Timer Lab

Example 1:

Write Embedded C code using ATmega328P µC to control led by Timer2.

Requirements:

The LED is connected to pin 0 in PORTC. -1
Connect the Led using **Positive Logic** configuration. -2
Configure the timer clock to F_CPU/1024. -3
Timing should be counted using Timer2 in Normal Mode. -4
Toggle the LED every half second. -5

Steps Main:

Configure the led pin to be an output pin. -1
Make LED off at the beginning(Positive Logic). -2
Enable global interrupts in MC by setting the I-Bit. -3
Start the timer. -4

Steps Timer init:

initial the value of timer counter register (TCNT) to zero. -1
Enable overflow interrupt. -2
Configure the timer control register -3
Non PWM mode FOC0=1 ✓
Normal Mode WGM01=0 & WGM00=0 ✓
Normal Mode COM00=0 & COM01=0 ✓
clock = F_CPU/1024 CS00=1 CS01=0 CS02=1 ✓

ISR: Increase the count of a global or static variable in ISR and check the count if it reach the required counts to obtain the required time (0.5 sec).

Example2:

Write Embedded C code using ATmega328P µC to control a 7-sgment using Timer2.

Requirements:

7-segment connected to PORTC. -1
Configure the timer clock to F_CPU/256. -2
Timing should be counted using Timer2 in Normal Mode. -3
Every second the 7-segment should be incremented by one. If the 7-segment reaches 9, overflow will occur. -4

Example 3:

Repeat Example 1 but use Timer2 Clear timer on compare (CTC) Mode.

- In this example CTC mode will act exactly as the timer in Example 1.

Example 4:

Write Embedded C code using ATmega328P µC to generate a 15.625 KHz clock using Timer2 CTC Mode.

Requirements:

1. Use Timer2 in CTC Mode with a clock equal to F_CPU/1024 clock.
2. Clock duty cycle is 50%.
3. Connect the output to speaker of kit

Steps Timer init:

1. initial the value of timer counter register (TCNT) to zero.
2. Set Compare value (OCR=250)
3. Set OC2A (PB3) Pin as output pin
4. Configure the timer control register
   - ✓ Non PWM mode FOC0=1
   - ✓ CTC Mode WGM01=1 & WGM00=0
   - ✓ Toggle OC2A on compare match COM00=1 & COM01=0
   - ✓ clock = CPU/1024 clock CS00=1 CS01=1 CS02=1
   - ✓ clock = F_CPU/128 CS20=1 CS21=0 CS22=1.

Example 5:

Assuming that a push button is connected to the pin ICP1 (PB0), the following program will read the Timer1 (TCNT1) value at every rising edge (button press) and place the time difference between consecutive presses on the serial monitor using the Serial.println() function.

Steps Main:

1. Clear the Timer1 counter (TCNT1) to start fresh.
2. Clear the Input Capture Flag (ICF1) to ensure the previous event does not interfere.
3. Configure ICP1 (PB0) to capture on the rising edge of the button press.
4. Wait for the first button press (Monitor ICF1).
5. Capture the current time in ICR1 and store it in a variable.
6. Clear the capture flag for the next event.
7. Wait for the next button press (Monitor ICF1 again).

Capture the new time in ICR1 and calculate the time difference (recent capture - previous capture). -8

Convert the time difference to milliseconds and display it on the serial monitor. -9