# Mục tiêu:

* Hiểu các mode hoạt động của timer
* Hiểu cách sử dụng timer để tao delay và tạo xung

# Tham khảo:

* Tài liệu hướng dẫn thí nghiệm, chương 4, 5
* Atmel-2505-Setup-and-Use-of-AVR-Timers\_ApplicationNote\_AVR130.pdf

# Bài 1

1. Viết chương trình con delay 1 ms sử dụng timer 0. Sử dụng chương trình con này để tạo xung 500 Hz trên chân PA0.
2. Mô phỏng, chỉnh sửa chương trình để tạo ra xung chính xác.

.EQU P\_OUT=0

.ORG 0

RJMP MAIN

.ORG 0x40

MAIN: LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

LDI R16, (1<<P\_OUT) ;đặt PA0 output

OUT DDRA, R16

LDI R17, 0x00 ;Timer0 mode NOR

OUT TCCR0A, R17

LDI R17, 0x00 ;mode NOR, dừng

OUT TCCR0B, R17

START: RCALL DELAY ;delay tạo độ rộng xung

IN R17, PORTA ;đọc PortA

EOR R17, R16 ;đảo bit PA0

OUT PORTA, R17

RJMP START ;lập vòng lại

DELAY: LDI R17, 0x83 ;nạp TCNT0=0x83= -125

OUT TCNT0, R17

LDI R17, 0x03 ;Timer0 chạy, hệ số chia N=64

OUT TCCR0B, R17

WAIT: IN R17, TIFR0 ;đọc thanh ghi cờ Timer0

SBRS R17, TOV0 ;chờ cờ TOV0=1 Timer0 tràn

RJMP WAIT ;cờ TOV0=0 tiếp tục chờ

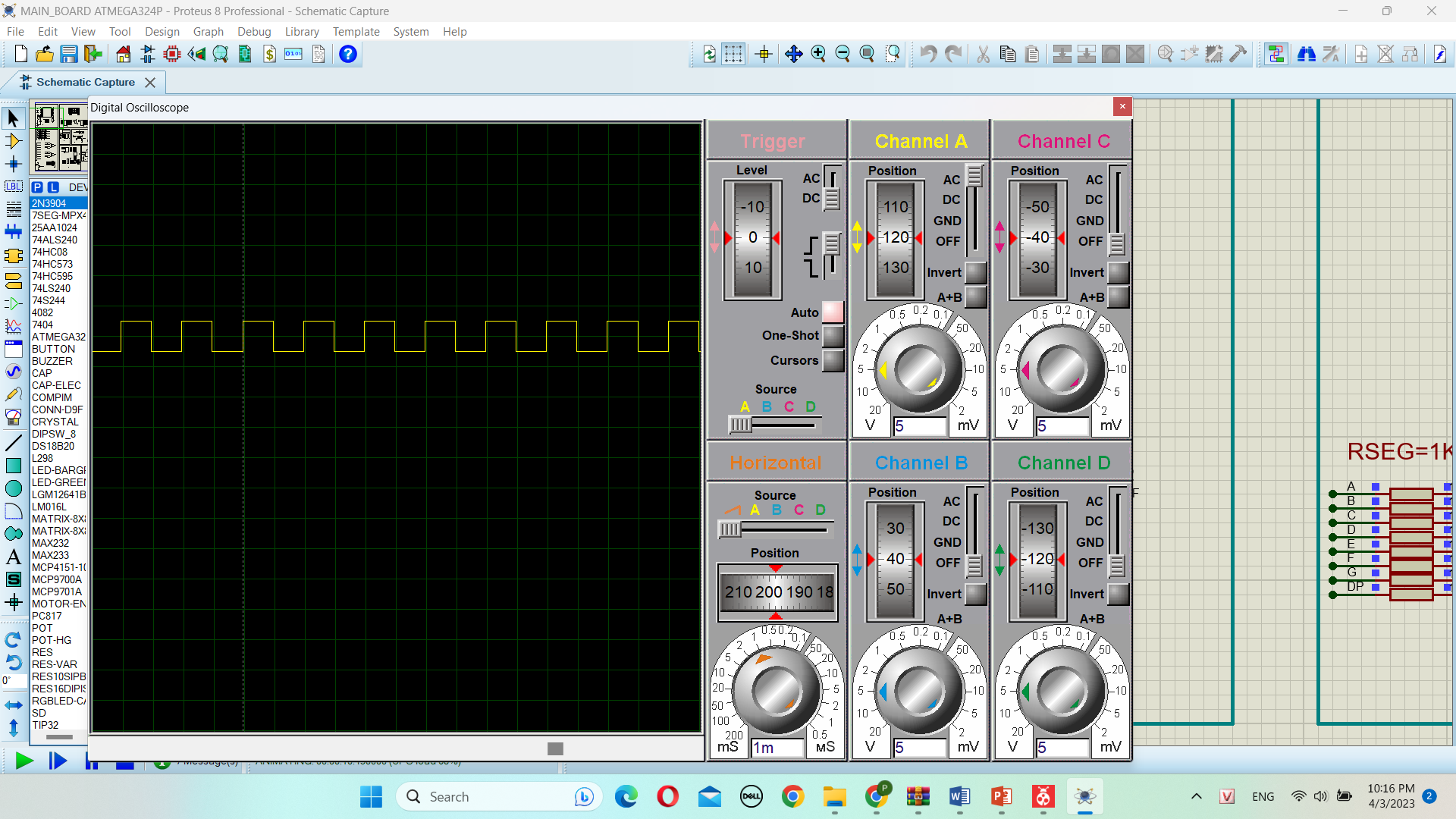
OUT TIFR0, R17 ;nạp bit TOV0=1 xóa cờ TOV0

LDI R17, 0x00 ;dừng Timer0

OUT TCCR0B, R17

RET

1. Kết nối chân PA0 vào oscilloscope để kiểm tra



# Bài 2

1. Viết chương trình tạo 1 xung vuông 64 us sử dụng timer 0 ở chế độ Normal mode. Ngõ ra sử dụng chân OC0.

.EQU P\_OUT=3

.ORG 0

RJMP MAIN

.ORG 0x40

MAIN: LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

LDI R16, (1<<P\_OUT) ;đặt PB3 output

OUT DDRB, R16

LDI R17, 0x00 ;Timer0 mode NOR

OUT TCCR0A, R17

LDI R17, 0x00 ;mode NOR, dừng

OUT TCCR0B, R17

START: RCALL DELAY ;3MC delay tạo độ rộng xung

IN R17, PORTB ;1MC đọc PortB

EOR R17, R16 ;1MC đảo bit PB3

OUT PORTB, R17 ;1MC

RJMP START ;2MC lập vòng lại

DELAY: LDI R17, 0x18 ;1MC nạp TCNT0=0x18= -232 (256-24= 232)

OUT TCNT0, R17 ;1MC

LDI R17, 0x01 ;1MC Timer0 chạy, hệ số chia N=1

OUT TCCR0B, R17 ;1MC

WAIT: IN R17, TIFR0 ;1MC đọc thanh ghi cờ Timer0

SBRS R17, TOV0 ;2/1MC chờ cờ TOV0=1 Timer0 tràn

RJMP WAIT ;2MC cờ TOV0=0 tiếp tục chờ

OUT TIFR0, R17 ;1MC nạp bit TOV0=1 xóa cờ TOV0

LDI R17, 0x00 ;1MC dừng Timer0

OUT TCCR0B, R17 ;1MC

RET ;4MC

1. Viết chương trình thực hiện tạo xung vuông có chu kỳ 64 us sử dụng timer 1 ở chế độ CTC mode. Ngõ ra sử dụng chân OC0.

.EQU P\_OUT=3

.ORG 0

RJMP MAIN

.ORG 0x40

MAIN: LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

LDI R16, (1<<P\_OUT) ;PB3 output

OUT DDRB, R16

LDI R17, 255 ;nạp giá trị so sánh

STS OCR1AL, R17

LDI R17, 0x00 ;Timer1 mode CTC4

STS TCCR1A, R17

LDI R17, 0x09 ;Timer1 chạy, N=1 (\*)

STS TCCR1B, R17

START: IN R17, TIFR1 ;đọc thanh ghi cờ 1MC

SBRS R17, OCF1A ;chờ cờ OCF0A=1 2/1MC

RJMP START ;OCF1A=0 quay lại 2MC

OUT TIFR1, R17 ;xóa cờ 1MC

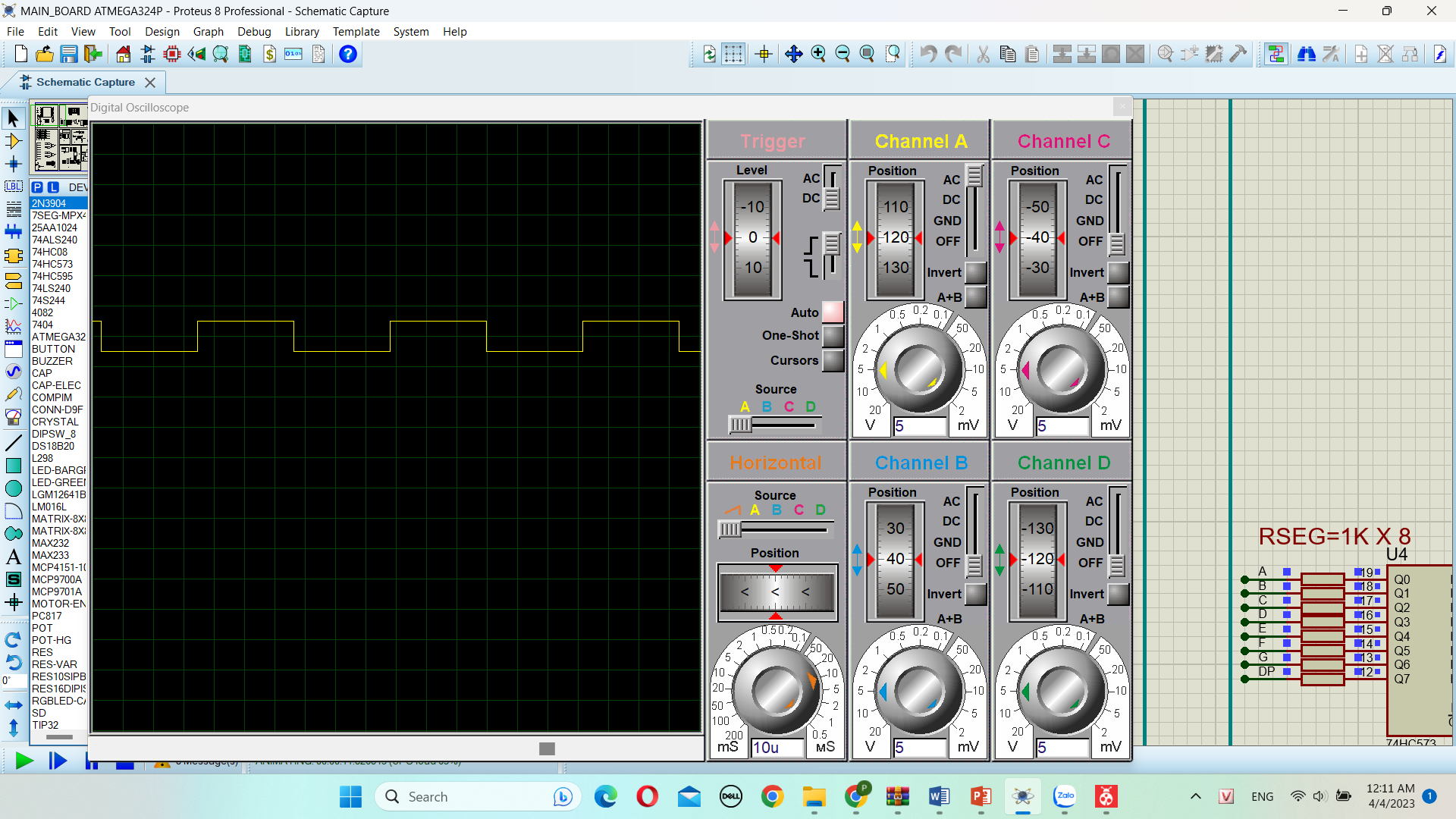
IN R17, PORTB ;đọc PortB 1MC

EOR R17, R16 ;đảo bit PB3 1MC

OUT PORTB, R17 ;xuất ra PortB 1MC

RJMP START ;lặp vòng lại 2MC

1. Kết nối chân OC0 ra oscilloscope và quan sát

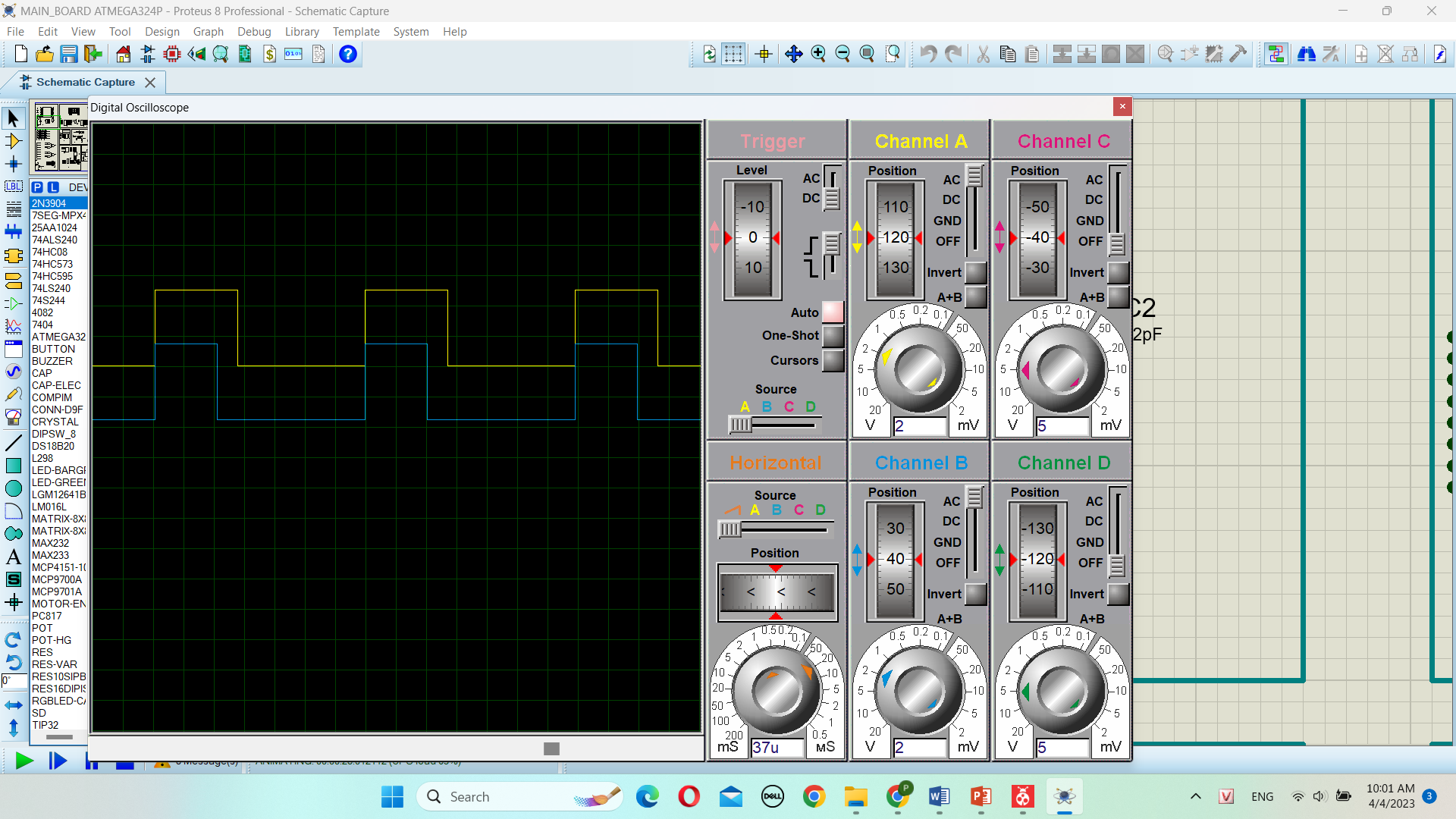


# Bài 3

1. Cho chương trình sau tạo 2 xung PWM trên OC0A và OC0B

|  |
| --- |
| .org 00  call initTimer0  start:  rjmp start  initTimer0:  // Set OC0A (PB3) and OC0B (PB4) pins as outputs  ldi r16, (1 << PB3) | (1 << PB4);  out DDRB,r16  ldi r16, (1 << COM0B1)|(1 << COM0A1) | (1 << WGM00)|(1 << WGM01)  out TCCR0A,r16 // setup TCCR0A  ldi r16, (1 << CS01)  out TCCR0B,r16 // setup TCCR0B  ldi r16, 100  out OCR0A,r16 //OCRA = 100  ldi r16, 75  out OCR0B,r16 //OCRB = 75  ret |

1. Kết nối chân OC0A và chân OC0B ra 2 kênh đo của oscilloscope, đo và ghi nhận, giải thích dạng sóng thu được



# Bài 4

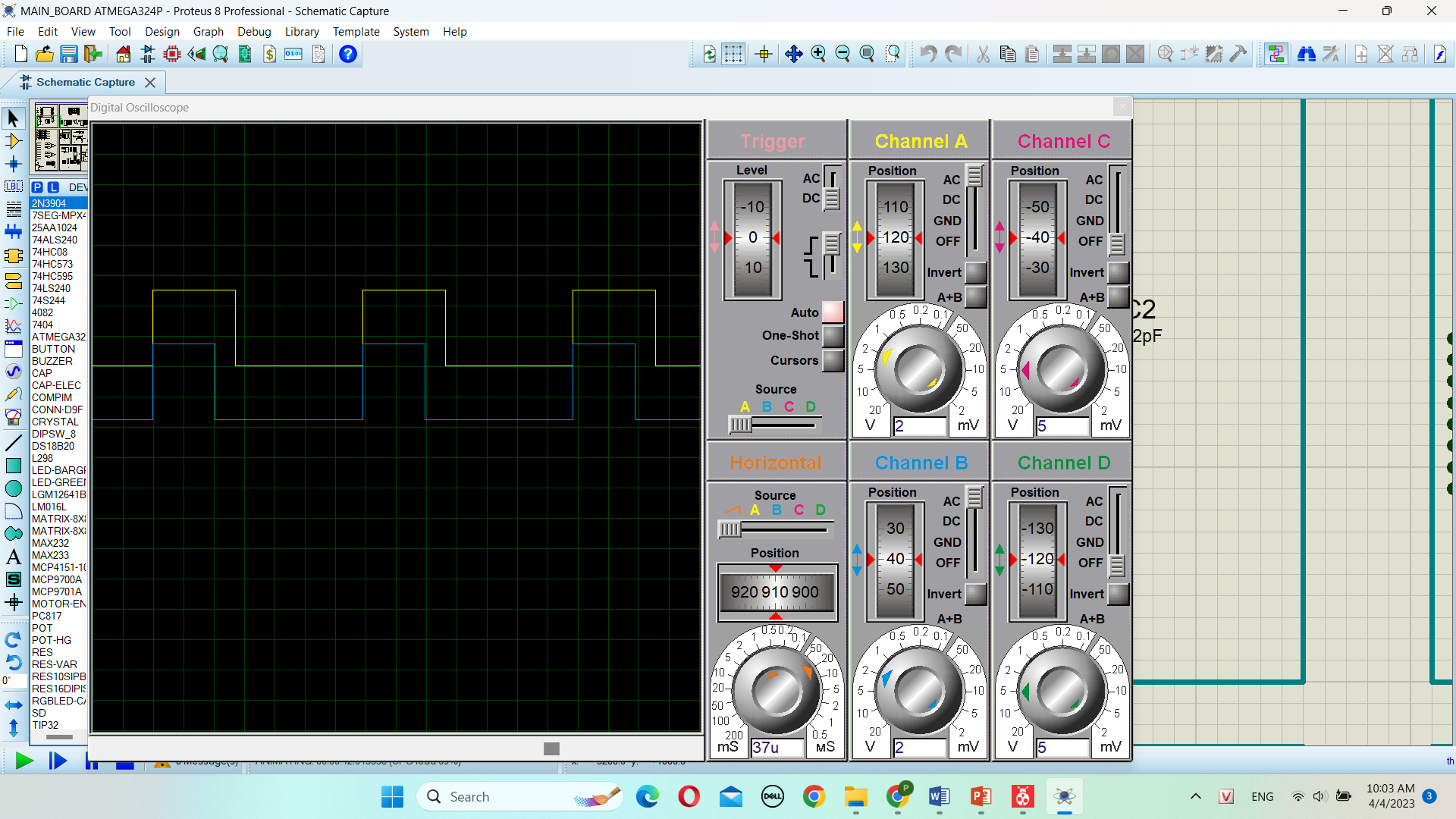
1. Sửa chương trình trên ứng với các trường hợp khác nhau của TCCR0A và TCCR0B

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TCCR0A | | | | | | | | TCCR0B | | | | | | | |
|  | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | COM0A1 | COM0A0 | COM0B1 | COM0B0 |  |  | WGM01 | WGM00 | FOC0A | FOC0B |  |  | WGM02 | CS02 | CS01 | CS00 |
| 2 | 1 | 0 | 1 | 0 |  |  | 1 | 1 |  |  |  |  | 0 | 0 | 1 | 0 |
| 3 | 1 | 0 | 1 | 0 |  |  | 1 | 1 |  |  |  |  | 1 | 0 | 1 | 0 |
| 4 | 1 | 0 | 1 | 0 |  |  | 0 | 1 |  |  |  |  | 0 | 0 | 1 | 0 |

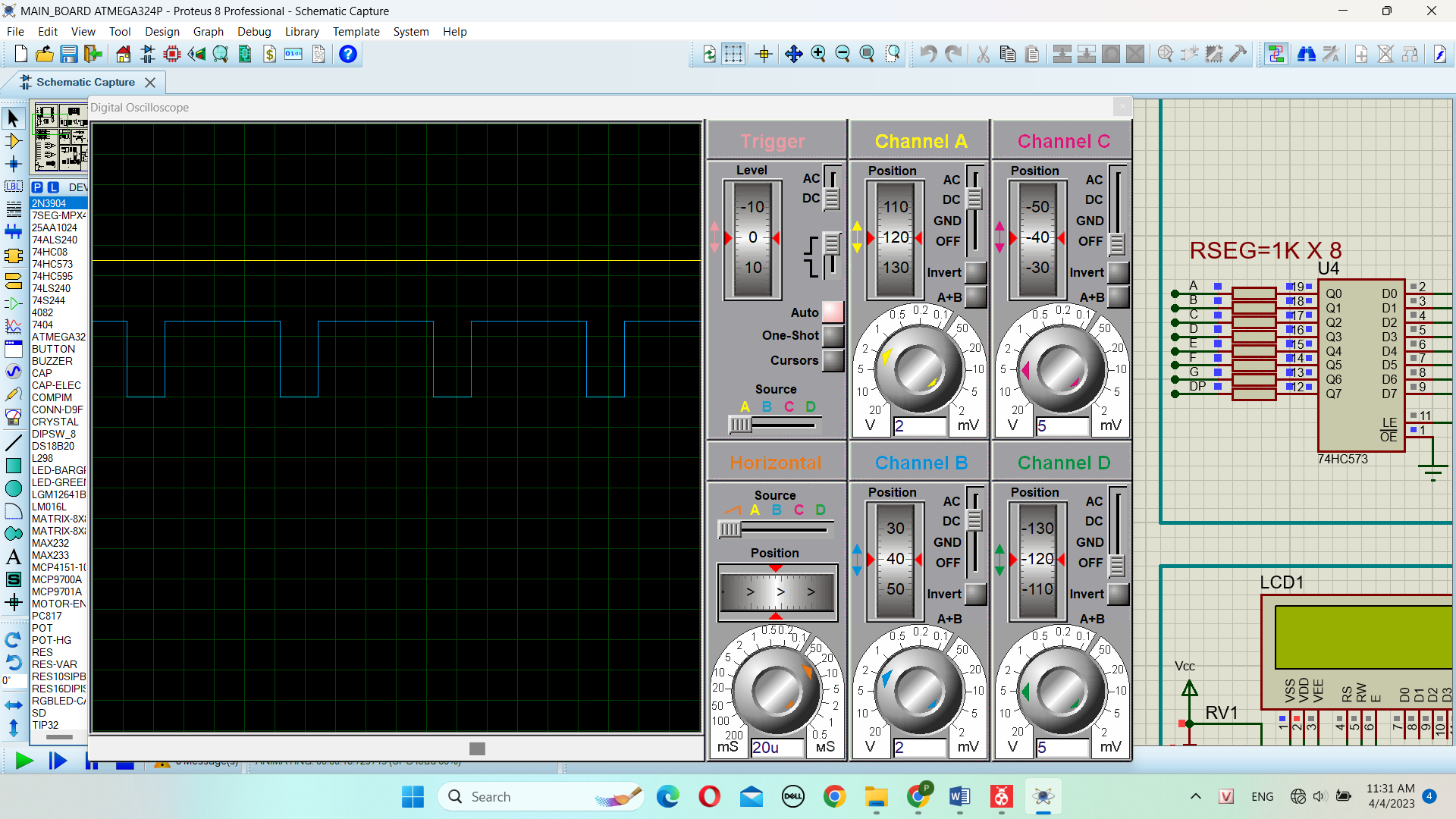
TH\_1: Mode FPWM (TOP: $FF)  
TH\_2: Mode FPWM (TOP: OCR0A)  
TH\_3: Mode PCPWM (TOP: $FF)

1. Kết nối chân OC0A và chân OC0B ra 2 kênh đo của oscilloscope, đo và ghi nhận, giải thích dạng sóng thu được

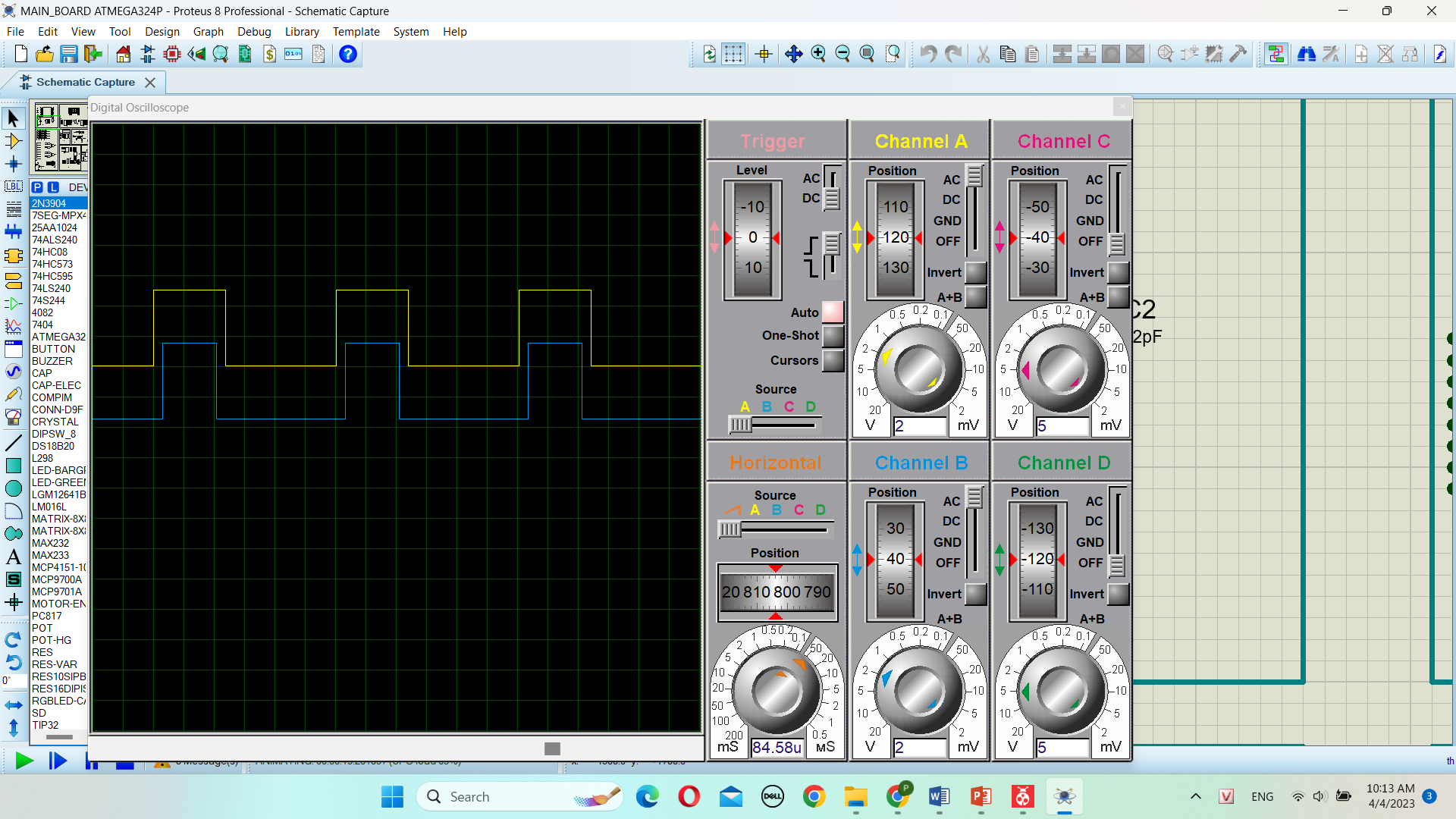
TH\_1: Mode FPWM (TOP: $FF)



TH\_2: Mode FPWM (TOP: OCR0A)



TH\_3: Mode PCPWM (TOP: $FF)



# Bài 5

1. Viết chương trình tạo ra 1 xung tần số 1Khz, duty cycle 25% trên chân OC0B

.EQU P\_OUT=4 ;gán ký hiệu P\_OUT=5

.EQU OUTPORT= PORTB

.EQU SETPORT= DDRB

.EQU TP\_H= -31 ;giá trị đặt trước mức 1

.EQU TP\_L= -94 ;giá trị đặt trước mức 0

.ORG 0

RJMP MAIN

.ORG 0x40

MAIN: LDI R16, HIGH(RAMEND)

OUT SPH, R16

LDI R16, LOW(RAMEND)

OUT SPL, R16

LDI R16, (1<<P\_OUT) ;đặt PB4 output

OUT SETPORT, R16

LDI R17, 0x00 ;Timer0 mode NOR

OUT TCCR0A, R17

LDI R17, 0x00 ;mode NOR, dừng

OUT TCCR0B, R17

START:SBI OUTPORT, P\_OUT ;output=1 1MC

LDI R17,TP\_H ;nạp TP\_H 1MC

RCALL DELAY\_T0 ;ctc chạy Timer0 3MC

CBI OUTPORT, P\_OUT ;output=0 1MC

LDI R17, TP\_L ;nạp TP\_L 1MC

RCALL DELAY\_T0 ;ctc chạy Timer0 3MC

RJMP START ;2MC

DELAY\_T0:

OUT TCNT0, R17 ; 1MC

LDI R17, 0x03 ;chạy, chia N=64 1MC

OUT TCCR0B, R17 ; 1MC

WAIT: IN R17,TIFR0 ; 1MC

SBRS R17, TOV0 ; 2/1MC

RJMP WAIT ; 2MC

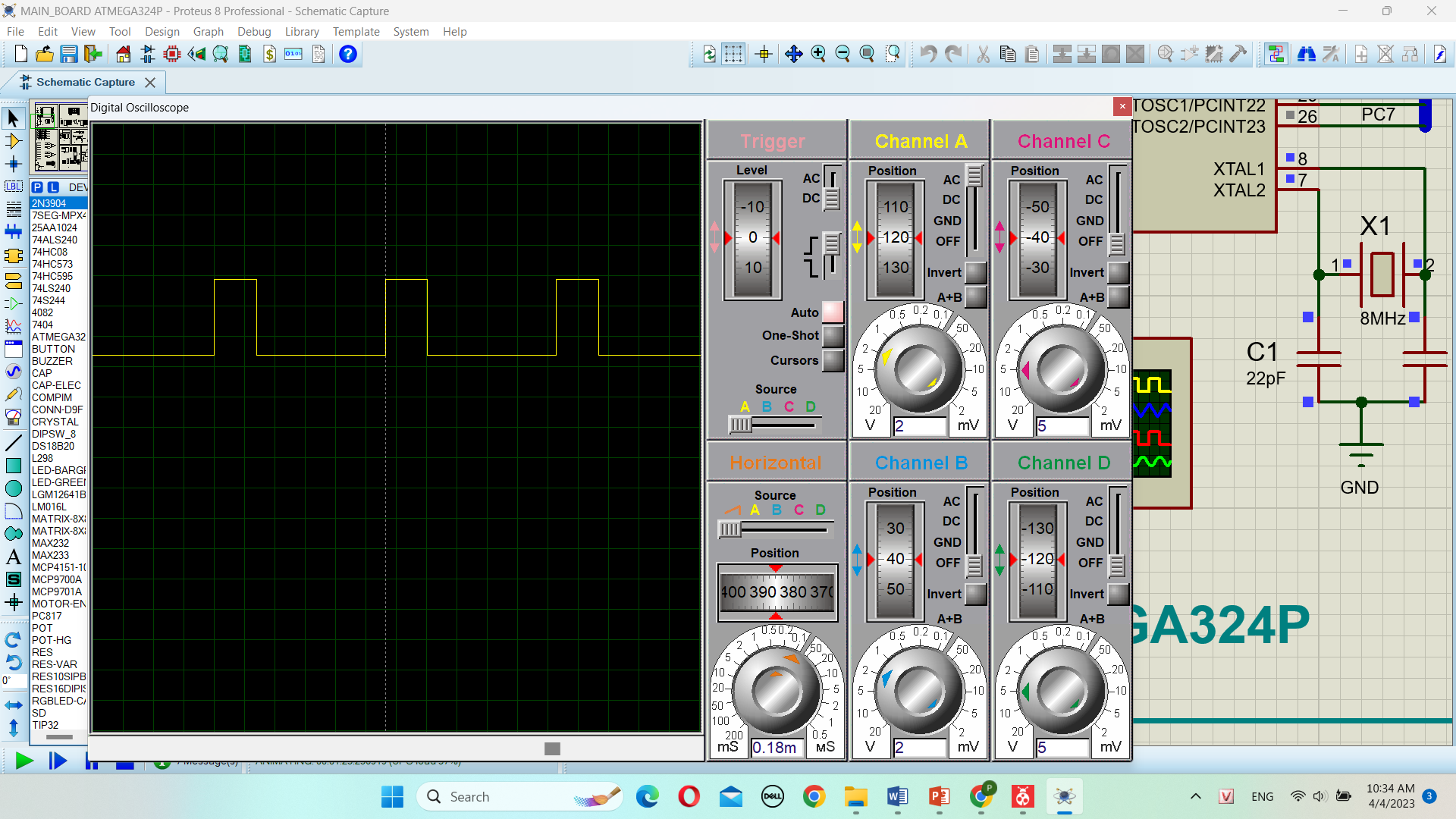
OUT TIFR0, R17 ; 1MC

LDI R17, 0x00 ;dừng 1MC

OUT TCCR0B,R17 ; 1MC

RET ; 4MC

1. Kết nối vào Oscilloscope và đo dạng sóng ngõ ra



C) Kết nối OC0B vào kênh R của 1 LED RGB. Viết chương trình để tăng duty cycle trên OC0B từ 0% lên 100% rồi lại giảm xuống 0, sau 10 ms duty cycle tăng/giảm 1%.

.ORG 0

RJMP XUAT\_NHAP

.ORG 0X40

XUAT\_NHAP: SBI DDRB,4

CBI PORTB,4

STACK: LDI R16,HIGH(RAMEND)

OUT SPH,R16

LDI R16,LOW(RAMEND)

OUT SPL,R16

SET\_TIMER:

LDI R16,0X23 ; timer0 mode FPWM3, dao BIT NGO RA KHI DAT GIA TRI SO SANH

OUT TCCR0A,R16

LDI R16,124 ; nap gia tri TOP cho OCR0A

OUT OCR0A,R16

LDI R16,0X0B

OUT TCCR0B,R16

MAIN: LDI R16,0

LDI R17,1

LDI R18,2

START:

LOOP1: RCALL DELAY\_10MS

ADD R16,R17

OUT OCR0B,R16

RCALL DELAY\_10MS

ADD R16,R18

OUT OCR0B,R16

RCALL DELAY\_10MS

ADD R16,R17

OUT OCR0B,R16

RCALL DELAY\_10MS

ADD R16,R17

OUT OCR0B,R16

CPI R16,125

BREQ LOOP2

RJMP LOOP1

LOOP2: RCALL DELAY\_10MS

SUB R16,R17

OUT OCR0B,R16

RCALL DELAY\_10MS

SUB R16,R18

OUT OCR0B,R16

CPI R16,0

BREQ START

RJMP LOOP2

DELAY\_10MS:

LDI R19,80

L1: LDI R20,250

AGAIN: NOP

DEC R20

BRNE AGAIN

DEC R19

BRNE L1

RET