# khung doi TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TRUYỂN THÔNG THÁI NGUYÊN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Khoa công nghệ điện tử và truyền thông

**

**BÁO CÁO TIỂU LUẬN MÔN HỌC**

**KỸ THUẬT VI XỬ LÝ VÀ ỨNG DỤNG**

**Đề tài**

**MẠCH CẢNH BÁO AN NINH SỬ DỤNG 8051**

Nhóm sinh viên thực hiện: Lê Minh Hữu \*.

Dương Văn Nam.

Lê Thanh Lam.

Lớp: KTMT K19A.

Giáo viên giảng dạy: Ths.Mai Thị Kim Anh.

**Thái Nguyên, tháng 3 năm 2022**

**CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI**

Hiện nay có rất nhiều các loại vi điều khiển của các hang khác nhau, ví dụ như họ 8051, AVR của hang Atmel, PIC của hang Microchip… Trong các loại vi điều khiển thì phổ biến nhất ở Việt Nam hiện nay là chip vi điều khiển thuộc họ 8051. Do họ 8051 là một trong những chip điều khiển xuất hiện sớm nhất nên nó là nền tảng cho sự phát triển của các họ vi điều khiển thế hệ sau. Hơn nữa nó được sử dụng trong hầu hết các loại đồ điện tử dân dụng ở nước ta hiện nay cho nên lập trình cho vi điều khiển đã được đưa vào chương trình học cho sinh viên.

Cuộc sống của chúng ta tồn tại cùng lúc với nhiều thực thể vật lý, những thứ chúng ta nhận biết được như là các động cơ học, tác dụng nhiệt, của ánh sáng, của âm thanh, mùi vị… Nhằm mục đích con người nhận biết rõ hơn các vận động trên cũng như nghiên cứu ra các loại cảm biến. Cảm biến là thiết bị điện tử cảm nhận những thay đổi từ môi trường bên ngoài và biến đổi thành các tín hiệu để điều khiển các thiết bị khác. Ngày nay có rất nhiều loại cảm biến đã được tạo ra, như cảm biến ánh sáng, cảm biến nhiệt độ, cảm biến siêu âm, cảm biến hồng ngoại… Trong đề tài này, chúng tôi dung cảm biến để phát hiện các vật thể than nhiệt có chuyển động qua lại hay còn gọi là PIR. Ứng dụng là hiện nay ở nhiều khu vực xảy ra tình trạng mất trộm rất nhiều cho nên ý tưởng của chúng tôi là làm mạch chống trộm đơn giản có thể gắn ở cửa ra vào, có chi phí thấp và để hiểu rõ hơn về nguyên lý hoạt động và muốn phát triển thêm về mô hình này, chúng tôi đã chọn đề tài cho bài tiểu luận này là thiết kế mạch chống trộm dùng PIR và 89C51.

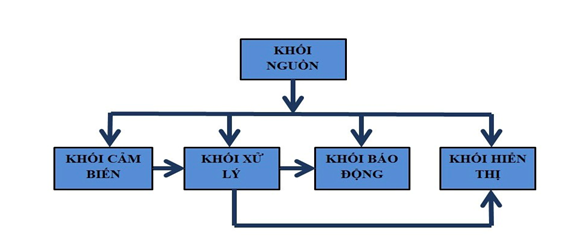
**CHƯƠNG 2. PHÂN TÍCH CÔNG NGHỆ THIẾT KẾ**

**2.1. Yêu cầu thiết kế**

Thiết kế mạch cảnh báo an ninh sử dụng 8051 kết hợp với cảm biến chuyển động hồng ngoại, hiển thị lên màn hình lcd 16x02 phát cảnh báo qua còi chíp và đèn led khi phát hiện vật thể phát tia hồng ngoại chuyển động. Thiết kế mạch khoa học, hiệu quả, tiết kiệm được chi phí thi công, dễ dàng sửa chữa, bảo trì.

**2.2. Giải pháp thiết kế**

**2.2.1. Sơ đồ khối**



*Hình 2.1.**Sơ đồ khối của hệ thống.*

**2.2.2. Phân tích chức năng và nhiệm vụ các khối**

* **Khối nguồn:** Có nhiệm vụ cung cấp nguồn 5v cho toàn mạch bao gồm các khối cảm biến, khối xử lý, khối báo động, khối hiển thị.
* **Khối cảm biến:** Module cảm biến PIR có nhiệm vụ cảm nhận sự xuất hiện của các thân nhiệt và xuất ra mức 0 hoặc mức 1 cho đầu vào của at89s51 trong khối xử lý để xử lí và đưa tới khối báo động
* **Khối xử lí:** Là vi điều khiển AT89S52 có nhiệm vụ xử lí tín hiệu từ cảm biến và đưa ra tín hiệu đến khối báo động
* **Khối báo động :** Khi có tín hiệu báo động được điều khiển từ khối xử lý còi chip sẽ phát ra báo động đồng thời đèn led đỏ và trắng bật.
* **Khối hiển thị:** Tiếp nhận thông tin vi điều khiển để hiển thị kết quả. Màn hình LCD sẽ hiển thị trạng thái khi an toàn hoặc hiển thị cảnh báo bị xâm nhập.

**2.3. Lựa chọn linh kiện**

**2.3.1. Giới thiệu về vi điều khiển AT89S52**

1. ***Tổng Quan về AT89S52***

AT89S52 là họ IC vi điều kliiền đo hãng Atmel sản xuất. Các sản phẩm AT89S52 thích hợp cho những ứng dụng điều khiển. Việc xử lý trên byte và các toán số học ở cấu trúc đữ liệu nhỏ được thực hiện bằng nhiều chế độ truy xuất đữ liệu nhanh trên RAM nội. Tập lệnh cung cấp một bảng tiện dụng của nhũng lệnh số học 8 bit gồm cả lệnh nhân và lệnh chia. Nó cung cấp những hỗ trợ mở rộng trên chip dùng cho những biến một bit như là kiểu dữ liệu riêng biệt cho phép quản lý và kiểm tra bit trực tiếp trong hộ thống điều khiển.

AT89S52 cung cấp những đặc tính chuẩn như: 8 KByte bộ nhớ chi đọc có thể xóa và lập trình nhanh (EPROM), 128 Byte RAM, 32 đường I/O, 3TIMER/COUNTER 16 Bit, 5 vectơ ngắt có cấu trúc 2 mức ngắt, một Port nối tiếp bán song công, 1 mạch dao động tạo xung Clock và bộ dao động ON-CHIP.

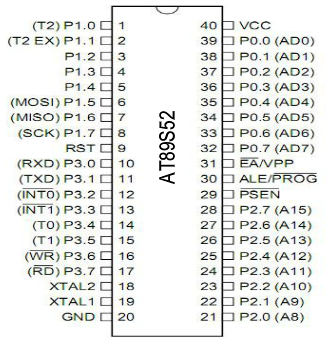
Các đặc điểm của Chip AT89S52 được tóm tắt như sau:

* 8 KByte bộ nhớ có thế lập trình nhanh, có khả năng tới 1000 chu kỳ ghi/xoá
* Tần số hoạt động từ: 0Hz đến 24 MHz
* Mức khóa bộ nhớ lập trình
* Bộ Timer/counter 16 Bit
* 128 Byte RAM nội.
* 4 Port xuất /nhập I/O 8 bit.
* Giao tiếp nối tiếp.
* 64 KB vùng nhớ mã ngoài
* 64 KB vùng nhớ dữ liệu ngoại.
* 4 ps cho hoạt động nhân hoặc chia.

1. **Mô tả chân AT89S52**

* Sơ đồ chân AT89S52

Mặc dù các thành viên của họ 8051 (ví đụ 8751, 89S52, 89C51, DS5000) đều có các kiểu đóng vỏ khác nhau, chẳng hạn như hai hàng chân DIP (Dual In-Line Pakage), dạng vỏ dẹt vuông QPF (Quad Flat Pakage) và dạng chip không có chân đỡ LLC (Leadless Chip Carrier) thì chúng đều có 40 chân cho các chức năng khác nhau như vào ra I/O, đọc RD, ghi WR, địa chỉ, dữ liệu và ngắt. Cần phải lưu ý một số hãng cung cấp một phiên bản 8051 có 20 chân với số yêu câu thấp hơn. Tuy nhiên hầu hết các nhà phát triển sử dụng chip đóng vỏ 40 chân với hai hàng chân DIP.



*Hình 2.2: Vi xử lý AT 89S52*

* Chức năng các chân at89S52

***+ Port 0:*** từ chân 32 đến chân 39 (PO.O \_P0.7). Port 0 có 2 chức năng: trong các thiết kế cỡ nhỏ không dùng bộ nhớ mở rộng nó có chức năng như các đường I/O, đối với thiết kế lớn có bộ nhớ mở rộng nó được kết hợp giữa bus địa chỉ và bus dữ liệu.

***+ Port 1:*** từ chân 1 đến chân 9 (P1.0 \_ P1.7). Port 1 là port I/O dùng cho giao tiếp với thiết bị bên ngoài nếu cần.

**+ Port 2:** từ chân 21 đến chân 28 (P2.0 \_P2.7). Port 2 là một port có tác dụng kép dùng như các đường I/O hoặc là byte cao của bus địa chỉ đối với các thiết bị dùng bộ nhớ mở rộng.

***+ Port 3:*** từ chân 10 đến chân 17 (P3.0 \_ P3.7). Port 3 là port có tác dụng kép. Các chân của port này có nhiều chức năng, có công dụng chuyển đổi có liên hệ đến các đặc tính đặc biệt cùa 89S52 như bảng sau:

|  |  |  |
| --- | --- | --- |
| Bit | Tên | Chức năng chuyền đối |
| P3.0 | RXĐ | Ngõ vào đừ liệu nối tiếp. |
| P3.1 | TXĐ | Ngõ xuất đừ liệu nối tiếp. |
| P3.2 | INTO | Ngõ vào ngắt cứng thứ 0. |
| P3.3 | INTI | Ngõ vào ngắt cứng thứ 1. |
| P3.4 | TO | Ngõ vào TIMER/ COUNTER thứ 0. |
| P3.5 | Tl | Ngõ vào của TIMER/ COUNTER thứ 1. |
| P3.6 | WR | Tín hiệu ghi dữ liệu lên bộ nhớ ngoài. |
| P3.7 | RĐ | Tín hiệu đọc bộ nhớ dữ liệu ngoài. |

***+ PSEN (Program store enable):*** Là tín hiệu ngõ ra có tác dụng cho phép đọc bộ nhớ chương trình mở rộng và thường được nối đến chân OE của Eprom cho phép đọc các byte mã lệnh.

PSEN ở mức thấp trong thời gian 89S52 lấy lệnh. Các mã lệnh của chương trình được đọc từ Eprom qua bus dữ liệu, được chốt vào thanh ghi lệnh bên trong 89S52 để giải mã lệnh. Khi 89S52 thi hành chương trình trong ROM nội, PSEN ở mức cao.

**+ *ALE (Address Latch Enable):*** Khi 89S52 truy xuất bộ nhớ bên ngoài, Port 0 có chức năng là bus địa chỉ và dữ liệu do đó phải tách các đường dữ liệu và địa chi. Tín hiệu ra ALE ờ chân thứ 30 dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và đữ liệu khi kết nối chúng với IC chốt.

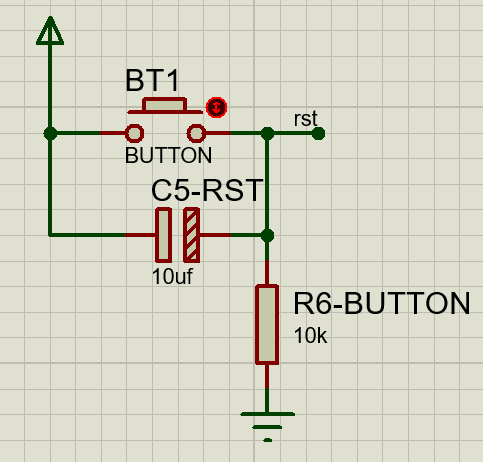
Tín hiệu ở chân ALE là một xung trong khoảng thời gian port 0 đóng vai trò là địa chỉ thấp nên chốt địa chi hoàn toàn tự động.

***+ EA (External Access):*** Tín hiệu vào EA (chân 31) thường được mắc lên mức 1 hoặc mức 0. Nếu ờ mức 1, 89S52 thi hành chương trình từ ROM nội. Nếu ở mức 0, 89S52 thi hành chương trình từ bộ nhớ mở rộng. Chân EA được lấy làm chân cấp nguồn 21V khi lập trình cho Eprom trong 89S52.

***+ RST (Reset):*** Khi ngõ vào tín hiệu này đưa lên mức cao ít nhất 2 chu kỳ máy, các thanh ghi bên trong được nạp những giá trị thích hợp đế khởi động hệ thống. Khi cấp điện mạch phải tự động reset.

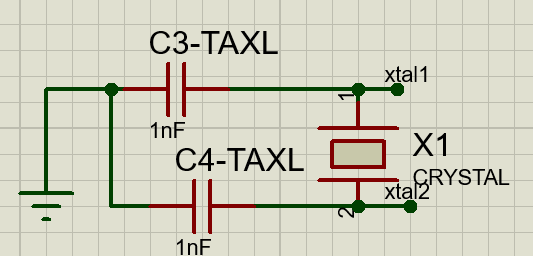
Các giá trị tụ và điện trở được chọn là:

R=10K; c=10uF; nút bấm.



*Hình 2.3: Sơ đồ nối chân reset cho AT89S52..*

***+ Các ngõ vào bộ dao động XI, X2:***Bộ tạo dao động được tích hợp bên trong at89S52. Khi sử dụng at89S52, người ta chỉ cần nối thêm thạch anh và các tụ. Tần số thạch anh tùy thuộc vào mục đích của người sử dụng.



*Hình 2.4: Ngõ tạo xung dao động.*

**2.3.2. Tìm hiểu về cảm biến PIR**

**Tìm hiểu về cảm biến PIR**

Cảm biến thân nhiệt chuyển động PIR (Passive infrared sensor) HC-SR505 Mini có kích thước nhỏ gọn chỉ 10mm, được sử dụng để phát hiện chuyển động của các vật thể phát ra bức xạ hồng ngoại chuyển động: con người, con vật, các vật phát nhiệt,...

Cảm biến thân nhiệt chuyển động PIR HC-SR505 gồm có 3 chân: 1 chân nối với nguồn VCC, 1 chân nối GND, chân còn lại là chân tín hiệu (chân out). PIRsẽ xuất ra tín hiệu mức cao (High) khi phát hiện vật thể nhiệt chuyển động trong vùng quét, tín hiệu này sau đó sẽ được giữ ở mức cao trong khoảng thời gian trễ T sau khi kích hoạt, lúc này nếu cảm biến vẫn bắt được tín hiệu sẽ vẫn duy trì chân tín hiệu mức cao trong thời gian trễ T, chỉ khi trong khoảng thời gian trễ T mà cảm biến không bắt được tín hiệu thì chân tín hiệu cảm biến mới trở về mức thấp (Low).



*Hình 2.5: Cảm biến PIR.*

**Thông số kỹ thuật:**

* Điện áp hoạt động: 4.5~20VDC
* Dòng tiêu thụ: <60uA
* Mức tín hiệu: High 3.3V / Low 0V
* Trigger: repeatable trigger
* Thời gian trễ T sau khi kích hoạt: 8s + -30%
* Góc quét: Max 100 độ (hình nón có tâm là cảm biến)
* Khoảng cách bắt: 3 meters
* Đường kính thấu kính: 10mm
* Kích thước: 10 x 23mm

**2.3.3. Các linh kiện khác sử dụng trong mạch:**

1. **Điện trở**

Điện trở là đại lượng vật lý đặc trưng cho tính chất cản trở dòng điện của một vật thể dẫn điện. Nó được định nghĩa là tỉ số của hiệu điện thế giữa hai đầu vật thể đó với cường độ dòng điện đi qua nó:



Hình 2.6: Điện trở.

u : là hiệu điện thế giữa hai đầu vật dẫn điện, đo bằng vôn (V).

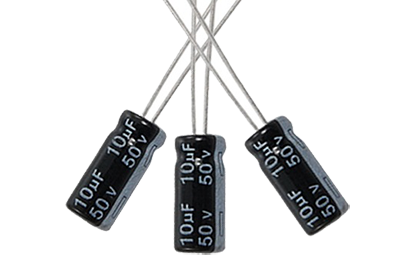
I : là cường độ dòng điện đi qua vật dẫn điện, đo bằng ampe (A).

R : là điện trở của vật dẫn điện, đo bang Ohm (Q).

Mục đích: Điện trở được dùng để chế tạo ra địch mức điện áp giữa hai điểm khác nhau của mạch.

1. **Tụ điện**

Tụ điện phẳng gồm hai bàn phẳng kim loại diện tích đặt song song và cách nhau một khoảng d.

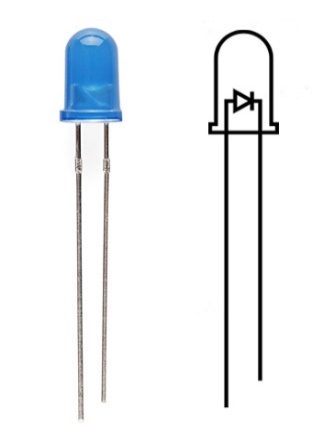
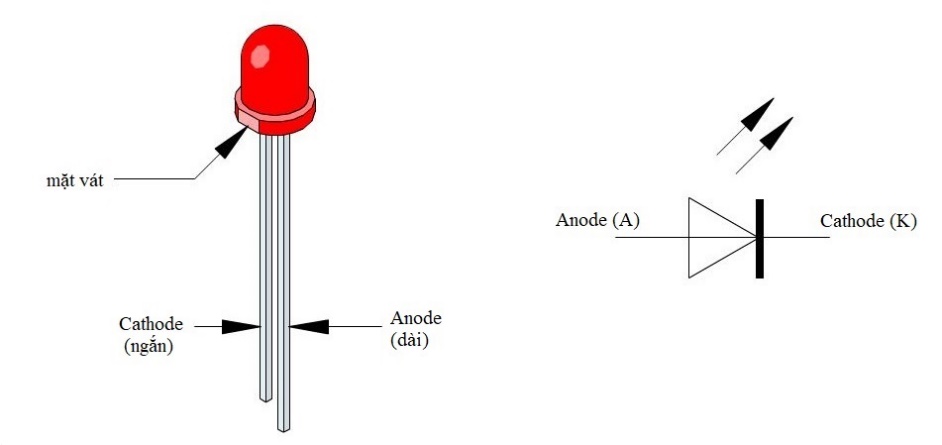
*Hình 2.7: Tụ điện.*

Cường độ điện trường bên trong tụ có trị số:



£ 0 = 8.86.1012 c2/ N.m2 là hằng số điện môi của chân không.

£ là hằng số điện môi tương đối của môi trường; đối với chân không £ =1, giấy tầm dầu = 3,6; gốm = 5,5; mica = 4-5-5

1. **Led**

##### *Hình 2.7: Hình ảnh thực và kí hiệu của LED.*

LED (Light Emitting Diode), diode phát quang, là diode có khả năng phát ra ánh sáng. LED cũng được cấu tạo từ một khối bán dẫn loại p ghép nối với một khối bán dẫn loại n bằng công ngh

Khi phân cực thuận bằng một điện áp thích hợp, LED sẽ phát ra ánh sáng. Ánh sáng phát ra có thể là hồng ngoại, ánh sáng nhìn thấy hay tử ngoại phụ thuộc vào hợp chất hóa học bên trong nó

* Cách xác định chân cực:
* Dựa vào độ dài chân cực (chưa bị chỉnh sửa): Chân dài là chân Anode (+), chân ngắn hơn là chân Cathode (-).
* Dựa vào mặt vát : Chân gần mặt vát trên thân LED hơn là chân Cathode (-), chân còn lại là Anode (+).
* Dựa vào bản cực: Chân nào có bản cực lớn hơn là chân Cathode (-), chân còn lại là Anode (+).

**d.** **Nguồn addapter 5v**



##### *Hình 2.8: Hình ảnh thực tế của nguồn adapter.*

Adapter là bộ chuyển đổi nguồn điện giữa các thiết bị điện tử sử dụng các nguồn điện khác 220V xuống một điện áp thấp hơn. Nói một cách dễ hiểu là chuyển đổi các thiết bị từ không tương thích trở nên tương thích với nhau được gọi chung là adapter

Các thiết bị điện tử sẽ được hoạt động một cách ổn định khi được adapter cung cấp một nguồn điện thích hợp.

Thông qua adapter, nguồn đầu vào AC (xoay chiều) sẽ được biến đổi thành nguồn ra DC (một chiều). Adapter sẽ hoạt động một cách ổn định nếu như bạn cung cấp cho nó một nguồn điện đầu vào từ 110 – 220V thành đầu ra chuẩn 5V.

**e. Nút bấm**

##### *Hình 2.9: Hình ảnh thực tế nút bấm.*

Nút nhấn là một loại linh kiện dùng để chuyển đổi trạng thái được xác lập trước theo người sử dụng.

**f. Công tắc**



##### *Hình 2.10: Hình ảnh thực tế công tác.*

Công tắc: Bật/ tắt cảm biến pir. Công tắc bật lên PIR không hoạt động, hệ thống ở trạng thái bình thường an toàn.

**g. Còi chip**

- Buzzer hay còn gọi là còi chíp, là thiết bị phát ra âm thanh (tiếng bíp bíp) hay dùng trong các mạch điện tử.  
- Cấu tạo của Buzzer gồm 2 chân, chân dài là chân (+) và chân ngắn là chân (-). Trong quá trình sử dụng cần chú ý mắc đúng chân để tránh làm hỏng buzzer.  
- Ứng dụng này sử dụng Buzzer 5V, có khả năng phát ra âm thanh có tần số tối đa 2.5kHz.



##### *Hình 2.11: Hình ảnh thực tế còi chip.*

**h. Màn hình LCD 1602**

[Màn hình LCD 1602 xanh lá](https://nshopvn.com/product/man-hinh-lcd-1602-xanh-la/) sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến, nhiều code mẫu và dễ dàng sử dụng hơn nếu đi kèm mạch chuyển tiếp I2C

### **THÔNG SỐ KỸ THUẬT**

* Điện áp hoạt động là 5V.
* Kích thước: 80 x 36 x 12.5mm.
* Chữ trắng, nền xanh dương.
* Khoảng cách giữa hai chân kết nối là 0.1 inch tiện dụng khi kết nối với Breadboard.
* Tên các chân được ghi ở mặt sau của màn hình LCD hổ trợ việc kết nối, đi dây điện.
* Có đèn led nền, có thể dùng biến trở hoặc PWM điều chình độ sáng để sử dụng ít điện năng hơn.
* Có thể được điều khiển với 6 dây tín hiệu.
* Có bộ ký tự được xây dựng hỗ trợ tiếng Anh và tiếng Nhật, xem thêm HD44780 datasheet để biết thêm chi tiết.



##### *Hình 2.12: Hình ảnh thực tế LCD.*

**2.4. Ngôn ngữ lập trình và phần mềm phụ trợ**

***2.4.1. Ngôn ngữ lập trình***

Để lập trình cho vi điều khiển chúng ta có nhiều phương pháp cũng như ngôn ngữ, nhưng phổ biến nhất vẫn là ngôn ngữ C. C là một ngôn ngữ lập trình bậc cao, khi sử dụng người lập trình không cần hiểu sâu sắc về cấu trúc của bộ vi điều khiển, có thể sử dụng các đoạn chương trình có sẵn đã xây dụng trước đó dễ dàng hơn mà chỉ cần sửa chữa một phần hoặc có thể sử dụng toàn bộ.

Khi sử dụng ngôn ngữ C người lập trình sẽ phải nắm rõ một số cú pháp khai báo, nắm rõ về các kiểu dữ liệu.

-Kiểu dữ liệu:

|  |  |  |
| --- | --- | --- |
| Kiểu | Số Byte | Khoảng giá trị |
| Char | 1 | -128-+127 |
| Unsigned char | 1 | 0-255 |
| Int | 2 | -32768- +32767 |
| Unsigned int | 2 | 0-65535 |
| Long | 4 | -2147483648 - +2147483647 |
| Unsigned long | 4 | 0 – 4294697295 |
| Float | 4 |  |

- Khai báo biến:

Cú pháp: Kiểu\_dữ\_liệu Vùng\_nhớ Tên\_biến\_at\_Địa\_chir;

Ví dụ: Unsigned char data x;

- Định nghĩa lại khiểu:

Cú pháp: typedef Kiểu\_dữ\_liệu Tên\_biến;

Ví Dụ: typerdef int n5[5];

- Khai báo mảng:

Cú pháp: Tên\_kiểu Vùng\_nhớ Tên\_mảng[số\_phần\_tử];

Khi bỏ trống số phần tử mảng ta sẽ có mảng có số phần tử bất kì.

Ví dụ: Unsigned data a[5], b[2][3];

- Khai báo biến con trỏ:

Cú pháp: Kiểu\_dữ\_liệu Vùng\_nhớ \*Tên\_biến;

Ví dụ: int \*int\_prt;

Long data \*long\_prt;

Cấu trúc của một trương trình C gồm 4 phần chính:

1. Khai báo chỉ thị tiền xử lý
2. Khai báo các biến toàn cục
3. Khai báo nguyên mẫu các hàm
4. Xây dựng các hàm và chương chình chính.

Ví dụ:

#include<regx51.h>

#include<string.h>

#define Led1 P1\_0

Unsigned char code Led\_arr[3];

Unsigned char data dem;

Unsigned int xdata X;

Void delay(unsigned int n);

Bit kiemtra(unsigned int a);

Void delay(unsigned int n);

{

Khai báo cục bộ;

Chương chình trễ;

}

Void main()

{

Khai báo biến cục bộ;

Trương trình chính;

}

Bit kiemtra(unsigned int a)

{

Khai báo biến cục bộ;

Chương trình kiểm tra biến a;

}

- Ngoài ra trong chương trình còn có các chỉ thị tiền xử lý và chú thích.

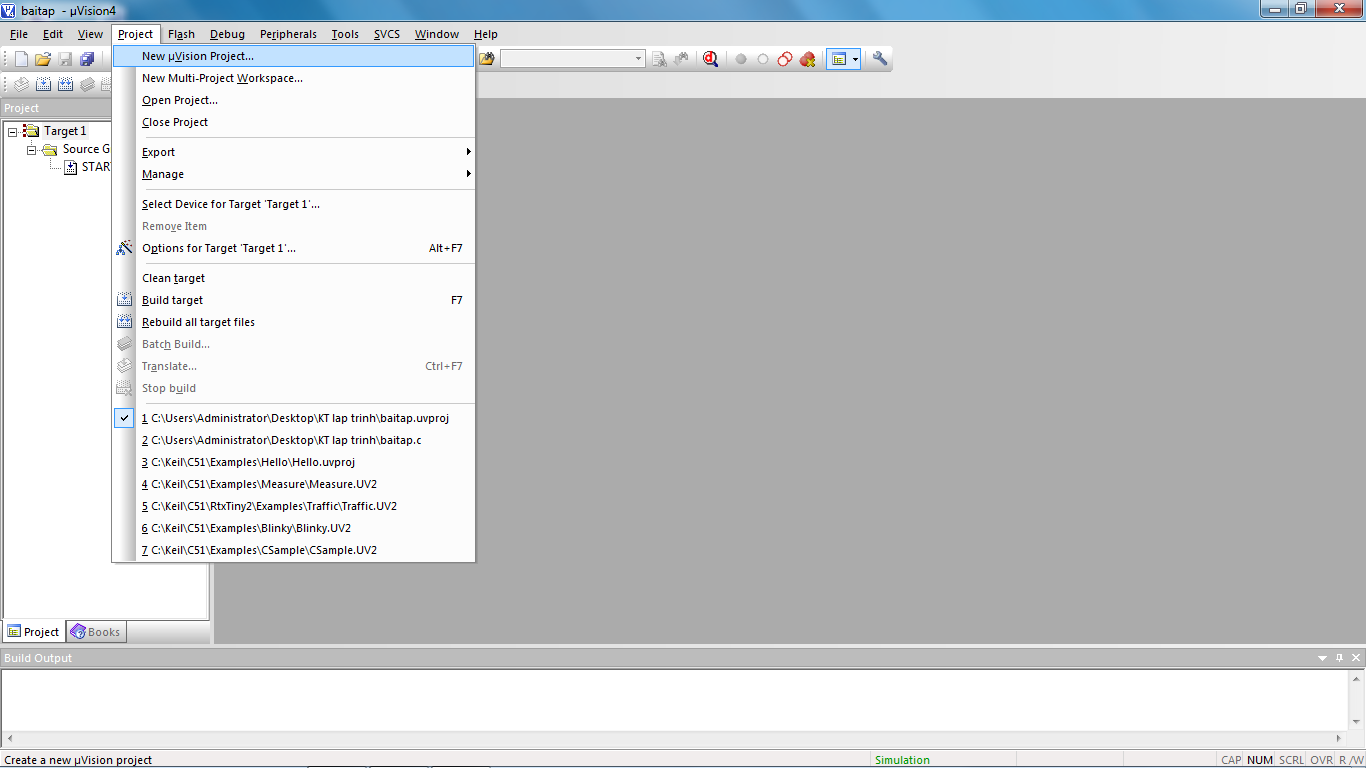
***2.4.2. Phần mềm phụ trợ***

Keil C hay còn gọi là Keil 8051 là 1 công cụ lập trình hỗ trợ khá đầy đủ để người dùng soạn thảo và biên dịch chương trình dành cho các vi điều khiển thuộc họ 8051 bằng ngôn ngữ C và Assembly.

Sử dụng Keil C để viết, biên dịch chương trình cho 8051 cần thực hiện các bước:

**B1:** Mở Keil C và tạo 1 project mới:

* Cài đặt và chạy Keil C, vào Project chọn New Project.



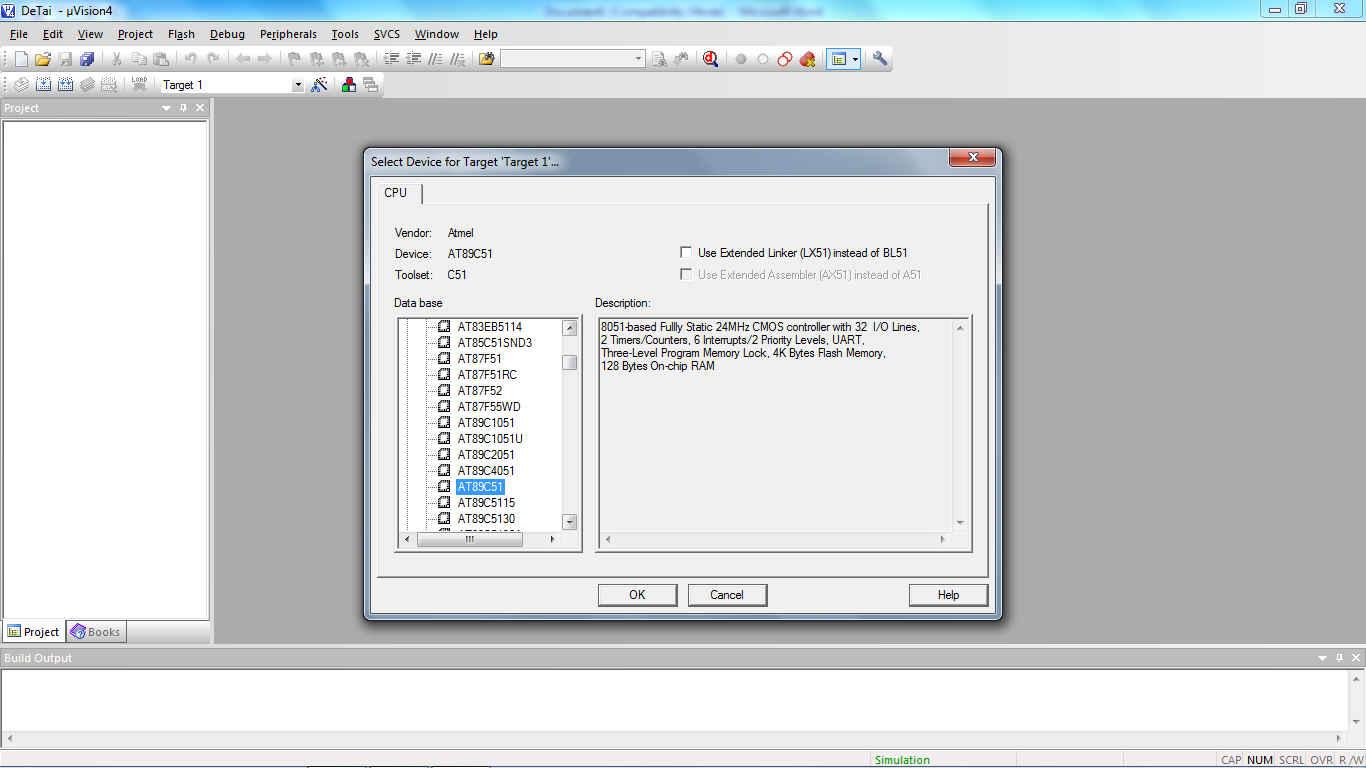
*Hình 2.13: Tạo project*

* Chọn đường dẫn đến thư mục để lưu và đặt tên cho Project.



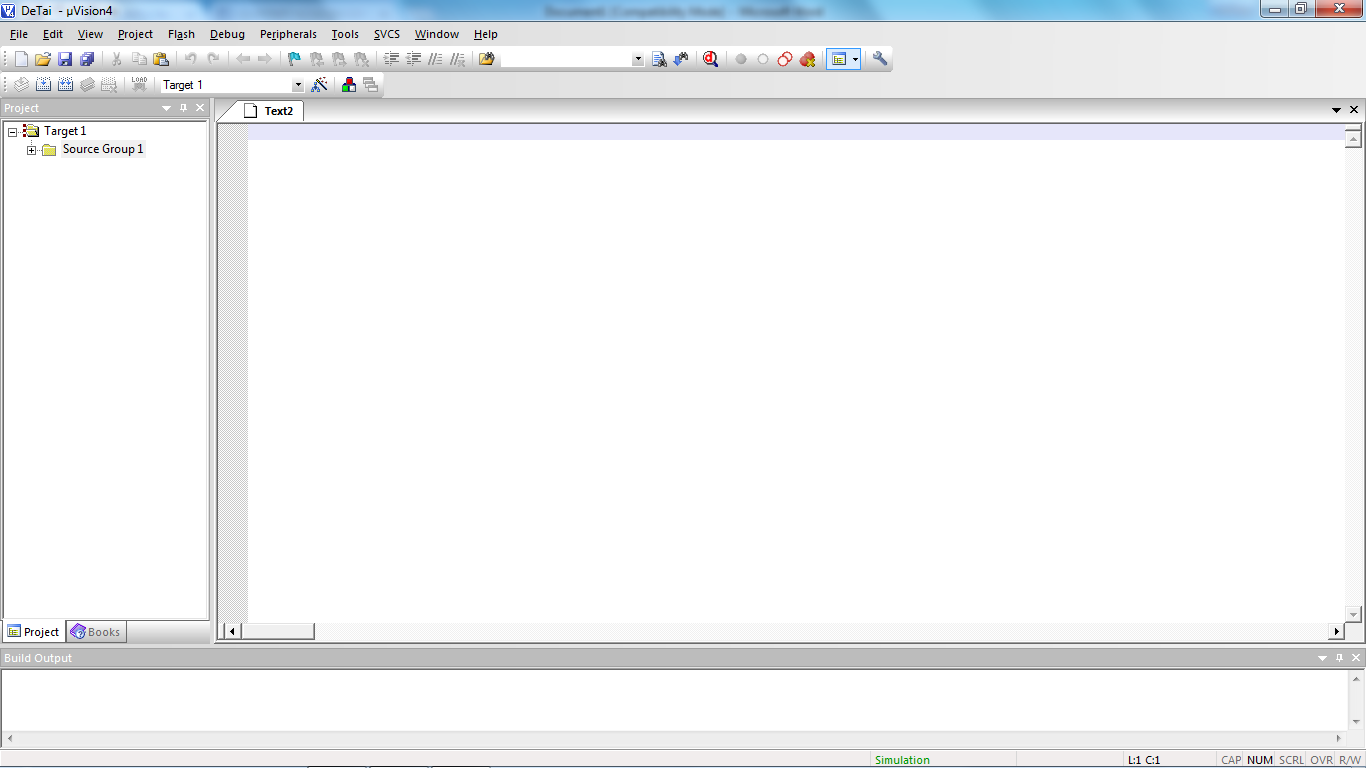
*Hình 2.14: Đặt tên project trong keil c*

* Chọn loại Chip: chọn Atmel => AT89C51, chọn Yes.



*Hình 2.15: Chọn chip vi điều khiển*

* Vào file => New.

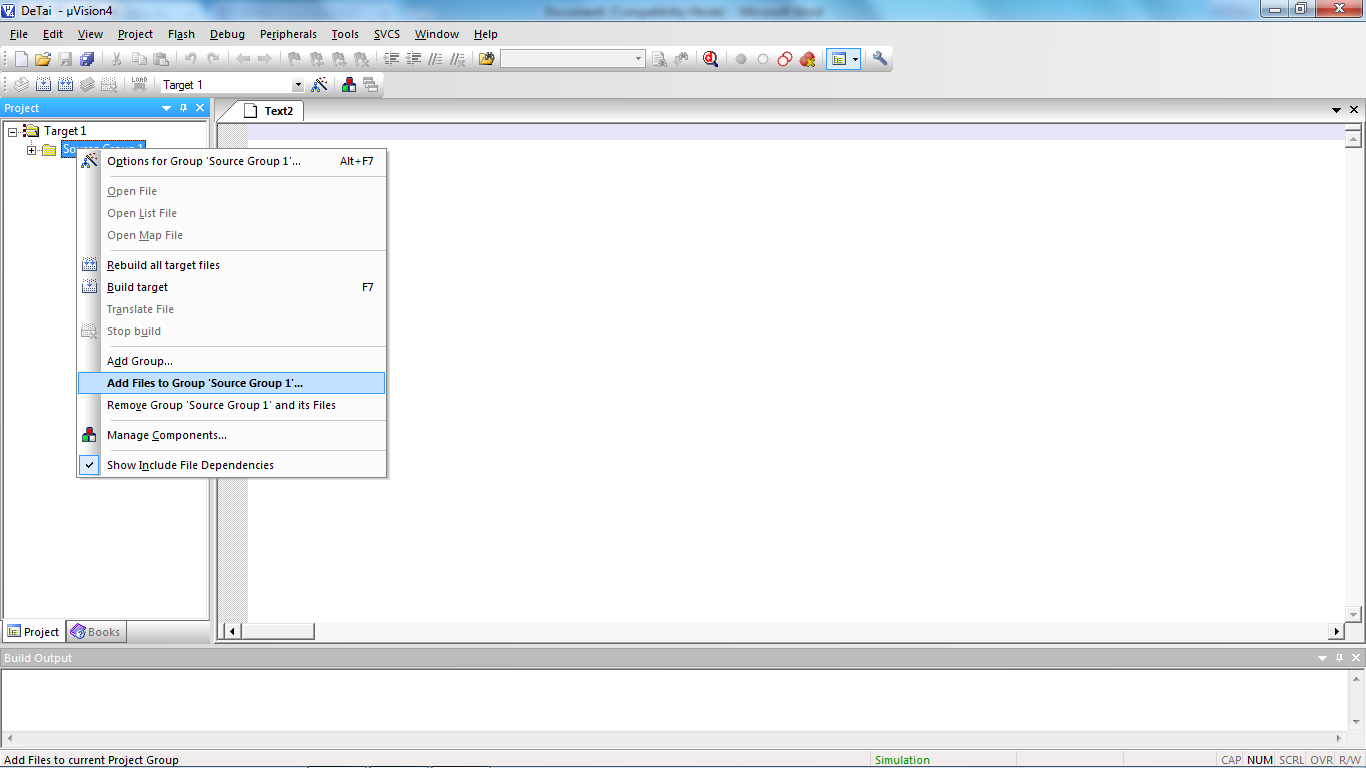


*Hình 2.16: Tạo file .c trong keil c*

* Sau đó lại vào file => Save, chọn đường dẫn tời thư mục đã lưu Project, đặt tên cho file với đuôi là *.c*, chọn Save.

**B2.** Add file .c vào project:

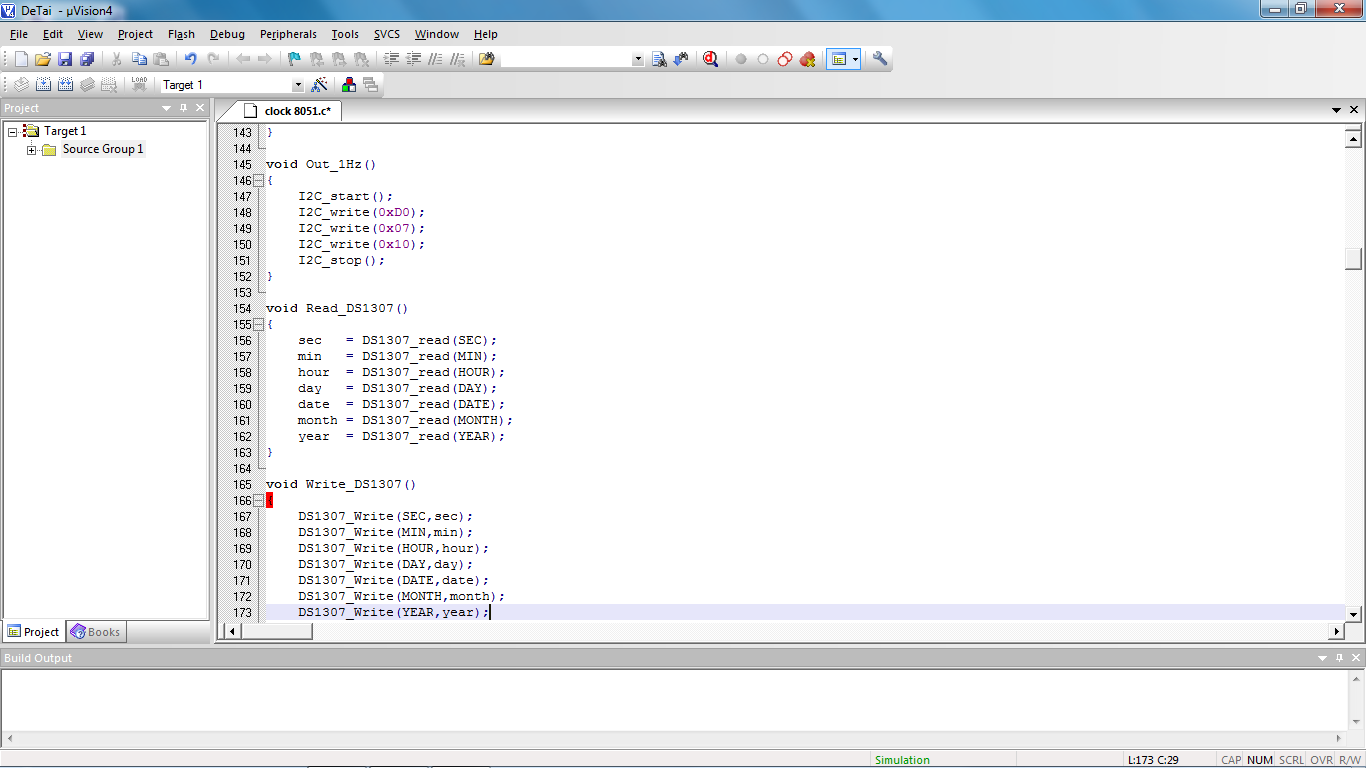
* Trong giao diện Keil C, ở không gian làm việc của Project: Chuột phải vào phần “Source Group 1” -> Add files to Group “Source Group 1”:



*Hình 2.17: Add file .c*

* Chọn đường dẫn đến thư mục Project => chọn file.C vừa tạo ở bước 2 => Add

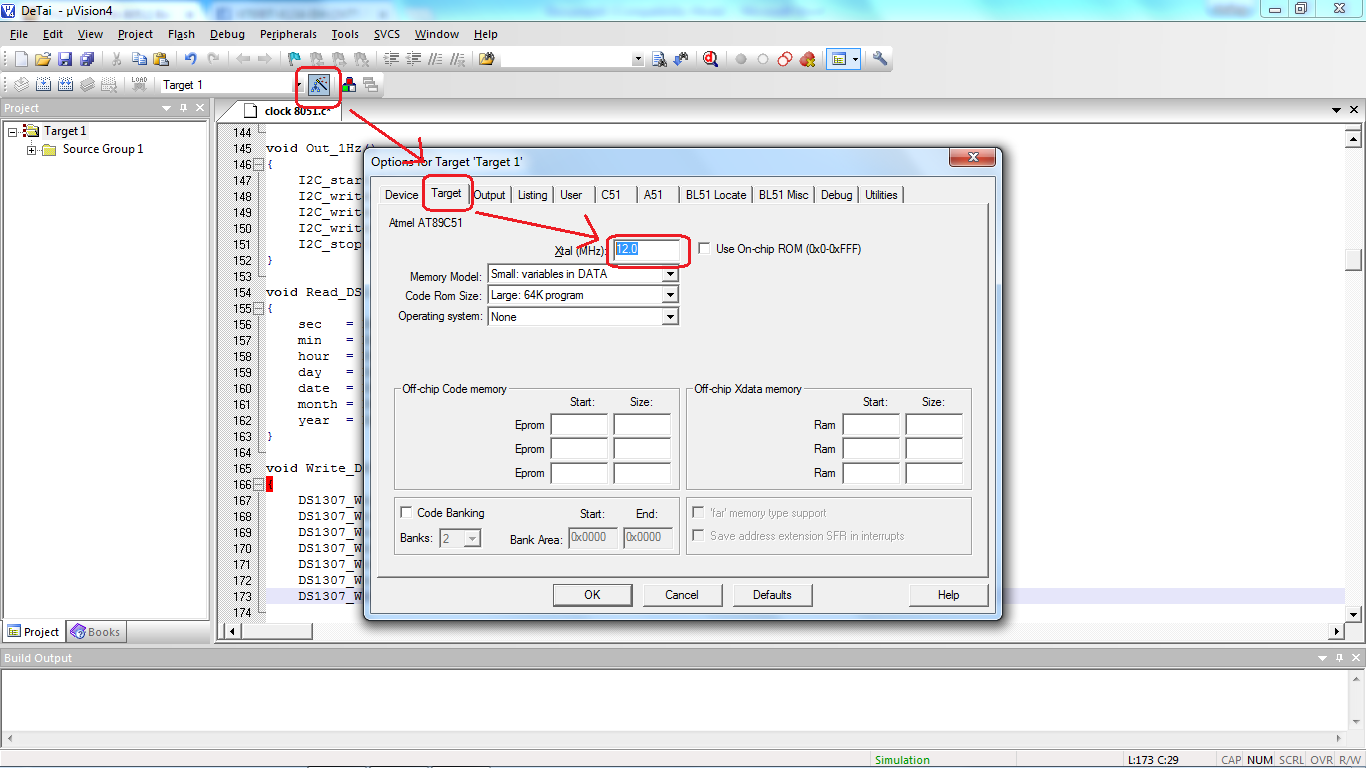
**B3.** Viết chương trình cho 8051.



*Hình 2.18: Viết code trên keil c*

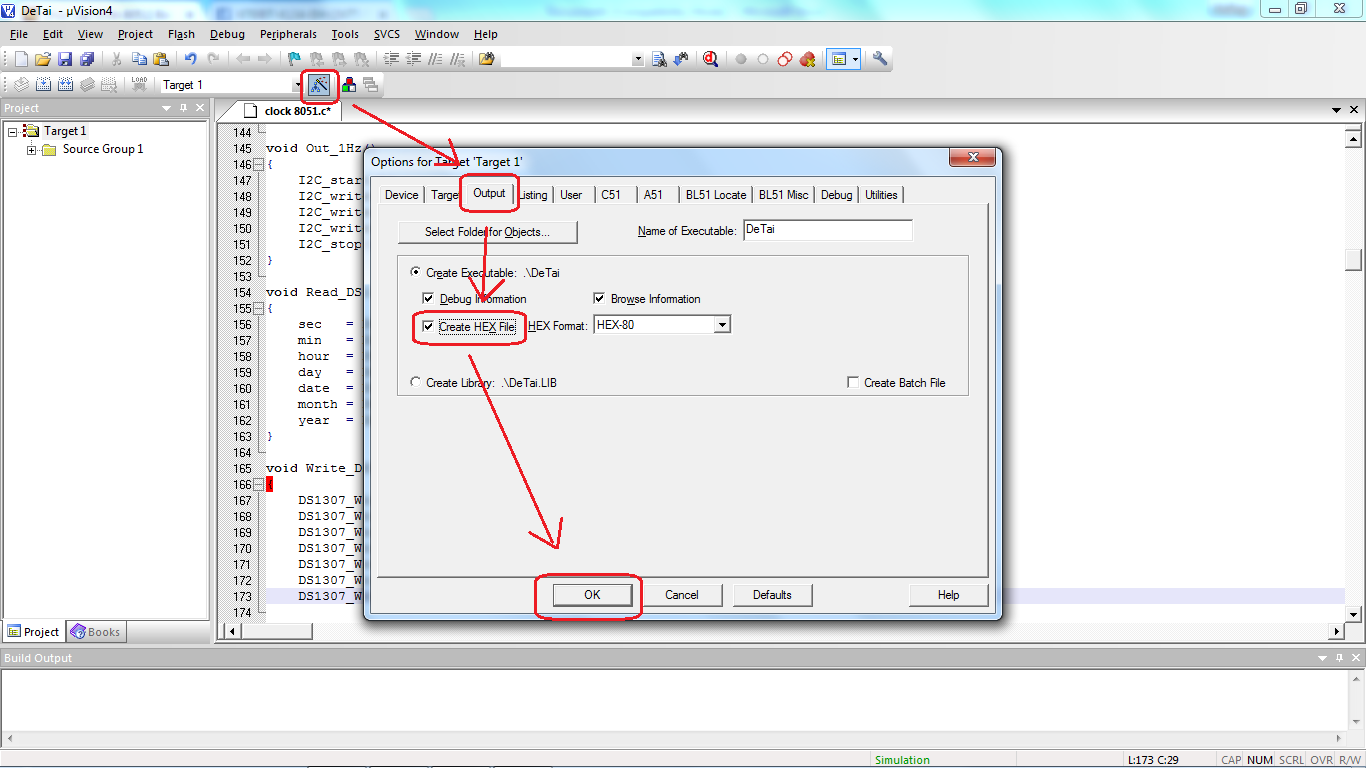
**B4.** Tạo file .hex

* Sửa tần số thạch anh:



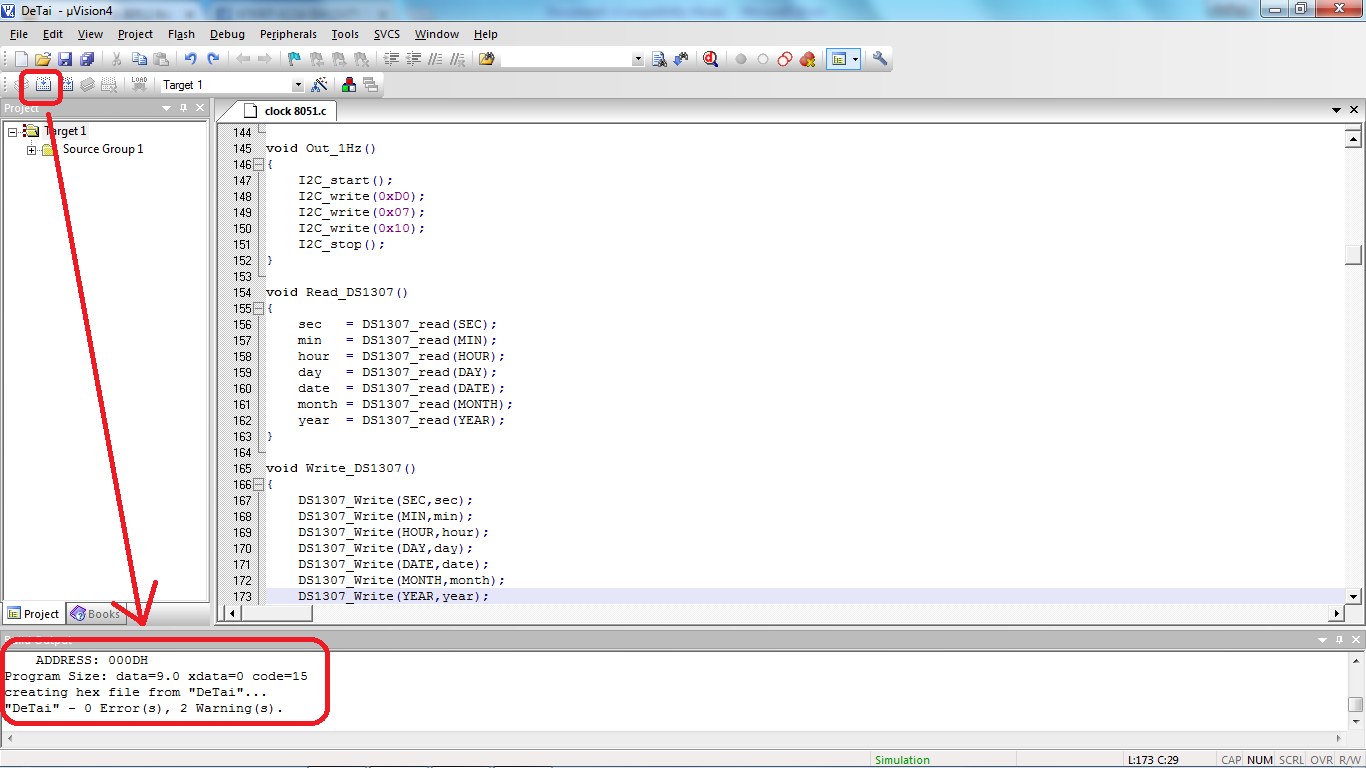
*Hình 2.19: Chỉnh sửa tần số thạch anh*

Tạo file .hex:



*Hình 2.20: Tạo file hex đầu ra*

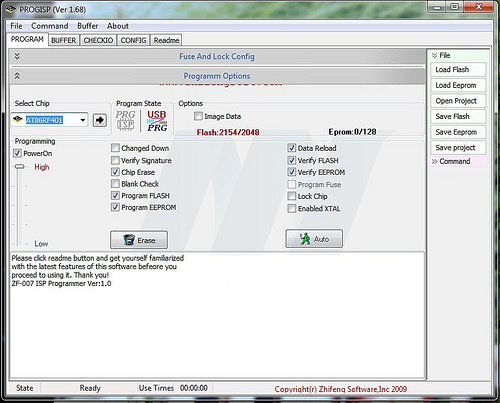
* Biên dịch chương trình: Chọn Build hoặc nhấn F7. Nếu thành công sẽ có thông báo: “Creating hex file from …”, 0 Error. Và ta sẽ có 1 file Hex được tạo ra cùng thư mục với project.



*Hình 2.21: Biên dịch chương trình*

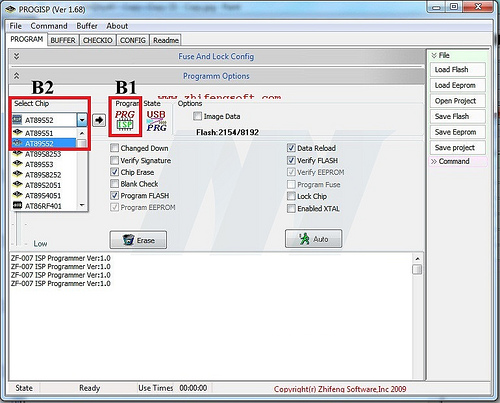
**B5.** Nạp chương trình cho vi điều khiển: Tùy từng mạch nạp chuyên dụng mà ta cần cài đặt Driver, và phần mềm nạp Chip khác nhau. Trong bài, nhóm thực hiện nạp chương trình cho 8051 bằng phần mềm Progisp:

* Chạy phần mềm Progisp:



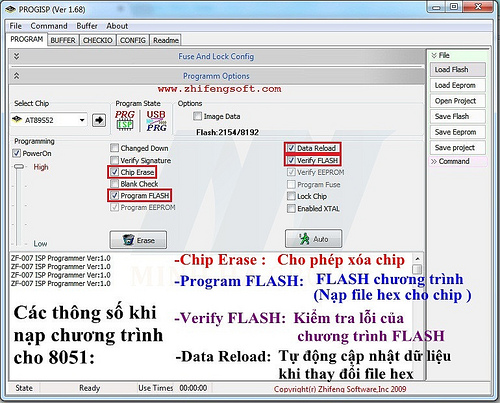
*Hình 2.22: Nạp chương trình qua phần mềm nạp*

* 1:Chương trình nhận mạch nạp USBISP.  
  2:Chọn loại chip ( Mình chọn 89S52 ).



*Hình 2.23: Chọn loại chip cần nạp*

* Các thông số khi nạp chương trình cho 8051



*Hình 2.24: Chỉnh sửa thông số khi nạp*

* Xóa chương trình cũ có trong chip:



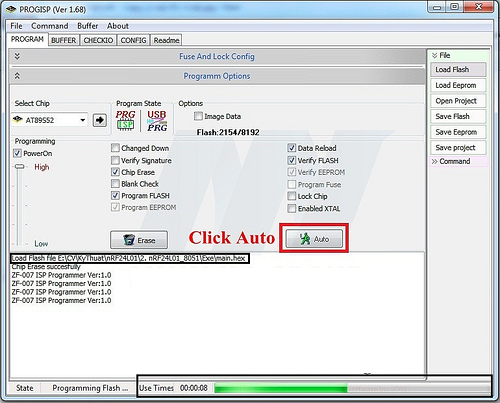
*Hình 2.25: Xóa chương trình trong chíp cũ*

* Click **Load Flash**, Tìm file **.hex** đã tạo chọn **Open.**



*Hình 2.26: Chọn file hex nạp vào chip*

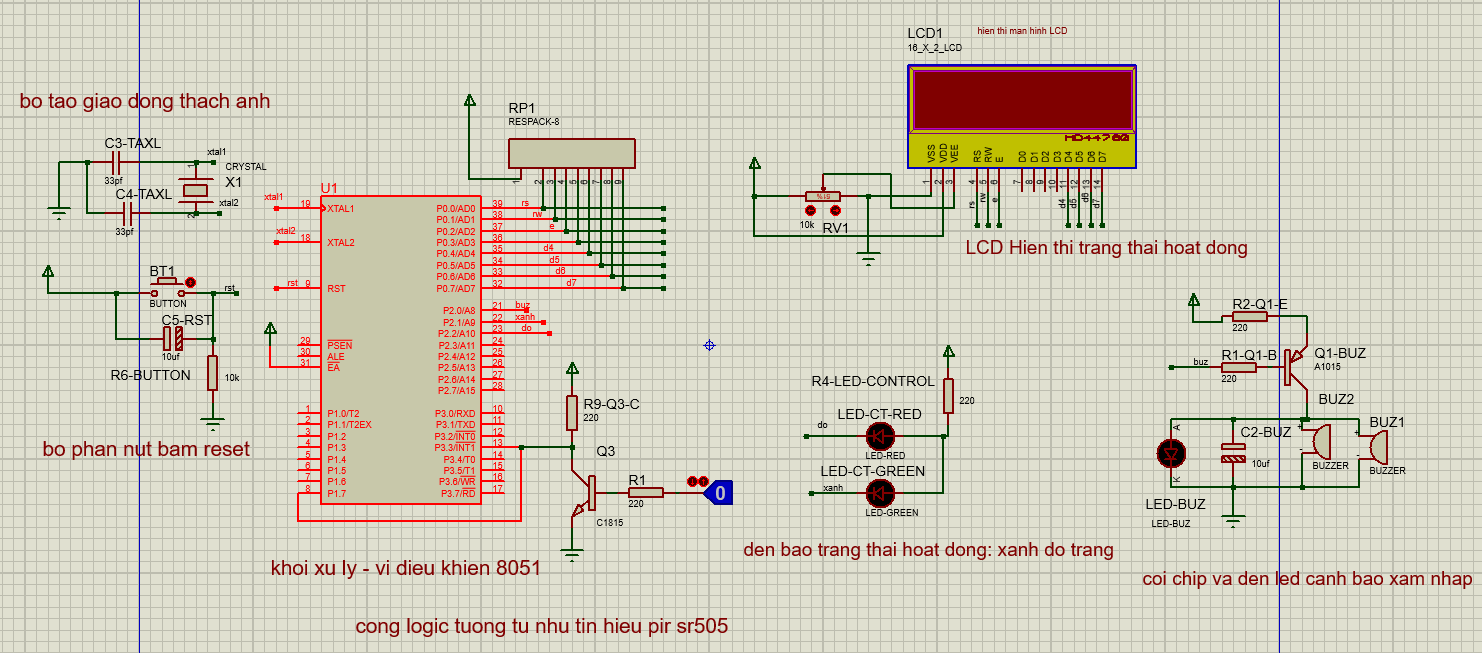
* Click **Auto** để load chương trình, chờ hoàn tất là xong



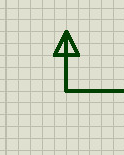
*Hình 2.27: Nhấn auto để nạp chương trình*

**CHƯƠNG 3. THỰC THI THIẾT KẾ MẠCH CẢNH BÁO AN NINH SỬ DỤNG 8051**

**3.1. Thiết kế phần cứng**

****

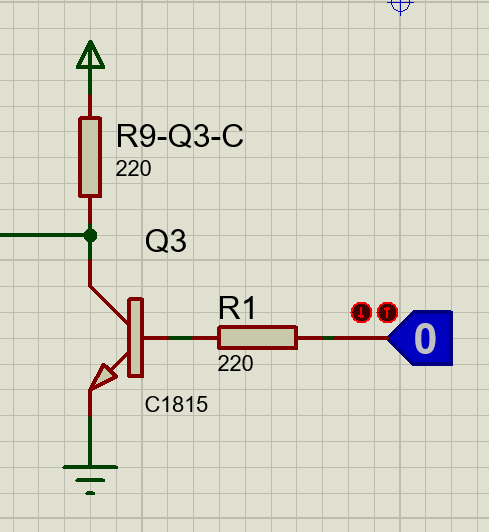
**a, Khối nguồn:**



*Hình 3.1: Mô phỏng nguồn trong proteus*

-Sử dụng nguồn 5v, cấp nguồn cho các khối cảm biến, khối xử lý, khối báo động, khối hiển thị.

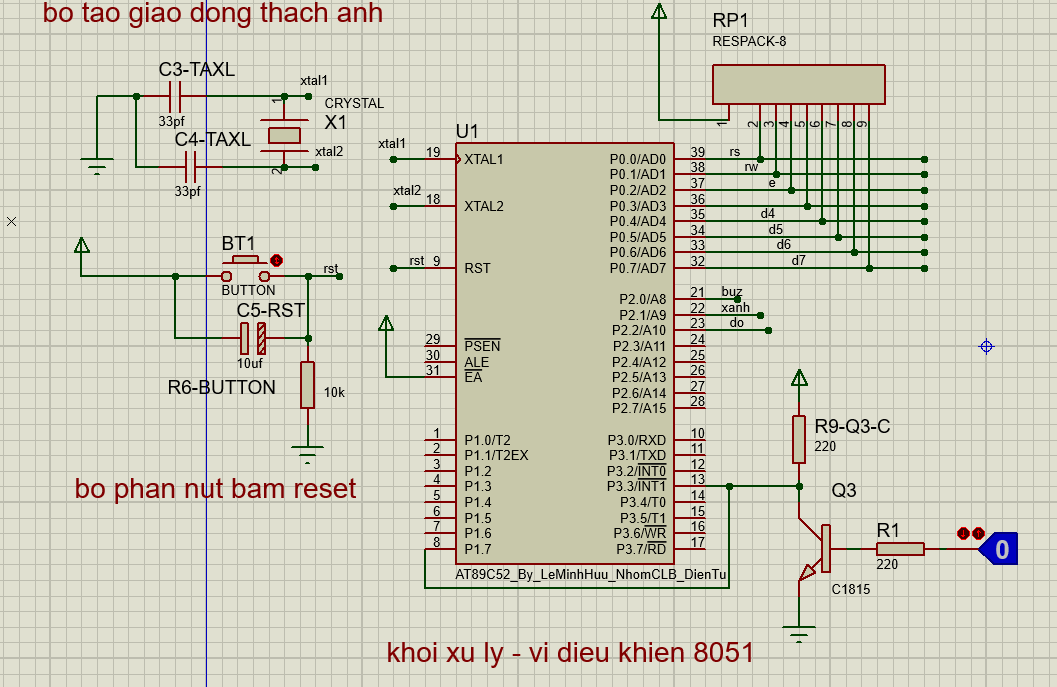
**c, Khối cảm biến**

****

*Hình 3.2: Mô phỏng khối cảm biến pir*

* Cảm biến PIR khi phát hiện ra thân nhiệt, chân out sẽ phát ra mức 1. Vi điều khiển cho phép ngắt ở mức 0 nên ta dùng một cổng not lắp bởi mạch gồm 1 transistor và 2 điện trở như trên để đảo tín hiệu vào vi điều khiển chuyển thành mức 0. Vì cảm biến PIR điều khiển bằng một chân ở mức 0 hoặc 1 nên trong mô phỏng có thể thay thế bằng cổng logicstate để dễ điều khiển.

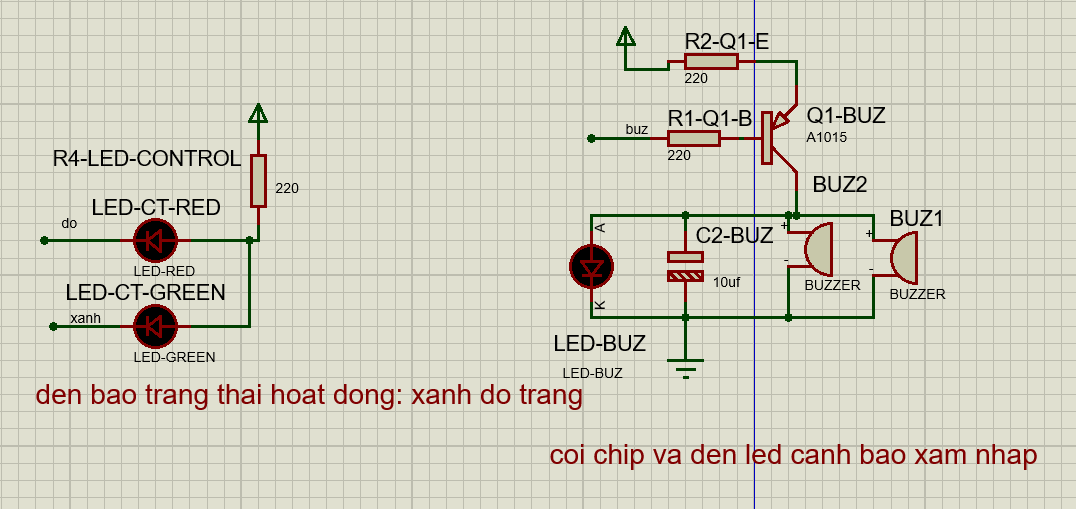
**b, Khối xử lý:**

****

*Hình 3.3: Mô phỏng khối xử lý vi điều khiển 8051*

Khối xử lý gồm có một vi điều khiển 8051 – at89s52, bộ phận thạch anh 11,0592MHz để tạo dao động và bộ phận nút bấm reset chip, điện trở treo.

**c, Khối báo động**

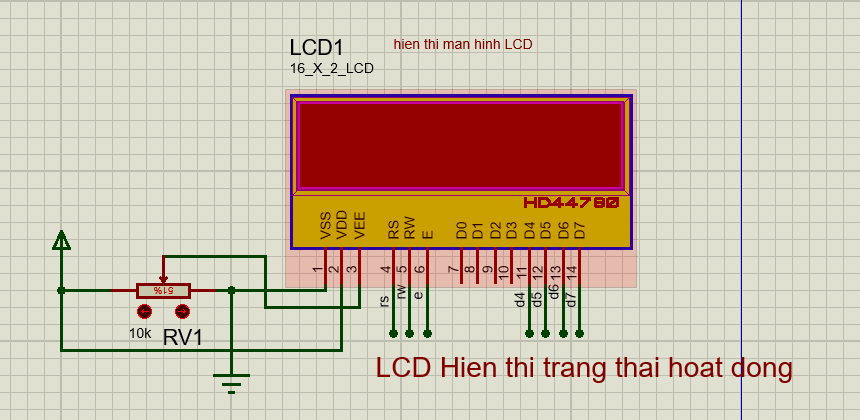
****

*Hình 3.4: Mô phỏng khối báo động*

-Khi có báo động còi chip kêu đồng thời led màu đỏ nhấp nháy, led trắng sáng liên tục, led xanh thì tắt. Quá trình lặp lại đến khi hết tín hiệu ngắt.

-Khi hết tín hiệu ngắt thì trở lại trạng thái an toàn ban đầu: Led xanh sáng, 2 led còn lại và còi chip tắt.

**d, Khối hiển thị:**

****

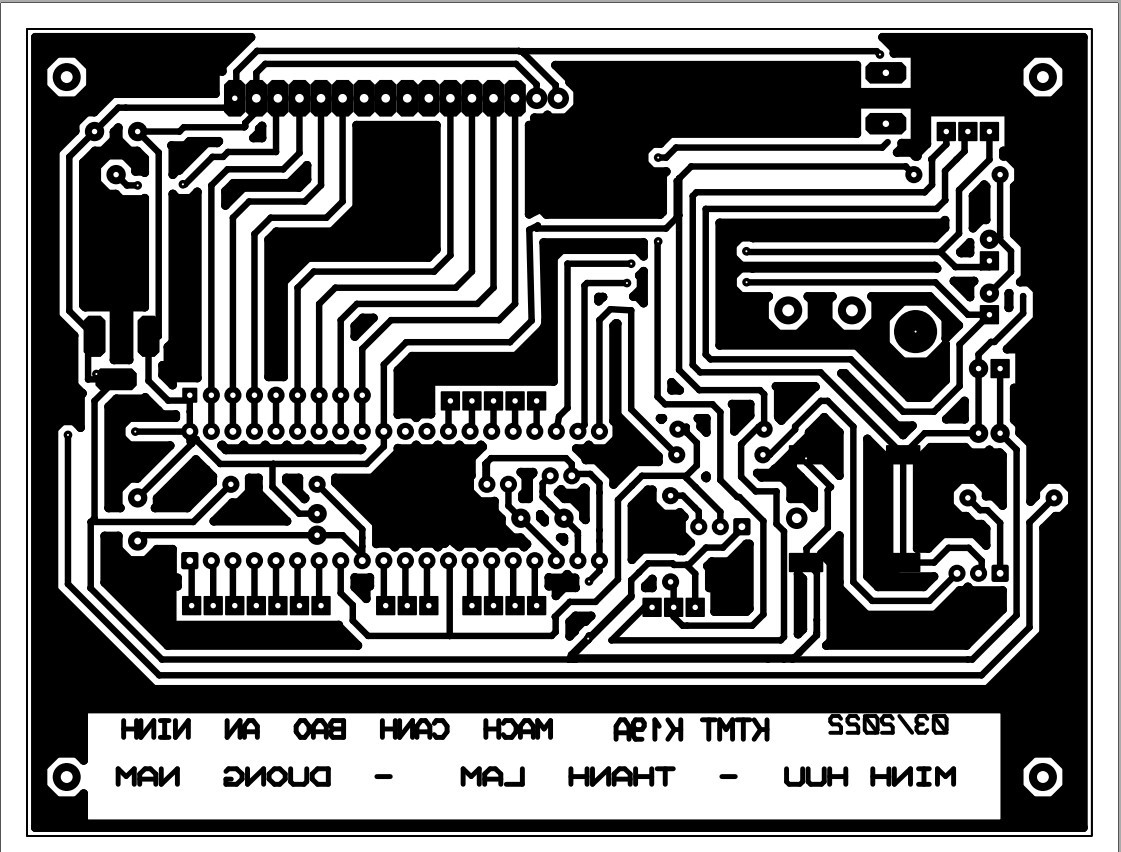
*Hình 3.5: Mô phỏng khối hiển thị lên lcd 1602*

**-** Khối hiển thị LCD có nhiệm vụ hiển thị trạng thái an toàn hay hiển thị cảnh báo xâm nhập

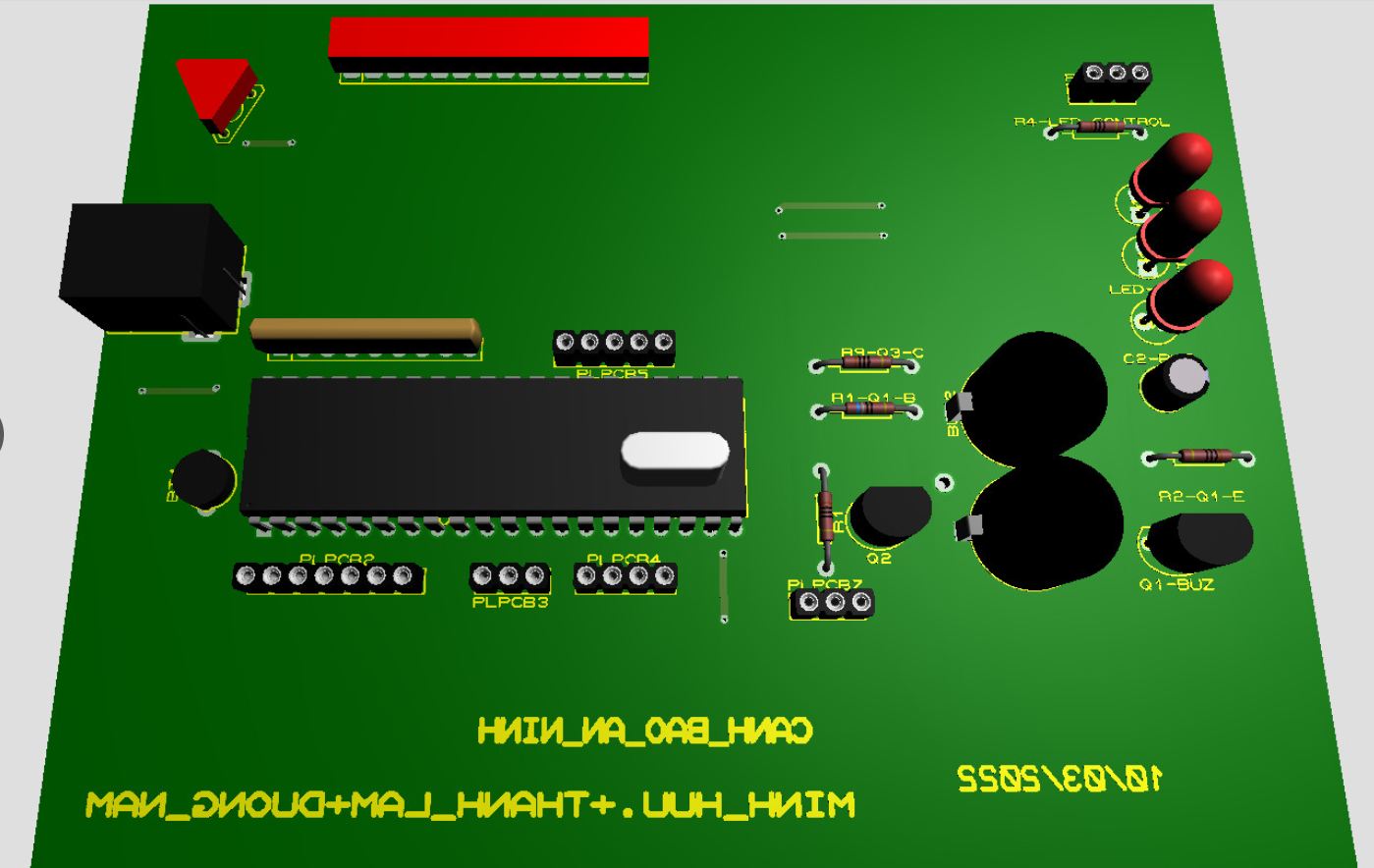
**3.1.1. Sơ đồ nguyên lý**

*Hình 3.6: Sơ đồ nguyên lý*

**3.1.2. Sơ đồ mạch in**

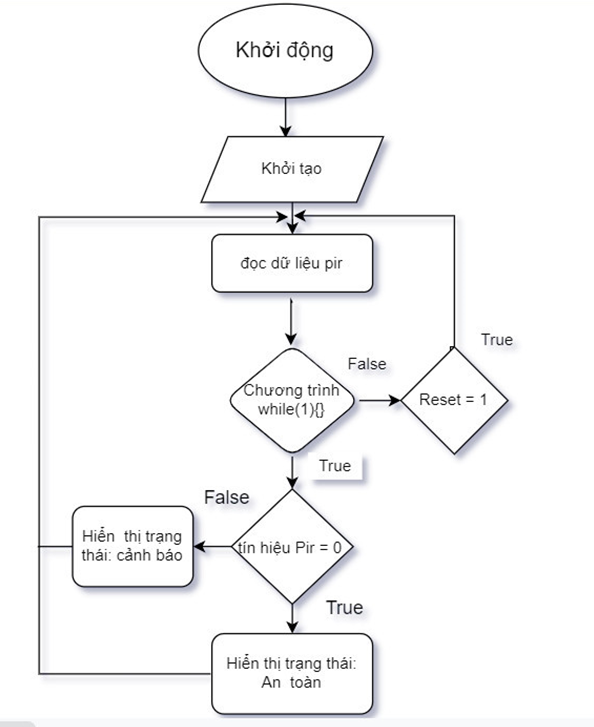


*Hình 3.7: Mạch in*

*Hình 3.8: Mạch mô phỏng 3d*

**3.2. Thiết kế phần mềm**

**3.2.1. Lưu đồ thuật toán**



*Hình 3.9: Lưu đồ thuật toán*

- Giải thích:

Sau khi bật hệ thống thì sẽ khởi tạo các chân điều khiển, ngắt, LCD. Sau khi khởi tạo xong thì hệ thống bắt đầu nhận tín hiệu từ chân cảm biến PIR và thực hiện chương trình. while(1) đúng thì thực hiện hiển thị trạng thái an toàn: LCD hiển thị “HE THONG AN TOAN”, còi chip led đỏ, trắng tắt, led xanh bật. Mỗi lần hiển thị xong thì hệ thống sẽ chờ tín hiệu ngắt và lặp lại thao tác hiển thị trạng thái an toàn.

Khi có tín hiệu ngắt, nếu hệ thống xác định là ngắt Reset thì hệ thống quay trở lại bước khởi tạo ban đầu và tiếp tục thực hiện chương trình. Nếu hệ thống xác định là ngắt ngoài thì sẽ hiển thị cảnh báo: LCD hiển thị “BI XAM NHAP ”, còi chip, led đỏ và trắng bật, led xanh tắt. Hệ thống sẽ hiển thị cảnh báo cho đến khi hết tín hiệu ngắt sẽ quay trở lại trạng thái ban đầu (hiển thị trạng thái an toàn^^).

**3.2.2. Mã nguồn chương trình**

//ham thuc hien ngat

void ngatPir(void) interrupt 2{ //ngat ngoai 1

int b = 9999, a = 2000;

LCD\_Clear();

LCD\_Gotoxy(0,0);

LCD\_Puts(" BI XAM NHAP ! ");

LCD\_Gotoxy(0,1);

LCD\_Puts(" -- WARINGING --");

LCD\_Gotoxy(0,0);

LCD\_Puts(" BI XAM NHAP ! ");

P2\_2 = 0; //bat led trang sang lien tuc

while(a--){}

P2\_0 = 0; //bat led do va coi chip

P2\_1 = 1; //tat xanh do chan 2.1

while(b--){}

while(a--){}

P2\_0 = 1; //tat led do va coi chip

}

//============================================================

//kiem tra dkien led

void antoan(){

delay\_ms(100);

led\_green = on; //bat den xanh

buzzer = led\_red = off; //tat den do, coii

LCD\_Clear(); //xoa man hinh lcd

LCD\_Gotoxy(0,0); //in ra chuoi tai hang 1 vi tri bat dau thu 0

LCD\_Puts("NGAY MOI VUI VE");

LCD\_Gotoxy(0,1);//in ra chuoi tai hang 2 vi tri bat dau thu 0

LCD\_Puts("HE THONG AN TOAN");

delay\_ms(1000);

led\_green = on; //bat den xanh

P2\_2 = 1; //tat led do chan 2.2

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ctr chinh\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//==========================================================

void main(){

IE = 0x84; //1 0 0 0 0 1 0 0//khoi tao ngat ngoai 1

IT1 = 0; //ngat ngoai 1 chan p3.3 INT1. IT1 = 0 ngat theo muc , IT = 1 ngat theo suon

LCD\_Init();//Khoi tao LCD

delay\_ms(1000);

LCD\_Puts("HUU - LAM - NAM");//Gui chuoi len LCD

delay\_ms(1000);

LCD\_Clear(); //xoa man hinh lcd

LCD\_Gotoxy(2,0);

LCD\_Puts("LOP KTMT K19A");

LCD\_Gotoxy(0,1);

LCD\_Puts(" LE HUU XIN CHAO");

delay\_ms(5000);

buzzer = led\_green = led\_red = on;

delay\_ms(1000);

buzzer = led\_green = led\_red = off;

while(1){

antoan(); //goi ham an toan

}

}

**3.3. Một số hình ảnh của sản phẩm**

****

*Hình 3.10: Mạch cảnh báo an ninh thực tế*

****

*Hình 3.11: Mạch cảnh báo an ninh*

**KẾT LUẬN VÀ ĐÁNH GIÁ**

Sau một thời gian nỗ lực học tập kết hợp với sự chỉ bảo nhiệt tình của cô Mai Thị Kim Anh cũng như các thầy cô giáo, nhóm chúng em đã hoàn thành đề tài “Thiết kế mạch cảnh báo an ninh sử dụng 8051”.

Trong đề tài thiết kế sử dụng phần mềm mô phỏng proteus dễ dàng cho việc vẽ mô phỏng hoạt động của mạch điện, thiết kế mạch in cùng với các công cụ hỗ trợ lập trình và giao tiếp với vi điều khiển AT89S52. Phần mềm hỗ trợ chi tiết và thao tác với chúng nhanh chóng.

- Đánh giá:

**Ưu điểm:**

- Mạch hoạt động ổn định

- Cách sử dụng đơn giản

- Có thể phát hiện vật thể phát tia hồng ngoại chuyển động trong phạm vi tiếp nhận của cảm biến pir

**Nhược điểm**:

-Khi khởi động hệ thống lần đầu tiên mà đã có tín hiệu ngắt, màn hình LCD sẽ không hiển thị chữ mà chỉ bật sáng led và còi chip. Giải pháp: Khi bật hệ thống lên thì bật công tắc để PIR không hoạt động. Sau khi khởi động hoàn toàn hệ thống thì mới tắt công tắc để PIR hoạt động.

-Mạch chưa kết nối với cách hệ thống khác nên làm giảm hiệu quả cảnh báo.

Sau một thời gian miệt mài nghiên cứu đề tài, chúng em đã hoàn thành và học hỏi được rất nhiều kiến thức lí thuyết cũng như kinh nghiệm thực hành, được hiểu rõ và chi tiết hơn về các linh kiện điện tử, hiểu rõ về một số họ vi điều khiển, biết xây dựng, thiết kế mạch theo yêu cầu sử dụng. Và đặc biệt hơn, thông qua việc nghiên cứu này, chúng em đã củng cố thêm về cách tổ chức, phân chia cũng như làm việc theo nhóm.

Để đề tài hoàn thiện hơn và nâng cao kinh nghiệm học tập, đúc rút ra nhiều kiến thức hơn cho những đề tài sau, chúng em rất mong nhận được sự đóng góp ý kiến của thầy cô và các bạn.

*Chúng em xin chân thành cám ơn!*

**HƯỚNG PHÁT TRIỂN**

Trong tương lai, nhóm em sẽ tìm hiểu kỹ hơn và mở rộng đề tài với mạch cảnh báo an ninh sử dụng 8051 cho hoàn thiện hơn, cụ thể là:

* Gửi thông tin cảnh báo xâm nhập lên điện thoại.
* Kết hợp với mở cửa bằng mật khẩu.
* Điều khiển bóng đèn công suất lớn qua rơ le.

**TÀI LIỆU THAM KHẢO**

1. Bài giảng môn KỸ THUẬT VI XỬ LÝ VÀ ỨNG DỤNG
2. Tra cứu, tham khảo trên INTERNET

**PHỤ LỤC**

MÃ NGUỒN CHƯƠNG TRÌNH

//code vdk 8051 mach canh bao an ninh

#include <REGX52.H>

/\*\*\*\*\*\*\*\*Khai bao chan giao tiep\*\*\*\*\*\*\*\*\*\*\*\*/

#define LCD\_RS P0\_0

#define LCD\_RW P0\_1

#define LCD\_EN P0\_2

#define LCD\_D4 P0\_4

#define LCD\_D5 P0\_5

#define LCD\_D6 P0\_6

#define LCD\_D7 P0\_7

#define on 0

#define off 1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//khai bao chan chuc nang

sbit buzzer = P2^0; //chan noi voi coi chip

sbit led\_green = P2^1; //led bao trang thai an toan

sbit led\_red = P2^2; //led bao trang thai xam nhap

sbit input\_data = P1^7; //chan tin hieu vao tu pir rs 505

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delay\_us(unsigned int t){

unsigned int i;

for(i=0;i<t;i++);

}

void delay\_ms(unsigned int t){

unsigned int i,j;

for(i=0;i<t;i++)

for(j=0;j<125;j++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ctr giao tiep LCD 16x2 4bit\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void LCD\_Enable(void){

LCD\_EN =1;

delay\_us(3);

LCD\_EN=0;

delay\_us(50);

}

//Ham Gui 4 Bit Du Lieu Ra LCD

void LCD\_Send4Bit(unsigned char Data){

LCD\_D4=Data & 0x01;

LCD\_D5=(Data>>1)&1;

LCD\_D6=(Data>>2)&1;

LCD\_D7=(Data>>3)&1;

}

// Ham Gui 1 Lenh Cho LCD

void LCD\_SendCommand(unsigned char command){

LCD\_Send4Bit(command >>4);/\* Gui 4 bit cao \*/

LCD\_Enable();

LCD\_Send4Bit(command); /\* Gui 4 bit thap\*/

LCD\_Enable();

}

void LCD\_Clear(){// Ham Xoa Man Hinh LCD

LCD\_SendCommand(0x01);

delay\_us(10);

}

// Ham Khoi Tao LCD

void LCD\_Init(){

LCD\_Send4Bit(0x00);

delay\_ms(20);

LCD\_RS=0;

LCD\_RW=0;

LCD\_Send4Bit(0x03);

LCD\_Enable();

delay\_ms(5);

LCD\_Enable();

delay\_us(100);

LCD\_Enable();

LCD\_Send4Bit(0x02);

LCD\_Enable();

LCD\_SendCommand( 0x28 ); // giao thuc 4 bit, hien thi 2 hang, ki tu 5x8

LCD\_SendCommand( 0x0c); // cho phep hien thi man hinh

LCD\_SendCommand( 0x06 ); // tang ID, khong dich khung hinh

LCD\_SendCommand(0x01); // xoa toan bo khung hinh

}

void LCD\_Gotoxy(unsigned char x, unsigned char y){

unsigned char address;

if(!y)address=(0x80+x);

else address=(0xc0+x);

delay\_us(1000);

LCD\_SendCommand(address);

delay\_us(50);

}

void LCD\_PutChar(unsigned char Data){//Ham Gui 1 Ki Tu

LCD\_RS=1;

LCD\_SendCommand(Data);

LCD\_RS=0 ;

}

void LCD\_Puts (char \*s){//Ham gui 1 chuoi ky tu

while (\*s){

LCD\_PutChar(\*s);

s++;

}

}

//============================================================

//ham thuc hien ngat

void ngatPir(void) interrupt 2{ //ngat ngoai 1

int b = 9999, a = 2000;

LCD\_Clear();

LCD\_Gotoxy(0,0);

LCD\_Puts(" BI XAM NHAP ! ");

LCD\_Gotoxy(0,1);

LCD\_Puts(" -- WARINGING --");

LCD\_Gotoxy(0,0);

LCD\_Puts(" BI XAM NHAP ! ");

P2\_2 = 0; //bat led trang sang lien tuc

while(a--){}

P2\_0 = 0; //bat led do va coi chip

P2\_1 = 1; //tat xanh do chan 2.1

while(b--){}

while(a--){}

P2\_0 = 1; //tat led do va coi chip

}

//============================================================

//kiem tra dkien led

void antoan(){

delay\_ms(100);

led\_green = on; //bat den xanh

buzzer = led\_red = off; //tat den do, coii

LCD\_Clear(); //xoa man hinh lcd

LCD\_Gotoxy(0,0); //in ra chuoi tai hang 1 vi tri bat dau thu 0

LCD\_Puts("NGAY MOI VUI VE");

LCD\_Gotoxy(0,1);//in ra chuoi tai hang 2 vi tri bat dau thu 0

LCD\_Puts("HE THONG AN TOAN");

delay\_ms(1000);

led\_green = on; //bat den xanh

P2\_2 = 1; //tat led do chan 2.2

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ctr chinh\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//==========================================================

void main(){

IE = 0x84; //1 0 0 0 0 1 0 0//khoi tao ngat ngoai 1

IT1 = 0; //ngat ngoai 1 chan p3.3 INT1. IT1 = 0 ngat theo muc , IT = 1 ngat theo suon

LCD\_Init();//Khoi tao LCD

delay\_ms(1000);

LCD\_Puts("HUU - LAM - NAM");//Gui chuoi len LCD

delay\_ms(1000);

LCD\_Clear(); //xoa man hinh lcd

LCD\_Gotoxy(2,0);

LCD\_Puts("LOP KTMT K19A");

LCD\_Gotoxy(0,1);

LCD\_Puts(" LE HUU XIN CHAO");

delay\_ms(5000);

buzzer = led\_green = led\_red = on;

delay\_ms(1000);

buzzer = led\_green = led\_red = off;

while(1){

antoan(); //goi ham an toan

}

}

//THE END