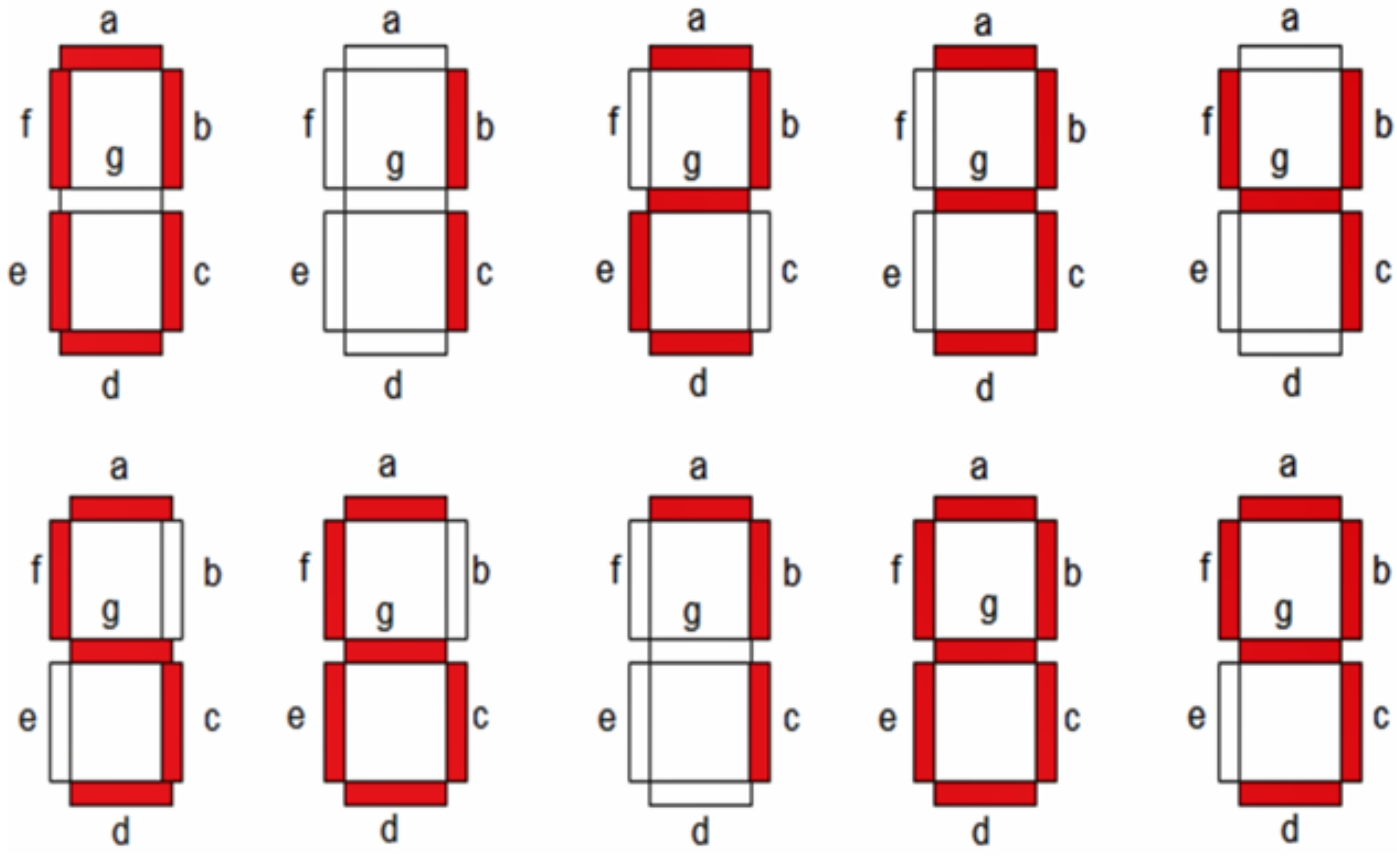
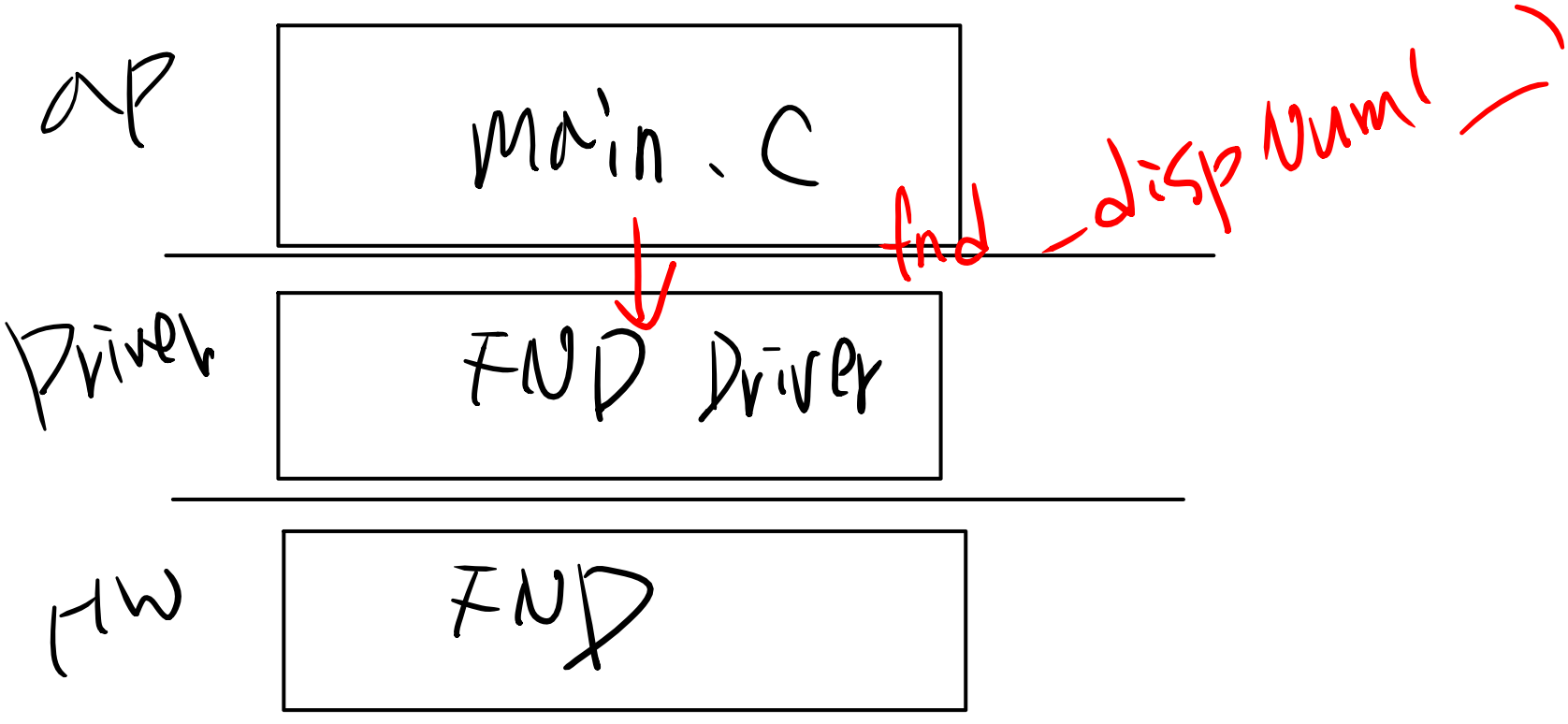
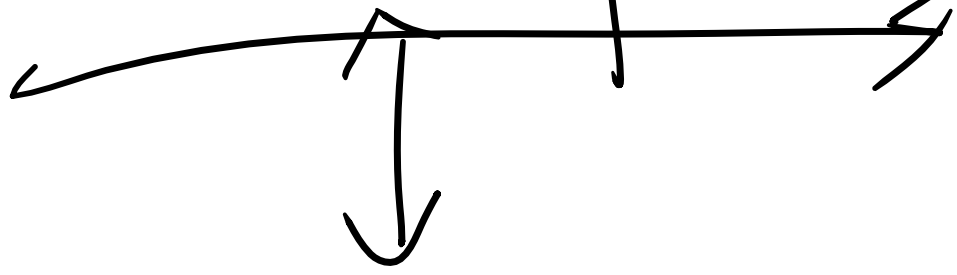

a
f
b
g
e
c
d


AP
main.c
fnd_dispNum(
Driver
FND Driver
HW
FND

주기적으로 FND-display() 함수를

수행하게끔 하겠다.

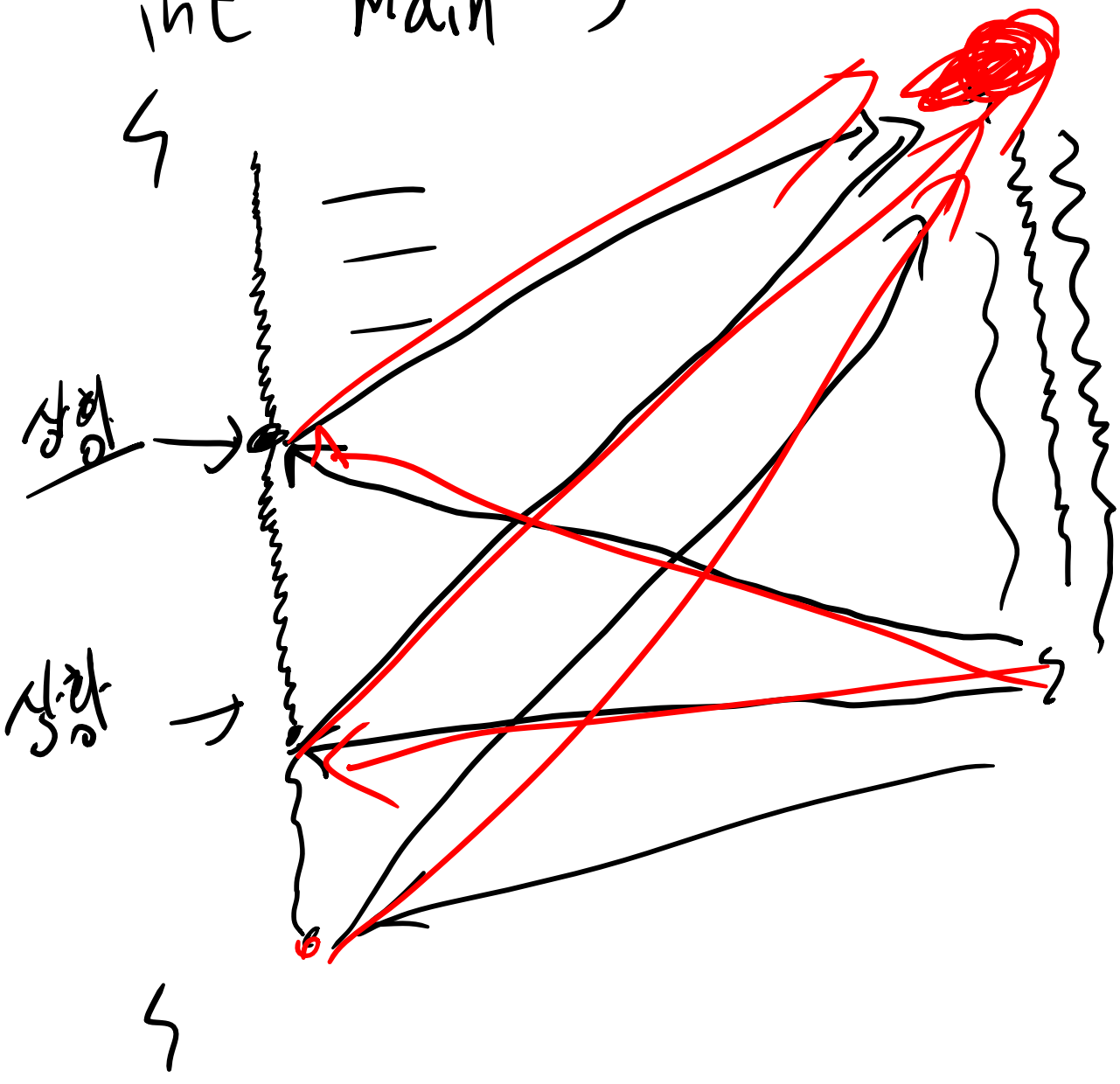
⇒ Interrupt



polling

int main( )

ISR( )



Interrupt, 중지(?) 발생조건

① Timer / Counter

② 외부 인터럽트

③ 내부 interface 하나

ATmega128A MCU

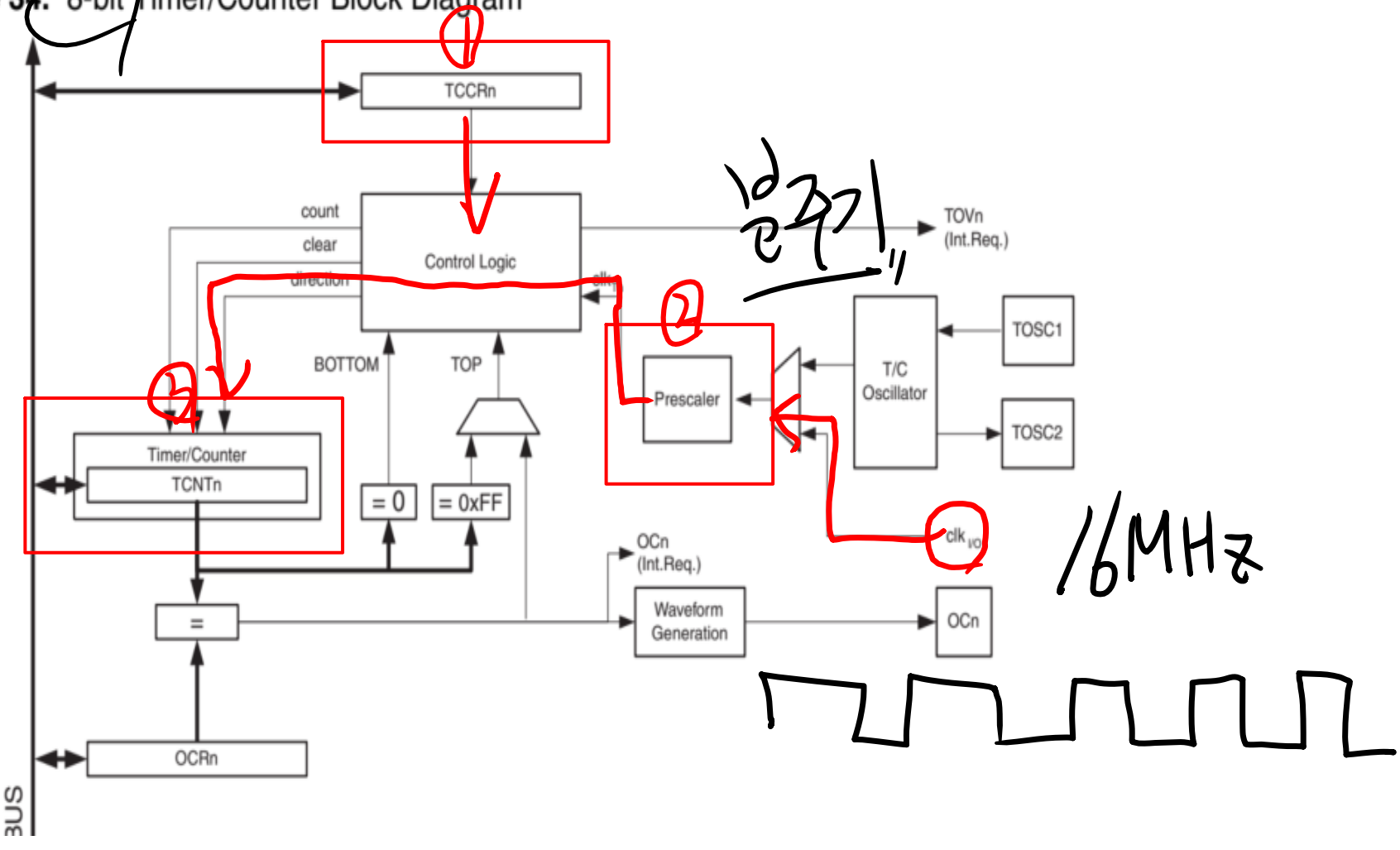
총 4EA Timer/Counter

0, 1, 2, 3.

- 0, 2 : 8bit
- 1, 3 : 16bit

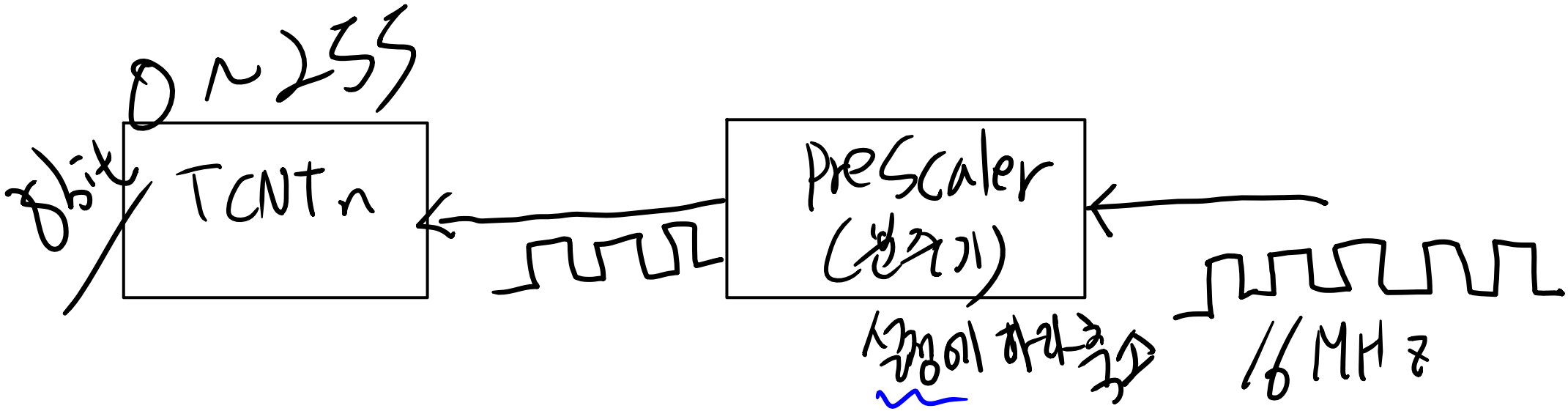
Ø.2 해당

Figure 34. 8-bit Timer/Counter Block Diagram



분주기

16MHz

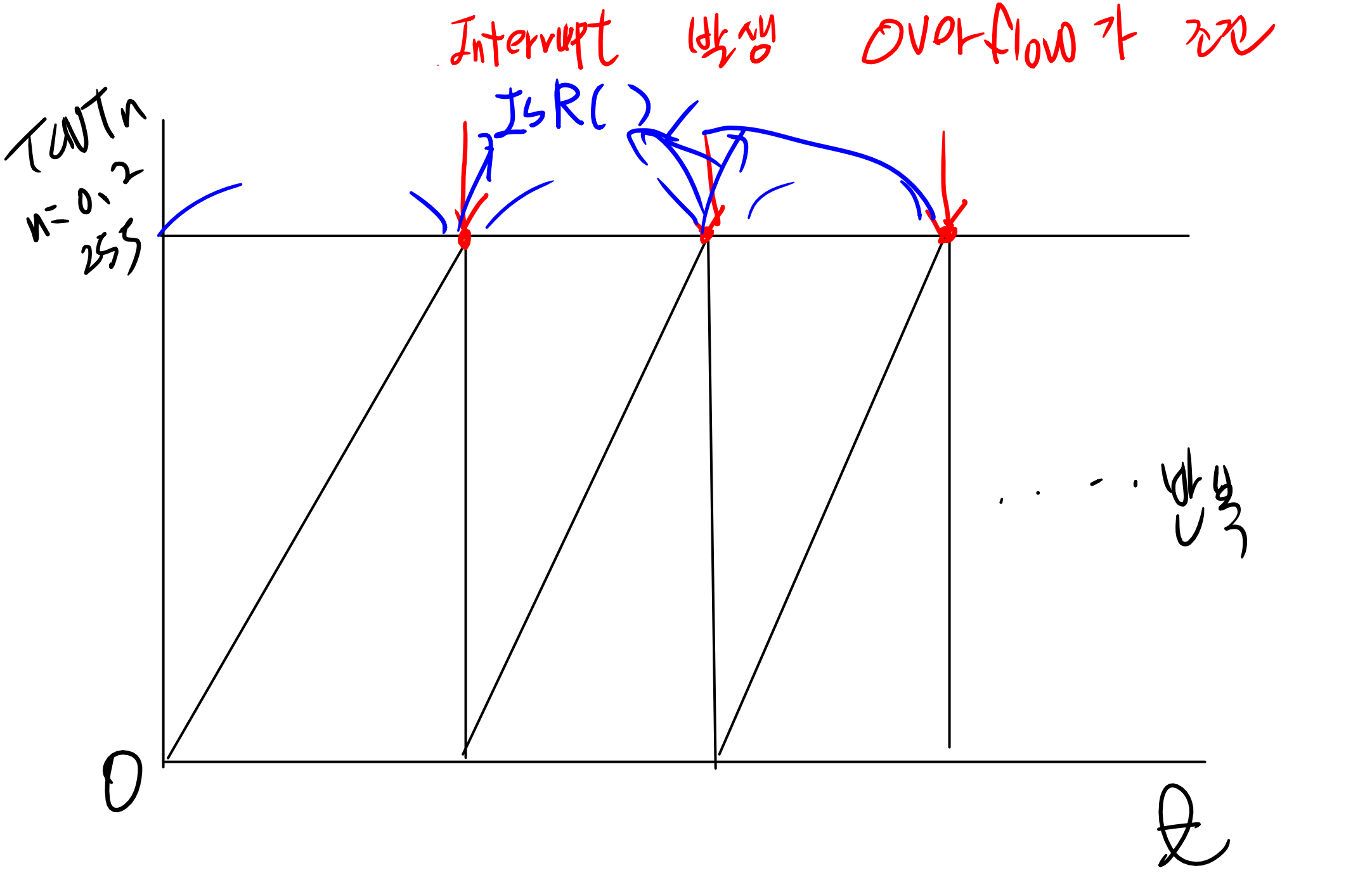


설정에 따라 다름

1/64

Interrup 발생 주기는?

1.024 ms


Interrupt 발생 overflow 가 될
ISR( )
TCNTn
n=0, 2
255
0
t

Timer/Counter Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

5b.

Table 56. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk_{T0S}/(No prescaling)
0	1	0	clk_{T0S}/8 (From prescaler)
0	1	1	clk_{T0S}/32 (From prescaler)
1	0	0	clk_{T0S}/64 (From prescaler)
1	0	1	clk_{T0S}/128 (From prescaler)
1	1	0	clk_{T0S}/256 (From prescaler)
1	1	1	clk_{T0S}/1024 (From prescaler)

Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer/Counter n을 사용하려면 한다면

조건: overflow Interrupt

① prescaler 설정

② Time/Counter Enable 설정 (TIMSK)

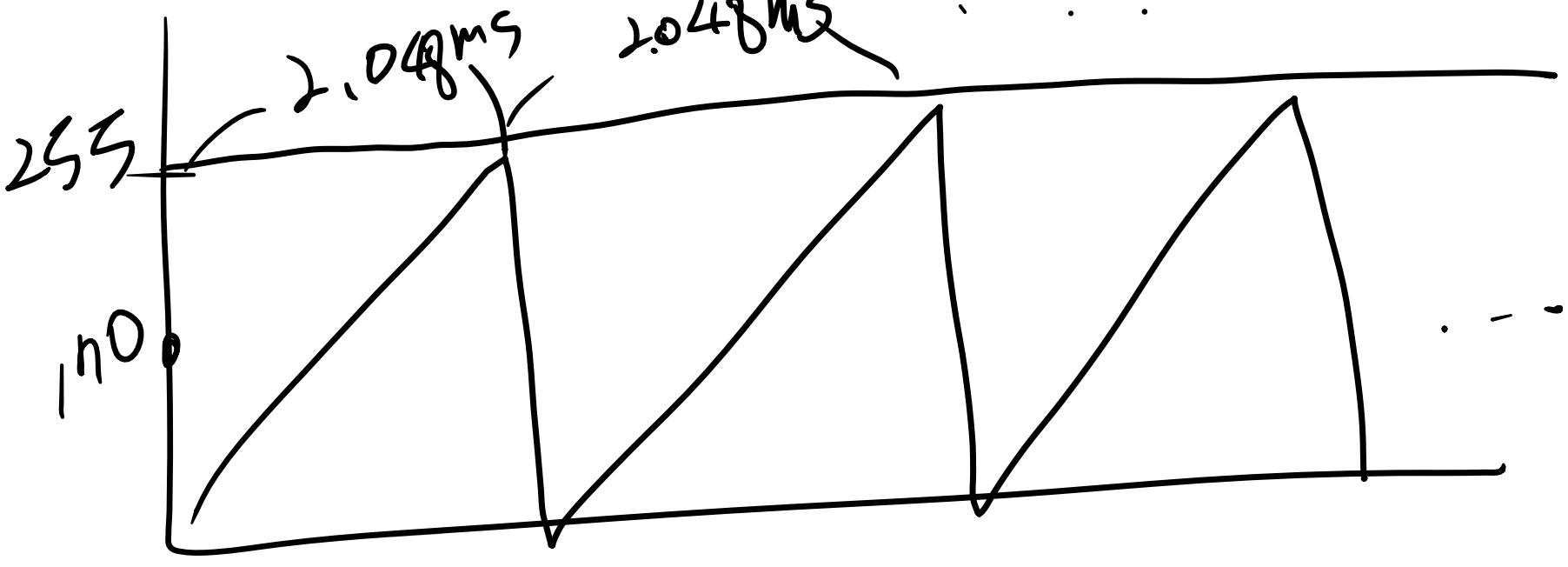
③ Global Interrupt Enable 설정

④ ISR( )

Interrup 주기 계산

내가 정확히 1ms 마다 Interrupt를 발생

조건: prescaler 1/128

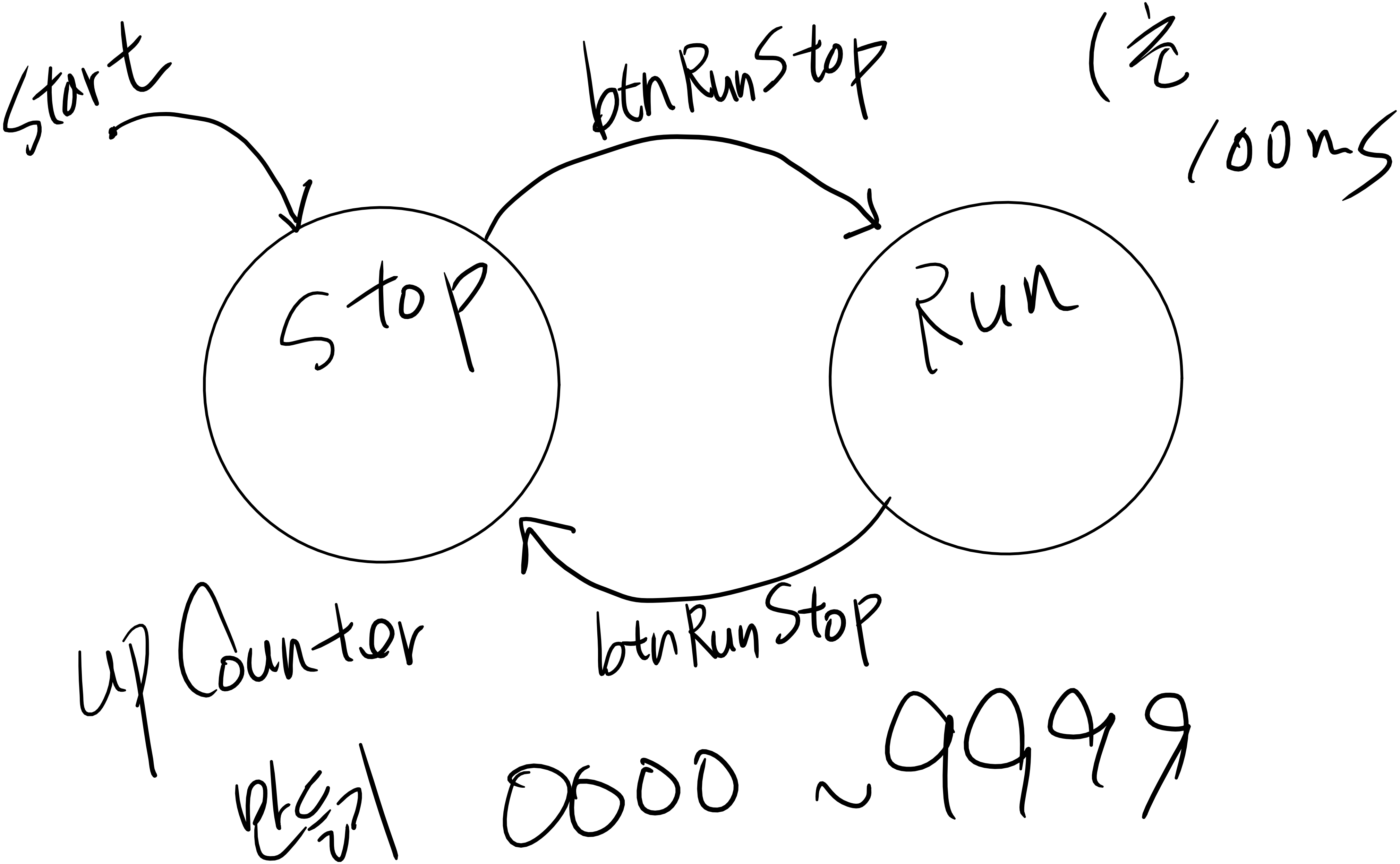


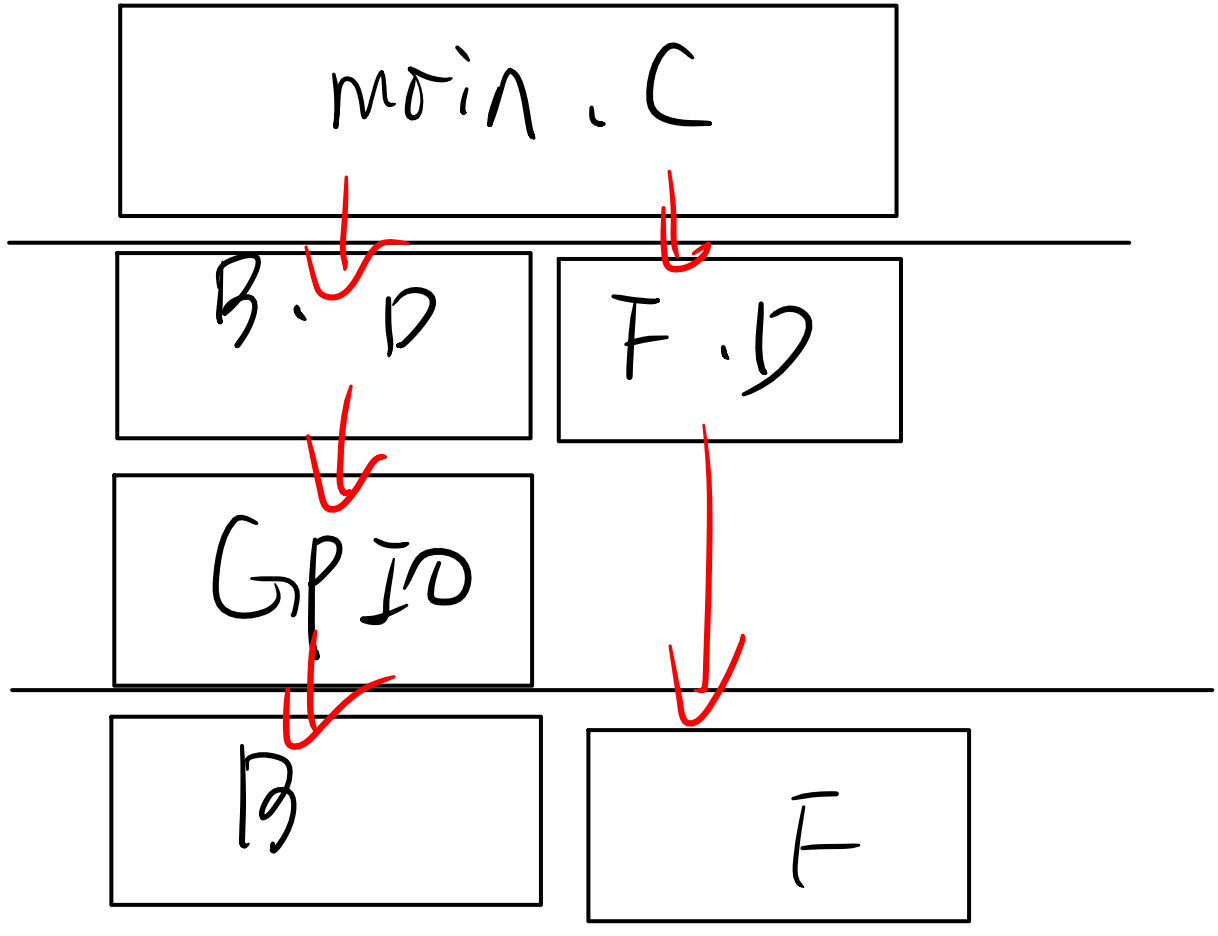
main.c

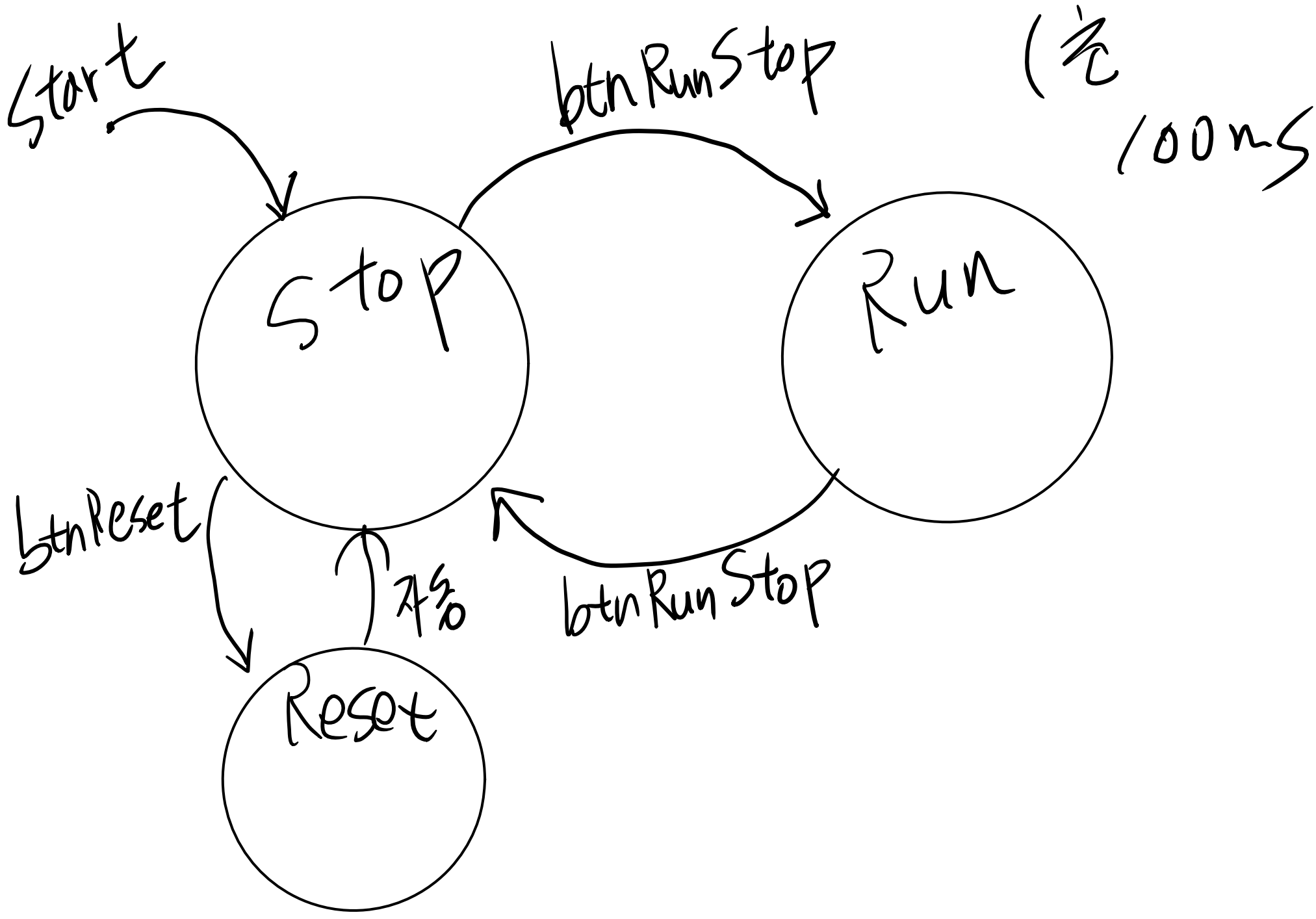
```
Counter = 5123;
fnd_set_fndData(Counter)
```

FND.C

fndData


Start
btnRunStop
100ms
Stop
Run
btnRunStop
up Counter
0000 ~ 9999


main.c
B.D
F.D
GPIO
B
F


Start
btnRunStop
Stop
Run
btnReset
btnRunStop
Reset