

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO ĐỒ ÁN II

ĐỀ TÀI:

THIẾT KẾ ĐỒNG HỒ SỐ SỬ DỤNG MSP430

GVHD: TS. Vương Hoàng Nam

Nhóm sinh viên thực hiện:

Đinh Thị Thu Hà	20131150
Nguyễn Thanh Hà	20131161
Nguyễn Thị Hà	20131163
Nguyễn Xuân Quyền	20133193
Đỗ Xuân Sơn	20133299
Bùi Văn Tâm	20133416
Nguyễn Thị Thanh Tâm	20133430

Hà Nội, 06/2017

LỜI NÓI ĐẦU

Ngành điện tử - truyền thông là một trong những ngành mũi nhọn của cuộc Cách mạng lần thứ 4, đây là ngành khoa học – kỹ thuật phát triển nhanh, có bước tiến nhảy vọt trong những năm gần đây. Trong khi lĩnh vực truyền thông giúp truyền tải thông tin một cách dễ dàng hữu ích thì lĩnh vực điện tử tạo ra những sản phẩm với vô vàn ứng dụng hữu ích từ đời sống, khoa học, quân sự...

Với tư cách là sinh viên ngành điện tử - truyền thông thì việc thiết kế ra một mạch điện tử không quá phức tạp là điều cần phải thực hiện được vì đây là một trong những kiến thức căn bản giúp sinh viên hiểu được nhiều hơn về thiết kế cũng như lập trình cho vi điều khiển. Vì vậy, trong khuôn khổ học phần “Đồ án 2”, nhóm sinh viên được giao cho nghiên cứu thiết kế “Đồng hồ số sử dụng MSP430”. Đây là một đề tài hay giúp nhóm sinh viên tìm hiểu thêm được về một loại vi điều khiển mới, cũng như củng cố thêm các kỹ năng liên quan đến thiết kế mạch điện tử, kỹ năng làm việc nhóm, tra cứu tài liệu...

Tuy nhiên, do kiến thức còn hạn chế, nên sản phẩm thu được còn chưa được hoàn thiện, nhóm sinh viên mong nhận được ý kiến góp ý bổ sung của các thầy hướng dẫn

Nhóm sinh viên xin chân thành cảm ơn thầy Vương Hoàng Nam và thầy Đặng Khánh Hòa đã hướng dẫn, hỗ trợ để nhóm sinh viên có thể hoàn thiện được học phần này.

MỤC LỤC

LỜI NÓI ĐẦU	1
MỤC LỤC	2
DANH MỤC HÌNH ẢNH	3
PHẦN 1: MỞ ĐẦU	4
1.1. Lý do chọn đề tài.....	4
1.2. Mục đích nghiên cứu.....	4
1.3. Đối tượng nghiên cứu	4
1.4. Nhiệm vụ nghiên cứu	4
1.5. Phạm vi nghiên cứu.....	4
1.6. Phương pháp nghiên cứu.....	4
1.7. Phân công công việc	5
PHẦN 2: TRIỂN KHAI ĐỀ TÀI	6
2.1. Tìm hiểu về vi điều khiển MSP430 và KIT MSP430	6
2.1.1. Giới thiệu tổng quát về MSP430	6
2.1.2. Sơ đồ chức năng của MSP430G2553.....	7
2.1.3. Sơ đồ các chân của MSP430G2553	9
2.1.4. Giới thiệu về KIT MSP430	12
2.2. Xác định yêu cầu, tính toán sử dụng các linh kiện hỗ trợ	17
2.2.1. Xác định yêu cầu.....	17
2.2.2. Tính toán sử dụng các linh kiện hỗ trợ	17
2.2.3. Yêu cầu phần mềm	18
2.3. Thiết kế mạch, vẽ mạch in	18
2.3.1. Thiết kế mạch:.....	18
2.3.2. Vẽ mạch in:	21
2.4. Xây dựng code theo từng chức năng.....	22
2.5. Test mạch, kiểm tra, sửa lỗi	38
2.6. Hoàn thiện mạch, xây dựng báo cáo	38
KẾT LUẬN	39
TÀI LIỆU THAM KHẢO	40

DANH MỤC HÌNH ẢNH

<i>Hình 1: Sơ đồ chức năng của MSP430G2553</i>	<i>7</i>
<i>Hình 2: Sơ đồ các chân của MSP430G2553</i>	<i>9</i>
<i>Hình 3: KIT MSP430G2</i>	<i>12</i>
<i>Hình 4: Các chân tín hiệu của Kit MSP430</i>	<i>14</i>
<i>Hình 5: Mạch khối nguồn</i>	<i>18</i>
<i>Hình 6: Mạch khối điều chỉnh</i>	<i>19</i>
<i>Hình 7: Mạch khối hiển thị</i>	<i>19</i>
<i>Hình 8: Mạch khối điều khiển</i>	<i>20</i>
<i>Hình 9: Mạch khối loa</i>	<i>20</i>
<i>Hình 10: Sơ đồ mạch hoàn chỉnh</i>	<i>21</i>
<i>Hình 11: Mạch in</i>	<i>21</i>

PHẦN 1: MỞ ĐẦU

1.1. Lý do chọn đề tài

Thiết kế mạch và lập trình cho vi điều khiển là một trong những kiến thức căn bản mà bất kì kỹ sư điện tử nào cũng cần phải nắm được.

Trong khuôn khổ học phần “Đồ án 2”, dưới sự gợi ý của thầy Vương Hoàng Nam, nhóm được chọn đề tài thiết kế “Đồng hồ số sử dụng MSP430”. Đề tài này giúp nhóm sinh viên vừa nghiên cứu được về phần cứng, diễn hình như nghiên cứu về MSP430, vừa học cách lập trình cho phần cứng đó.

1.2. Mục đích nghiên cứu

Nhóm sinh viên nghiên cứu đề tài này với mục đích chính là hoàn thiện học phần “Đồ án 2”, tuy nhiên đề tài này cũng giúp nhóm củng cố thêm kiến thức về vi điều khiển, lập trình cho phần cứng.

1.3. Đối tượng nghiên cứu

Đối tượng nghiên cứu chính của đề tài là về vi điều khiển MSP430: tìm hiểu về vi điều khiển, lập trình cho vi điều khiển.

1.4. Nhiệm vụ nghiên cứu

- Tìm hiểu về chip MSP430
- Thiết kế đồng hồ số bằng cách lập trình cho chip MSP430

1.5. Phạm vi nghiên cứu

Thiết kế đồng hồ số ra được sản phẩm thực tế, không chỉ dựa trên mô phỏng.

1.6. Phương pháp nghiên cứu

- Phân tích và tổng hợp
- Kiểm tra bằng thực nghiệm: chạy thử trực tiếp trên kit.

1.7. Phân công công việc

Nội dung công việc	Thành viên thực hiện
Tìm hiểu về MSP430.	Cả nhóm
Xây dựng yêu cầu chức năng, tính toán sử dụng các linh kiện hỗ trợ.	Đỗ Xuân Sơn, Nguyễn Xuân Quyền
Thiết kế mạch, vẽ mạch in.	Bùi Văn Tâm, Nguyễn Thanh Hà
Xây dựng code theo từng chức năng.	Bùi Văn Tâm, Đinh Thị Thu Hà
Test mạch, kiểm tra sửa lỗi	Cả nhóm
Hoàn thiện mạch, xây dựng báo cáo	Nguyễn Thị Hà, Nguyễn Thị Thanh Tâm

PHẦN 2: TRIỂN KHAI ĐỀ TÀI

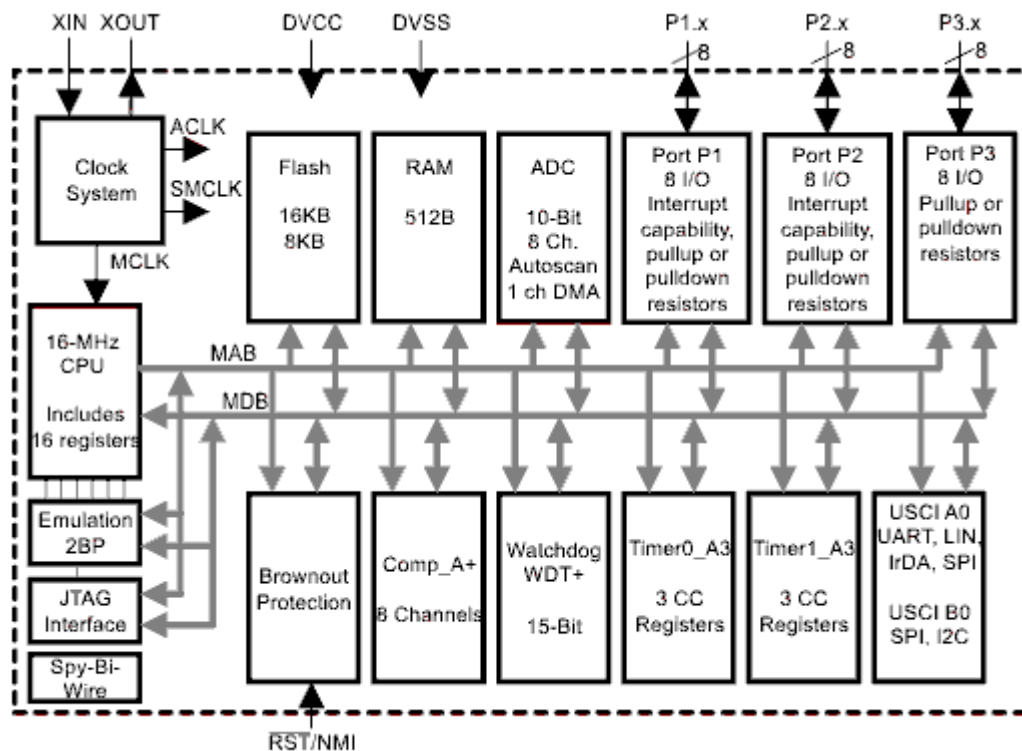
2.1. Tìm hiểu về vi điều khiển MSP430 và KIT MSP430

2.1.1. Giới thiệu tổng quát về MSP430

- MSP430 là dòng vi điều khiển với hệ thống bộ định thời và các cổng ngoại vi linh hoạt được kết nối với nhau theo cấu trúc VON-NEUMANN, có các Bus liên kết như: Bus địa chỉ bộ nhớ (MAB), Bus dữ liệu bộ nhớ (MDB). Đây là dòng VDK ra đời sau AVR hay PIC, nên nó có các đặc điểm tiên tiến hơn những người tiền nhiệm như tốc độ tính toán nhanh hơn (max 25MIPS), tiết kiệm điện hơn hẳn (dùng cho các ứng dụng dùng pin), nhiều module giao tiếp hơn, phần cứng mềm dẻo hơn, giá/tính năng thấp, thay đổi xung clock bằng phần mềm được, các chế độ ngủ ổn định....
- MSP430 có một số phiên bản như: MSP430x1xx, MSP430x2xx, MSP430x3xx, MSP430x4xx, MSP430x5xx. Dưới đây là những đặc điểm tổng quát của họ vi điều khiển MSP430:
- Cấu trúc sử dụng nguồn thấp giúp kéo dài tuổi thọ của Pin
 - Duy trì $0.1\mu A$ dòng nuôi RAM.
 - Chỉ $0.8\mu A$ real- time clock.
 - $250\mu A$ / MIPS.
- Bộ tương tự hiệu suất cao cho các phép đo chính xác
 - 12 bit hoặc 10 bit ADC-200 kskp, cảm biến nhiệt độ, Vref.
 - 12 bit DAC.
 - Bộ giám sát điện áp nguồn.
- 16 bit RISC CPU cho phép được nhiều ứng dụng, thể hiện một phần ở kích thước Code lập trình.
 - Thanh ghi lớn nên loại trừ được trường hợp tắc nghẽn tập tin khi đang làm việc.
 - Thiết kế nhỏ gọn làm giảm lượng tiêu thụ điện và giảm giá thành.
 - Tối ưu hóa cho những chương trình ngôn ngữ bậc cao như C, C++
 - Có 7 chế độ định địa chỉ.
 - Khả năng ngắt theo véc tơ lớn.

- Trong lập trình cho bộ nhớ Flash cho phép thay đổi Code một cách linh hoạt, phạm vi rộng, bộ nhớ Flash còn có thể lưu lại như nhật ký của dữ liệu.
- Do nhóm sử dụng KIT Launchpad MSP430G2553 nên nhóm sẽ trình bày những tìm hiểu của nhóm về vi điều khiển MSP430G2553.

2.1.2. Sơ đồ chức năng của MSP430G2553



Hình 1: Sơ đồ chức năng của MSP430G2553

Vi điều khiển MSP430G2553 gồm 6 khối với các chức năng cụ thể như sau:

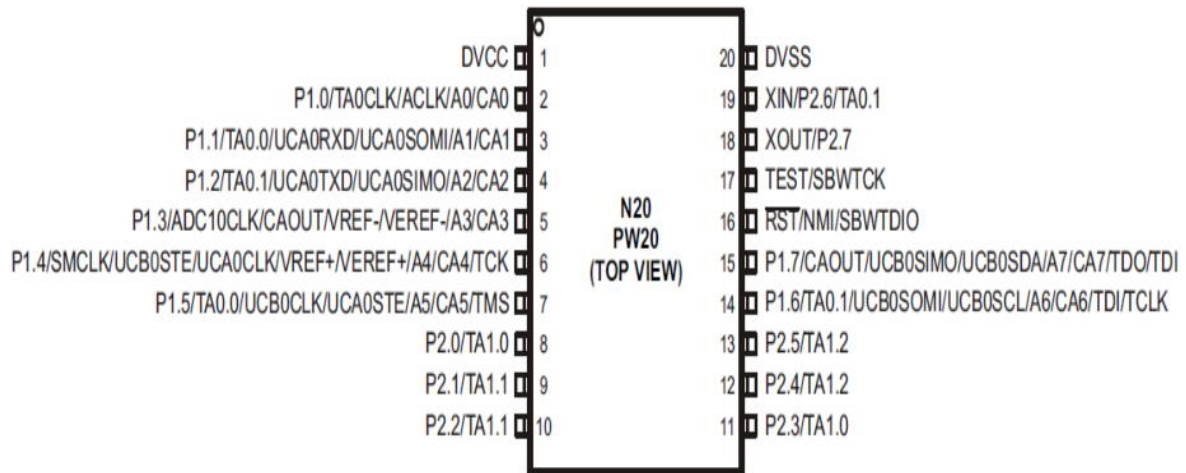
- Khối xử lý trung tâm bao gồm:
 - Khối tính toán số học/logic (ALU).
 - Khối giải mã lệnh và các mạch hỗ trợ xử lý ngắt, tái khởi động
 - Các thanh ghi bao gồm thanh ghi đếm chương trình PC, con trỏ ngăn xếp SP, thang ghi trạng thái (SR), thanh ghi tạo hàng số CG và 12 thanh ghi đa năng

- Bộ nhớ chương trình: Là bộ nhớ không mất dữ liệu khi mất điện. Trước kia là ROM, nay sử dụng FLASH. Chíp MSP430G2553 chỉ có 16KB.
- Bộ nhớ dữ liệu: RAM truy xuất tùy ý nhưng dữ liệu bị xóa khi mất điện. (Hiện đã có bộ nhớ dữ liệu không bị xóa khi mất điện).
- Các cổng nhập/xuất: Kết nối với các hệ thống khác.
- Đường BUS dữ liệu và BUS địa chỉ: Để truyền dữ liệu và lệnh giữa các khối.
- Khối xung nhịp: Tạo xung đồng bộ các khối.

Ngoài ra, vi điều khiển còn có 8 khối thường gặp khác:

- Khối định thời (Timer): Đếm thời gian chính xác. Các vi điều khiển hiện nay có ít nhất 2 khối này.
- Khối định thời cảnh báo: Là khối kiểm soát lỗi chương trình theo thời gian. Khối này sẽ tái khởi động chíp khi chương trình bị lỗi.
- Khối giao tiếp tuần tự: Kết nối với các IC khác bằng cách truyền từng bit.
- Khối nhớ dữ liệu không bay hơi: Lưu trữ dữ liệu ngay cả khi mất điện. Thường dùng để lưu cấu hình thiết bị như địa chỉ IP trong các ADSL router
- Khối biến đổi tương tự - số: Cho phép chuyển đổi tín hiệu tương tự sang dạng số.
- Khối biến đổi số - tương tự: Cho phép chuyển đổi tín hiệu tương tự sang dạng số, thường dùng để điều khiển động cơ bằng phương pháp xung số (PWM).
- Đồng hồ thời gian thực: Lưu giữ giá trị năm tháng ngày.
- Bộ nạp và chạy chương trình: Cho phép nạp chương trình từ máy tính vào bộ nhớ chương trình

2.1.3. Sơ đồ các chân của MSP430G2553



Hình 2: Sơ đồ các chân của MSP430G2553

- DVCC (1): Chân cấp nguồn VCC, dùng để cấp nguồn 3,3V cho chip. Nguồn có thể dao động trong khoảng 1,8V –3,6V.
- P1.0/TA0CLK/A0/CA0 (2):
 - o Chân xuất/nhập số.
 - o TA0CLK: ngõ vào nhận dạng xung clock cho Timer0_A.
 - o ACLK: ngõ ra xung clock ACLK.
 - o A0: ngõ vào tương tự A0 của kênh ADC10.
 - o CA0: đầu vào CA0 của Comparator_A+
- P1.1/TA0.0/UCARXD/UCASOMI/A1/CA1(3):
 - o Chân xuất/nhập số.
 - o TA0.0: Timer0_A.
 - o UCARXD: chân nhận dữ liệu USCI_A0 trong chế độ UART.
 - o UCASOMI: ngõ ra USCI_A0 khi hoạt động ở chế độ slave, ngõ vào khi hoạt động ở chế độ master trong chế độ SPI.
 - o A1: ngõ vào tương tự A1 của kênh ADC10.
 - o CA1: đầu vào CA1 của Comparator_A+
- P1.2/TA0.0/UCATXD/UCASIMO/A2/CA2(4):
 - o Chân xuất/nhập số.
 - o TA0.0: Timer0_A.

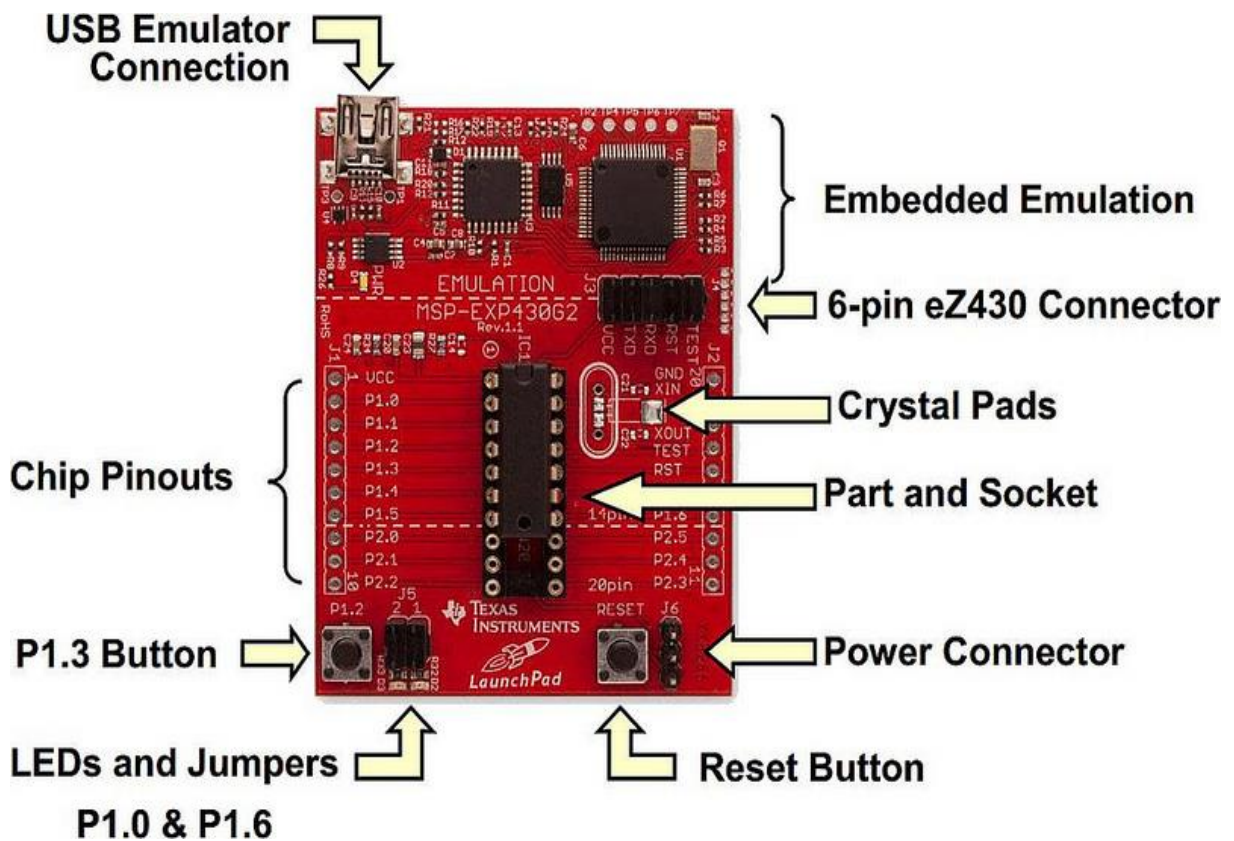
- UCA0TXD: chân truyền dữ liệu USCI_A0 trong chế độ UART.
- UCA0SIMO: ngõ vào USCI_A0 khi hoạt động ở chế độ slave, ngõ ra khi hoạt động ở chế độ master trong chế độ SPI.
- A2: ngõ vào tương tự A2 của kênh ADC10.
- CA2: đầu vào CA2 của Comparator_A+
- P1.3/ADC10CLK/CAOUT/VREF-/VEREF-/A3/CA3(5):
 - Chân xuất/nhập số.
 - ADC10CLK: đầu ra xung clock của ADC10.
 - CAOUT: đầu ra của Comparator_A+
 - VREF-/ VEREF-: điện áp tham chiếu âm VREF-/ VEREF-
 - A3: ngõ vào tương tự A3 của kênh ADC10.
 - CA3: đầu vào CA3 của Comparator_A+
- P1.4/SMCLK/UCA0CLK/ VREF+/VEREF+/A4/CA4/TCK(6):
 - Chân xuất/nhập số.
 - SMCLK: tín hiệu đầu ra SMCLK
 - VREF+/VEREF+: điện áp tham chiếu dương VREF+/VEREF+.
 - A4: ngõ vào tương tự A4 của kênh ADC10.
 - CA4: đầu vào CA4 của Comparator_A+
 - TCK: ngõ vào kiểm tra xung clock TCK.
- P1.5/TA0.0/UCB0CLK/UCA0STE/A5/CA5/TMS(7):
 - Chân xuất/nhập số.
 - UCB0CLK: xung clock UCSI_B0
 - UCA0STE: cho phép truyền dữ liệu UCSI_A0 trong chế độ slave.
 - A5: ngõ vào tương tự A5 của kênh ADC10.
 - CA5: đầu vào CA5 của Comparator_A+
 - TMS: ngõ ra lựa chọn chế độ kiểm tra
- P2.0/TA1.0(8):
 - Chân xuất/nhập số.
 - TA1.0: đầu ra Out0 ở chế độ so sánh của Timer0_A
- P2.1/TA1.1(9):
 - Chân xuất/nhập số.

- TA1.0: đầu ra Out1 ở chế độ so sánh của Timer0_A
- P2.2/TA1.1(10):
 - Chân xuất/nhập số.
 - TA1.1: đầu ra Out1 ở chế độ so sánh của Timer0_A
- P2.3/TA1.0(11):
 - Chân xuất/nhập số.
 - TA1.1: đầu ra Out0 ở chế độ so sánh của Timer0_A
- P2.4/TA1.2(12):
 - Chân xuất/nhập số.
 - TA1.1: đầu ra Out2 ở chế độ so sánh của Timer0_A
- P2.5/TA1.2(13):
 - Chân xuất/nhập số.
 - TA1.1: đầu ra Out2 ở chế độ so sánh của Timer0_A
- P1.5/TA0.1/UCB0SOMI/UCB0SCL/A6/CA6/TDI/TCLK(14):
 - Chân xuất/nhập số.
 - TA0.1: đầu ra Out1 của Timer0_A.
 - UCB0SCL: xung clock SCL I2C trong chế độ I2C.
 - UCB0SOMI: ngõ ra USCI_B0 khi hoạt động ở chế độ slave, ngõ vào khi hoạt động ở chế độ master trong chế độ SPI.
 - TDI: ngõ vào kiểm tra dữ liệu
 - TCLK: ngõ vào kiểm tra xung clock
 - A6: ngõ vào tương tự A6 của kênh ADC10.
 - CA6: đầu vào CA6 của Comparator_A+
- P1.6/CAOUT/UCB0SIMO/UCB0SDA/A7/CA7/TDO/TDI(15):
 - Chân xuất/nhập số.
 - UCB0SDA: chân duex liệu SDA I2C trong chế độ I2C.
 - UCB0SIMO: ngõ vào USCI_B0 khi hoạt động ở chế độ slave, ngõ ra khi hoạt động ở chế độ master trong chế độ SPI.
 - TDI: ngõ vào kiểm tra dữ liệu
 - TDO: ngõ ra kiểm tra dữ liệu.
 - A7: ngõ vào tương tự A7 của kênh ADC10.

- CA7: đầu vào CA7 của Comparator_A+
- $\overline{\text{RST}}/\text{NMI}/\text{SBWTDIO}(16)$:
 - Chân reset hoạt động ở mức thấp hoặc cấm ngắt.
- TEST/SBWTCCK(17):
- XOUT/P2.7(18):
- XIN/P2.6/TA0.1(19):
- DVSS (20): chân nối đất VSS.

2.1.4. Giới thiệu về KIT MSP430

a) Sơ lược cấu tạo Kit MSP430:



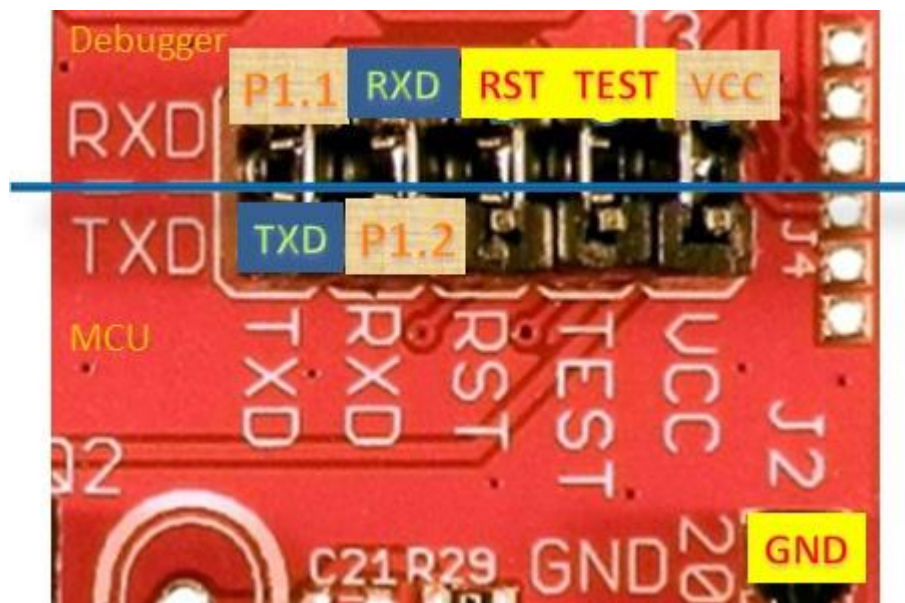
Hình 3: KIT MSP430G2

KIT MSP-EXP430G2 được phát triển bởi Texas Instruments hỗ trợ các dòng vi điều khiển MSP430G2xx. LaunchPad hỗ trợ cổng kết nối USB, 20 chân DIP socket. Nó là một công cụ hỗ trợ cho nạp code cũng như đánh giá và gỡ lỗi dễ dàng hơn.

Một Kit LaunchPad gồm hai thành phần, với GND được phủ chung:

- Nửa trên: (Embedded Emulation) là phần mạch nạp theo chuẩn spy-bi-wire Jtag (2 dây), kết hợp với chuyển đổi giao tiếp UART với máy tính. Trên cùng là đầu USBmini để nối với máy tính.
- Phía dưới là hàng Header để nối ra đối tượng cần giao tiếp, bao gồm các chân:
 - *TXD, RXD*: phục vụ giao tiếp UART với máy tính.
 - *RST, TEST*: phục vụ nạp và debug (sửa lỗi) theo chuẩn spy-bi-wire Jtag.
 - *VCC*: cấp nguồn 3V3 cho đối tượng (thường là nửa dưới LaunchPad).
- 6-pin eZ430 connector (J4): là 6 chân kết nối với mạch ngoài eZ430. Chức năng các chân như sau. Thứ tự chân từ trên xuống dưới, chân 1 bắt đầu từ phần bên Emulation.

Chân	Tín hiệu	Chức năng
1	TXT	Truyền dữ liệu ra cho của giao tiếp UART
2	VCC	Chân cấp điện áp cho khối giao tiếp với J4
3	TEST/SBTCK	Chế độ kiểm tra cho JTAG chân và Spy-Bi-Wire đầu vào đồng hồ kiểm tra trong quá trình lập trình và thử nghiệm
4	RST / SBWTDIO	Thiết lập lại, Spy-Bi-Wire đầu vào dữ liệu thử nghiệm / đầu ra trong quá trình lập trình và thử nghiệm
5	GND	Chân nối đất
6	TXD	UART nhận được dữ liệu đầu vào (giao tiếp UART từ bảng mạch eZ430 vào PC hoặc MSP430G2xx)



Hình 4: Các chân tín hiệu của Kit MSP430

- Nửa dưới: là một mạch phát triển MSP430 đơn giản, bao gồm:
 - Socket cắm MSP430 (khối IC1): gồm 20 chân cắm, hỗ trợ cho chip 14 chân hoặc 20 chân (hỗ trợ cho các dòng vdk MSP430G2xxx), Pad hàn thạch anh, nút nhấn Reset chip.
 - Nút nhấn gắn vào P1.3, hai Led hiển thị có jumper để gắn vào P1.0 và P1.6.
 - Hai hàng header để kết nối hai hàng chân của chip ra ngoài:
 - VCC (chân 1) VSS (chân 20) dùng để cấp nguồn 3,3V cho chip. Nguồn có thể dao động trong khoảng 1,8V – 3,6 V
 - P1.0–P1.7, P2.0, and P2.7 là 2 cổng nhập, xuất số. Mỗi cổng 8 chân 8 bit), gọi tắt là P1 và P2. Các khối chức năng cũng sử dụng các chân này khi cần nhờ cấu hình thanh ghi chọn khối P1SEL và P2 SEL.
 - Một hàng header nguồn *GND-GND-VCC* để lấy nguồn 3,3V trên LaunchPad cấp cho các khối liên kết với KIT.

b) Bộ nhớ KIT MSP430

Kit không có bộ nhớ riêng, bộ nhớ của kit nằm chính trong bộ nhớ của chip kết nối với khối IC1. KIT hỗ trợ các vi điều khiển dòng MSP430G2xx, các vi điều khiển này có bộ nhớ: 2-16KB ROM, 265 B hoặc 512 B RAM, 256 Byte Flash.

• **Chuẩn giao tiếp UART**

- Có thể nói đây là một trong những giao tiếp phổ biến và dễ sử dụng nhất, dùng để truyền nhận dữ liệu giữa 2 hay nhiều thiết bị khác nhau không nhất thiết là vi điều khiển. UART là chuẩn truyền thông đồng bộ (nghĩa là 2 thiết bị giao tiếp với nhau cần phải biết được các thông số của khung truyền như

tốc độ, độ dài khung, số bit stop, bit parity....). Khi muốn dùng giao tiếp UART thì chúng ta phải khai báo hàm khởi tạo giá trị cho các thanh ghi UART, cụ thể:

- Chọn các chân dùng UART (VĐK MSP430G2553 là P1.1 và P1.2) để ở chức năng giao tiếp ngoại vi, tùy vào thứ tự ưu tiên chúng ta chọn thanh ghi PxSEL và PxSEL2 cho phù hợp.
- Định dạng khung truyền dữ liệu, số bit stop, bit parity, số bit dữ liệu bằng thanh ghi UCA0CTL0
- Cài đặt nguồn cấp xung clk
- Định tần số giao tiếp (Chú ý là module trên kit Launchpad chỉ hỗ trợ duy nhất tần số 9600, muốn dùng tần số khác các bạn buộc phải nối với module UARTto Com bên ngoài).
- Set cờ cho phép ngắt nhận tín hiệu.

2.2. Xác định yêu cầu, tính toán sử dụng các linh kiện hỗ trợ

2.2.1. Xác định yêu cầu

a) Yêu cầu chức năng:

- Hiển thị giờ
- Hiển thị ngày/tháng/năm
- Hiển thị nhiệt độ.
- Chỉnh ngày, giờ.
- Hẹn giờ.
- Chỉnh độ tương phản.
- Đồng hồ đếm ngược.

b) Yêu cầu phi chức năng:

- Sử dụng vi điều khiển MSP430.
- Đồng hồ chạy tốt ổn định
- Giá thành hợp lý, không quá cao.

2.2.2. Tính toán sử dụng các linh kiện hỗ trợ

- Điều khiển bằng MSP430.
- Nguồn cấp 3.3V, với nguồn cấp cho VĐK này chúng ta hoàn toàn có thể sử dụng nguồn cấp 5V qua LM1117(IC ổn áp tuyến tính).
- Hiển thị tất cả các thông số trên LED đồng hồ số chuyên dụng, dùng 74HC595 để giảm số chân điều khiển.
- Dùng IC BQ32000 làm đồng hồ giữ nhịp.
- Xây dựng menu chứa phần chỉnh giờ, ngày, tháng, năm, chỉnh độ tương phản, hẹn giờ.
- 4 nút nhấn: Up, Down, Enter, Exit để có thể vào ra trong menu.
- Giữ nút Up trong vài giây để vào Menu.
- Chức năng tự động trở về chương trình chính (Hiện giờ) khi không có nút nào được nhấn trong 1 phút. Dùng timer để định thời gian cho tác vụ này.
- Thêm chân TX của UART để dùng khi Test code.

2.2.3. Yêu cầu phần mềm

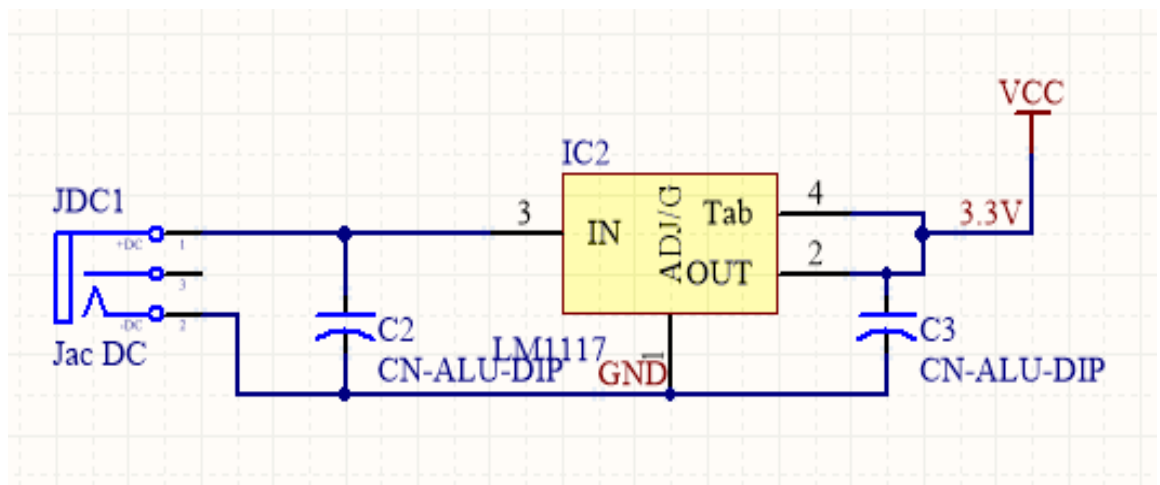
- Phần mềm vẽ mạch: Altium
- Lập trình: CCS
- Phần mềm đọc UART.
- Datasheet: MSP430G2553, MSP43x2xx, LM1117, LED đồng hồ
- Thư viện MSP430: I2C, UART, Flash, Clock, Switch, ADC.

2.3. Thiết kế mạch, vẽ mạch in

2.3.1. Thiết kế mạch:

a) Khối nguồn:

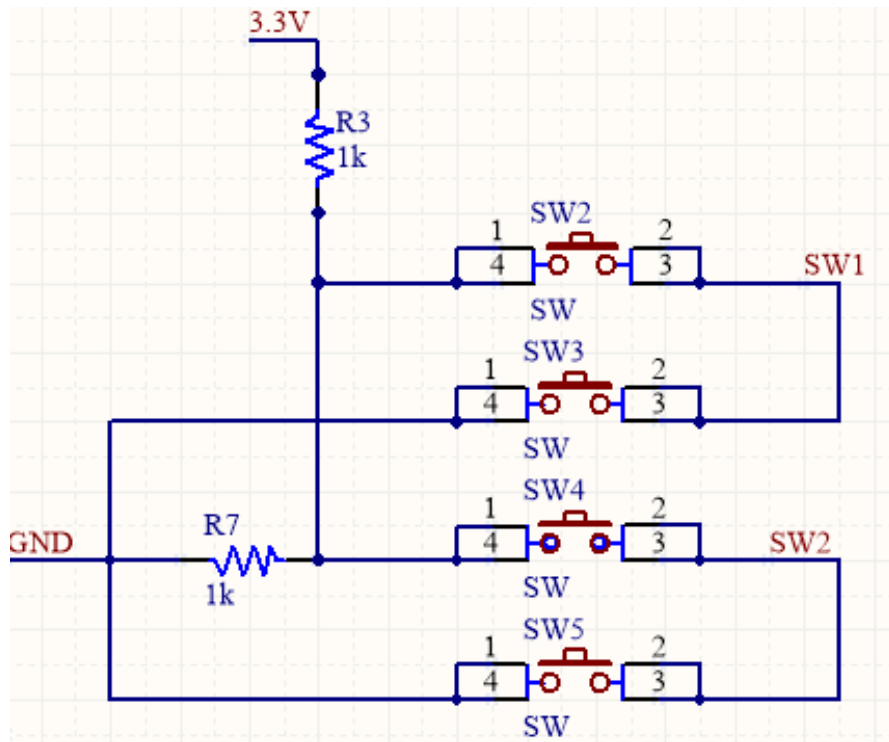
Nguồn gồm có LM1117 và Jack cắm một chiều JDC nên khá đơn giản. Nguồn cấp cho vi điều khiển là 3.3V.



Hình 5: Mạch khối nguồn

b) *Khối điều chỉnh:*

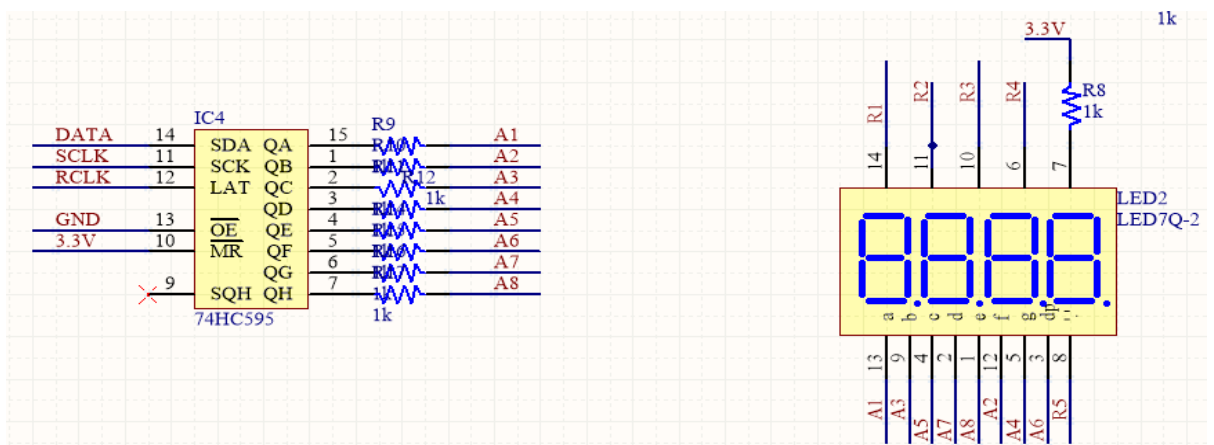
- Bao gồm 4 nút bấm để điều chỉnh ngày giờ, hẹn giờ, điều chỉnh độ tương phản theo yêu cầu chức năng đã xác định ở trên.
- Do nút bấm còn thừa 2 chân nên ta đưa ADC vào để giảm lỗi.



Hình 6: Mạch khối điều chỉnh

c) *Khối hiển thị:*

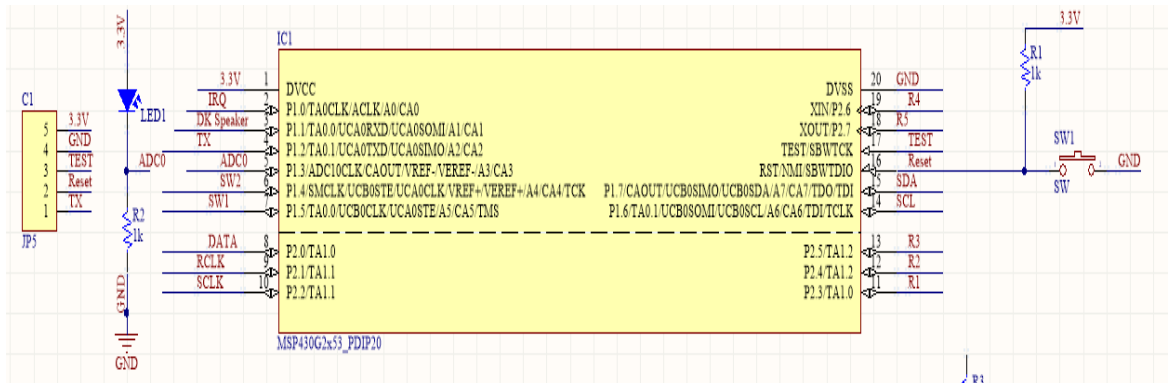
Sử dụng LCD để hiển thị ngày giờ, đồng thời sử dụng IC 74HC595 để tiết kiệm số chân vi điều khiển.



Hình 7: Mạch khối hiển thị

d) **Khởi điều khiển:**

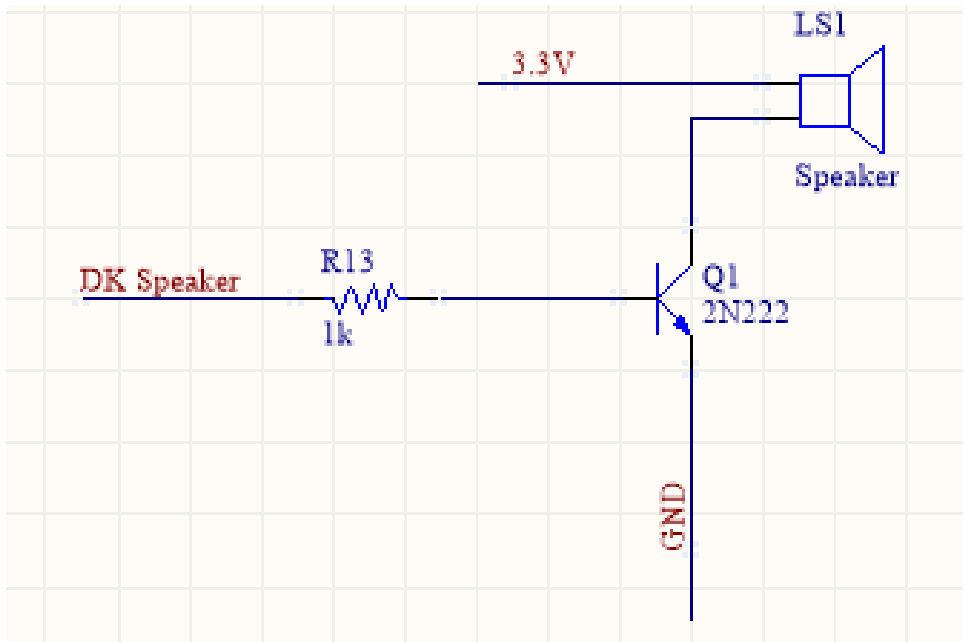
Sử dụng vi điều khiển MSP430G2553 vì có sẵn kit và bộ nhớ lớn.



Hình 8: Mạch khởi điều khiển

e) **Khởi âm thanh:**

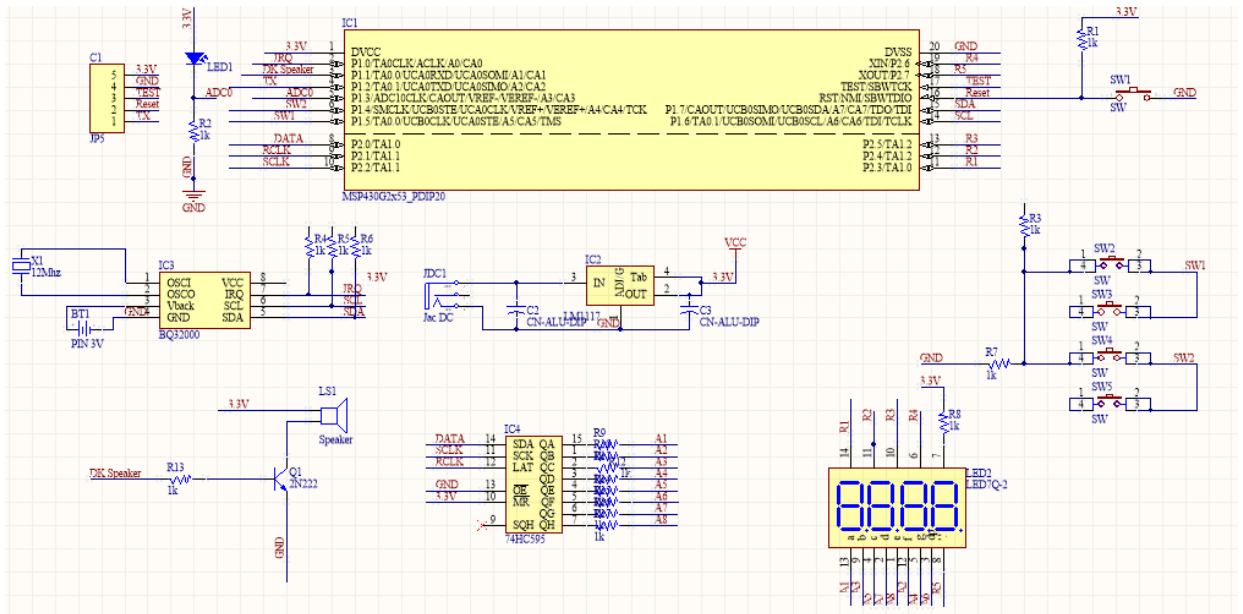
Sử dụng loa để hẹn giờ báo thức.



Hình 9: Mạch khởi loa

f) Ghép nối các khối lại với nhau ta được mạch hoàn chỉnh:

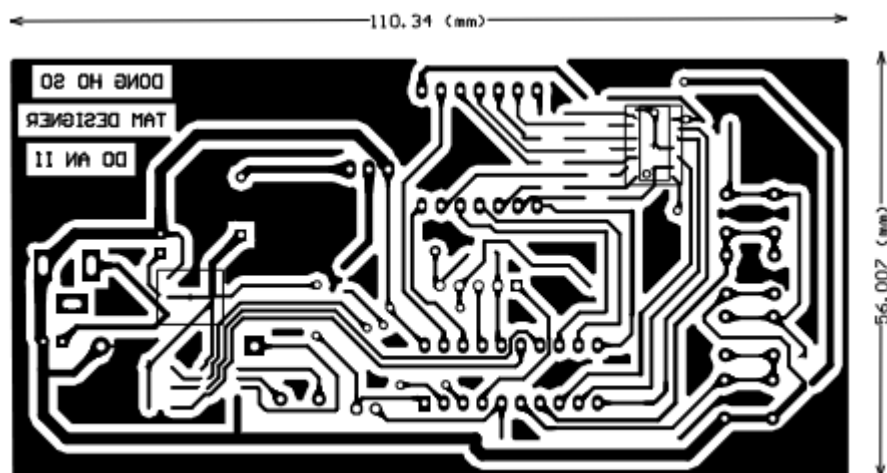
Sau khi thiết kế mạch cho từng khối, ta ghép nối chúng lại với nhau để có được mạch hoàn chỉnh.



Hình 10: Sơ đồ mạch hoàn chỉnh

2.3.2. Vẽ mạch in:

Sau khi thiết kế được mạch hoàn chỉnh, ta vẽ mạch in để thực hiện lắp đặt mạch, mạch in đạt yêu cầu kích thước nhỏ gọn, các đường đi dây được sắp xếp gọn gàng, đạt yêu cầu.



Hình 11: Mạch in

2.4. Xây dựng code theo từng chức năng

Sử dụng phần mềm CCS (C Composer Studio).

- Phần mềm C Composer Studio (CCS) là một phần mềm lập trình hầu hết cho các dòng MSP430 dễ dàng sử dụng với trình biên dịch Eclipse và miễn phí với ứng dụng < 16KB.
- CCS cho phép lập trình với hầu hết các vi điều khiển hiện có.
- Đối với MSP430 thì CCS cho phép lập trình với ngôn ngữ C, thực hiện debug và nạp chương trình trực tiếp cho vi điều khiển thông qua mạch nạp tích hợp trên KIT MSP430.

- **Hàm hiển thị giờ:**

```
void Hien_thi_gio()
{
    delay_led_on=delay_led_on_chuan/2+Light/2;
    delay_led_off=delay_led_off_chuan/2+(250-Light)/2;

    GIO10_ON;
    if(Dau_hai_cham)P2OUT&=~BIT7;    //Bat dau hai cham
    else P2OUT|=BIT7;    //Tat dau hai cham
    Write_74HC595_8bit(soled[gio10BCD]);
    _delay_us(delay_led_on);
    P2OUT|=BIT7; //Tat dau hai cham
    Write_74HC595_8bit(0xFF);
    _delay_us(delay_led_off);

    GIO_ON;
    Write_74HC595_8bit(soled[gioBCD]);
    _delay_us(delay_led_on);
    Write_74HC595_8bit(0xFF);
    _delay_us(delay_led_off);

    PHUT10_ON;
    Write_74HC595_8bit(soled[phut10BCD]);
    _delay_us(delay_led_on);
    Write_74HC595_8bit(0xFF);
    _delay_us(delay_led_off);

    PHUT_ON;
    Write_74HC595_8bit(soled[phutBCD]);
    _delay_us(delay_led_on);
    Write_74HC595_8bit(0xFF);
    _delay_us(delay_led_off);
}
```

- **Hàm hiển thị ngày:**

```
void Hien_thi_ngay()
{
    Enable_timerA0=1;    //Cho phép dem timer
    Dem_timerA0=0;
    while(press==0x02)
    {
        if(Dem_timerA0==0)
        {
            Doc_thoi_gian();
#ifdef USE_UART
            UART_Write_Char(10);                //Ky tu xuong dong
            sprintf(buff,"Date:%d:%d:20%d",ngay,thang,nam);
            UART_Write_String(buff);
#endif
            Dem_timerA0++;
            SWITCH_CheckKey();
        }
        if(Dem_timerA0>=5)Dem_timerA0=0; //Sau 1s thì quét lại nút bấm

        GIO10_ON;
        Write_74HC595_8bit(soled[ngay10BCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        GIO_ON;
        Write_74HC595_8bit(soled[ngayBCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        dau_cham=1;
        PHUT10_ON;
        Write_74HC595_8bit(soled[thang10BCD]);
        _delay_us(delay_led_on);
        dau_cham=0;
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;
        Write_74HC595_8bit(soled[thangBCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
    Dem_timerA0=0;
    //Hien thi nam
    while(Dem_timerA0<4)
    {
        GIO10_ON;
        Write_74HC595_8bit(soled[2]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
}
```



```

        GIO_ON;
        Write_74HC595_8bit(soled[0]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT10_ON;
        Write_74HC595_8bit(soled[nam/10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;
        Write_74HC595_8bit(soled[nam%10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
    Enable_timerA0=0;
    Dem_timerA0=0;
}

```

- **Hàm đọc thời gian:**

```

void Doc_thoi_gian()
{
    unsigned char data[8];
    I2C_USCI_Read_Word(0x00,data,8);
    giay = data[1];
    phut = data[2];
    gio  = data[3];
    ngay = data[5];
    thang= data[6];
    nam  = data[7];

    ngay10BCD =(data[5]&0x30)>>4;
    ngayBCD   =data[5]&0x0F;
    thang10BCD=(data[6]&0x10)>>4;
    thangBCD  =data[6]&0x0F;

    giay = ((giay&0x70)>>4)*10 + (giay & 0x0F);
    phut = ((phut&0x70)>>4)*10 + (phut & 0x0F);
    gio  = ((gio&0x30)>>4)*10  + (gio  & 0x0F);
    ngay = ((ngay&0x30)>>4)*10 + (ngay & 0x0F);
    thang= ((thang&0x10)>>4)*10+ (thang& 0x0F);
    nam  = ((nam&0xF0)>>4)*10  + (nam  & 0x0F);
}

```

- **Hàm đọc nhiệt độ:**

```

void Doc_nhiet_do()
{
    long Temperature=0;
    Enable_timerA0=1;
    Dem_timerA0=0;
    while(press==0x04)
    {
        if(Dem_timerA0==0)
        {
            Temperature=
(Temp[7]+Temp[6]+Temp[5]+Temp[4]+Temp[3]+Temp[2]+Temp[1]+Temp[0])>>3;
            Temperature = (Temperature*3125/128-10180)/25;
#ifdef USE_UART
                UART_Write_Char(10);                //Ky tu xuong dong
                sprintf(buff,"Temp:%d *C",Temperature);
                UART_Write_String(buff);
#endif

                SWITCH_CheckKey();
                Dem_timerA0++;
        }
        if(Dem_timerA0>=5)Dem_timerA0=0;

        if(Temperature>0)
        {
            GPIO_ON;
            Write_74HC595_8bit(soled[Temperature/10]);
            _delay_us(delay_led_on);
            Write_74HC595_8bit(0xFF);
            _delay_us(delay_led_off);

            PHUT10_ON;
            Write_74HC595_8bit(soled[Temperature%10]);
        }
        else
        {
            GPIO10_ON;
            Write_74HC595_8bit(soled[11]);    // Dau -
            _delay_us(delay_led_on);
            Write_74HC595_8bit(0xFF);
            _delay_us(delay_led_off);

            GPIO_ON;
            Write_74HC595_8bit(soled[-Temperature/10]);
            _delay_us(delay_led_on);
            Write_74HC595_8bit(0xFF);
            _delay_us(delay_led_off);

            PHUT10_ON;
            Write_74HC595_8bit(soled[-Temperature%10]);
        }

        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        //Hien *C
    }
}

```

```

        PHUT_ON;
        Write_74HC595_8bit(soleled[12]); /*C
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
    Enable_timerA0=0;
    Dem_timerA0=0;
}

```

- **Hàm chỉnh độ tương phản:**

```

void Chinh_do_tuong_phan()
{
    while(1)
    {
        Hien_thi_gio();
        SWITCH_Key();
        if(press==1)Dem_timerA0=0;
        else if(press== 2) //Tang do tuong phan
        {
            Dem_timerA0=0;
            if(delay_led_off_chuan>10)delay_led_off_chuan-=10;
            else delay_led_off_chuan=10;
            delay_led_on_chuan = 255-delay_led_off_chuan;
#ifdef USE_UART
            UART_Write_Char(10);                //Ky tu xuong dong
            sprintf(buff,"Delay_on:%d
Delay_off:%d",delay_led_on_chuan,delay_led_off_chuan);
            UART_Write_String(buff);
#endif
        }
        else if(press == 4) //Giam do tuong phan
        {
            Dem_timerA0=0;
            if(delay_led_off_chuan<250)delay_led_off_chuan+=10;
            else delay_led_off_chuan=250;
            delay_led_on_chuan = 255 - delay_led_off_chuan;
#ifdef USE_UART
            UART_Write_Char(10);                //Ky tu xuong dong
            sprintf(buff,"Delay_on:%d
Delay_off:%d",delay_led_on_chuan,delay_led_off_chuan);
            UART_Write_String(buff);
#endif
        }
        else if(press == 8) //Thoat
        {
            Dem_timerA0=0;
            press=0;
            break;
        }
        if(Dem_timerA0>=120)break;
    }
    Flash_Write_Byte(SegmentD,delay_led_on_chuan,0);
    Flash_Write_Byte(SegmentD,delay_led_off_chuan,1);
    Dem_timerA0=0;
    Enable_timerA0=0;
}

```

- **Hàm chỉnh giờ:**

```
void Chinh_gio()
{
    unsigned char temp=0;    //Phan chuc nang phim 2 va 3 chinh gio va phut
    while(1)
    {
        //Hien thi gio
        delay_led_on=delay_led_on_chuan/2+Light/2;
        delay_led_off=delay_led_off_chuan/2+(250-Light)/2;

        GIO10_ON;
        P2OUT&=~BIT7;    //Bat dau hai cham

        if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[gio10BCD]);
        _delay_us(delay_led_on);
        P2OUT|=BIT7; //Tat dau hai cham
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        GIO_ON;

        if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[gioBCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT10_ON;

        if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[phut10BCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;

        if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[phutBCD]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        SWITCH_Key();
        if(press!=0)
        {
            Enable_timerA0=1;
            Dem_timerA0=0;
            //__bic_SR_register(GIE);
            switch(press)
            {
                case 1:
                {
                    if(temp==1)temp=0;
                    else temp=1;
                    break;
                }
                case 2:    //Tang phut/gio
                {
                    if(temp==0) //Gio
                    {
```

```

        if(gio==23)gio=0;
        else gio++;

I2C_USCI_Write_Byte(0x02,((gio/10)<<4)+gio%10);
    }
    else //Phut
    {
        if(phut==59)phut=0;
        else phut++;

I2C_USCI_Write_Byte(0x01,((phut/10)<<4)+phut%10);
    }
    break;
}
case 4: //Giam phut/gio
{
    if(temp==0) //Gio
    {
        if(gio==0)gio=23;
        else gio--;

I2C_USCI_Write_Byte(0x02,((gio/10)<<4)+gio%10);
    }
    else //Giam phut
    {
        if(phut==0)phut=59;
        else phut--;

I2C_USCI_Write_Byte(0x01,((phut/10)<<4)+phut%10);
    }
    break;
}
case 8:
{
    Dem_timerA0=120;
    break;
}
}
//__bis_SR_register(GIE);
}
if(Dem_timerA0>=120)break;
}
press=0;
Dem_timerA0=0;
Enable_timerA0=0;
}

```

- **Hàm chỉnh ngày:**

```
void Chinh_ngay()
{
    unsigned char temp=0;    //Phan chuc nang phim 2 va 3 chinh gio va phut
    while(1)
    {
        ngay10BCD =ngay/10;
        ngayBCD   =ngay%10;
        thang10BCD=thang/10;
        thangBCD  =thang%10;

        //Hien thi ngay
        GIO10_ON;

        if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[ngay10BC
D]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        GIO_ON;

        if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[ngayBCD]
);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        dau_cham=1;
        PHUT10_ON;

        if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[thang10B
CD]);
        _delay_us(delay_led_on);
        dau_cham=0;
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;

        if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[thangBCD
]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        SWITCH_Key();
        if(press!=0)
        {
            Enable_timerA0=1;
            Dem_timerA0=0;
            //__bic_SR_register(GIE);           //Ngat timer
            switch(press)
            {
                case 1:
                {
                    if(temp==1)temp=0;
                    else temp=1;
                }
            }
        }
    }
}
```

```

        break;
    }
    case 2:      //Tang ngay/thang
    {
        if(temp==0)    //ngay
        {
            if((thang==1)|| (thang==3)|| (thang==5)|| (thang==7)|| (thang==8)|| (thang==
10)|| (thang==12))
            {
                if(ngay==31)ngay=1;
                else ngay++;
            }
            else
            if((thang==4)|| (thang==6)|| (thang==9)|| (thang==11))
            {
                if(ngay==30)ngay=1;
                else ngay++;
            }
            else if(thang==2)
            {
                if(nam%4==0)
                {
                    if(ngay==29)ngay=1;
                    else ngay++;
                }
                else
                {
                    if(ngay==28)ngay=1;
                    else ngay++;
                }
            }
        }

        I2C_USCI_Write_Byte(0x04,((ngay/10)<<4)+ngay%10);
    }
    else    //thang
    {
        if(thang==12)
        {
            thang=1;
            nam++;
        }
        else thang++;

        I2C_USCI_Write_Byte(0x05,((thang/10)<<4)+thang%10);
        I2C_USCI_Write_Byte(0x06,((nam/10)<<4)+nam%10);
    }
    break;
}
case 4: //Giam ngay/thang
{
    if(temp==0)        //ngay
    {
        if(ngay==1)
        {
            if((thang==1)|| (thang==3)|| (thang==5)|| (thang==7)|| (thang==8)|| (thang==
10)|| (thang==12))ngay=31;

```

```

else
if((thang==4)|| (thang==6)|| (thang==9)|| (thang==11)) ngay=30;
else
if((thang==2)&&(nam%4==0)) ngay=29;
else ngay=28;
}
else ngay--;

I2C_USCI_Write_Byte(0x04,((ngay/10)<<4)+ngay%10);
}
else //Giam thang
{
if(thang==1)
{
thang=12;
nam--;
}
else thang--;

I2C_USCI_Write_Byte(0x05,((thang/10)<<4)+thang%10);
I2C_USCI_Write_Byte(0x06,((nam/10)<<4)+nam%10);
}
break;
}
case 8:
{
Dem_timerA0=120;
break;
}
}
//__bis_SR_register(GIE);
#ifdef USE_UART
UART_Write_Char(10); //Ky tu xuong dong
sprintf(buff,"Date:%d:%d:20%d",ngay,thang,nam);
UART_Write_String(buff);
#endif
}
if(Dem_timerA0>=120)break;
}
press=0;
Dem_timerA0=0;
Enable_timerA0=0;
}

```


- **Hàm kiểm tra báo thức:**

```
void Kiem_tra_bao_thuc()
{
    unsigned char i=0;
    //Kiem tra bao thuc
    Enable_timerA0=1;
    Dem_timerA0=0;
    while(1)
    {
        Hien_thi_gio();
        if(i<20)i++;
        else i=0;

        if(i==0)
        {
            SWITCH_CheckKey();
            P1OUT ^=BIT1;
            if(press!=0)
            {
                Ngat_chuong();
                break;
            }
        }
        if(Dem_timerA0>=600)break;        //Keu trong 5 phut,qua thi tu tat
    }
    Ngat_chuong();
    Enable_timerA0=0;
    Dem_timerA0=0;
}
```

- **Hàm hẹn giờ:**

```
void Hen_gio()
{
    unsigned char temp=0;        //Phan chuc nang phim 2 va 3 chinh gio va phut
    //Chu y bit 7 cua gio_BT =1 thi khong cho bao thuc,bao 0 thi cho bao thuc
    if(gio_BT[0]>23)temp=2;        //Neu co ngat bao thuc bat thi chuyen sang trang thai
2-tat bao thuc
    Enable_timerA0=1;            //Cho phep dem timer
    Dem_timerA0=0;
    while(1)
    {
        // Chuong trinh hoat dong nhu sau
        // Nhan phim 1 de chon chinh gio,phut hay tat bao thuc
        // Nhan 2 de tang len,3 de giam xuong,4 de thoat
        // Truoc khi thoat du lieu bao thuc duoc luu vao Flash tai o 2(gio),3(phut)

        // Hien thi gio bao thu
        if(temp!=2)
        {
            GIO10_ON;
            P2OUT&=~BIT7;        //Bat dau hai cham

            if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[gio_BT[0]/10]);
            _delay_us(delay_led_on);
        }
    }
}
```

```

        P2OUT|=BIT7; //Tat dau hai cham
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        GIO_ON;

    if((temp!=0)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[gio_BT[0]%10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT10_ON;

    if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[phut_BT[0]/10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;

    if((temp!=1)|| (giay%2==0)|| (Vua_nhan))Write_74HC595_8bit(soled[phut_BT[0]%10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
    else //Tat bao thuc,hien chu "OFF"
    {
        GIO_ON;
        Write_74HC595_8bit(soled[0]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT10_ON;
        Write_74HC595_8bit(soled[13]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;
        Write_74HC595_8bit(soled[13]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);
    }
    SWITCH_Key(); //Quet ban phim
    if(press!=0)
    {
        Dem_timerA0=0;
        switch(press)
        {
            case 1:
            {
                if(temp==2)temp=0;
                else temp++;
                if(temp==2)
                {
                    gio_BT[0]=BIT7;
                    phut_BT[0]=BIT7;

```

```

        }
        else
        {
            if(gio_BT[0]>23){
                gio_BT[0]^=BIT7;
                phut_BT[0]^=BIT7;
            }
        }
        break;
    }
    case 2:      //Tang phut/gio
    {
        if(temp==0) //Gio
        {
            if(gio_BT[0]==23)gio_BT[0]=0;
            else gio_BT[0]++;
        }
        else if(temp==1) //Phut
        {
            if(phut_BT[0]==59)phut_BT[0]=0;
            else phut_BT[0]++;
        }
        break;
    }
    case 4: //Giam phut/gio
    {
        if(temp==0)      //Gio
        {
            if(gio_BT[0]==0)gio_BT[0]=23;
            else gio_BT[0]--;
        }
        else if(temp==1) //Giam phut
        {
            if(phut_BT[0]==0)phut_BT[0]=59;
            else phut_BT[0]--;
        }
        break;
    }
    case 8:
    {
        Flash_Write_Byte(SegmentD,gio_BT[0],2);
        Flash_Write_Byte(SegmentD,phut_BT[0],3);
        Dem_timerA0=120;
        break;
    }
    }
    }
    if(Dem_timerA0>=120)break;
}
press=0;
Dem_timerA0=0;
Enable_timerA0=0;
}

```

- **Hàm đồng hồ đếm ngược:**

```
void Dong_ho_dem_nguoc()
{
    Reset:
    //Chu y bit 7 cua gio_BT =1 thi khong cho bao thuc,bao 0 thi cho bao thuc
    phut_dem_nguoc=0;
    giay_dem_nguoc=0;
    Enable_timerA0=1;           //Cho phep dem timer
    Dem_timerA0=0;
    Co_dem_nguoc=0;
    while(1)
    {
        // Chuong trinh hoat dong nhu sau
        // Nhan phim 1 bat dau
        // Nhan 2 de tang phut,3 de tang giay,4 de thoat
        // Hien thi gio bao thu

        GPIO10_ON;
        P2OUT&=~BIT7;           //Bat dau hai cham
        if(phut_dem_nguoc>=10)Write_74HC595_8bit(soleled[phut_dem_nguoc/10]);
        _delay_us(delay_led_on);
        P2OUT|=BIT7; //Tat dau hai cham
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        GPIO_ON;
        Write_74HC595_8bit(soleled[phut_dem_nguoc%10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT10_ON;
        Write_74HC595_8bit(soleled[giay_dem_nguoc/10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        PHUT_ON;
        Write_74HC595_8bit(soleled[giay_dem_nguoc%10]);
        _delay_us(delay_led_on);
        Write_74HC595_8bit(0xFF);
        _delay_us(delay_led_off);

        SWITCH_Key();           //Quet ban phim
        if(press!=0)
        {
            Dem_timerA0=0;
            if((press==1)&&(Co_dem_nguoc==0))
            {
                Co_dem_nguoc=1;
                Enable_timerA0=0;
                Dem_timerA0=0;
            }
            else if((press==2)&&(Co_dem_nguoc==0))
            {
                if(phut_dem_nguoc<99)phut_dem_nguoc++;
                else phut_dem_nguoc=0;
            }
        }
    }
}
```

```

    }
    else if((press==4)&&(Co_dem_nguoc==0))
    {
        if(giay_dem_nguoc<59)giay_dem_nguoc++;
        else giay_dem_nguoc=0;
    }
    else if(press==8)break;
}
if(Dem_timerA0>=120)break;
if(Co_dem_nguoc&&(!giay_dem_nguoc)&&(!phut_dem_nguoc))
{
    Do_chuong();
    _delay_cycles(300000);
    Ngat_chuong();
    _delay_cycles(300000);
    Do_chuong();
    _delay_cycles(300000);
    Ngat_chuong();
    _delay_cycles(300000);
    Do_chuong();
    _delay_cycles(300000);
    Ngat_chuong();
    goto Reset;
}
}

Co_dem_nguoc=0;
press=0;
Dem_timerA0=0;
Enable_timerA0=0;
}

// Timer A0 interrupt service routine
#pragma vector=TIMER0_A0_VECTOR
__interrupt void Timer_A0 (void)
{
    if((Dem_timerA0<65535)&&(Enable_timerA0==1))Dem_timerA0++;
}

#pragma vector = WDT_VECTOR
__interrupt void Watchdog (void)
{
    WDTCTL |= WDTPW;
}

#pragma vector = PORT1_VECTOR
__interrupt void Interrupt_PORT1(void)
{
    if(P1IFG&BIT0)
    {
        //Giam thoi gian luu vua nhan nut
        if(Vua_nhan)Vua_nhan--;
        //Giam thoi gian cua dong ho dem nguoc
        if(Co_dem_nguoc)
        {
            if(giay_dem_nguoc>0)giay_dem_nguoc--;
            else {
                if(phut_dem_nguoc){

```

```

        phut_dem_nguoc--;
        giay_dem_nguoc=59;
    }
}
//Nhay dau cham
Dau_hai_cham ^=0x01;
if((phut==0)&&(giay==0))Doc_thoi_gian();
else if(giay<59)giay++;
    else
    {
        giay=0;
        if(phut<59)phut++;
        else
        {
            phut=0;
            if(gio<23)gio++;
            else gio=0;
        }
    }

phut10BCD = phut/10;
phutBCD = phut%10;
if(gio<10)gio10BCD=10;           //Khong hien thi
else gio10BCD = gio/10;
gioBCD = gio%10;
/*
UART_Write_Char(10);           //Ky tu xuong dong
sprintf(buff,"Time:%d:%d:%d",gio,phut,giay);
UART_Write_String(buff);
*/
P1IFG &=~BIT0;           //xoa co ngat tai chan P1.0
}
}

```

2.5. Test mạch, kiểm tra, sửa lỗi

Sau khi xây dựng code, ta nạp code vào MSP430 rồi chạy thử. Tuy nhiên, nhóm sinh viên vẫn chưa tìm ra được nguyên nhân khiến màn hình hiển thị không đúng với mong muốn, nên project vẫn chưa được hoàn thiện.

2.6. Hoàn thiện mạch, xây dựng báo cáo

Sau khi hoàn tất các công đoạn trên, nhóm sinh viên tham gia vào xây dựng báo cáo.

KẾT LUẬN

Sau quá trình tìm hiểu, và triển khai thực hiện, nhóm sinh viên đã thiết kế được đồng hồ số sử dụng vi điều khiển MSP430 với các chức năng mà nhóm xác định. Tuy nhiên, do kiến thức còn hạn chế, nên sản phẩm làm ra còn chưa được hoàn thiện, chạy đúng với yêu cầu đặt ra.

Trong quá trình thực hiện đề tài, nhóm sinh viên xin chân thành cảm ơn thầy Vương Hoàng Nam và thầy Đặng Khánh Hòa đã giúp đỡ nhóm hoàn thiện được đề tài. Chúng em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

[1] <http://www.ti.com/product/msp430g2553>, **Datasheet MSP430G2553**

[2] <http://www.hitechlab.vn>

[3] Huỳnh Ngọc Phụng, **Cơ sở lý thuyết về MSP430.**