THE MICROPROCESSORS & MICROCONTROLLERS

Instructor : The Tung Than
Student: Bach Hoang Phan Thanh
ID: 21520599

PRATICE EXERCISE #3:
USING INTERRUPT

1. Design Result

Task:

+ Present and draw a flowchart to handle 2 buttons with the following functions:

o Button A: Pause/Resume stopwatch

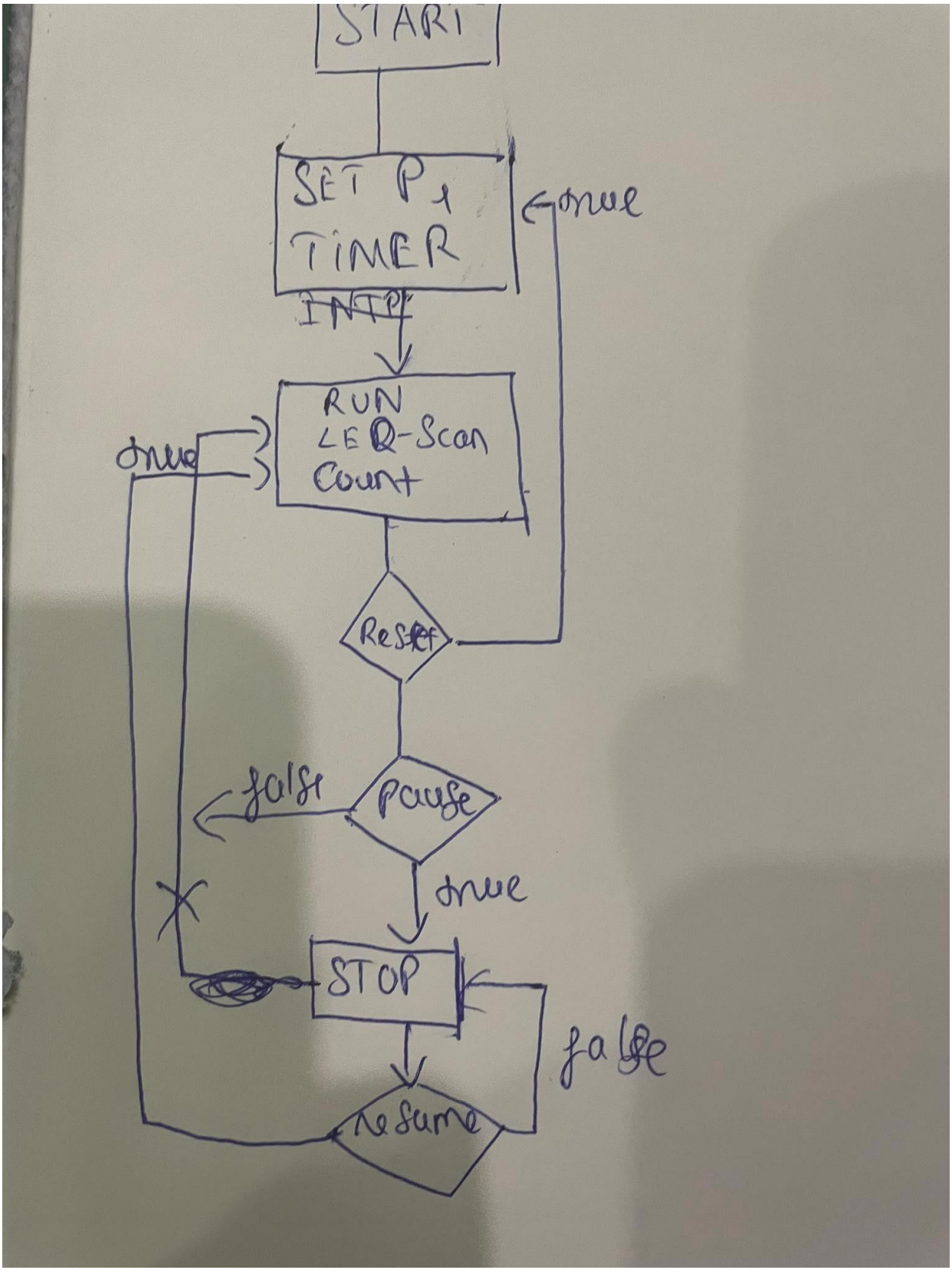
o Button B: Reset the stopwatch..

+ Using AT89C51/AT89C52 in combination with 4 7-Segment LED modules and 2 buttons above, design a Sport clock circuit with the ability to count accurately to 1% of seconds, counting range from 00.00 seconds to 99.99 seconds and has 2 buttons to control Pause/Resume and Reset.

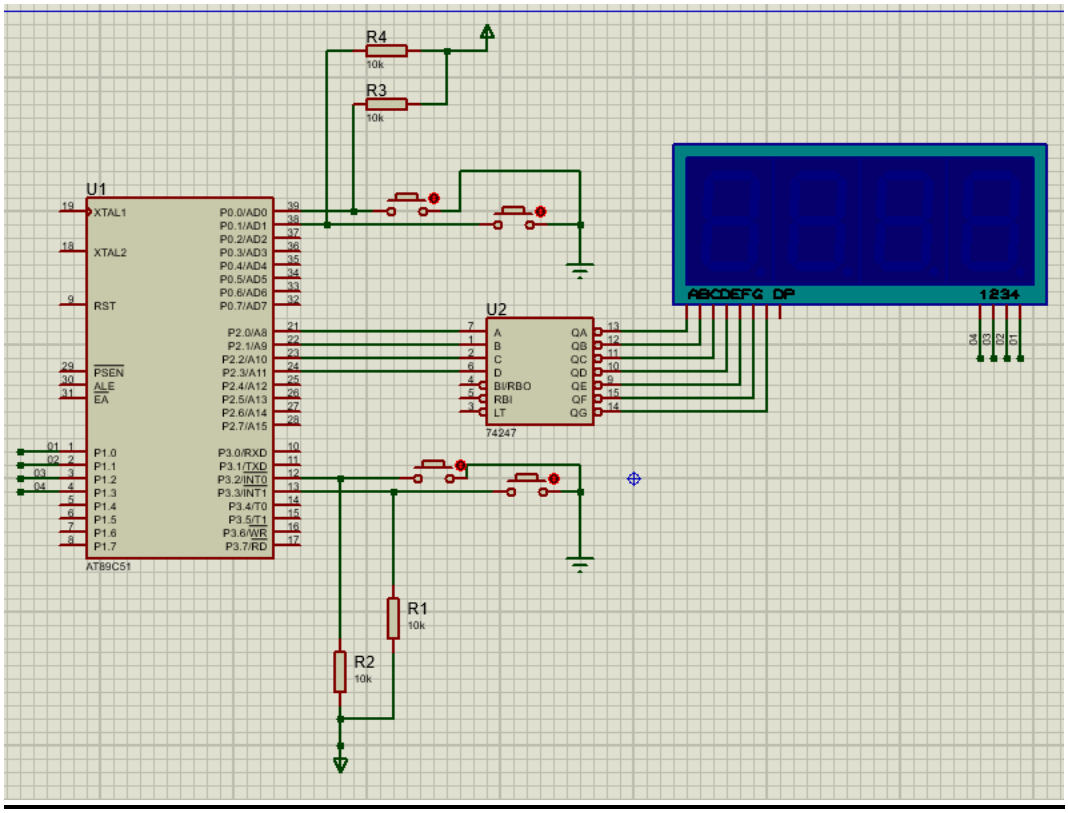
+ Add 2 buttons to the Sport watch with the following function: o Button C: Increase the number of seconds counting to 1 second o Button D: Decrease the number of seconds to 1 second

Picture of Result:

Flowchart:


START
SET P1
TIMER
true
INTP
RUN
LED-Scan
Count
true
Reset
false
Pause
true
STOP
false
Resume

Result:


R4
10k
R3
10k
U1
XTAL1
XTAL2
RST
PSEN
ALE
EA
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
AT89C51
U2
A
B
C
D
BI/RBO
RBI
LT
QA
QB
QC
QD
QE
QF
QG
74247
ABCDEFG DP
1234
R1
10k
R2
10k

2. Explain the operating principle of the effects

- Google Drive link: https://drive.google.com/drive/folders/1MjT-Dsfz1p1rsbeuOTYqRH1pB7SMWmWp?usp=sharing
- Source code, include English Explanation:
- (Many of the code I reuse from Lab2 so the explanation have to be shortened)

Source Code (Include English Explanation)

```
ORG 0000H; Reset
        LJMP MAIN

ORG 0003h; Interrupt 0
        LJMP BUTTONA

ORG 000Bh; Timer 0
        LJMP DISPLAY

ORG 0013h; Interrupt 1
        LJMP BUTTONB

ORG 001Bh; Timer 1
LJMP UP

ORG 0030H; Main process
MAIN:
START:
MOV P1, #0001b

;;Set Timer
MOV TMOD, #11h; Set mode 1 for 2 timers
SETB TR0
SETB TR1
MOV IE, #8Fh; Enable interrupts
SETB IT0; Set falling-edge type for ext. interrupt 0
SETB IT1; Set falling-edge type for ext. interrupt 1

LOOP: ;; This LOOP use for check if BUTTON C or BUTTON D pushed
CHECK_BUT1:
JB P0.0, CHECK_BUT2
```

```
        ACALL BUTTONC
CHECK_BUT2:
        JB P0.1, NOTREG
        ACALL BUTTOND
NOTREG:
        SJMP LOOP


BUTTONC: ;; to increase second
        JNB P0.0, $
        INC R2
        CJNE R2, #0Ah, DONEC
        MOV R2, #00h
        INC R3
        CJNE R3, #0Ah, DONEC
        MOV R3, #00h
DONEC:
RET


BUTTOND: ;; to decrease second
        JNB P0.1, $
        DEC R2 ;
        CJNE R2, #0FFh, DONED
        MOV R2, #09h
        DEC R3
        CJNE R3, #0FFh, DONED
        MOV R3, #09h
DONED:
RET

UP: ;;run the clock, increase each value of 4-led
        INC R0
UP_R0:
        CJNE R0, #10d, UP_R1
        MOV R0, #00d
        INC R1
UP_R1:
        CJNE R1, #10d, UP_R2
        MOV R1, #00d
```

```
        INC R2
UP_R2:
        CJNE R2, #10d, UP_R3
        MOV R2, #00d
        INC R3
UP_R3:
        CJNE R3, #10d, FINISH
        MOV R3, #00d
FINISH:
        ACALL TIMER_1
RETI

DISPLAY: ;; to display
        MOV A, P1
        RR A
        ACALL CHECK
        MOV P1, A
        ACALL SELECT_LED

        ACALL TIMER_2
RETI

SELECT_LED: ;;move the value to display on all 4 7-segments LED
        CJNE A, #0001b, LEFT1
        MOV P2, R0 ;; R0 move into P2 to display on LED
        LJMP END_SELECT
LEFT1:
        CJNE A, #0010b, LEFT2
        MOV P2, R1 ;; R1 move into P2 to display on LED
        LJMP END_SELECT
LEFT2:
        CJNE A, #0100b, LEFT3
        MOV P2, R2 ;; R2 move into P2 to display on LED
        LJMP END_SELECT
LEFT3:
        MOV P2, R3 ;; R3 move into P2 to display on LED
END_SELECT:
RET

CHECK:
```

```
        CJNE A, #80h, END_CHECK
        MOV A, #08h
        END_CHECK:
RET

BUTTONA: ;; to pause the clock
        CPL TR1 ;;TR1 go to 0, Timer stop
RETI

BUTTONB: ;; reset BUTTON - all led value will be 0
        MOV R0, #00h
        MOV R1, #00h
        MOV R2, #00h
        MOV R3, #00h
RETI
TIMER_1:
        MOV TH0, #HIGH(-1000); Set delay for 1000us for timer0
        MOV TL0, #LOW(-1000)
        SETB TR0; Start timer 0
RET
TIMER_2:
        MOV TH1, #HIGH(-10000) ;;loads the high byte of the initial value
for Timer 1 to delay for 10000us
        MOV TL1, #LOW(-10000)   ;; loads the low byte of the initial value
for Timer 1 to delay for 10000us
        SETB TR1; Start timer 1
RET
END
```

3. Extra

The main difference between level-triggered and edge-triggered interrupts is that level-triggered interrupts are continuously active as long as the condition that generated the interrupt is present, while edge-triggered interrupts are active only when the interrupt signal changes state.

In summary, level-triggered interrupts are useful when the condition generating the interrupt is continuous and long-lasting, while edge-triggered interrupts are useful when the condition generating the interrupt is short-lived or occurs only once.