A.pdf>
3. <https://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>
4. <https://people.csail.mit.edu/albert/bluez-intro/>
5. <http://www.ti.com/lit/an/slva704/slva704.pdf>
6. <http://hccc.ee.ccu.edu.tw/courses/bt/vg/rfcomm.pdf>
7. <https://doc.lagout.org/programming/python/Head%20First%20Python%2C%20First%20Edition%20%282010%29.pdf>
8. <https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>







