

03

진행상황 및 문제점

정유경 : eQEP 이론 및 소스코드 구현

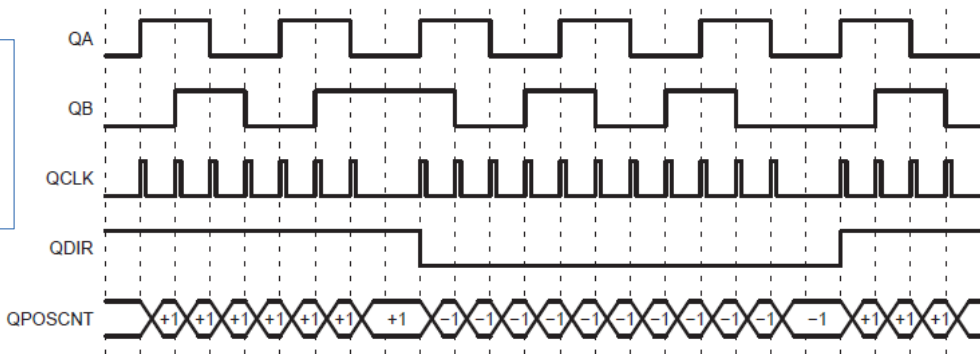
eQEP란?

- MCU는 2개의 eQEP모듈을 지원한다.
- 모터제어에 활용되는 회로로 프로젝트에서는 초기 위치로부터의 상대 위치 이동 정보를 알려주는 Incremental 방식의 로터리 엔코더 출력신호를 이용한다.
- 외부 엔코더에서 Quadrature Encoder Pulse인 A, B를 입력 받아서, QCLK, QDIR로 디코드한다.
- eQEP 모듈이 내장한 타이머 카운터로 펄스 수를 세어, 현재 모터의 위치와 방향, 속도를 알 수 있다.

Quadrature Encoder는 2개의 엔코더 신호가 서로 90도 위상 차를 가진다. eQEP가 Quadrature 모드로 동작할 때 각 엔코더 신호의 각 상승/하강에지에서 QCLK이 생성되어 4체배가 된다.

홀 센서를 이용하여 회전 수, 이동거리, 속도를 구할 수 있다. 두 개의 홀 센서를 이용하여 모터의 회전 방향을 알 수 있다.

Figure 34-7. Quadrature-clock and Direction Decoding



측정방법

- 속도측정시 저속, 고속에 따라 측정방법이 다르다. (68: 고속, 69:저속)

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity may be written as:

$$v(k) \approx \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad (68)$$

$$v(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad (69)$$

where

$v(k)$: Velocity at time instant k

$x(k)$: Position at time instant k

$x(k-1)$: Position at time instant $k-1$

T : Fixed unit time or inverse of velocity calculation rate

ΔX : Incremental position movement in unit time

$t(k)$: Time instant " k "

$t(k-1)$: Time instant " $k-1$ "

X : Fixed unit position

ΔT : Incremental time elapsed for unit position movement.

68: 정해진 샘플링 주기 내에 몇 개의 펄스가 포함되는지 확인한다.
저속상태에서 오차가 발생한다.

69: 펄스 간 시간을 계산하여 속도를 측정한다.
(QCAP 이용). 저속에 유리

ENCODER PART LEAD WIRE 색상별 역할

NO	COLOR	역할
1	RED	+5V
2	YELLOW	A상출력
3	BLUE	GROUND
4	WHITE	B상출력

eQEP의 입력

Quadrature-clock mode / Direction-count mode

1. QEPA/XCLK 과 QEPB/XDIR

- Quadrature-clock mode일 때, 엔코더는 90도의 위상차이가 나는 두 개의 구형파 신호를 제공한다. 위상 관계는 입력 샤프트의 회전 방향을 결정하는데 사용, 인덱스 위치로부터의 eQEP 펄스의 수는 상대 위치 정보를 유도하기 위해 사용된다.
- Direction-count mode 일때 QEPA 핀은 클럭 입력을 제공하고 QEPB 핀은 방향 입력을 제공한다(일부 엔코더)

2. eQEPI : 인덱스 또는 제로 마커

eQEP 인코더의 인덱스 출력에 연결되어 각 회전마다 위치 카운터를 리셋한다.

3. QEPS : 스트로브 입력

범용적으로 쓰인다. 스트로브 핀에서 원하는 이벤트 발생(ex. 모터가 정의 된 위치에 도달) 시 위치 카운터를 초기화하거나 래치할 수 있다.

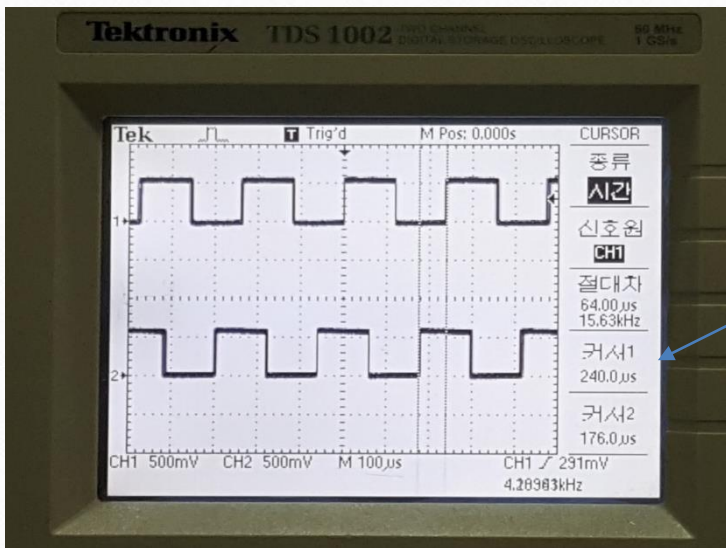
Encoder Spec 분석

- 증분형, 분해능: 432P/R (P/R : 1회전 당 펄스 수, 라인 수)

ex. 실제로는 Quadrature mode 일때, $432 \times 4 = 1728$, $360/1728 = 0.25$ 도의 분해능

ex. 분당 2800rpm으로 작동하는 모터에 연결되는 432라인의 엔코더는 $432 \times 2800/60 = 20.160\text{kHz}$ 의 주파수를 발생시킨다. 따라서 QEPA 또는 QEPB 출력의 주파수를 측정하여 모터의 속도를 결정

ex. 인덱스 펄스(1회전시 발생하는 펄스)는 $2800/60 = 46.67\text{kHz}$



모터에 인가하는 전압을 증가시키면, 펄스 폭 증가

If 모터에 5V 인가

$240\mu\text{s} \rightarrow 4.17\text{kHz} = (\text{분당회전수})\text{rpm} / 60 \times 432$
따라서 (분당회전수) = 578rpm

eQEP의 디코더와 상태도

Figure 34-6. Quadrature Decoder State Machine

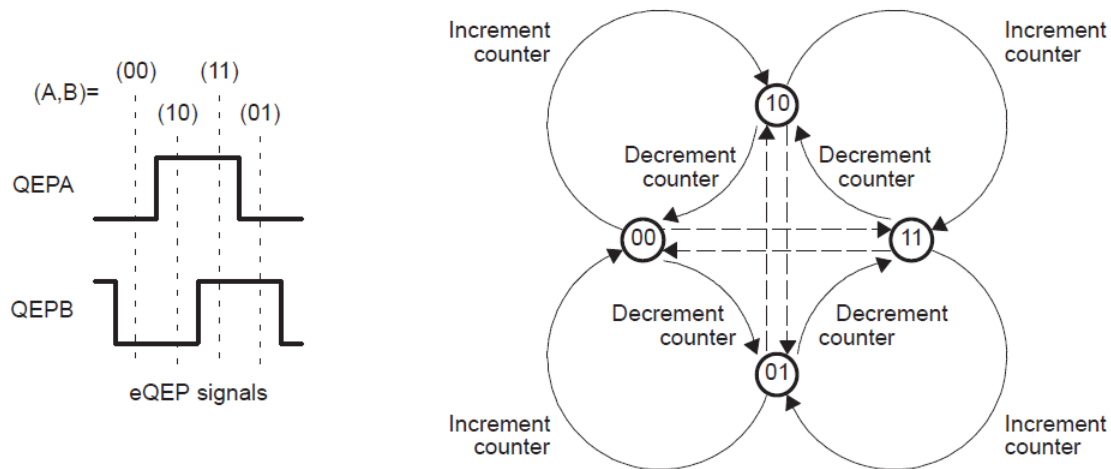
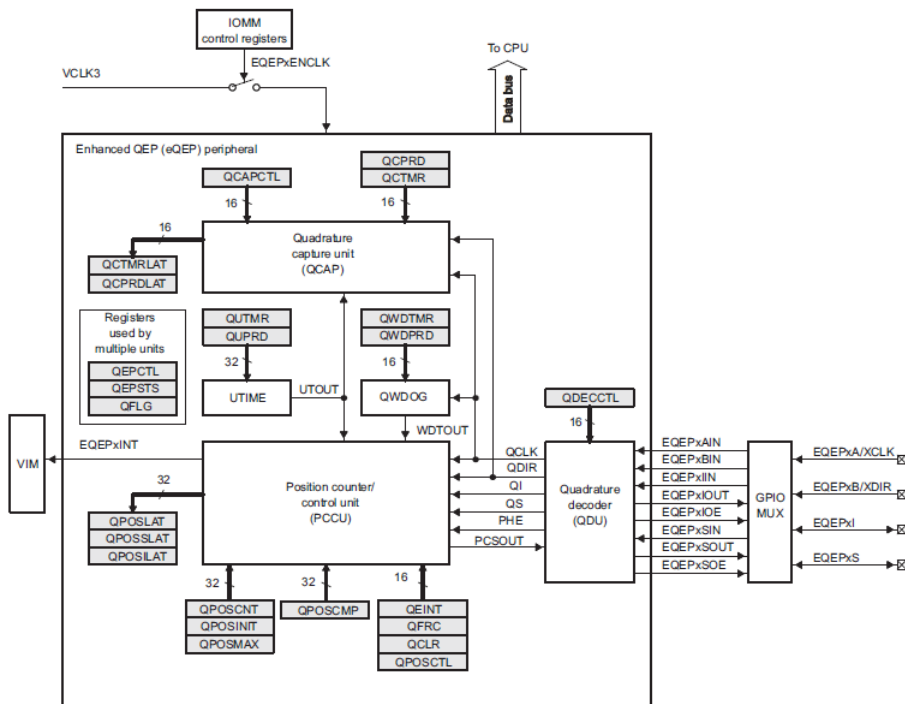


Figure 34-4. Functional Block Diagram of the eQEP Peripheral



- 직교 니코더 장치 (QDU)
- 위치 카운터 및 위치 측정 용 제어 장치 (PCCU)
- 저속 측정 (QCAP)을 위한 직교 예지 캡처 유닛
- 속도 / 주파수 측정을 위한 단위 시간 베이스 (UTIME)
- 스тол감지를 위한 위치 독 타이머 (QWDOG)

코드 구현

1. 하드웨어 구성

[V9, W9] EQEPxA, EQEPxB

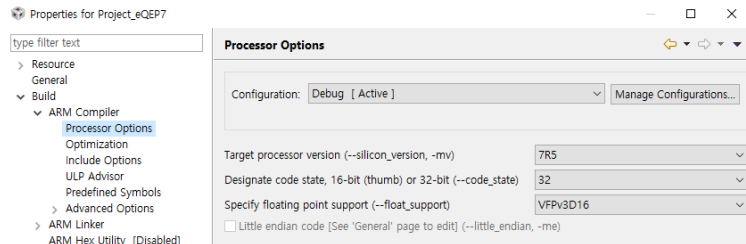
[V10] EQEPI

[F3] EQEPS

2. Halcogen 설정

eQEP Driver enable하고 Pinmux 설정한 후 eQEPA를 다음과 같이 설정한다.

CCS에서 build시 VFP instruction required 에러 해결방법



EQEP1 EQEP2

General Configuration

Position Counter Mode: QUADRATURE_COUNT

External clock rate: RESOLUTION_1x

Select QDIR: CLOCKWISE

☐ Invert QEPxA Polarity

☐ Invert QEPxB Polarity

☐ Invert QEPx Polarity

☐ Gate Index Pin with Strobe

☐ Swap Quadrature Clock Input

Compare Output Configurations

Sync Output Pin Select: INDEX_PIN

Shadow Load Mode: QPOSCNT_EQ_QPSCMP

Sync Output Polarity: ACTIVE_HIGH

☐ Enable Sync Output

☐ Enable Position Compare Shadow

Compare Value: 0x00000000

Sync Pulse Width: 0x00 x 4 VCLK4

Position Counter Configuration

Counter Init Index Event: RISING_EDGE

Counter Init Strobe Event: DIRECTION_DEPENDENT

Position Counter Reset On: MAX_POSITION

Counter Latch Index Event: RISING_EDGE

Counter Latch Strobe Event: RISING_EDGE

Max Position Count: 0x00000000

☐ Init Counter on Index Event

☐ Init Counter on Strobe Event

☐ Enable SW Initialization

Init Position Count to: 0x00000000

Interrupt Configuration

☒ Position counter error Interrupt

☒ Quadrature phase error Interrupt

☒ Quadrature direction change Interrupt

☐ Watchdog time out Interrupt

☒ Position counter underflow Interrupt

☒ Position counter overflow Interrupt

☐ Position-compare ready Interrupt

☐ Position-compare match Interrupt

☒ Strobe event latch Interrupt

☒ Index event latch Interrupt

☐ Unit time out interrupt

Capture Configuration

Capture Timer Prescaler: PS_8

Unit Pos Event Prescaler: PS_512

Cap Timer Pos Mode: ON_POSITION_COUNTER_READ

☐ Init Counter on Strobe Event

Unit Init Period: 0x00000000

Watchdog Configuration

Watchdog Timer Value: 0x0000


```
#include <HL_eqep.h>
#include <HL_hal_stdtypes.h>
#include <HL_reg_eqep.h>
#include <HL_sys_core.h>
#include <stdio.h>

#define UNIT_POSITION_X 60U

int main(void)
{
    uint16 deltaT = 0U;
    float velocity = 0U;
    /* EQEP initialization based on GUI Configuration. */
    QEPIInit();

    /* Enable Position Counter */
    eqepEnableCounter(eqepREG1);

    /* Enable Unit Timer. */
    eqepEnableUnitTimer(eqepREG1);

    /* Enable capture timer and capture period latch. */
    eqepEnableCapture(eqepREG1);

    while(1)
    {
        /* Status flag is set to indicate that a new value is latched in the QCPRD register. */
        if((eqepREG1->QEPSTS & 0x80U) != 0U)
        {
            /* Elapsed time between unit position events */
            deltaT = eqepREG1->QCPRD;

            velocity = (float)(UNIT_POSITION_X/deltaT);

            /* Clear the Status flag. */
            eqepREG1->QEPSTS |= 0x80U;
            printf("deltaT: %d\n", deltaT);
            printf("velocity: %f\n", velocity*1000000000000);
        }
    }
}
```

```
[CortexR5] deltaT: 522
velocity: 0.000000
deltaT: 14078
velocity: 0.000000
deltaT: 10048
velocity: 0.000000
deltaT: 11147
velocity: 0.000000
deltaT: 9576
velocity: 0.000000
deltaT: 12005
velocity: 0.000000
deltaT: 13427
velocity: 0.000000
deltaT: 11082
velocity: 0.000000
deltaT: 12894
velocity: 0.000000
deltaT: 11382
velocity: 0.000000
deltaT: 9027
velocity: 0.000000
deltaT: 16033
velocity: 0.000000
deltaT: 13585
velocity: 0.000000
deltaT: 14656
velocity: 0.000000
deltaT: 19876
velocity: 0.000000
deltaT: 13423
velocity: 0.000000
deltaT: 9148
velocity: 0.000000
deltaT: 9913
velocity: 0.000000
deltaT: 9780
velocity: 0.000000
```

문제점

1. eQEP모듈의 각 레지스터에 대한 이해 필요
2. eQEPI, eQEPS의 기능에 대한 이해 필요
3. velocity 계산과정에서 에러발생원인 분석 필요

*. 540rpm = 3m/s lidar 속도

04 진행계획

진행계획

1. eQEP 모듈의 기능분석 및 코드구현
 2. MCU 주변회로
-