

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

\_\_\_\_\_ \* \_\_\_\_\_



# BÀI TẬP LỚN

Môn: Vi xử lý

**ĐỀ TÀI: Xây dựng hệ thống tưới cây tự động**

Sinh viên : Nguyễn Hồng Ngọc

Phạm Minh Tâm

Lớp : KSTN-CNTT-K60

Giáo viên hướng dẫn: TS. Ngô Lam Trung

*Hà Nội, ngày 9 tháng 6 năm 2018*

# Mục lục

Lời nói đầu .....	3
Chương I: Giới thiệu về vi điều khiển AT89S52 .....	4
1.1 Tổng quan về họ vi điều khiển Intel 8051 .....	4
1.2 Giới thiệu chung về AT89s52 .....	4
1.3 Chi tiết về AT89S52 .....	5
1.3.1 Sơ đồ khối của AT89S52.....	5
1.3.2 Cấu tạo chân của vi điều khiển .....	6
Chương II: Giới thiệu hệ thống báo động tưới cây và các công cụ thực hiện .....	9
2.1 Hệ thống báo động tưới cây .....	9
2.2 Các công cụ sử dụng để thiết kế hệ thống .....	10
2.2.1 Bộ AT89S52 Development Kit .....	10
2.2.2 Proteus Design Suit 7.8 .....	11
2.2.3 Keil $\mu$ Vision4.....	12
2.2.4 IC Programmer v2 .....	12
Chương III: Thiết kế lập trình hệ thống .....	14
3.1 Thiết kế phần cứng.....	14
3.2 Lập trình phần mềm .....	14
3.2.1 Khởi tạo ngắt.....	14
3.2.2 Trạng thái ban đầu .....	15
3.2.3 Sự kiện ấn nút .....	15
3.2.4 Sự kiện khô nguy hiểm .....	17
3.2.5 Sự kiện tưới cây.....	18
Chương IV : Kết luận .....	19
Tài liệu tham khảo .....	20

# Lời nói đầu

Trong thời buổi hiện nay, khoa học kĩ thuật và công nghiệp máy tính đang phát triển mạnh mẽ và đã trở thành một phần quan trọng không thể thiếu trong bất cứ một ngành nghề lĩnh vực cuộc sống nào. Các thiết bị công nghệ xuất hiện ngày một nhiều hơn, chứa bên trong nó những vi mạch và vi xử lý phức tạp, hiện đại. Đi kèm với sự phức tạp của phần cứng, lập trình phần mềm cho các thiết bị vi xử lý này cũng trở lên phức tạp hơn rất nhiều. Vì thế, muốn làm việc tốt với phần cứng của thiết bị công nghệ nào cũng đòi hỏi cần có sự hiểu biết kĩ càng về các vi xử lý và kiến thức lập trình vi xử lý bằng Assembly.

Tuy nhiên, việc lập trình vi xử lý là không hề đơn giản. Với yêu cầu càng cao thì hệ thống sẽ càng phức tạp và việc lập trình với từng phần của hệ thống sẽ càng khó khăn. Kể cả với những yêu cầu đơn giản như đóng mở một đèn LED theo chu kì thời gian cũng đòi hỏi hệ thống vi xử lý phải có đầy đủ các khối: bộ vi xử lý, bộ nhớ, các IC ngoại vi. Hệ thống tạo ra cũng thường rất phức tạp, bên cạnh đó các vi xử lý thường đi kèm với dữ liệu dạng byte, word trong khi các đối tượng trong điện tử công nghiệp thường điều khiển theo các bit.

Chính vì sự phức tạp này mà các nhà sản xuất đã tích hợp một ít bộ nhớ cùng các thiết bị ngoại vi và vi xử lý để tạo thành một IC gọi là vi điều khiển. Với vi điều khiển, người sử dụng không cần thiết phải biết một lượng kiến thức quá nhiều như khi làm việc với vi xử lý, qua đó dễ dàng và thuận tiện hơn khi lập trình với nó. Sự ra đời của các vi điều khiển đã đánh dấu một bước tiến lớn trong ngành điều khiển công nghiệp điện tử, giúp cho việc thiết kế các mạch hệ thống nhúng trở lên dễ dàng và tiện lợi hơn rất nhiều so với trước.

Có rất nhiều những hãng chế tạo vi điều khiển với đặc tính và chức năng khác nhau, một trong số đó là Atmel. Một trong số các vi điều khiển phổ biến của Atmel đó là bộ AT89s52 thuộc họ vi điều khiển Intel 8051. Thông qua môn học này, để hiểu rõ hơn về cách thức lập trình vi điều khiển nói chung và AT89s52 nói riêng, chúng em đã thực hiện đề tài “Thiết kế hệ thống báo động tưới cây bằng vi điều khiển”. Dưới đây là phần báo cáo cho sản phẩm mà nhóm đã thực hiện. Báo cáo bao gồm các phần: Giới thiệu về vi điều khiển AT89s52, Giới thiệu về hệ thống báo động tưới cây và các công cụ, Thiết kế lập trình hệ thống và Kết luận.

**Nhóm sinh viên**

Nguyễn Hồng Ngọc,

Phạm Minh Tâm.

# Chương I: Giới thiệu về vi điều khiển AT89S52

## 1.1 Tổng quan về họ vi điều khiển Intel 8051

**Intel 8051** (hay có tên gọi khác là MCS 51) là vi điều khiển đơn tinh thể (không nhúng với CPU) kiến trúc Harvard, lần đầu tiên được sản xuất bởi Intel năm 1980. Trong những năm 1980 và đầu những năm 1990, Intel 8051 là một vi điều khiển rất nổi tiếng, được ứng dụng rộng rãi trong nhiều hệ thống nhúng khác nhau.

Kiến trúc cơ bản bên trong Intel 8051 bao gồm các đặc tính sau:

- Một ALU 8-bit, một thanh tích lũy và một thanh ghi 8-bit, do đó nó là một vi điều khiển 8-bit
- Bus dữ liệu 8-bit - có thể truy cập 8 bits dữ liệu trong một hoạt động.
- Bus địa chỉ 16-bit - có thể truy cập 216 vị trí nhớ 64kB (65536 vị trí) cho mỗi bộ nhớ RAM và ROM
- RAM trên chip - 128 bytes (bộ nhớ dữ liệu)
- ROM trên chip - 4 kbytes (bộ nhớ chương trình)
- 32 chân I/O riêng biệt (4 nhóm mỗi nhóm 8 chân I/O) có thể được truy cập riêng rẽ.
- Hai bộ định thời/đếm 16-bit
- Đơn vị thu/phát bất đồng bộ phổ biến UART hoàn toàn song công.
- Sáu nguyên nhân ngắt với 2 mức ưu tiên.
- Chế độ tiết kiệm năng lượng (trên một số phiên bản).

Tiếp theo đó, Intel đã nâng cấp và cho ra đời các chip hiện đại hơn như 8052, 8053, 8055. Hiện nay, Intel không còn sản xuất các loại vi điều khiển thuộc họ 8051 nữa, thay vào đó các nhà sản xuất thứ hai được cấp phép để cung cấp loại vi điều khiển này. Các vi điều khiển mới này chứa các thiết bị hiện đại hơn, với lõi phối hợp 8051, được sản xuất bởi hơn 20 nhà sản xuất độc lập, như Atmel, Maxim IC (công ty con của Dallas Semiconductor), NXP Semiconductors (Philips Semiconductor trước đây), Winbond, Silicon Laboratories, Texas Instruments và Cypress Semiconductor.

## 1.2 Giới thiệu chung về AT89s52

Vi điều khiển dòng 8051 được sử dụng rộng rãi nhất trên thế giới cũng như ở Việt Nam đó là các vi điều khiển của hãng Atmel. Các mã số vi điều khiển cũng được Atmel thay đổi khi sản xuất. Mã số 80 trong “80Cxx” chuyển thành 89, chẳng hạn như chip 80C52 của Intel khi sang Atmel là chip AT89C52 với tính năng tương tự. Các dòng vi điều khiển phổ biến khác của Atmel là AT89C51, AT89C53 và AT89C55. Sau một khoảng thời gian cải tiến và phát triển, Atmel tung ra thị trường dòng vi điều khiển mang số hiệu “89Sxx” với nhiều cải tiến và đặc biệt là có thêm khả năng nạp chương trình theo chế độ nối tiếp rất đơn giản và tiện lợi cho người sử dụng.

Vi điều khiển AT89S52 tương thích với dòng Intel 8051 có các đặc tính cơ bản sau:

- Vi điều khiển CMOS 8-bit

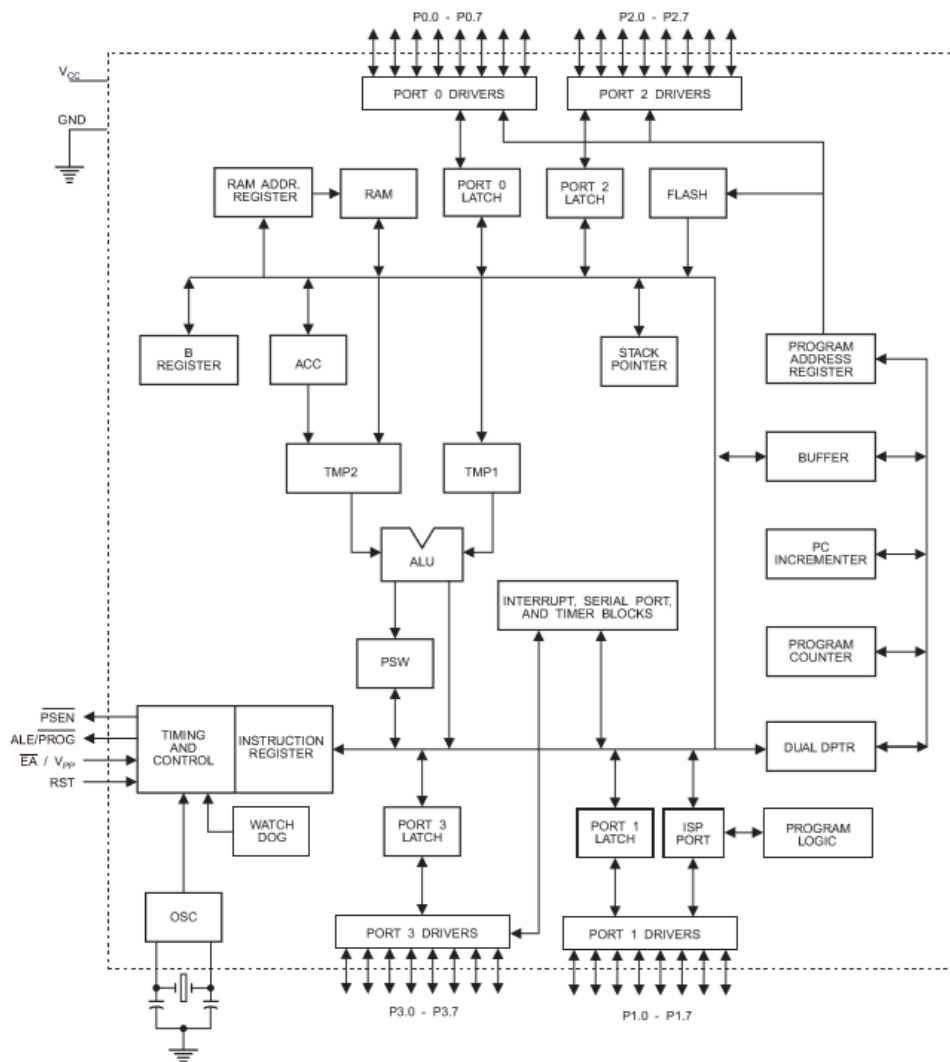
- 8K bytes bộ nhớ trong lập trình được
- Vùng điện áp từ 4.0V đến 5.5V
- Bộ nhớ RAM 256 x 8-bit
- 32 cổng vào ra lập trình được
- 3 bộ Timer 16-bit
- 8 nguồn ngắt
- Kênh vào ra nối tiếp UART
- Watchdog timer

AT89S52 là một bộ vi điều khiển mạnh mẽ thuộc dòng 8051 (MCS-51) với nhiều tính năng tiện lợi. Vi điều khiển AT89S52 ứng dụng rất tốt vào các hệ thống nhỏ, đơn giản với yêu cầu tiết kiệm năng lượng cao, phù hợp với nhiều thiết bị vào ra. Với đề tài xây dựng hệ thống báo động tưới cây, bộ vi điều khiển AT89S52 là rất phù hợp và đem lại hiệu quả cao về năng lượng.

## 1.3 Chi tiết về AT89S52

### 1.3.1 Sơ đồ khối của AT89S52

Sơ đồ khối của AT89S52 như sau:



Thành phần chính của vi điều khiển AT89S52 là bộ xử lý trung tâm CPU bao gồm:

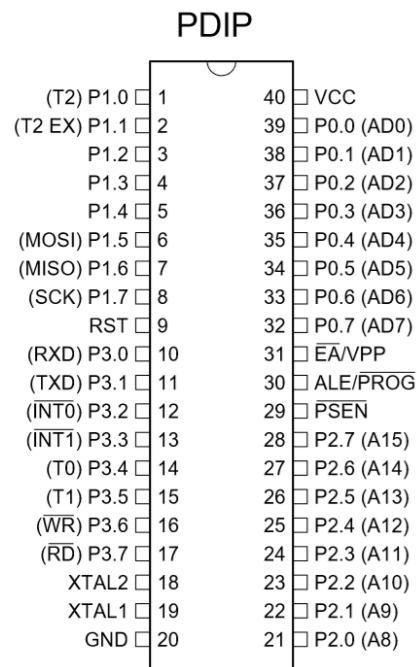
- Các thanh ghi chức năng đặc biệt SFR
- Đơn vị logic số học ALU
- Cổng vào ra I/O
- Bộ nhớ chương trình và bộ nhớ dữ liệu

Một số điều cơ bản về vi điều khiển AT89S52:

- Đơn vị xử lý trung tâm nhận trực tiếp xung từ bộ dao động hoặc từ một tín hiệu được đưa vào từ bên ngoài.
- Chương trình đang chạy có thể cho dừng lại nhờ một khối điều khiển ngắt bên trong. Các nguồn ngắt có thể là: các sự kiện ở bên ngoài, sự tràn bộ đếm định thời, hoặc cũng có thể là giao diện nối tiếp.
- Ba bộ định thời 16 bit hoạt động như một bộ đếm.
- Các cổng port0 đến port3 được sử dụng vào mục đích điều khiển.
- Giao diện nối tiếp chứa một bộ truyền và một bộ nhận làm việc độc lập với nhau.
- Bộ nhớ RAM và ROM để lưu trữ dữ liệu và lệnh còn các thanh ghi sử dụng để lưu trữ thông tin trong quá trình xử lý.

### 1.3.2 Cấu tạo chân của vi điều khiển

Tùy theo nhu cầu của người sử dụng mà có 3 kiểu chân được Atmel đưa ra cho vi điều khiển AT89S52, đó là PDIP, PLCC và TQFP. Ở đề tài này, nhóm đã sử dụng kiểu PDIP. Sơ đồ chân như sau:



Mô tả về các chân như sau:

- ❖ Nhóm chân nguồn: vận hành với nguồn đơn 5V
  - VCC được nối vào chân 40

- GND được nối vào chân 20 (hay nối Mass)
- ❖ Nhóm chân dao động: gồm chân 18 và 19 (chân XTAL1 và XTAL2), cho phép ghép nối thạch anh vào mạch dao động bên trong vi điều khiển, được sử dụng để nhận luồng xung clock từ bên ngoài để hoạt động, thường được ghép nối với thạch anh và các tụ để tạo nguồn xung clock ổn định.
  - XTAL1: Ngõ vào đến mạch khuếch đại dao động đảo và ngõ vào đến mạch tạo xung clock bên trong.
  - XTAL2: Ngõ ra từ mạch khuếch đại dao động đảo.
- ❖ Chân chọn bộ nhớ chương trình: chân 31 (EA/VPP) dùng để xác định chương trình thực hiện được lấy từ ROM nội hay ROM ngoài.
  - Chân 31 nối Mass: Sử dụng bộ nhớ chương trình bên ngoài vi điều khiển
  - Chân 31 nối VCC: sử dụng bộ nhớ chương trình 4Kb bên trong vi điều khiển.
- ❖ Chân reset RST: Ngõ vào RST ở chân 9 là ngõ vào reset dùng để thiết lập lại trạng thái ban đầu cho vi điều khiển. Hệ thống sẽ được thiết lập lại các giá trị ban đầu nếu ngõ này ở mức 1, tối thiểu 2 chu kỳ máy.
- ❖ Chân cho phép bộ nhớ chương trình PSEN: (program store enable) tín hiệu được xuất ra ở chân 29 dùng để truy vấn bộ nhớ chương trình ngoài. Chân này thường được nối vào chân OE (output enable) của ROM ngoài. Khi vi điều khiển làm việc với bộ nhớ chương trình ngoài, chân này phát ra tín hiệu kích hoạt ở mức thấp và được kích hoạt 2 lần trong một chu kỳ máy. Khi thực thi 1 chương trình ở ROM nội, chân này được duy trì ở mức logic không tích cực (logic 1). Không cần thiết kết nối chân này khi không sử dụng đến.
- ❖ Chân ALE: chân cho phép chốt địa chỉ - chân 30. Khi vi điều khiển truy xuất bộ nhớ từ bên ngoài, port0 vừa có chức năng là bus địa chỉ, vừa có chức năng là bus dữ liệu, do đó phải tách các đường dữ liệu và địa chỉ. Tín hiệu ở chân ALE được dùng làm tín hiệu điều khiển để giải đa hợp các đường địa chỉ và các đường dữ liệu khi kết nối chúng với IC chốt. Các xung tín hiệu ALE có tốc độ bằng 1/6 lần tần số dao động đưa vào vi điều khiển, như vậy có thể dùng tín hiệu ở ngõ ra ALE cung cấp cho các phần khác của hệ thống.
- ❖ Nhóm chân điều khiển vào ra:
  - Port0 (P0): Gồm 8 chân (32-39) có 2 chức năng:
    - Chức năng I/O: Các chân này được dùng để nhận tín hiệu từ bên ngoài vào để xử lý hoặc xuất tín hiệu ra để điều khiển
    - Chức năng bus dữ liệu và địa chỉ: (AD7-AD0) 8 chân này còn làm nhiệm vụ lấy dữ liệu từ ROM hoặc RAM ngoài, hoặc để định địa chỉ bộ nhớ ngoài.
  - Port1 (P1): Gồm 8 chân (1-8) chỉ có chức năng làm đường I/O
  - Port2 (P2): Gồm 8 chân (21-28) có hai chức năng:
    - Chức năng I/O
    - Chức năng bus địa chỉ cao (A8-A15) khi kết nối với bộ nhớ ngoài có dung lượng lớn (cần 2 bytes để đánh địa chỉ)
  - Port3 (P3): Gồm 8 chân (10-17) chức năng I/O. Với mỗi một chân có một chức năng riêng:
    - P3.0 RxD: Ngõ vào nhận dữ liệu nối tiếp

- P3.1 TxD: Ngõ xuất dữ liệu nối tiếp
- P3.2 INT0: Ngõ vào ngắt cứng 0
- P3.3 INT1: Ngõ vào ngắt cứng 1
- P3.4 T0: Ngõ vào của Timer 0
- P3.5 T1: Ngõ vào của Timer 1
- P3.6 WR: Ngõ điều khiển ghi dữ liệu lên bộ nhớ ngoài
- P3.7 RD: Ngõ điều khiển đọc dữ liệu từ bộ nhớ ngoài
- P1.0 T2: Ngõ vào của Timer 2
- P1.1 T2X: Ngõ nạp lại/thu nhận của Timer 2



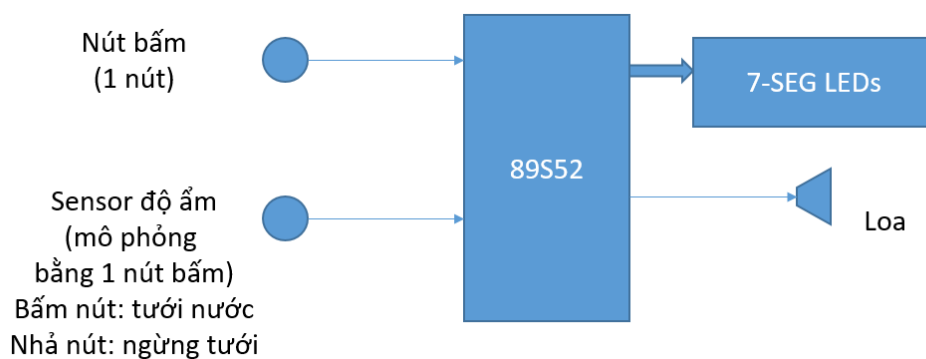
## Chương II: Giới thiệu hệ thống báo động tưới cây và các công cụ thực hiện

### 2.1 Hệ thống báo động tưới cây

Hệ thống báo động tưới cây được sử dụng để thông báo, nhắc nhở người dùng khi nào thì cần phải tưới cây. Hệ thống hay thiết bị này bao gồm:

- ❖ 4 bộ LED 7 thanh để hiển thị
- ❖ Một nút bấm điều khiển
- ❖ Loa báo hiệu
- ❖ Cảm biến độ ẩm

Sơ đồ mô phỏng của thiết bị như sau:

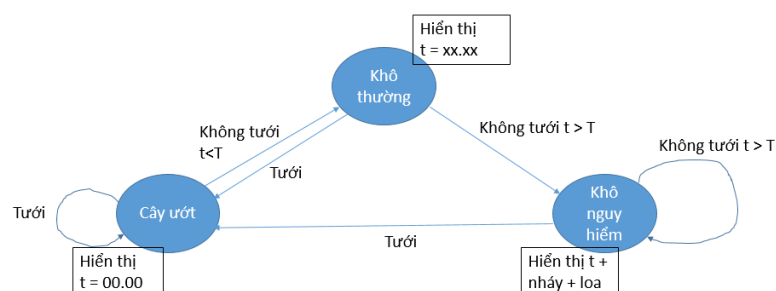


Chức năng của thiết bị như sau:

- ❖ Có sensor cảm biến độ ẩm (output = 1: đất ẩm, = 0: đất khô).
- ❖ Hiển thị thời gian (theo giờ) từ thời điểm đất bị khô gần nhất.
- ❖ Nhắc nhở đất bị khô quá thời gian cho phép: nháy đèn và phát tiếng kêu báo động. Thời gian này gọi là thời gian chờ tưới.
- ❖ Chỉ dùng 1 phím để set thời gian chờ tưới cây.

Hoạt động thiết bị: Cây có 3 trạng thái: ướt (được tưới nước), khô thường (không được tưới trong khoảng thời gian  $t < T$ ), và khô nguy hiểm ( $t > T$ ). Trạng thái được detect bởi sensor kết hợp timer đo thời gian.  $T$  là tham số tính theo giây, cấu hình được.

Sơ đồ trạng thái:



Cách sử dụng thiết bị thông qua nút điều khiển:

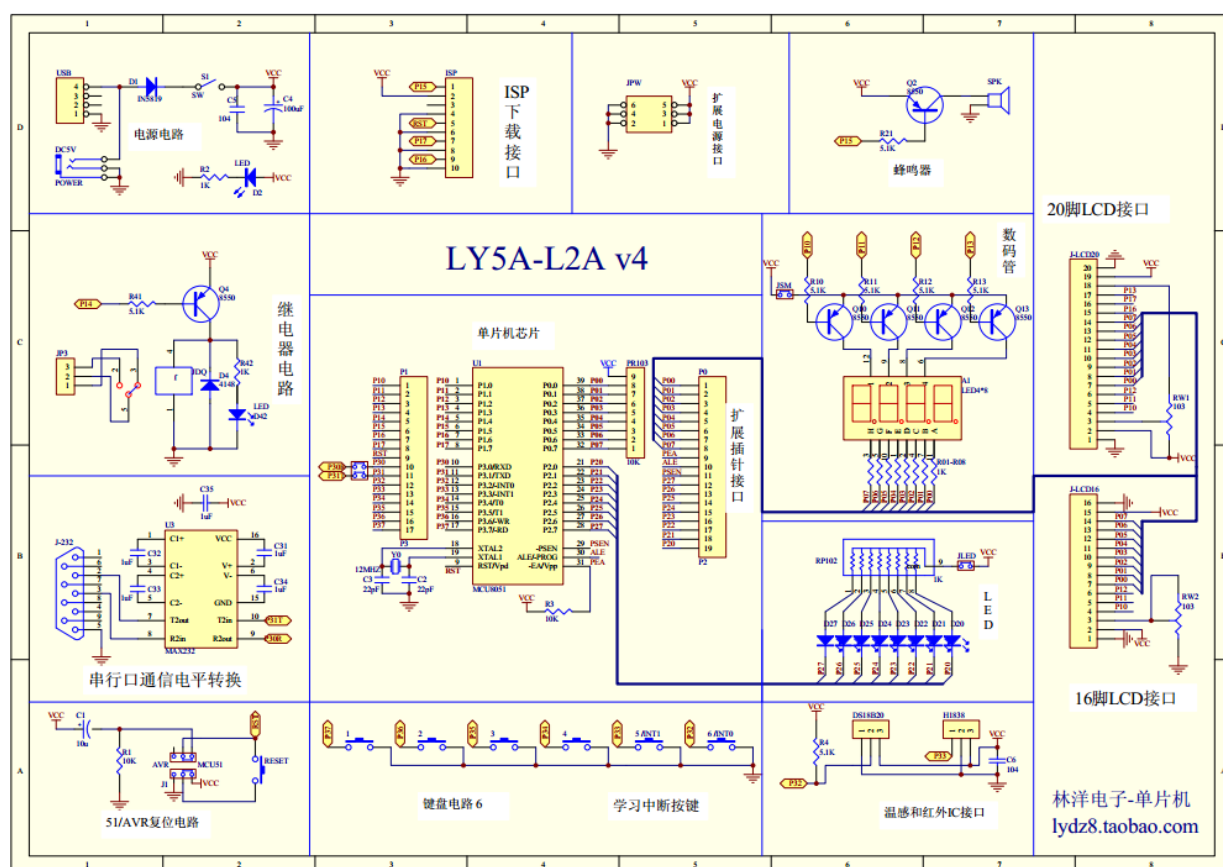
- ❖ Cấu hình tham số t: chỉ sử dụng 1 nút duy nhất để giao tiếp với mạch. Nút có thể đưa vào 2 lệnh: short-press, và long-press (giống đồng hồ điện tử casio).
- ❖ Màn hình bình thường hiển thị giá trị t. Khi short-press thì màn hình hiển thị T, ấn lần nữa thì sẽ lại quay về hiển thị T
- ❖ Khi màn hình đang hiện T và có long-press → chuyển sang trạng thái set giá trị cho T, màn hình sẽ hiện T + nhấp nháy chậm. Ấn short-press để set giá trị cho T. Long-press lần nữa sẽ lưu giá trị T và chuyển về trạng thái hiển thị bình thường.
- ❖ Khi đang ở trạng thái set giá trị T, gặp short-press thì giá trị T được tăng lên. Khi T tăng lên 60 thì tự giảm về 1.

Ngoài ra, ở trong phần thiết kế hệ thống này, do không có sensor phát hiện độ ẩm nên nhóm sẽ mô phỏng lại sensor thông qua một nút bấm khác. Khi thiết bị báo động cần phải tưới cây, ấn nút bấm cũng tương đương với việc cảm biến phát hiện ra cây đã được tưới, và khi đó, trạng thái của thiết bị sẽ được reset lại và bộ đếm lại đếm lại từ đầu. Nói cách khác nút bấm này dùng để mô phỏng cho việc tưới cây.

## 2.2 Các công cụ sử dụng để thiết kế hệ thống

### 2.2.1 Bộ AT89S52 Development Kit

Cấu tạo của một bộ AT89S52 Development Kit (ở đây cụ thể là sử dụng bộ LY5A-L2A v4) như sau:



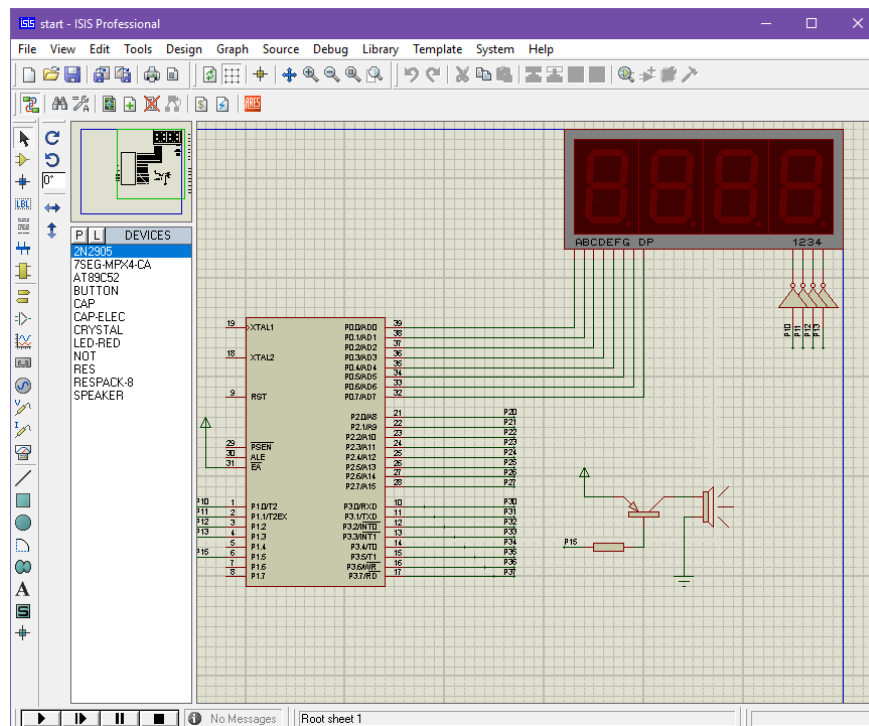
Bộ kit bao gồm:

- ❖ Bảng mạch: Để gắn các phần tử của bộ Kit.
- ❖ Vi điều khiển AT89S52: là vi điều khiển chính trong mạch. Có các thông số cơ bản như đã nêu ở phần trên.
- ❖ Thạch anh 12MHz: Là nguồn tạo xung nhịp dao động clock ổn định với tần số 12MHz cho dao động của 8051. Thạch anh được gắn vào chân XTAL1 và XTAL2 (chân 18 và 19) của AT89S52.
- ❖ Tụ gốm: Lọc nhiễu cho dao động của thạch anh. Hai tụ gốm 33 $\mu$ F sẽ được nối một đầu với chân thạch anh, đầu còn lại đấu ra Mass.
- ❖ Tụ hóa: Tụ hóa 10 $\mu$ F được gắn đầu âm vào chân RST, đầu dương lên nguồn. Khi cấp điện cho mạch, tụ sẽ phóng điện khiến chân RST bật lên mức cao, khi đó toàn bộ hệ thống sẽ được nạp lại từ đầu. Khi đang vận hành, tụ ngăn dòng điện vào chân RST
- ❖ Các điện trở: Hạn chế dòng điện và phân cực cho Transistor.
- ❖ Transistor: Có công dụng như một khóa.
- ❖ Các đèn LED: bao gồm LED đơn và LED 7 thanh cùng với loa đóng vai trò làm các thiết bị ra.
- ❖ Các nút bấm: đóng vai trò như thiết bị vào điều khiển.
- ❖ Các cổng vào ra ngoài.
- ❖ Một số dây nối cùng bộ nạp chương trình thông qua chuẩn USB.

Ở bài này, nhóm sẽ sử dụng bộ AT89S52 Development Kit này để thực hiện thiết bị báo động tưới cây. Về cơ bản, bộ Kit đã cung cấp đầy đủ các linh kiện cần thiết để tạo ra một thiết bị đơn giản.

## 2.2.2 Proteus Design Suit 7.8

Giao diện của phần mềm:

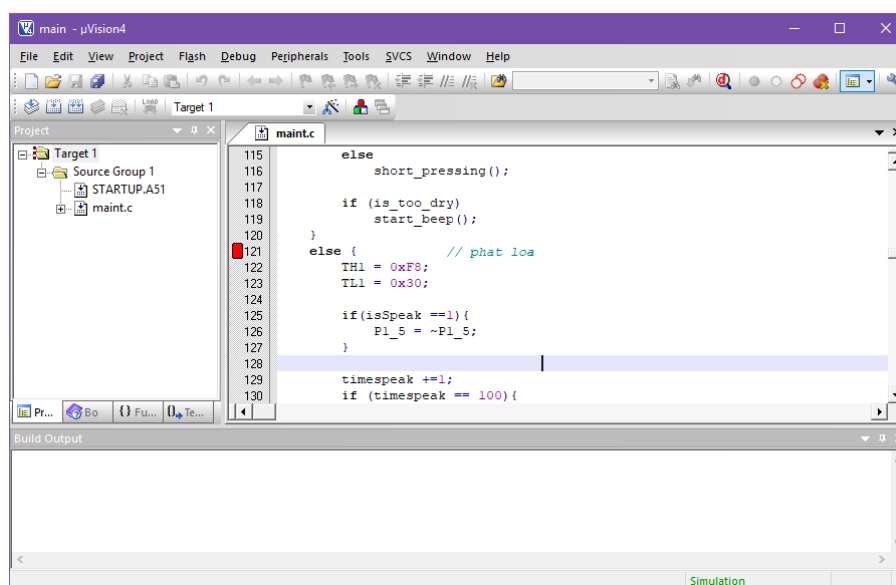


Trước khi thực hiện trên mạch thật, thiết bị sẽ được mô phỏng lại thông qua phần mềm Proteus Design Suit 7.8. Proteus Design Suit là một bộ công cụ phần mềm độc quyền được sử dụng chính vào công việc tự động hóa và thiết kế điện tử. Phần mềm này được sử dụng chủ yếu bởi các kỹ sư thiết kế điện tử và kỹ thuật viên để tạo ra sơ đồ và bản in điện tử để sản xuất bo mạch in.

Phần mềm sẽ giúp mô phỏng thiết bị báo động tưới cây đơn giản với đầy đủ các mạch và linh kiện cần thiết. Khi thực hiện, ta chỉ cần lập trình phần mềm rồi tải vào mô hình giả lập trên Proteus để chạy thử.

### 2.2.3 Keil $\mu$ Vision4

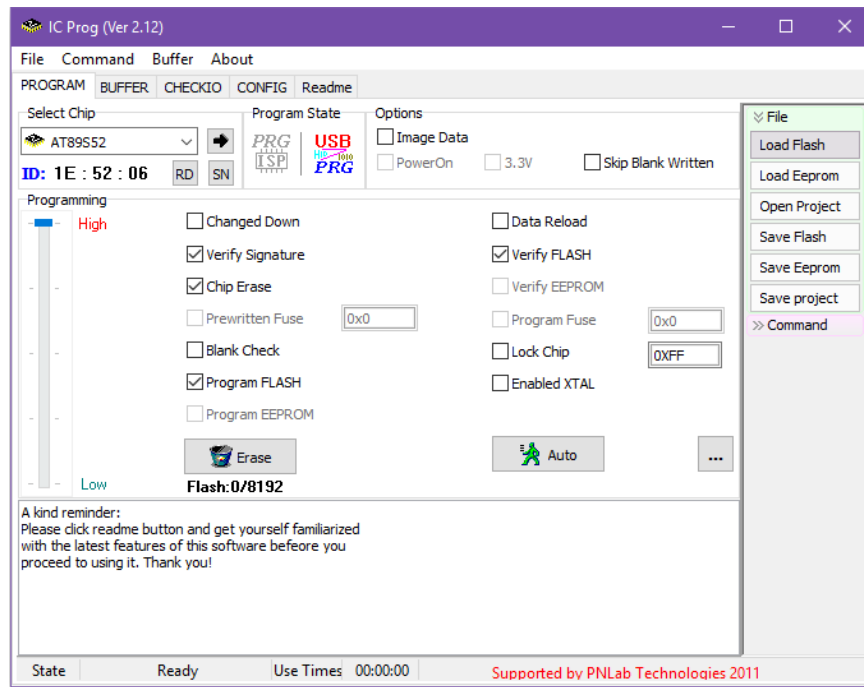
Giao diện phần mềm:



Phần mềm  $\mu$ Vision4 được phát triển bởi Keil là một IDE dùng để lập trình cho các mạch vi điều khiển khác nhau. Với IDE này, các thư viện tập lệnh dành cho các vi điều khiển đã được cung cấp sẵn. Ngôn ngữ được sử dụng để lập trình là C, sau đó phần mềm sẽ biên dịch ra file có đuôi .HEX (file mã Hexa cơ số 16). File mã Hex này sẽ được sử dụng để nạp vào vi điều khiển AT89S52 thông qua bộ nạp.

### 2.2.4 IC Programmer v2

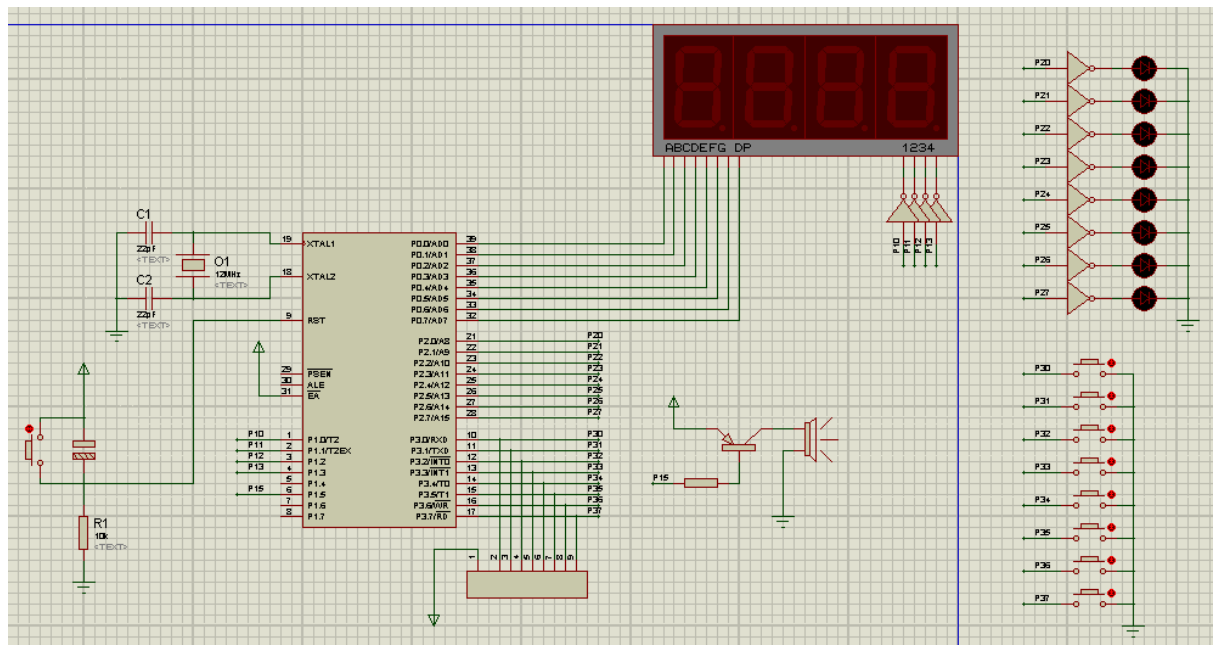
Giao diện chương trình:



Phần mềm IC Programmer v2 được sử dụng để nạp chương trình vào bộ vi điều khiển AT89S52. Sau khi kết nối bộ Kit với máy tính thông qua bộ nạp, chương trình sẽ tự động nhận bộ vi điều khiển. Để sử dụng, ta chọn loại chip ở giao diện phần mềm, chọn mã Hex để nạp vào mạch và ấn Auto để bắt đầu nạp. Phần mềm này cung cấp một giao diện đơn giản và có cơ chế sử dụng tương đối dễ dàng.

## Chương III: Thiết kế lập trình hệ thống

### 3.1 Thiết kế phần cứng



Sử dụng mạch STC89C52 LY5A-L2A (AT89S52).

Phần cứng được mô phỏng trên Proteus 7.8 như hình trên.

### 3.2 Lập trình phần mềm

#### 3.2.1 Khởi tạo ngắt

Nhóm dùng cơ chế ngắt của AT89S52 để giải quyết bài toán trên.

Vi xử lý AT89S52 có 6 ngắt bao gồm:

- 2 ngắt ngoài: INT0, INT1.
- 3 ngắt timer: Timer0, Timer1, Timer2.
- 1 ngắt của UART: Serial Port

Nhóm chỉ sử dụng 4 ngắt để giải quyết bài toán:

- ✓ Ngắt Timer0 với vai trò đếm thời gian cho vi xử lý. Cứ mỗi khi ngắt Timer0 được gọi biến đếm thời gian lại tăng lên 1. Biến đếm thời gian có tác dụng như một đồng hồ chuẩn để thực hiện các tác vụ như đếm giờ và phân biệt thời gian ấn nút.
- ✓ Ngắt Timer1 tác dụng để bắt sự kiện nhấn nút và điều khiển âm thanh phát ra từ loa.
- ✓ Ngắt INT0 Là nút điều khiển của bài toán với 2 chế độ short-press và long-press.
- ✓ Ngắt INT1 là mô phỏng của sensor độ ẩm với chế độ :
  - Bấm nút: tưới nước

- Nhả nút: ngừng tưới

Đầu tiên các biến được sử dụng trong chương trình sẽ được khai báo, khởi tạo các giá trị cần thiết và kích hoạt chế độ ngắt trong vi xử lý.

Khởi tạo chế độ ngắt trong vi xử lý :

- TMOD = 0x11; chỉ định cả 2 ngắt Timer0 và Timer1 đều hoạt động ở chế độ 1 (16bit).
- IP = 0x1A; thiết lập lại mức ưu tiên ngắt với ngắt Timer0 và Timer1 có ưu tiên cao hơn.
- EA = 1; bật chế độ ngắt trong vi xử lý
- EX0 = 1; cho phép ngắt ngoài INT0
- EX1 = 1; cho phép ngắt ngoài INT1
- ET0 = 1; cho phép ngắt Timer0
- TR0 = 1; timer0 sẽ đếm
- ET1 = 1; cho phép ngắt Timer1
- TR1 = 0; timer1 không đếm

Với ngắt Timer0 các thanh ghi TH0 và TL0 được gán giá trị lần lượt là 0x3c và 0xb0. Như vậy cứ sau 50ms thì ngắt Timer0 được gọi một lần. Khi đó giá trị của biến counter0 là biến đếm thời gian được tăng lên 1.

$$\frac{(0xFFFF - 0x3CB0 + 1) * 12}{12 * 10^6} = 0.05s = 50ms$$

### 3.2.2 Trạng thái ban đầu

Khi khởi động, thiết bị sẽ ở trạng thái khô thường. Ở trạng thái này sẽ hiển thị thời gian từ khi dừng tưới đến hiện tại. LED sẽ hiển thị ở dạng XX.XX với 2 chữ số đầu là thời gian tính bằng phút, 2 chữ số sau là thời gian tính bằng giây.

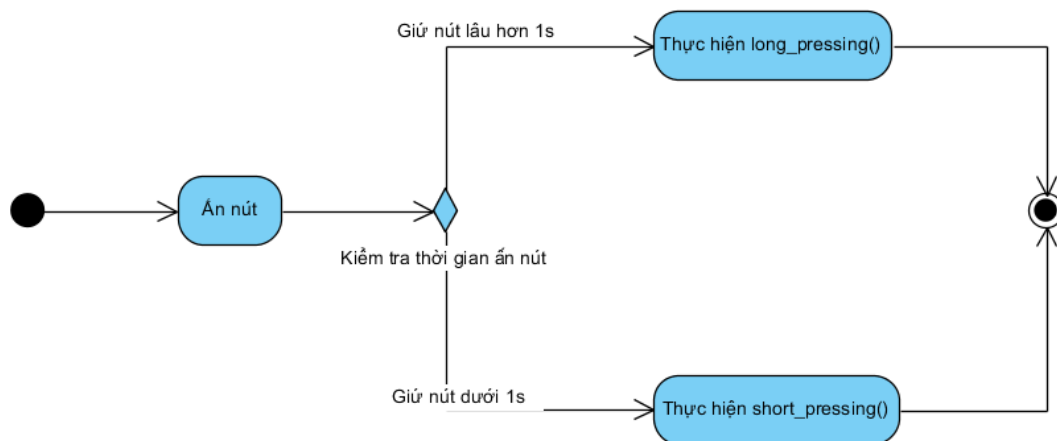
Trạng thái hiển thị thời gian T sẽ hiển thị thời gian giới hạn từ lúc dừng tưới cho đến khi chuyển sang trạng thái khô nguy hiểm. Thời gian T sẽ được tính bằng phút

Trạng thái set thời gian T sẽ hiển thị thời gian T được set. Khi nhấn nút T sẽ tăng lên T+1, nếu T tăng lên 60 sẽ tự động giảm về 1

### 3.2.3 Sự kiện ấn nút

Sử dụng nút ứng với ngắt ngoài INTO để làm nút ấn điều khiển chương trình.

Biểu đồ hoạt động của luồng sự kiện ấn nút



Khi nhấn nút ngắt INT0 được gọi và lưu thời gian bắt đầu nhấn nút, đặt biến chỉ sự kiện nhấn nút pressing = 1.

```

void external0() interrupt 0 {
    if (!pressing) {
        pressing = 1;
        pressing_start = counter0;
    }
    // 2ms
    TR1 = 0;
    TH1 = 0xF8;
    TL1 = 0x30;
    TR1 = 1;
}
  
```

Khi nhả nút, sau 2ms ngắt Timer1 sẽ được gọi.

Trong hàm xử lý ngắt Timer1

```

if (có sự kiện ấn nút)
    {tính thời gian nhấn nút};
    if (thời gian ấn nút >= 1s):
  
```



```

        long_pressing();

    else:

        short_pressing();

```

Nếu thời gian ấn nút  $\geq 1s$ , hàm `long_pressing()` sẽ được gọi. Hàm này sẽ thực hiện chuyển qua lại giữa trạng thái set thời gian T và trạng thái hiển thị thời gian.

```

if (đang ở trạng thái set T):

    {Chuyển sang trạng thái hiển thị}

else:

    {Chuyển sang trạng thái set T}

```

Nếu thời gian ấn nút  $< 1s$ , hàm `short_pressing()` sẽ được gọi. Hàm này sẽ thực hiện chuyển qua lại giữa trạng thái hiển thị thời gian T và trạng thái hiển thị thời gian t nếu đang ở chế độ hiển thị, nếu đang ở chế độ set T thì sẽ thực hiện chức năng điều chỉnh thời gian T.

```

if (đang ở trạng thái hiển thị):

    if (đang ở trạng thái hiển thị T):

        {Chuyển sang trạng thái hiển thị t}

    else:

        {Chuyển sang trạng thái hiển thị T}

else:

    {Điều chỉnh thời gian T}

```

### 3.2.4 Sự kiện khô nguy hiểm

Nếu không tưới trong khoảng thời gian  $t > T$  thì sẽ chuyển đến trạng thái khô nguy hiểm. Ở trạng thái khô nguy hiểm màn hình sẽ hiển thị thời gian t và nhấp nháy, loa sẽ kêu.

Để kiểm tra  $t > T$ , sử dụng ngắt Timer0, cứ sau 50ms sẽ so sánh thời gian không tưới t với T.

```

if (đang ở trạng thái khô thường && t >= T):

    {chuyển sang trạng thái khô nguy hiểm (gọi hàm too_dry())}

```

Khi ở trạng thái khô nguy hiểm, thực hiện nhấp màn hình hiển thị và phát loa:

```

void too_dry() {

    {Bắt đầu nhấp màn hình};

    {Bắt đầu phát loa}
}

```

```
}
```

Khi nhấp nháy màn hình sử dụng ngắt Timer0, các trạng thái sáng tối sẽ chuyển qua lại sau 500ms để tạo hiệu ứng nhấp nháy.

Sử dụng ngắt Timer1 để phát loa. Khởi tạo TH1 và TL1 là 0xF8, 0x30. Sau

$$\frac{(0xFFFF - 0xF830 + 1) * 12}{12 * 10^6} = 0.002s = 2ms \quad \text{ta sẽ đảo cực chân nối với loa để phát âm}$$

thanh. Cứ sau 100 nhịp thì lại dừng lại một khoảng thời gian.

```
void timer1() interrupt 3 {  
    if (không có sự kiện nhấn nút):  
        TH1 = 0xF8;  
        TL1 = 0x30;  
        if(isSpeak == 1){  
            P1_5 = ~P1_5;  
        }  
        timespeak += 1;  
        if (timespeak == 100){  
            isSpeak = !isSpeak;  
            timespeak = 0;  
        }  
}
```

### 3.2.5 Sự kiện tưới cây

Nút ứng với ngắt INT1 sẽ được mô hình như sensor độ ẩm.

Khi nhấn nút tương ứng với sự kiện tưới nước. Khi đó hệ thống sẽ ở trạng thái cây ướt.

Nếu không ấn nút nữa, hệ thống chuyển sang trạng thái khô thường và bắt đầu hiển thị thời gian t.

```
void external1() interrupt 2 {  
    t = 0;  
    {Chuyển về trạng thái khô thường}  
}
```



**Chi tiết code có thể xem tại :**

<https://github.com/tampmbk/CuoikyVXL/blob/master/cuoiki/maint.c>

## Chương IV : Kết luận

Như vậy, với bài tập lớn này, nhóm đã thực hiện được thành công một thiết bị báo động tưới cây thông qua bộ AT89S52 Development Kit. Qua đó, nhóm đã hiểu và nắm vững hơn các kiến thức về lập trình bộ vi điều khiển nói chung và vi điều khiển họ Intel 8051 nói riêng. Các tập lệnh của vi điều khiển, các kĩ thuật xử lý vào ra, xử lý ngắt cũng đã được thực hiện và ôn tập lại. Trên hết, với việc lập trình trên mạch thật và tạo ra sản phẩm thật, nhóm đã có nhiều kiến thức bổ ích về ứng dụng của vi điều khiển trong thực tế cũng như ứng dụng của ngành điện tử.

Tuy nhiên thiết bị báo động tưới cây chỉ là một thiết bị đơn giản để minh họa cho các kỹ thuật khi làm việc với vi điều khiển. Tiềm năng khai thác của AT89S52 là lớn hơn rất nhiều so với một thiết bị đơn giản như báo động tưới cây. Hiểu được điều này, nhóm mong muốn trong tương lai sẽ tìm hiểu nhiều hơn nữa về AT89S52 và xây dựng các thiết bị, hệ thống phức tạp hơn như máy tính điện tử.

## Tài liệu tham khảo

1. Wikipedia – Intel 8051 - [https://en.wikipedia.org/wiki/Intel\\_MCS-51](https://en.wikipedia.org/wiki/Intel_MCS-51)
2. AT89S52 data sheet - [https://www.keil.com/dd/docs/datashts/atmel/at89s52\\_ds.pdf](https://www.keil.com/dd/docs/datashts/atmel/at89s52_ds.pdf)
3. Proteus Manual - [https://www.ele.uva.es/~jesman/BigSeti/ftp/Cajon\\_Desastre/Software-Manuales/EBook%20-%20Proteus%20Manual.pdf](https://www.ele.uva.es/~jesman/BigSeti/ftp/Cajon_Desastre/Software-Manuales/EBook%20-%20Proteus%20Manual.pdf)
4. Walter A. Triebel, Avtar Singh - The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications - 1997.
5. Ngô Lam Trung - Slide môn Kỹ thuật vi xử lý