**Name: LƯƠNG THÀNH VỸ (2151280)**

**TRẦN TRUNG TOÀN (2151265)**

# objective:

* Execute instructions to create delays using subroutines.
* Perform communication with shift registers.

# references:

* Lab manual chapter 1-2.

# EXPERIMENT 1:

1. Use the following program:

|  |
| --- |
| .include "m324PAdef.inc"  .org 00  ldi r16,0x01  out DDRA, r16  start:  sbi PORTA,PINA0  cbi PORTA, PINA0  rjmp start |

Connect PA0 to a measurement channel on the TEST STATION and measure pulse forms using an oscilloscope.

# EXPERIMENT 2:

1. Write a subroutine Delay1ms and use it to write a program to generate a 1KHz square wave on PA0.
2. Use this subroutine to write subroutines Delay10ms, Delay100ms, and Delay1s.
3. Use the Delay1s subroutine to write a program to blink/turn off an LED connected to PA0.

# EXPERIMENT 3:

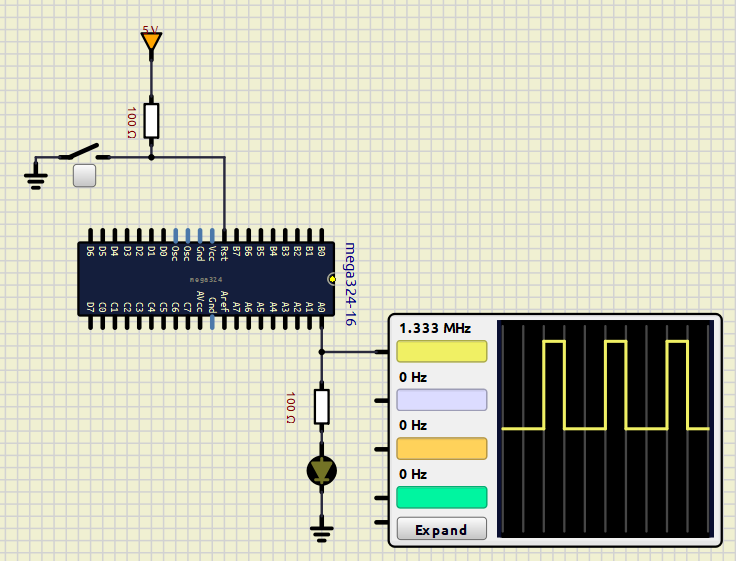
1. Connect the necessary signals from an AVR port to the control signals of the shift register on header J13. Connect the output of the shift register to a LED bar.
2. Using the sample programs from the experiment guide, write a program to create a gradually lit LED effect from left to right, then gradually turn them off from left to right after every 500ms.

# EXPERIMENT 1:

1. Answer the following questions:
   1. Capture a pulse waveform on PA0.

**ANSWER:**

We didn’t have access to an oscilloscope in the laboratory, so we captured the waveform using a simulation instead.



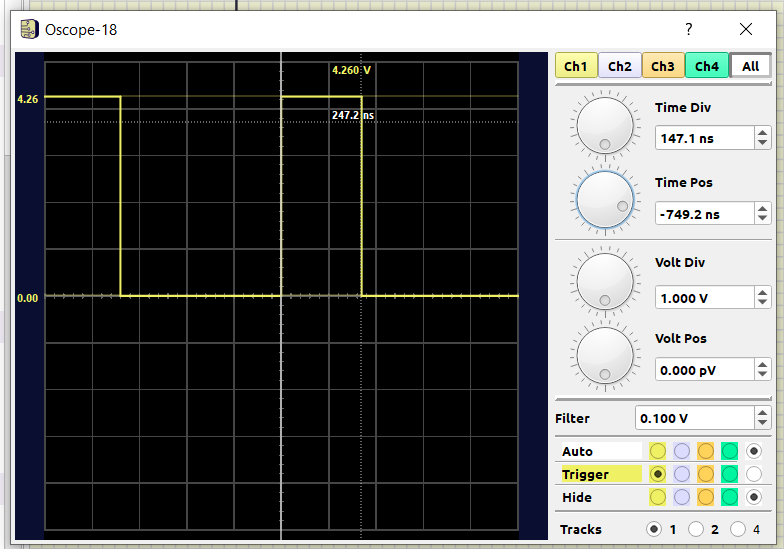
* 1. What is the frequency, duration of the high signal, and duration of the low signal?

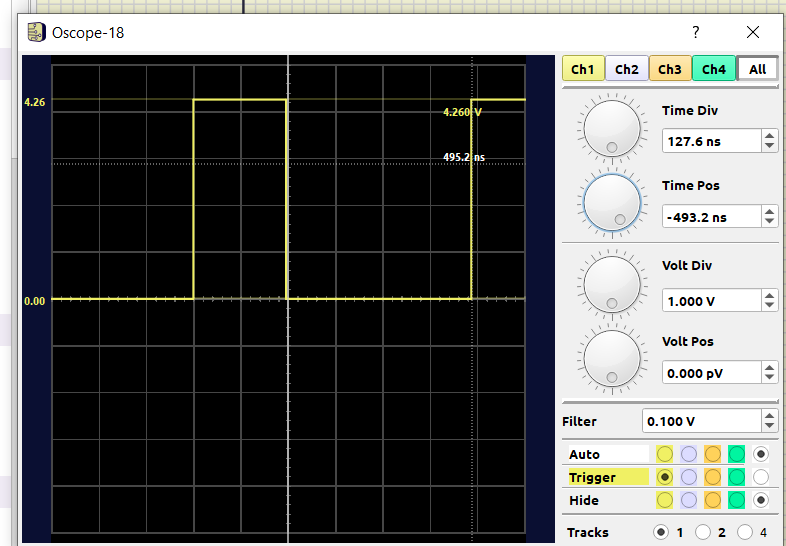
**ANSWER:**

|  |
| --- |
| .include "m324PAdef.inc"  .org 00  ldi r16,0x01  out DDRA, r16  start:  sbi PORTA,PINA0  cbi PORTA, PINA0  rjmp start |

**ANSWER:**

The frquency observed is 1.333 MHz. And the HIGH duration of the signal is 247 ns, the LOW duration is 496 ns.





* 1. Explain the measured results.

**ANSWER:**

The code creates a loop that toggles pin PA0 on and off using two consecutive instructions, generating a PWM waveform. Each loop iteration consists of three instructions: SBI, CBI, and RJMP, each requiring 2 cycles to execute. After setting the signal HIGH with SBI, the signal goes LOW 2 cycles later with CBI. However, it takes 4 cycles for the signal to go HIGH again, due to the RJMP and SBI instructions within the loop. This results in a LOW duration that is twice as long as the HIGH duration, creating a 33% duty cycle PWM.

The total period of the waveform is 6 cycles. Operating at 8 MHz, each cycle takes 0.125 µs, giving a period . Thus, the frequency is:

.

The HIGH duration of the signal is, and the LOW duration is us. These result matching the waveform observed on the oscilloscope.

# EXPERIMENT 2:

1. Answer the following questions:
   1. How to calculate the number of machine cycles needed to execute the Delay1ms subroutine. Present a simulation image.

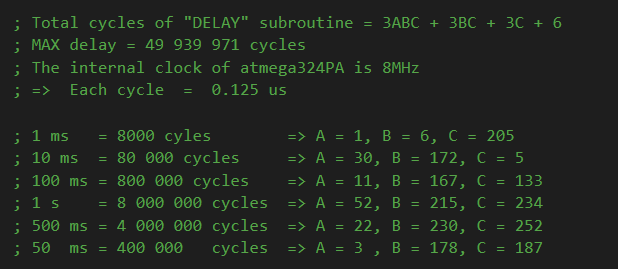
**ANSWER:**

At an operating frequency of 8 MHz, the AVR takes 0.125 µs per cycle. To achieve a delay of 1 ms, we need to create a subroutine that executes cycles.

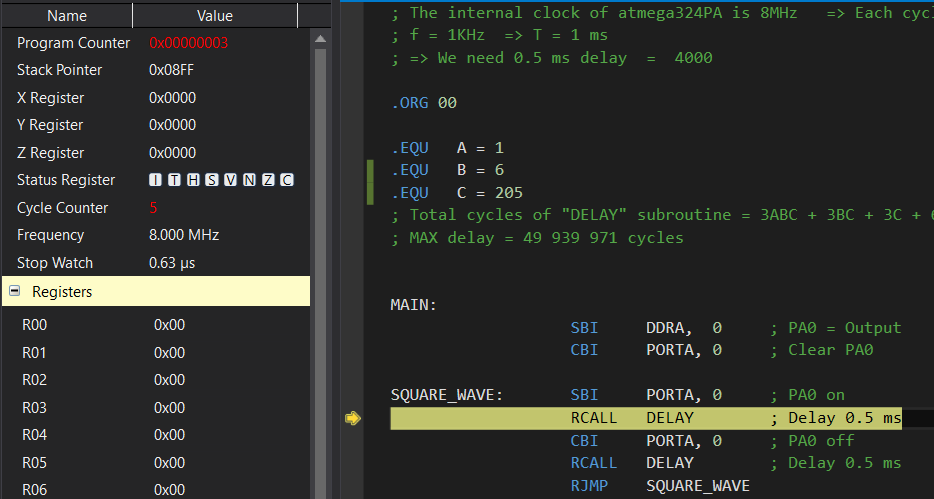
In our subroutine (shown in the source code below), we use 3 nested loops to generate a delay, with each loop iterating A, B, and C times, respectively. After writing the subroutine, we count the cycles executed by each instruction and loop iteration. From this, we derive a function for the delay in cycles as: 3ABC+3BC+3C+6

The PA0 pin experiences a 1 ms delay between each toggle, producing a square wave with a period of 2 ms. The frequency of this waveform is

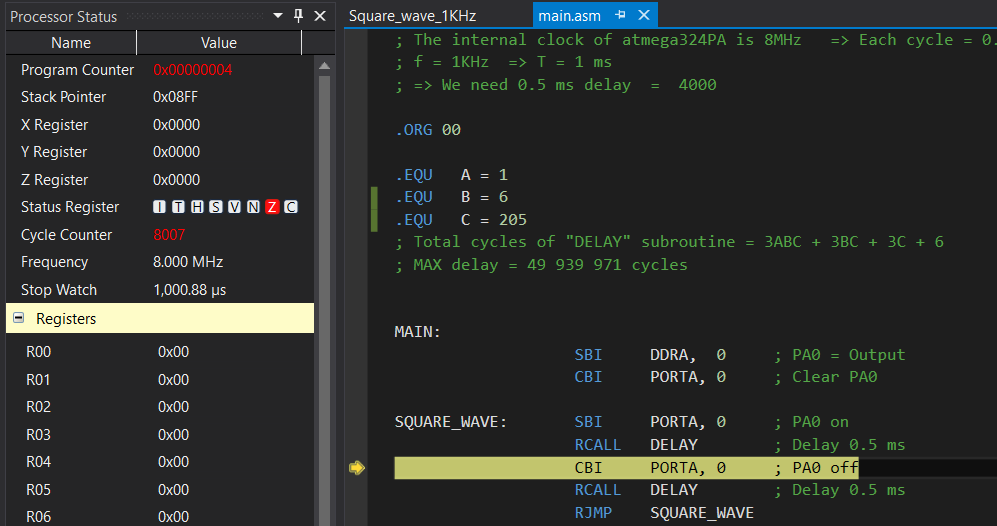
To achieve an 8000-cycle delay, we can assign random values to any two of A, B, or C and solve for the third. Alternatively, we can use a programming language like Python or C to iterate through all possible values of A, B, and C (from 0 to 255) and find the combination with the smallest difference from the target cycle count. Using this method, we have the delay subdroutine for these amount of delay as show below:



* 1. Image of a 1KHz square wave on PA0.
* **Before DELAY Suubroutine:**



* **After DELAY subroutine:**



* **Waveform on oscilloscope:**

A computer screen shot of a circuit board

Description automatically generated

* 1. What is the error?

**ANSWER:**

The observed frequency is 499.7 Hz, indicating an error of 0.3 Hz, or (, which is highly precise . This slight imperfection may be due to the RJMP instruction within the loop generating the waveform.

1. Source code for 2.c with comments.

**ANSWER:**

|  |
| --- |
| ; The internal clock of atmega324PA is 8MHz => Each cycle = 0.125 us  ; f = 1KHz => T = 1 ms  ; => We need 1 ms delay = 8000  ; 1 ms = 8000 cyles => A = 1, B = 6, C = 205  .ORG 00  .EQU A = 1  .EQU B = 6  .EQU C = 205  ; Total cycles of "DELAY" subroutine = 3ABC + 3BC + 3C + 6  MAIN:  SBI DDRA, 0 ; PA0 = Output  CBI PORTA, 0 ; Clear PA0  SQUARE\_WAVE: SBI PORTA, 0 ; PA0 on  RCALL DELAY ; Delay 1 ms  CBI PORTA, 0 ; PA0 off  RCALL DELAY ; Delay 1 ms  RJMP SQUARE\_WAVE    DELAY: ; # of Cycle of Instr  LDI R22, C ; +1  L2: LDI R21, B ; +1 }  L1: LDI R20, A ; +1 }  L0: DEC R20 ; +1 } L0 = 3A  BRNE L0 ; +2 }  ; -1 } L1 = B\* (L0 + 4 -1) = 3AB + 3B  DEC R21 ; +1 }  BRNE L1 ; +2 }  ; -1 } L2 = C\*(L1+4-1) = 3ABC + 3BC + 3C  DEC R22 ; +1 }  BRNE L2 ; +2 }  ; -1 } +3 and +3 for first 3 LDI  RET ; +4 }  ; ====> Total cycle of "DELAY" = 3ABC + 3BC + 3C + 6 |

**RESULT ON EXPERIMENT KIT:**

**In this Experiment, we modify the DELAY subroutine to delay 1s, which create the sqaure. This cause the LED to turn on and off each after 1 second.**

<https://drive.google.com/file/d/10wZPJvxVltNhq05imKVNCjv54i7hUQFc/view?usp=sharing>

# EXPERIMENT 3:

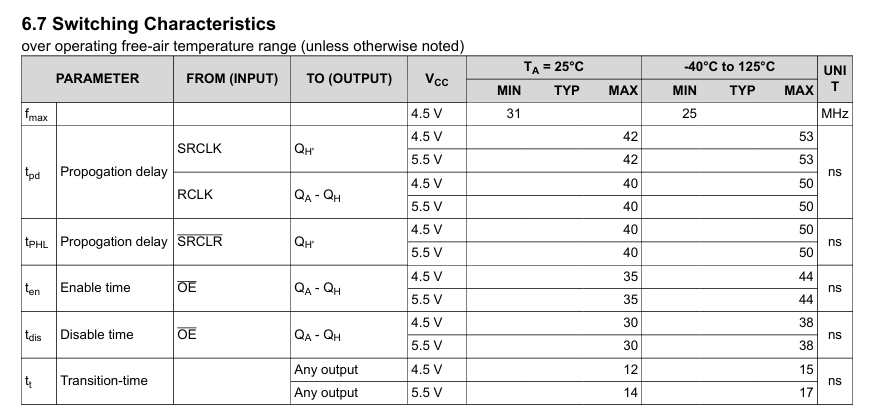
1. Answer the following questions:
   1. Describe the connections on the experimental kit.

**ANSWER:**

On the experiment kit, there are 4 pins for the SRCLK, RCLK, OE, and DS signals to control the 74HC595 shift register. However, since the 74HC595 is a right shift register, we can only display LEDs lighting up from right to left, as the serial input bit is shifted in from the right. To make the LEDs light up from left to right, we need to reverse the connections between the shift register outputs and the LEDs. This means connecting the LSB output to the far-left LED and the MSB output to the far-right LED.

* 1. According to the datasheet of 74HC595, what is the highest clock frequency it can operate at?

**ANSWER:**



The parameter fmax specified in the switch characteristic is 31 MHz at room temperature (T = 25oC). This means that maximum operating frequency of 74HC595 IC at Normal condition is 32 MHz.

* 1. How do you expand the display to 16 LEDs?

**ANSWER:**

To display 16 LEDs, we’ll need two 74HC595 shift registers. We can connect these two 8-bit shift registers by linking the Q7' (Serial Out) pin of the first register to the DS (Serial In) pin of the second register. Both registers will share the same RCLK, SRCLK clock signals, and OE signal, effectively creating a 16-bit shift register. We then control the RCLK, SRCLK, and OE signals as we would for a single 8-bit shift register, but the output now extends to 16 bits. This allows us to connect and control 16 LEDs.

However, on the experiment kit, there is only one LED bar and a single 74HC595 IC, so we displayed output on just 8 LEDs.

1. Source code with comments.

**ANSWER:**

|  |
| --- |
| ; The internal clock of atmega324PA is 8MHz  ; => Each cycle = 0.125 us  ; 500 ms = 4 000 000 cycles => A = 22, B = 230, C = 252  ; PA0 = SRCLK  ; PA1 = RCLK  ; PA2 = DS or SER (serial input)  ; PA3 = SRCLR (active LOW clear shift register)  .ORG 00  MAIN:  LDI R16, 0x0F ; Config P0 -> P3 as OUTPUT  OUT DDRA, R16  CBI PORTA, 3 ; Pulse SRCLR pin to (clear shift register)  SBI PORTA, 3 ; SRCLR pin is ACTIVE LOW  ; Turn on LED from Right to LEFT  CHASE\_UP: LDI R17, 0x08 ; Counter for number of bit sent (8-bit)  SBI PORTA, 2 ; PA2 = DS = 1    LOOP\_UP: SBI PORTA, 0 ; pulse PA0 = SRCLK  CBI PORTA, 0  SBI PORTA, 1 ; pulse PA1 = RCLK  CBI PORTA, 1  RCALL DELAY ; Delay 500 ms  DEC R17 ; Check if we have sent 8-bit of data yet ?  BRNE LOOP\_UP  RJMP CHASE\_DOWN  ; Turn off LED from Right to LEFT  CHASE\_DOWN: LDI R17, 0x08 ; Counter for number of bit sent (8-bit)  CBI PORTA, 2 ; PA2 = DS = 0    LOOP\_DOWN:  SBI PORTA, 0 ; pulse PA0 = SRCLK  CBI PORTA, 0  SBI PORTA, 1 ; pulse PA1 = RCLK  CBI PORTA, 1  RCALL DELAY ; Delay 500 ms  DEC R17 ; Check if we have sent 8-bit of data yet ?  BRNE LOOP\_DOWN  RJMP CHASE\_UP  DELAY: ; # of Cycle of Instr  LDI R22, 22 ; +1  L2: LDI R21, 230 ; +1 }  L1: LDI R20, 252 ; +1 }  L0: DEC R20 ; +1 } L0 = 3A  BRNE L0 ; +2 }  ; -1 } L1 = B\* (L0 + 4 -1) = 3AB + 3B  DEC R21 ; +1 }  BRNE L1 ; +2 }  ; -1 } L2 = C\* (L1 + 4 -1) = 3ABC + 3BC + 3C  DEC R22 ; +1 }  BRNE L2 ; +2 }  ; -1 } +3 and +3 for the first 3 LDI  RET ; +4 }  ; ====> Total cycle of "DELAY" = 3ABC + 3BC + 3C + 6 |

**RESULT ON EXPERIMENT KIT:**

<https://drive.google.com/file/d/1jlv9pvt4eus4BAlVC4lm9vk41Wagdr98/view?usp=sharing>