Line Tracer 09

- Tachometer -

This lecture is based on

- Timers
- Tachomter

1. Tachomter

What is Tachometer?

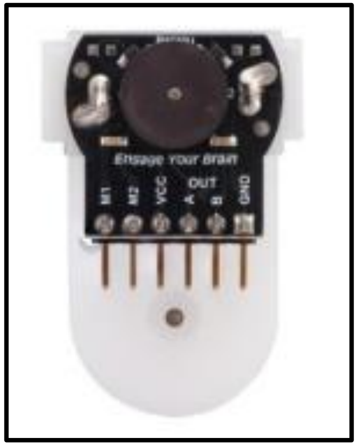
Definition

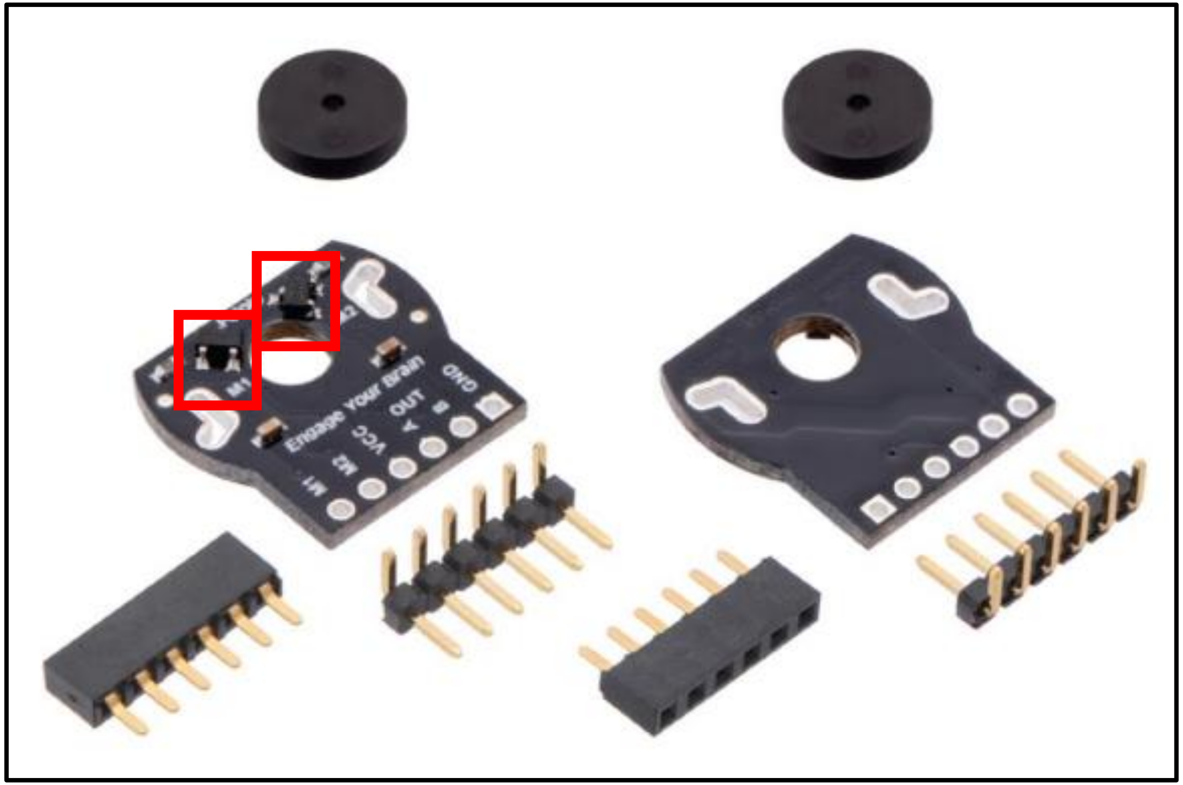
“ A tachometer is an instrument measuring the rotation of a shaft or disk”

What can we do with a tachometer?

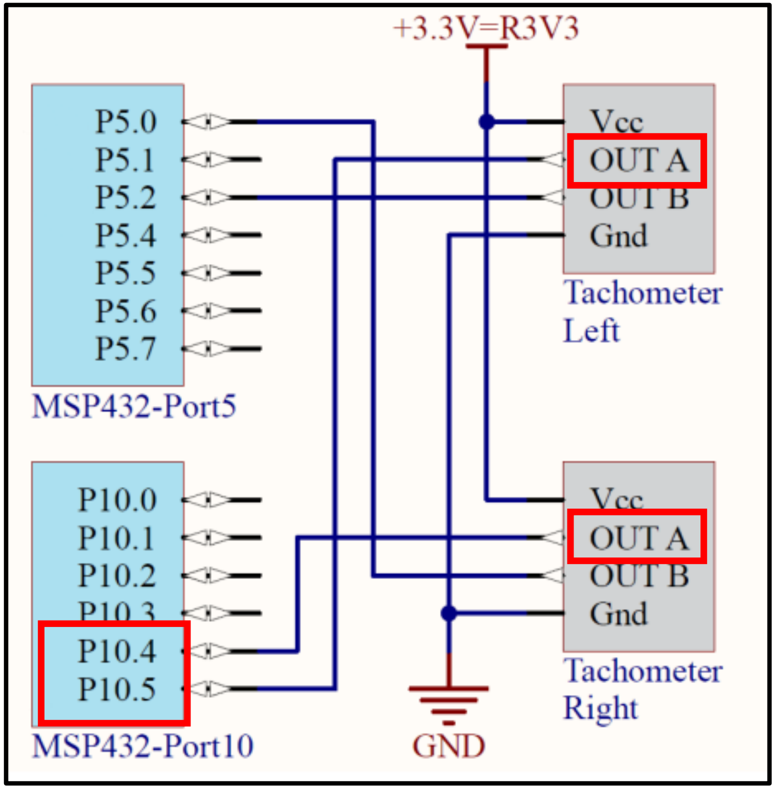
- We can measure RPM (rotations per minute) of the robot

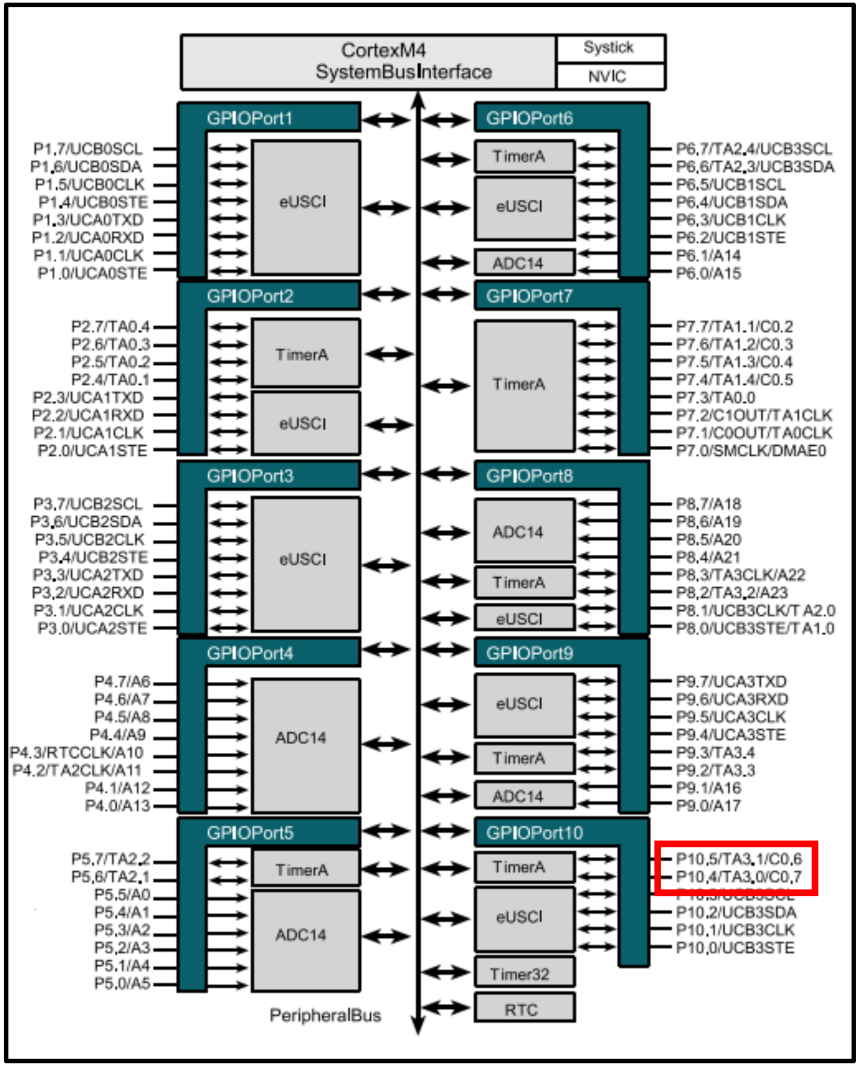
TI-RSLK Tachometer



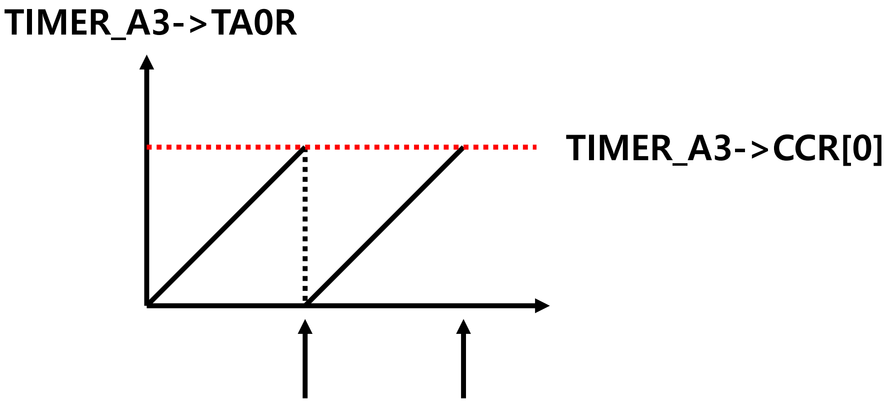


TI-RSLK Tachometer





Timer A Input Capture

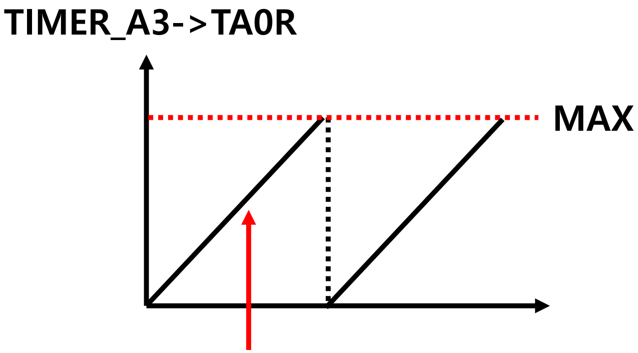


TIMER_A3->TA0R

TIMER_A3->CCR[0]

Interrupt Occurs

Output Compare



TIMER_A3->TA0R

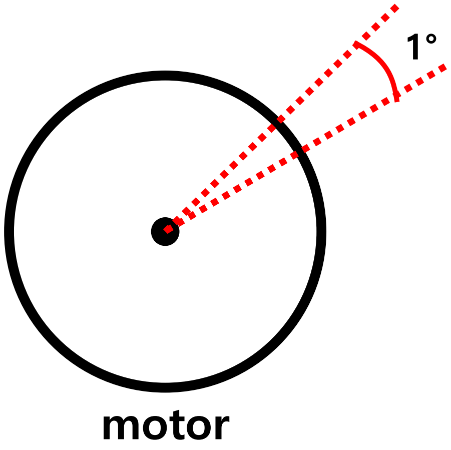
MAX

When an external event occurs,
TIMER_A3->CCR[0] = TIMER_A3->TA0R &
Timer interrupt occurs

Input Capture

When an event occurs, the counter value is stored in the CCR!

Tachometer & Timer’s Input Capture



Timer Interrupt Occurs

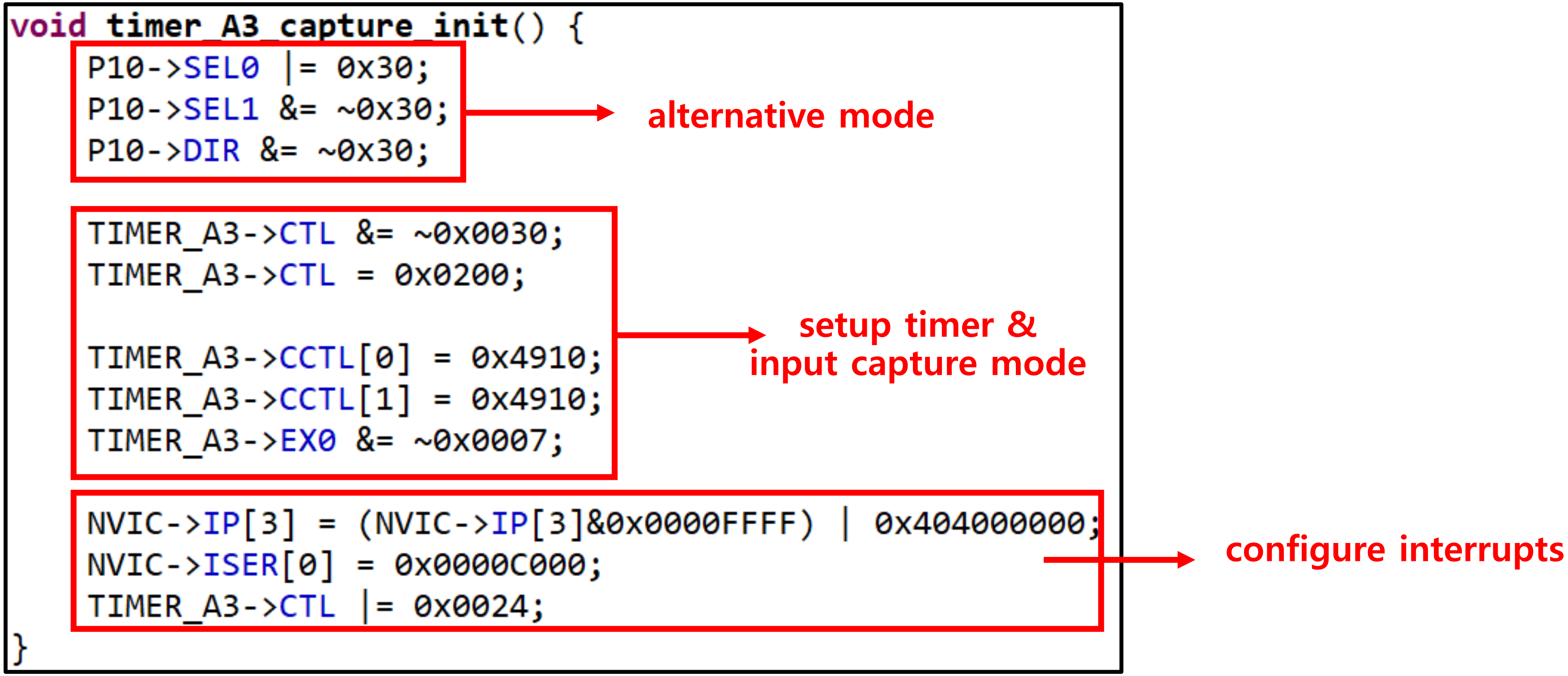
CCR = TA0R

Period
= Previous CCR – Current CCR

Timer Interrupt Handler

2. Implementation

Initialization



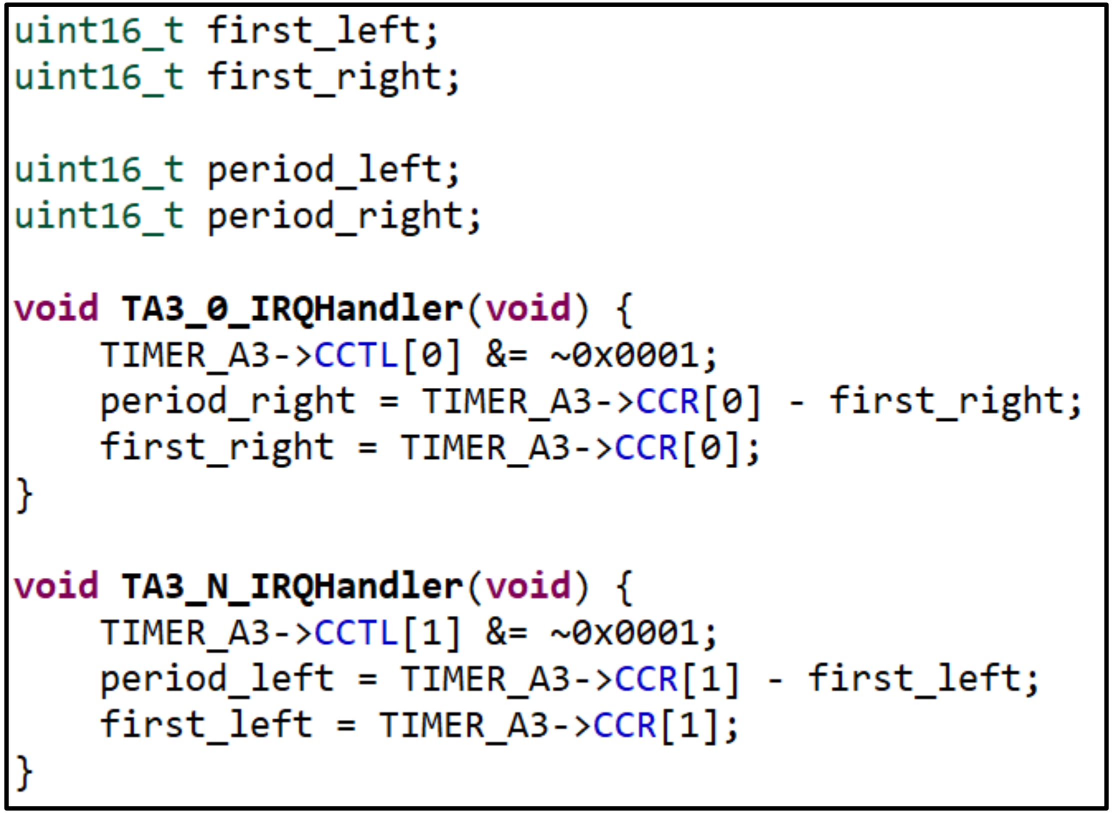
```
void timer_A3_capture_init() {
    P10->SEL0 |= 0x30;
    P10->SEL1 &= ~0x30;
    P10->DIR &= ~0x30;

    TIMER_A3->CTL &= ~0x0030;
    TIMER_A3->CTL = 0x0200;

    TIMER_A3->CCTL[0] = 0x4910;
    TIMER_A3->CCTL[1] = 0x4910;
    TIMER_A3->EX0 &= ~0x0007;

    NVIC->IP[3] = (NVIC->IP[3]&0x0000FFFF) | 0x40400000;
    NVIC->ISER[0] = 0x0000C000;
    TIMER_A3->CTL |= 0x0024;
}
```

alternative mode

setup timer &
input capture mode

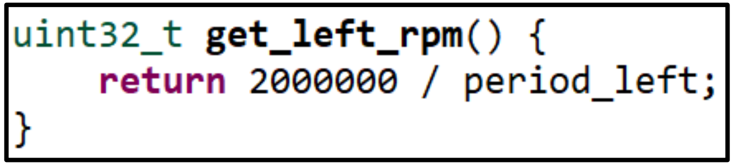
configure interrupts

Interrupt Handler



```
uint16_t first_left;
uint16_t first_right;

uint16_t period_left;
uint16_t period_right;

void TA3_0_IRQHandler(void) {
    TIMER_A3->CCTL[0] &= ~0x0001;
    period_right = TIMER_A3->CCR[0] - first_right;
    first_right = TIMER_A3->CCR[0];
}

void TA3_N_IRQHandler(void) {
    TIMER_A3->CCTL[1] &= ~0x0001;
    period_left = TIMER_A3->CCR[1] - first_left;
    first_left = TIMER_A3->CCR[1];
}
```

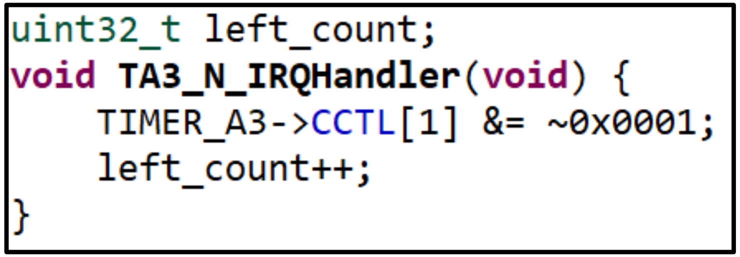
Measure RPM



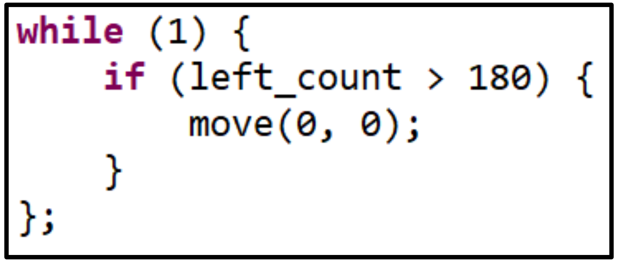
```
uint32_t get_left_rpm() {
    return 2000000 / period_left;
}
```

$$RPM = \frac{1°}{360°} * \frac{1}{period} * \frac{60s}{1m} * \frac{1}{timer\ cycle} * \frac{1{,}000{,}000{,}000ns}{s}$$

180° Wheel Rotation



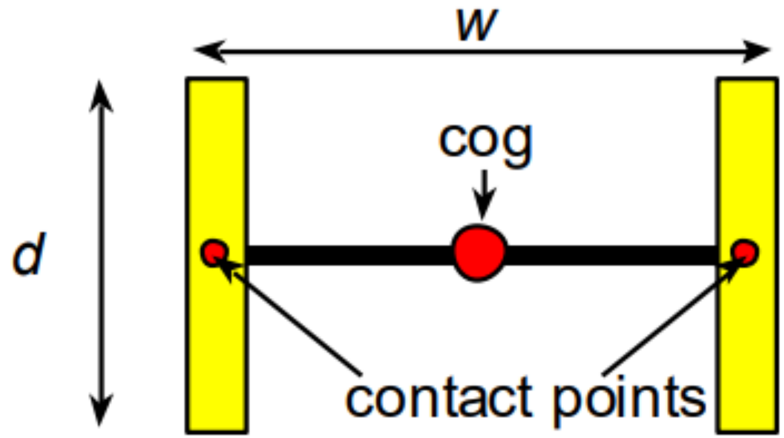
```
uint32_t left_count;
void TA3_N_IRQHandler(void) {
    TIMER_A3->CCTL[1] &= ~0x0001;
    left_count++;
}
```



```
while (1) {
    if (left_count > 180) {
        move(0, 0);
    }
};
```

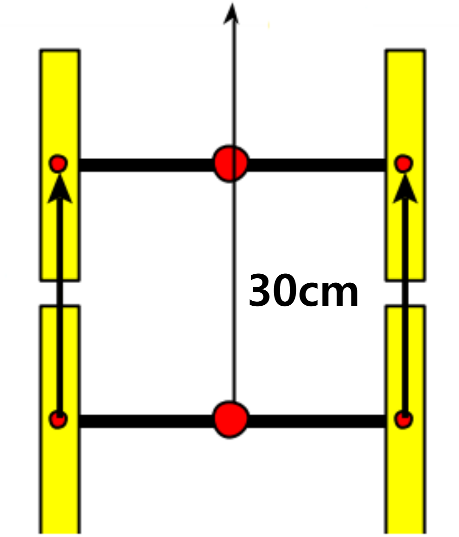
3. Activity
Please Upload LMS

Robot Spec



d = 7cm
w = 14cm

1. Go Straight 30cm



2. Rotate 30°(Assignments)