# OBJECTIVE:

* Understand and utilize external interrupts, timer interrupts.
* Scan 7-segment LED displays and LED matrices using timer interrupts.
* Understand how to control and measure motor speed

# References:

* Experiment guide, chapters 3, 4, 5, 7,12.
* Atmel-2505-Setup-and-Use-of-AVR-Timers\_ApplicationNote\_AVR130.pdf.

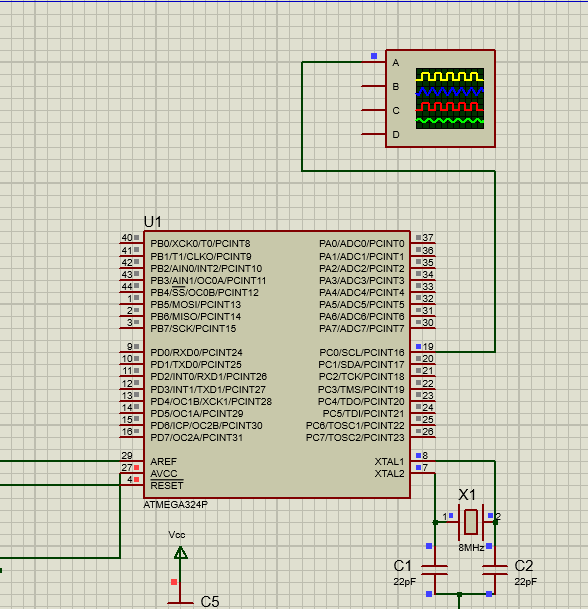
# EXPERIMENT 1:

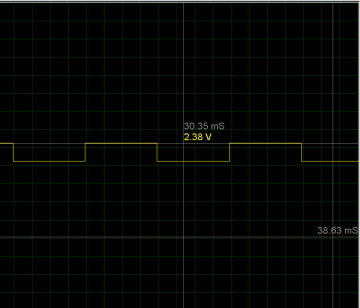
1. Programming to generate a 1 kHz frequency signal on pin PC0 using Timer 1 overflow interrupt. When Timer 1 overflows, the interrupt routine will toggle the PC0 pin and reset the counter register value.

| .EQU P\_OUT = 0  .ORG 0  RJMP MAIN  .ORG 0X001E  RJMP TIMER1\_OVF ; INTERRUPTS TIMER1 OVERFLOW  .ORG 0X40 ;  MAIN:  LDI R16, HIGH(RAMEND)  OUT SPH, R16  LDI R16, LOW(RAMEND)  OUT SPL, R16 ; SET UP THE SP  (STACK POINTER)  LDI R16,(1<<P\_OUT)  OUT DDRC, R16 ; PC0 = OUTPUT  LDI R17, HIGH(61536) ; TP\_H = 61536  STS TCNT1H, R17  LDI R17, LOW(61536) ; TP\_L = 61536  STS TCNT1L, R17  LDI R17, 0X00  STS TCCR1A, R17 ; TIMER 1 MODE NOR  LDI R17, 0X01  STS TCCR1B, R17 ; MODE NOR, KHONG CHIA,  START  SEI ; CHO PHEP NGAT TOAN CUC  LDI R17, (1 << TOIE1) ; CHO PHEP NGAT KHI TIMER1 TRAN  STS TIMSK1, R17  START: RJMP START ;----------------------------------------------------------------------- TIMER1\_OVF:  LDI R17, 0X00  STS TCCR1B, R17 ; STOP TIMER1  LDI R17, HIGH(61536)  STS TCNT1H, R17  LDI R17, LOW(61536) ; SET TCNT1  STS TCNT1L, R17  IN R17, PORTC ; READ PORTC  EOR R17, R16 ;  OUT PORTC, R17  LDI R17, 0X01  STS TCCR1B, R17 ; MODE NOR KHONG CHIA START  RETI |
| --- |

1. Connect PC0 to the oscilloscope to measure the waveform.

(Note: The clock frequency for the CPU on the experimental kit is 8 MHz.)





# EXPERIMENT 2:

1. Repeat exercise 1 using Timer 1 in CTC mode, utilizing the COMPARE\_MATCH interrupt, to generate a pulse with a frequency of 100 Hz on pin PC0.

| .EQU P\_OUT = 0  .ORG 0 RJMP MAIN  .ORG 0X001A  RJMP COMPARE\_MATCH ; INTERRUPTS COMPARE CHANNEL A  .ORG 0X40  MAIN:  LDI R16, HIGH(RAMEND)  OUT SPH, R16  LDI R16, LOW(RAMEND)  OUT SPL, R16 ; SET UP STACK POINTER (SP)  LDI R16, (1<<P\_OUT)  OUT DDRC, R16 ; PC0 = OUTPUT  LDI R17, HIGH(39999) ; TP\_H = 39999 STS OCR1AH, R17  LDI R17, LOW(39999) ; TP\_L = 39999 STS OCR1AL, R17  LDI R17, 0X00  STS TCCR1A, R17 ; TIMER 1 LDI R17, 0X09  STS TCCR1B, R17 ;  LDI R17, (1 << OCIE1A)  START: rjmp START  COMPARE\_MATCH:  LDI R17, 0X00  STS TCCR1B, R17 ; STOP TIMER1  LDI R17, HIGH(39999) ; TP\_H = 39999 STS OCR1AH, R17  LDI R17, LOW(39999) ; TP\_L = 39999 STS OCR1AL, R17  IN R17, PORTC ; DOC PORTC  EOR R17, R16 ; DAO BIT CHAN PC0 VOI R16 = 0X01 OUT PORTC, R17  LDI R17, 0X09  STS TCCR1B, R17 ; MODE CTC KHONG CHIA START RETI |
| --- |

1. Configure the timer to generate a COMPARE\_MATCH interrupt every 1 ms. Inside the interrupt, use a counter to count the number of interrupt occurrences and control pin PC0 to generate a pulse with a frequency of 100 Hz.

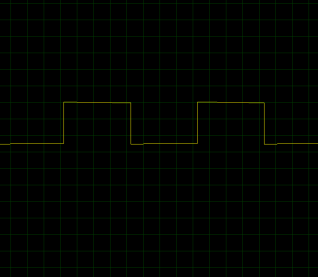
INC COUNT

CPI COUNT,5

BRNE RUN

Instructions: Increment the counter by 1 each time the interrupt occurs. If the counter reaches 5, toggle PC0 and reset the counter to 0.

1. Compile the program and observe the oscilloscope to verify the functionality of the program.



# EXPERIMENT 3:

1. Connect the necessary signals to control the 7-segment LED display module.
2. Utilize the COMPARE\_MATCH interrupt of Timer 1, as in Exercise 2, to display the numbers 1-2-3-4 on four 7-segment LED displays with a scanning frequency of 50 Hz. To measure the scanning frequency, toggle pin PC0 each time it switches to the next LED and measure this pulse on the oscilloscope.

(Refer to Chapter 4 of the experiment guide for further details.)

# EXPERIMENT 4:

Requirements:

1. Write a program to control the speed of a DC motor using PWM with a frequency of 1 kHz, using Timer 0. Control the speed increase/decrease using two buttons, where each button press increases/decreases the duty cycle by 5%. Allow the motor to start/stop and control the motor direction (forward/reverse) using two switches on a dip switch.

* Connect the motor to the kit.
* Connect the signals from the two switches on the dip switch to two AVR ports.
* Connect the signals from the two buttons to two AVR ports.
* Connect the signal from pin OC0B to a test point channel for measurement.
* Connect the signals from two port pins to control the forward/reverse direction to a single LED for status checking.

| .DEF FLAG\_REG = R18  .EQU SW\_FLG = 0  .EQU CONT\_IN = PINA  .EQU CONT\_ROTDC = PORTB  ; CONTROL THE DIRECTION ROTATE OF MORTOR  .EQU CONT\_ROTDC\_DDR = DDRB  .EQU MOTOR\_CTRL1 = 0  .EQU MOTOR\_CTRL2 = 1  .EQU MOTOR\_ENABLE = 6 ; 0C0B = PB4  .EQU SW1 = 0 ; BUTTON UP SPEED  .EQU SW2 = 1 ; BUTTON DOWN SPEED  .EQU SW3 = 2 ; SWITCH SET UP CHIEU QUAY DONG CO  .EQU SW4 = 3 ; SWITCH DUNG DONG CO  .EQU PMIN = 5 ; DUTY CYCLE = 5% IS MIN  .EQU PMAX = 118 ; DUTY CYCLE = 95% IS MAX  .EQU DELTA = 6 ; CHANGE (FAST/SLOW) 5% DUTY CYCLE  .ORG 0  RJMP MAIN  .ORG 0X40  MAIN:  LDI R16, HIGH(RAMEND)  OUT SPH, R16  LDI R16, LOW(RAMEND)  OUT SPL, R16 ; SET UP STACK POINTER (SP)  LDI R16, (1 << MOTOR\_ENABLE)  OUT DDRD, R16; OUTPUT CONTROL THE SPEED OF MOTOR  LDI R16, (1 << MOTOR\_CTRL1)|(1 << MOTOR\_CTRL2)  OUT CONT\_ROTDC\_DDR, R16 ; OUTPUT CONTROL THE DIRECT ROTATION OF MOTOR  LDI R16, 0 ; PORTA = INPUT CONTROL UP/DOWN SPEED AND ROTATION OF MORTOR DC  OUT DDRA, R16  LDI R16, 0X0F ; PULL UP RESISTOR  OUT PORTA, R16  START0:  LDI R16, 124 ; OCR0A : Fo = 1KHz  STS OCR2A, R16  LDI R16, PMIN ; OCR2B : DUTY CYCLE 5%  STS OCR2B, R16  LDI R16, 0B00100011 ; TIMER 2 MODE FPWM7,  STS TCCR2A, R16  LDI R16, 0B00001100 ; TIMER 2 MODE FPWM7, PRESCALER (N) = 64  STS TCCR2B, R16  LDI R19, DELTA ; DELTA IS THE NUMBER OF INCREASING/DECREASING DUTY CYCLE DUTY CYCLE  START:  IN R17, CONT\_IN  ANDI R17, (1 << SW3)|(1 << SW4) ; BIT SW3, SW4 OFF  SBRC R17, SW4  RJMP STOP\_MOTOR  SBRC R17, SW3  RJMP REVERSE  ;MOTOR ROTATES IN CLOW-WISE DIRECTION  SBI CONT\_ROTDC, MOTOR\_CTRL1  CBI CONT\_ROTDC, MOTOR\_CTRL2  RJMP CONTINUE  REVERSE:  CBI CONT\_ROTDC, MOTOR\_CTRL1  SBI CONT\_ROTDC, MOTOR\_CTRL2  RJMP CONTINUE  STOP\_MOTOR:  LDI R16, 0B00100011 ;TIMER0 MODE FPWM7  STS TCCR2A, R16  LDI R16, 0B00001000 ;TIMER0 MODE FPWM7  STS TCCR2B, R16  SBI PORTD, MOTOR\_ENABLE  CBI CONT\_ROTDC, MOTOR\_CTRL1  CBI CONT\_ROTDC, MOTOR\_CTRL2  RJMP START0  CONTINUE:  RCALL GET\_SW ;READ SW  SBRS FLAG\_REG, SW\_FLG ; SW = 1 -> PRESSED  RJMP START  CPI R17, 1 ; SW1 = UP PRESSED  BRNE SW2\_CHK ; CHECK SW2  LDS R17, OCR2B ; READ OCR2B  ADD R17, R19 ; R17 = R17 + DELTA  CPI R17, PMAX  BRCS UP\_SP  LDI R17, PMAX  UP\_SP:  STS OCR2B, R17 ;CAP NHAT OCR2B  RJMP START  SW2\_CHK:  CPI R17, 2 ; SW2 = DOWN PRESSED  BRNE START  LDS R17, OCR2B ;DOC OCR2B  SUB R17, R19 ; R17 = R17-DELTA  CPI R17, PMIN ; WIDTH OF LOW PULSE  BRCC DWN\_SP ; NOT YET -> UPDATE NEW VALUE  LDI R17, PMIN ; THE LIMIT = PMIN  DWN\_SP:  STS OCR2B, R17 ; UPDATE OCR2B  RJMP START  GET\_SW:  CBR FLAG\_REG, (1 << SW\_FLG) ; DELETE THE FLAG\_REG  BACK0:  LDI R16, 50  WAIT0:  IN R17, CONT\_IN  ANDI R17, (1 << SW1)|(1 << SW2) ; BIT SW1,SW2 OFF  CPI R17, (1 << SW1)|(1 << SW2)  BREQ EXIT\_SW ; RELEASE -> CANCLE  DEC R16 ; PRESSED  BRNE WAIT0  PUSH R17 ; SAVE CODE SW  BACK1:  LDI R16, 50  WAIT1:  IN R17, CONT\_IN  ANDI R17, (1 << SW1)|(1 << SW2) ; BIT SW1, SW2 OFF  CPI R17, (1 << SW1)|(1 << SW2)  BRNE BACK1 ;RELEASE SW  DEC R16  BRNE WAIT1  POP R17 ; RECOVER MA SW  CPI R17, (1 << SW2) ; SW1 = 0 PRESSED, SW2 = 1 RELEASED  BRNE SW2\_CODE  LDI R17, 1  RJMP SET\_FLG  SW2\_CODE:  CPI R17, (1 << SW1) ; SW2 = 0 PRESSED, SW1 = 1 RELEASE  BRNE EXIT\_SW ; NOT YET -> CANCLE  LDI R17, 2  SET\_FLG:  SBR FLAG\_REG,(1 << SW\_FLG) ; SET UP FLAG PRESS SW  EXIT\_SW:  RET |
| --- |

1. Compile, execute, and test the program by measuring the waveforms on an oscilloscope and observing the LED status when changing the dip switch and pressing the speed increase/decrease buttons.
2. Connect the PWM signal to MOTOR\_ENABLE and the control signals for the motor direction to MOTOR\_CTRL1 and MOTOR\_CTRL2 on J76 of the DC\_MOTOR module.
3. Test the operation of the system.
4. Measure the waveforms from the two A-B signals of the encoder and compare them in the two cases of the motor rotating forward or backward.

# EXPERIMENT 1:

1. Answer the questions:  
   a. In Normal mode, do we need to reset the count register when entering the Overflow interrupt?  
   b. Explain the values written to the timer configuration registers and prescaler.
2. Program source code with comments

| ........................................................................................................................ |
| --- |

# EXPERIMENT 2:

1. Trả Answer the questions:

a. In CTC mode, do we need to reset the count register when entering the COMPARE\_MATCH interrupt?

b. What are the advantages of this mode compared to the configuration in Exercise 1?  
c. Explain the values written to the timer configuration registers and prescaler.

1. Program source code with comments

| ........................................................................................................................ |
| --- |

# EXPERIMENT 3:

1. Answer the questions:  
   a. To achieve a scanning frequency of 50Hz, how long will one LED remain lit?  
   b. In that case, what will be the frequency of pin PC0 (toggled each time the LED switches)?  
   c. How many interrupt occurrences are required to switch the LED?
2. Program source code with comments

| ........................................................................................................................ |
| --- |

# EXPERIMENT 4:

1. Answer the questions:  
   a. Describe the connections on the kit.  
   b. Capture the waveform of the 2 encoder channels in both the forward and reverse rotation cases.
2. Program source code with comments

| ........................................................................................................................ |
| --- |