# Mục tiêu:

* Giao tiếp được với LED 7 đoạn
* Giao tiếp được với LED ma trận

# Tham khảo:

* Tài liệu hướng dẫn thí nghiệm, chương 4
* Atmel-2505-Setup-and-Use-of-AVR-Timers\_ApplicationNote\_AVR130.pdf

# Bài 2

1. Kết nối 1 port của AVR vào header J34. Kết nối 2 chân port khác vào tín hiệu nLE0 và nLE1 trên header J82. Set jumper để cấp nguồn cho LED 7 đoạn
2. Sử dụng các chương trình mẫu trong tài liệu hướng dẫn thí nghiệm, viết chương trình hiển thị số 0123 lên 4 LED 7 đoạn, sử dụng timer 0 để quét LED với tần số quét 50Hz.

# Bài 3

1. Kết nối port của AVR vào dip Switch, giả sử đó là PORTA
2. Viết chương trình hiện giá trị PORTA \* 9 lên 4 LED 7 đoạn.
3. Thay đổi giá trị dip switch và quan sát kết quả

# Bài 4

1. Kết nối các tín hiệu cần thiết để điều khiển LED ma trận. .
2. Sử dụng chương trình mẫu, chỉnh sửa nếu cần thiết để hiển thị chữ ‘A’ lên LED ma trận. Quét LED ma trận sử dụng timer để tạo delay với tần số quét 25 Hz.
3. Chỉnh sửa chương trình để đạt tần số quét là 125Hz.

Nhóm L09

Tổ 3

Trịnh Quang Huy 2110211

Nguyễn Ngọc Khanh 2111474

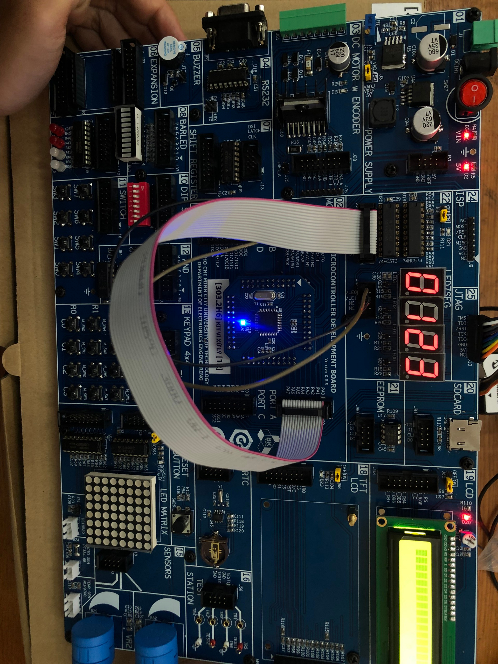
Trương Quốc Khánh 2111501

# Bài 2

1. Trả lời các câu hỏi
   1. Để có tần số quét là 50Hz, một LED sẽ sáng 1 lần trong bao lâu?

Một LED sẽ sáng 1 lần trong [1/(50Hz) ] = 0,02s để có tần số quét là 50Hz

* 1. Cấu hình timer như thế nào để có độ trễ này
     + Chọn timer0 mode Normal, hệ số chia N = 1, ta nạp giá trị ban đầu cho bộ đếm là -160 (tức là mỗi lần cờ TOV0 tràn khi bộ đếm đếm được 161 xung), ta sẽ có chương trình delay 0,02s.
     + Trong chương trình, nhóm em sử dụng delay 5ms để quét LED ổn định hơn nên chọn hệ số chia N = 8, giá trị ban đầu là -250, viết vòng lặp cho TOV0 tràn 20 lần.

1. Mã nguồn và chú thích

|  |
| --- |
| .DEF COUNT=R20  .EQU TP=-250 ;GIA TRI DAT TRUOC TIMER0  .EQU NUM=20 ;SO LAN CO TOV0 TRAN  .EQU OUTPORT=PORTA ;PA LA OUTPUT DATA  .EQU OUTPORT\_DDR=DDRA  .EQU CONT\_LED=PORTB ;PB LA TIN HIEU DIEU KHIEN LED  .EQU CONT\_LED\_DDR=DDRB  .EQU CONT\_ONLED=1 ;VI TRI BIT ON LED  .EQU LED\_DATA=0 ;VI TRI BIT ENABLE DATA  MAIN:  LDI R16,HIGH(RAMEND)  OUT SPH,R16  LDI R16,LOW(RAMEND)  OUT SPL,R16  LDI COUNT,NUM ;NAP GIA TRI DEM SO LAN CO TOV0 TRAN  LDI R17,0  OUT TCCR0A,R17 ;CHON MODE NOR  LDI R17,0  OUT TCCR0B,R17  LDI R17,$FF  OUT OUTPORT\_DDR,R17  OUT CONT\_LED\_DDR,R17    CBI CONT\_LED,LED\_DATA ;NGAT DUONG DAN DATA  CBI CONT\_LED,CONT\_ONLED ;NGAT KET NOI LED  START:  RCALL SCAN\_4LA  RJMP START  ;.............................  SCAN\_4LA:  LDI R18,4 ;R18=SO LAN QUET LED  LDI R19,$FE ;MA QUET LED ANODE CHUNG  LDI R21,3 ;GIA TRI BAT DAU  LOOP:  ;TAT HET LED  LDI R17,$FF  OUT OUTPORT,R17  SBI CONT\_LED,CONT\_ONLED  CBI CONT\_LED,CONT\_ONLED  MOV R17,R21 ;R21 LA SO HIEN THI TREN LED  RCALL BCD\_LED7A  OUT OUTPORT,R17  SBI CONT\_LED,LED\_DATA ;DUA MA LED HIEN THI RA NGOAI  CBI CONT\_LED,LED\_DATA  OUT OUTPORT,R19 ;CHI DINH VI TRI LED HIEN THI  SBI CONT\_LED,CONT\_ONLED  CBI CONT\_LED,CONT\_ONLED  RCALL DELAY\_5MS  ;QUET LED KE TIEP  LSL R19  INC R19  DEC R21 ;TANG SO HIEN THI CHO LED KE TIEP  DEC R18 ;GIAM SO LAN QUET LED  BRNE LOOP  RET  ;...............................  DELAY\_5MS:  LDI R17,TP  OUT TCNT0,R17 ;NAP GIA TRI BAT DAU DEM  LDI R17,$02  OUT TCCR0B,R17 ;TIMER0 CHAY, HE SO CHIA N=8  WAIT:  IN R17,TIFR0  SBRS R17,TOV0  RJMP WAIT  OUT TIFR0,R17 ;NAP LAI BIEN COUNT, XOA CO TOV0  LDI R17,0  OUT TCCR0B,R17 ;DUNG TIMER  DEC COUNT ;DEM SO LAN TRAN CO TOV0  BRNE DELAY\_5MS  LDI COUNT,NUM ;NAP LAI SO DEM  RET  ;................................  TABLE: .DB $C0,$F9,$A4,$B0,$99,$92  .DB $82,$F8,$80,$90,$88,$83  .DB $C6,$A1,$86,$8E  ;R17 LA OFFSET  ;KET QUA LUU LAI R17  BCD\_LED7A:  LDI ZH,HIGH(TABLE<<1)  LDI ZL,LOW(TABLE<<1)  ADD ZL,R17  CLR R16  ADC ZH,R16  LPM R17,Z  RET |

# Bài 3

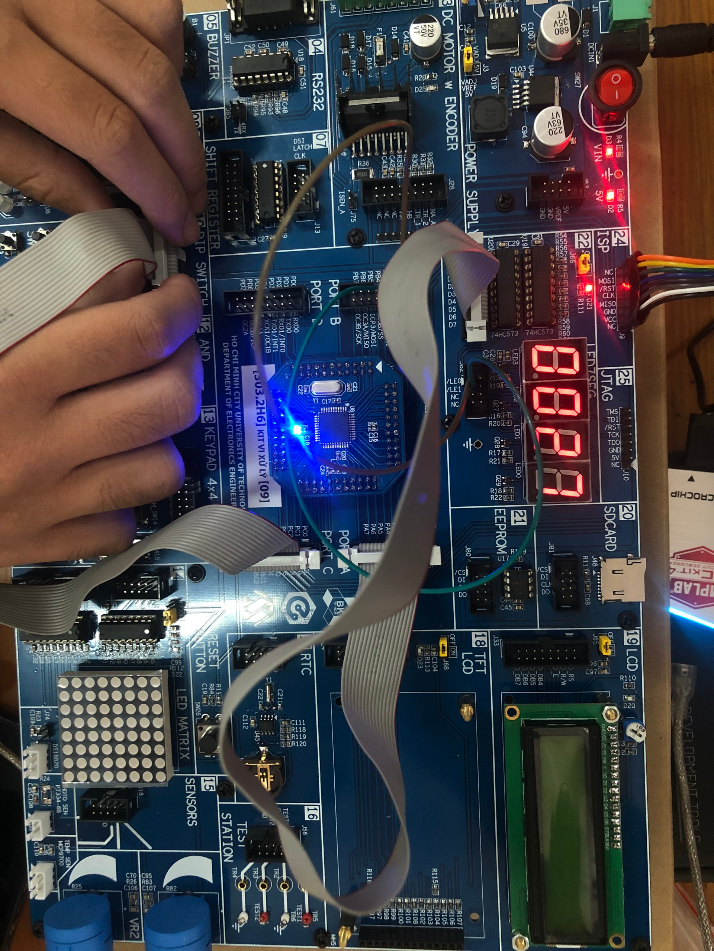
1. Trả lời các câu hỏi
   1. Giá trị PORTA \* 9 là số có bao nhiêu bit

PORTA có 8 bit, vậy nên PORTA \* 9 là số có 12 bit

* 1. Làm thế nào để hiển thị từng chữ số lên 4 LED?

Ta viết chương trình con chia cho 10 bằng thuật toán trừ số đó liên tục cho 10, để thu được:

* + - Lần thứ nhất gọi chương trình con chia cho 10, ta thu được số dư là BCD hàng đơn vị, phần thương là 3 chữ số hàng nghìn, hàng trăm và hàng chục.
    - Lần thứ hai gọi chương trình con chia cho 10, ta thu được số dư là BCD hàng chục, phần thương là 2 chữ số hàng nghìn và hàng trăm
    - Lần thứ ba gọi chương trình con chia cho 10, ta thu được số dư là BCD hàng trăm, phần thương là 1 chữ số hàng nghìn
    - Vậy, ta đã tách được các số ra, tiếp theo ta gọi bảng để truy mã 7 đoạn của từng chữ số, các phép toán đều được thực hiện trên 1 WORD nào đó vì PORTA \* 9 là số 12 bit.



1. Mã nguồn và chú thích

|  |
| --- |
| .DEF COUNT=R20  .EQU TP=-250 ;GIA TRI DAT TRUOC TIMER0  .EQU NUM=20 ;SO LAN CO TOV0 TRAN  .EQU INPORT=PINC ;PD LA DATA VAO  .EQU INPORT\_DDR=DDRC  .EQU OUTPORT=PORTA ;PA LA OUTPUT DATA  .EQU OUTPORT\_DDR=DDRA  .EQU CONT\_LED=PORTB ;PB LA TIN HIEU DIEU KHIEN LED  .EQU CONT\_LED\_DDR=DDRB  .EQU CONT\_ONLED=1 ;VI TRI BIT ON LED  .EQU LED\_DATA=0 ;VI TRI BIT ENABLE DATA  MAIN:  LDI R16,HIGH(RAMEND)  OUT SPH,R16  LDI R16,LOW(RAMEND)  OUT SPL,R16  LDI COUNT,NUM ;NAP GIA TRI DEM SO LAN CO TOV0 TRAN  LDI R17,0  OUT TCCR0A,R17 ;CHON MODE NOR  LDI R17,0  OUT TCCR0B,R17  LDI R17,0  OUT INPORT\_DDR,R17  LDI R17,$FF  OUT OUTPORT\_DDR,R17  OUT CONT\_LED\_DDR,R17    CBI CONT\_LED,LED\_DATA ;NGAT DUONG DAN DATA  CBI CONT\_LED,CONT\_ONLED ;NGAT KET NOI LED  START:    RCALL SCAN\_4LA  RJMP START  ;.............................  SCAN\_4LA:  LDI R25,4 ;R18=SO LAN QUET LED  LDI R19,$FE ;MA QUET LED ANODE CHUNG    LDI XL,LOW($0100)  LDI XH,HIGH($0100)  IN R17,INPORT  LDI R16,9 ;R17=R17\*9  MUL R17,R16  MOV R22,R0 ;R22 LA BYTE THAP  MOV R23,R1 ;R23 LA BYTE CAO  RCALL SPLIT\_16BIN\_BCD  ST X+,R18 ;CAT HANG DON VI  RCALL SPLIT\_16BIN\_BCD  ST X+,R18 ;CAT HANG CHUC  RCALL SPLIT\_16BIN\_BCD  ST X+,R18 ;CAT HANG TRAM  ST X+,R22 ;CAT HANG NGHIN  LDI XL,LOW($0100)  LDI XH,HIGH($0100)  LOOP:  ;TAT HET LED  LDI R17,$FF  OUT OUTPORT,R17  SBI CONT\_LED,CONT\_ONLED  CBI CONT\_LED,CONT\_ONLED  LD R21,X+ ;LAY LAI CAC SO HIEN THI (DONVI-CHUC-TRAM-NGHIN)  MOV R17,R21 ;R21 LA SO HIEN THI TREN LED  RCALL BCD\_LED7A  OUT OUTPORT,R17  SBI CONT\_LED,LED\_DATA ;DUA MA LED HIEN THI RA NGOAI  CBI CONT\_LED,LED\_DATA  OUT OUTPORT,R19 ;CHI DINH VI TRI LED HIEN THI  SBI CONT\_LED,CONT\_ONLED  CBI CONT\_LED,CONT\_ONLED  RCALL DELAY\_5MS  ;QUET LED KE TIEP  LSL R19  INC R19  DEC R25 ;GIAM SO LAN QUET LED  BRNE LOOP  RET  ;...............................  DELAY\_5MS:  LDI R17,TP  OUT TCNT0,R17 ;NAP GIA TRI BAT DAU DEM  LDI R17,$02  OUT TCCR0B,R17 ;TIMER0 CHAY, HE SO CHIA N=8  WAIT:  IN R17,TIFR0  SBRS R17,TOV0  RJMP WAIT  OUT TIFR0,R17 ;NAP LAI BIEN COUNT, XOA CO TOV0  LDI R17,0  OUT TCCR0B,R17 ;DUNG TIMER  DEC COUNT ;DEM SO LAN TRAN CO TOV0  BRNE DELAY\_5MS  LDI COUNT,NUM ;NAP LAI SO DEM  RET  ;................................  TABLE: .DB $C0,$F9,$A4,$B0,$99,$92  .DB $82,$F8,$80,$90,$88,$83  .DB $C6,$A1,$86,$8E  ;R17 LA OFFSET  ;KET QUA LUU LAI R17  BCD\_LED7A:  LDI ZH,HIGH(TABLE<<1)  LDI ZL,LOW(TABLE<<1)  ADD ZL,R17  CLR R16  ADC ZH,R16  LPM R17,Z  RET  ;.............................  ;CTC PHEP CHIA R17:R16  ;TRA VE R22=KQ (->BYTE THAP) ;R18= SO DU (<-BYTE THAP)  SPLIT\_16BIN\_BCD:  CLR R24 ;BIEN DEM  LOOP1:  INC R24  SUBI R22,10 ;R22 LA BYTE THAP  BRCC LOOP1  SUBI R23,1 ;SBCI R23,0// R23 LA BYTE CAO  BRCC LOOP1    LDI R16,10  CLR R17  ADD R22,R16 ;R22 LA BYTE THAP CHUA SO DU  ADC R23,R17  DEC R24 ;KET QUA  PUSH R22  PUSH R24  POP R22  POP R18  RET |

# Bài 4

1. Trả lời các câu hỏi
2. Mô tả kết nối trên kit

Dùng PORTD để xuất ra LED ma trận

PB3 là chân set clear signal

PB2 là chân set shift clock

PB1 là chân set latch pin

PB0 là chân set shift data pin

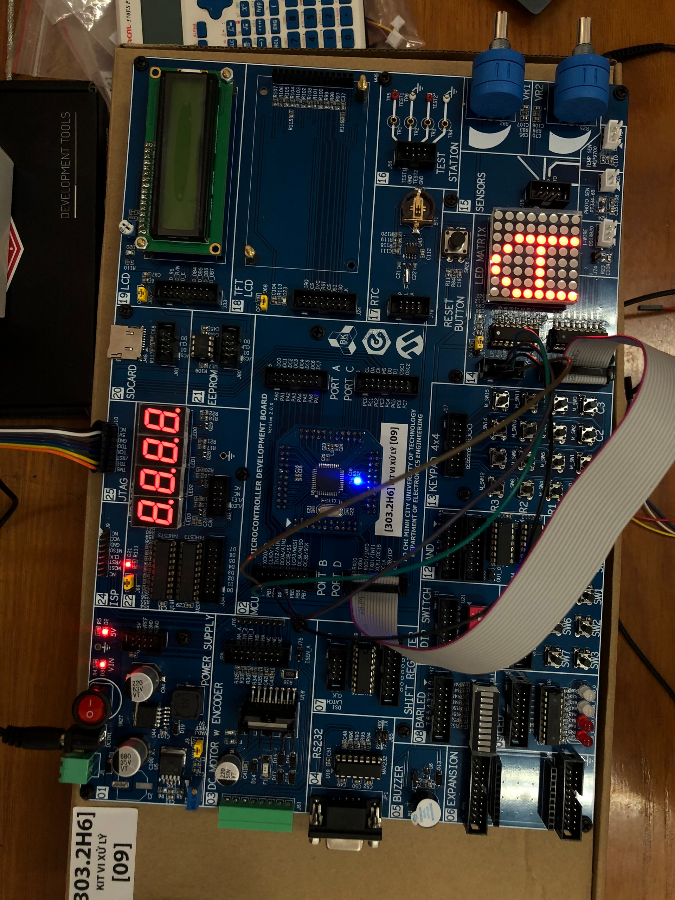
1. Để có tần số quét 25Hz thì một cột LED sáng trong bao lâu?

Để có tần số quét 25Hz thì một cột LED sáng trong [1/(25Hz)] = 0,04s

1. Sự khác nhau khi quét ở tần số 25Hz và 125Hz

Ở tần số quét 25Hz thì một cột LED sáng trong [1/(25Hz)] = 0,04s

Ở tần số quét 125Hz thì một cột LED sáng trong [1/(125Hz)] = 8ms

Như vậy, ở 125Hz LED sẽ được quét nhanh hơn, ít bị chập chờn vì tần số cao. ở 25Hz mắt người sẽ nhìn thấy LED chớp tắt mà không phải cảm giác sáng liên tục như ở 125Hz

1. Mã nguồn chương trình với chú thích

|  |
| --- |
| .ORG 0X0000 ; INTERRUPT VECTOR TABLE  RJMP RESET\_HANDLER ; RESET  .ORG 0X001A  RJMP TIMER1\_COMP\_ISR  RESET\_HANDLER:  ; INITIALIZE STACK POINTER  LDI R16, HIGH(RAMEND)  OUT SPH, R16  LDI R16, LOW(RAMEND)  OUT SPL, R16  CALL SHIFTREGISTER\_INITPORT  CALL SHIFTREGISTER\_CLEARDATA  CALL INITTIMER1CTC  ; ENABLE GLOBAL INTERRUPTS  SEI  CALL LEDMATRIX\_PORTINIT  MAIN:  JMP MAIN  .EQU CLEARSIGNALPORT = PORTB ; SET CLEAR SIGNAL PORT TO PORTB  .EQU CLEARSIGNALPIN = 3 ; SET CLEAR SIGNAL PIN TO PIN 3 OF PORTB  .EQU SHIFTCLOCKPORT = PORTB ; SET SHIFT CLOCK PORT TO PORTB  .EQU SHIFTCLOCKPIN = 2 ; SET SHIFT CLOCK PIN TO PIN 2 OF PORTB  .EQU LATCHPORT = PORTB ; SET LATCH PORT TO PORTB  .EQU LATCHPIN = 1 ; SET LATCH PIN TO PIN 1 OF PORTB  .EQU SHIFTDATAPORT = PORTB ; SET SHIFT DATA PORT TO PORTB  .EQU SHIFTDATAPIN = 0 ; SET SHIFT DATA PIN TO PIN 0 OF PORTB  ; INITIALIZE PORTS AS OUTPUTS  SHIFTREGISTER\_INITPORT:  PUSH R24  LDI R24, (1<<CLEARSIGNALPIN)|(1<<SHIFTCLOCKPIN)|(1<<LATCHPIN)|(1<<SHIFTDATAPIN);  OUT DDRB, R24 ; SET DDRB TO OUTPUT  POP R24  RET  SHIFTREGISTER\_CLEARDATA:  CBI CLEARSIGNALPORT, CLEARSIGNALPIN ; SET CLEAR SIGNAL PIN TO LOW  ; WAIT FOR A SHORT TIME  SBI CLEARSIGNALPORT, CLEARSIGNALPIN ; SET CLEAR SIGNAL PIN TO HIGH  RET  ; SHIFT OUT DATA  ;SHIFT OUT R27 TO BAR LED  SHIFTREGISTER\_SHIFTOUTDATA:  PUSH R18  CBI SHIFTCLOCKPORT, SHIFTCLOCKPIN ;  LDI R18, 8 ; SHIFT 8 BITS  SHIFTLOOP:  SBRC R27, 7 ; CHECK IF THE MSB OF SHIFTDATA IS 1  SBI SHIFTDATAPORT, SHIFTDATAPIN ; SET SHIFT DATA PIN TO HIGH  SBI SHIFTCLOCKPORT, SHIFTCLOCKPIN ; SET SHIFT CLOCK PIN TO HIGH  LSL R27 ; SHIFT LEFT  CBI SHIFTCLOCKPORT, SHIFTCLOCKPIN ; SET SHIFT CLOCK PIN TO LOW  CBI SHIFTDATAPORT, SHIFTDATAPIN ; SET SHIFT DATA PIN TO LOW  DEC R18  BRNE SHIFTLOOP  ; LATCH DATA  SBI LATCHPORT, LATCHPIN ; SET LATCH PIN TO HIGH  CBI LATCHPORT, LATCHPIN ; SET LATCH PIN TO LOW  POP R18  RET  ;LOOKUP TABLE FOR COLLUMN CONTROL  LEDMATRIX\_COL\_CONTROL: .DB 0X80, 0X40, 0X20, 0X10, 0X08, 0X04, 0X02, 0X01  ; LOOKUP TABLE FOR FONT  ;LEDMATRIX\_FONT\_A:  ;.DB $FF,$FF,$FF,$FF,$FF,$FF,$FF,$FF  LEDMATRIX\_FONT\_A:  .DB 0B11111100, 0B00010010, 0B00010001, 0B00010001, 0B00010010, 0B11111100, 0B00000000, 0B00000000  ; J38 CONNECT TO PORTD  ; CLEAR SIGNAL PIN TO PIN 0 OF PORTB  ; SHIFT CLOCK PIN TO PIN 1 OF PORTB  ; LATCH PIN TO PIN 0 OF PORTB  ; SHIFT DATA PIN TO PIN 3 OF PORTB  ; OUTPUT: NONE  .EQU LEDMATRIXPORT = PORTD  .EQU LEDMATRIXDIR = DDRD  .DSEG  .ORG SRAM\_START ;STARTING ADDRESS IS 0X100  LEDMATRIXBUFFER : .BYTE 8  LEDMATRIXCOLINDEX : .BYTE 1  .CSEG  .ALIGN 2  LEDMATRIX\_PORTINIT:  PUSH R20  PUSH R21  LDI R20, 0B11111111 ; SET PORT AS OUTPUT  OUT LEDMATRIXDIR, R20  LDI R20,0 ;COL INDEX START AT 0  LDI R31,HIGH(LEDMATRIXCOLINDEX)  LDI R30,LOW(LEDMATRIXCOLINDEX)  ST Z,R20  LDI R20,0  LDI R31,HIGH(LEDMATRIX\_FONT\_A << 1) ;Z REGISTER POINT TO FONTA VALUE  LDI R30,LOW(LEDMATRIX\_FONT\_A << 1)  LDI R29,HIGH(LEDMATRIXBUFFER) ; Y REGISTER POINT TO FONTA VALUE  LDI R28,LOW(LEDMATRIXBUFFER)  LDI R20,8  LEDMATRIX\_PORTINIT\_LOOP: ;COPY FONT TO DISPLAY BUFFER  LPM R21,Z+  ST Y+,R21  DEC R20  CPI R20,0  BRNE LEDMATRIX\_PORTINIT\_LOOP  POP R21  POP R20  RET  ; DISPLAY A COLLUMN OF LED MATRIX  ; INPUT: R27 CONTAINS THE VALUE TO DISPLAY  ; R26 CONTAIN THE COL INDEX (3..0)  ; OUTPUT: NONE  LEDMATRIX\_DISPLAY\_COL:  PUSH R16 ; SAVE THE TEMPORARY REGISTER  PUSH R27  CLR R16  OUT LEDMATRIXPORT,R16  CALL SHIFTREGISTER\_SHIFTOUTDATA  LDI R31,HIGH(LEDMATRIX\_COL\_CONTROL << 1)  LDI R30,LOW(LEDMATRIX\_COL\_CONTROL << 1)  CLR R16  ADD R30,R26  ADC R31,R16  LPM R27,Z  OUT LEDMATRIXPORT,R27  POP R27  POP R16 ; RESTORE THE TEMPORARY REGISTER  RET ; RETURN FROM THE FUNCTION  INITTIMER1CTC:  PUSH R16  LDI R16, HIGH(2500) ; LOAD THE HIGH BYTE INTO THE TEMPORARY REGISTER  STS OCR1AH, R16 ; SET THE HIGH BYTE OF THE TIMER 1 COMPARE VALUE  LDI R16, LOW(2500) ; LOAD THE LOW BYTE INTO THE TEMPORARY REGISTER  STS OCR1AL, R16 ; SET THE LOW BYTE OF THE TIMER 1 COMPARE VALUE  LDI R16, (1 << CS10)| (1<< WGM12) ; LOAD THE VALUE 0B00000101 INTO THE TEMPORARY REGISTER  STS TCCR1B, R16 ;  LDI R16, (1 << OCIE1A); LOAD THE VALUE 0B00000010 INTO THE TEMPORARY REGISTER  STS TIMSK1, R16 ; ENABLE THE TIMER 1 COMPARE A INTERRUPT  POP R16  RET  TIMER1\_COMP\_ISR:  PUSH R16  PUSH R26  PUSH R27  LDI R31,HIGH(LEDMATRIXCOLINDEX)  LDI R30,LOW(LEDMATRIXCOLINDEX)  LD R16,Z  MOV R26,R16  LDI R31,HIGH(LEDMATRIXBUFFER)  LDI R30,LOW(LEDMATRIXBUFFER)  ADD R30,R16  CLR R16  ADC R31,R16  LD R27,Z  CALL LEDMATRIX\_DISPLAY\_COL  INC R26  CPI R26,8  BRNE TIMER1\_COMP\_ISR\_CONT  LDI R26,0 ;IF R26 = 8, RESET TO 0  TIMER1\_COMP\_ISR\_CONT:  LDI R31,HIGH(LEDMATRIXCOLINDEX)  LDI R30,LOW(LEDMATRIXCOLINDEX)  ST Z,R26  POP R27  POP R26  POP R16  RETI |