eQEP TEST
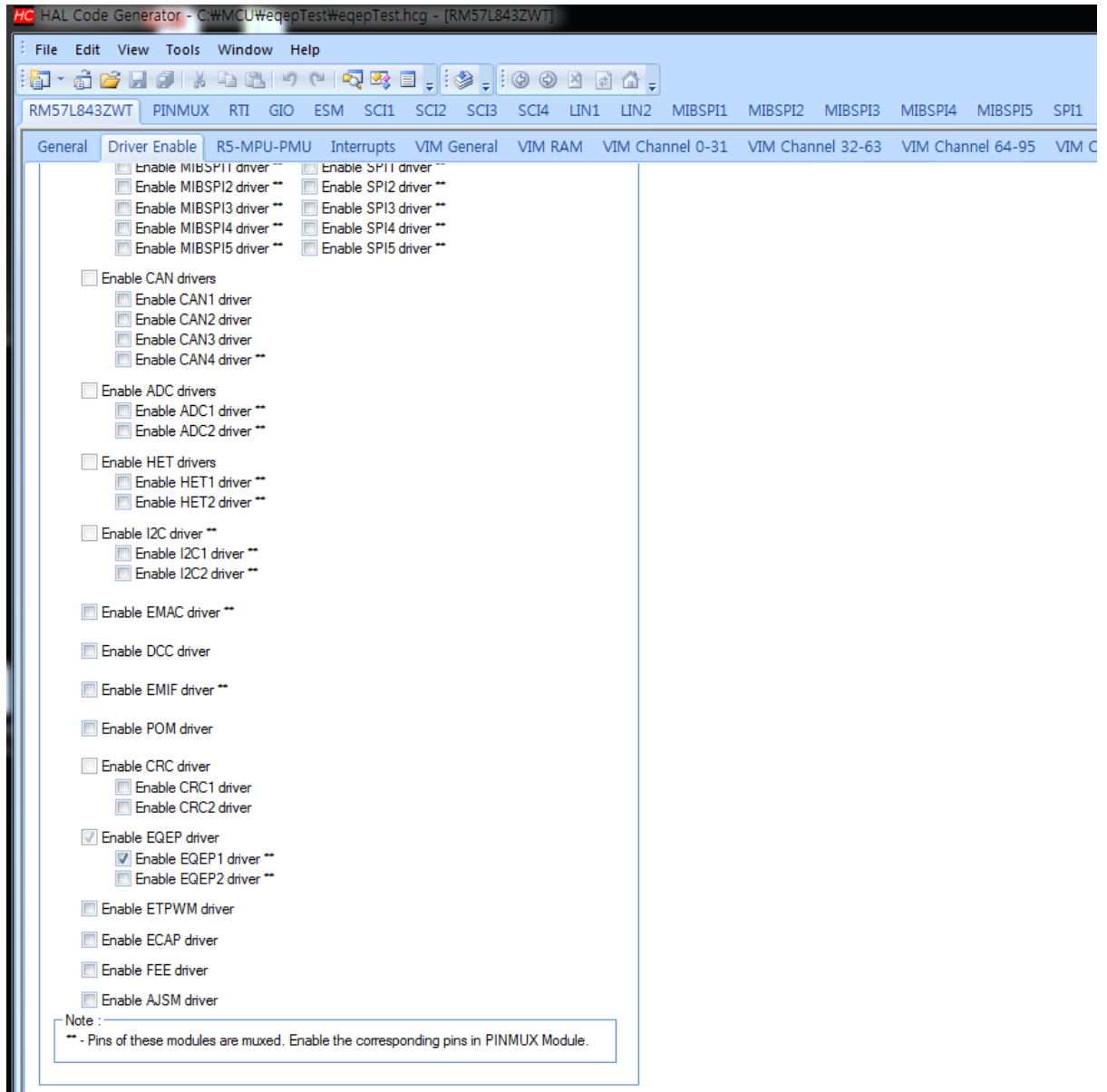
1. HALCOGEN

File   Edit   View   Tools   Window   Help

RM57L843ZWT   PINMUX   RTI   GIO   ESM   SCI1   SCI2   SCI3   SCI4   LIN1   LIN2   MIBSPI1   MIBSPI2   MIBSPI3   MIBSPI4   MIBSPI5   SPI1

General   Driver Enable   R5-MPU-PMU   Interrupts   VIM General   VIM RAM   VIM Channel 0-31   VIM Channel 32-63   VIM Channel 64-95   VIM C

☐ Enable MIBSPI1 driver **   ☐ Enable SPI1 driver **
☐ Enable MIBSPI2 driver **   ☐ Enable SPI2 driver **
☐ Enable MIBSPI3 driver **   ☐ Enable SPI3 driver **
☐ Enable MIBSPI4 driver **   ☐ Enable SPI4 driver **
☐ Enable MIBSPI5 driver **   ☐ Enable SPI5 driver **

☐ Enable CAN drivers
   ☐ Enable CAN1 driver
   ☐ Enable CAN2 driver
   ☐ Enable CAN3 driver
   ☐ Enable CAN4 driver **

☐ Enable ADC drivers
   ☐ Enable ADC1 driver **
   ☐ Enable ADC2 driver **

☐ Enable HET drivers
   ☐ Enable HET1 driver **
   ☐ Enable HET2 driver **

☐ Enable I2C driver **
   ☐ Enable I2C1 driver **
   ☐ Enable I2C2 driver **

☐ Enable EMAC driver **

☐ Enable DCC driver

☐ Enable EMIF driver **

☐ Enable POM driver

☐ Enable CRC driver
   ☐ Enable CRC1 driver
   ☐ Enable CRC2 driver

☑ Enable EQEP driver
   ☑ Enable EQEP1 driver **
   ☐ Enable EQEP2 driver **

☐ Enable ETPWM driver

☐ Enable ECAP driver

☐ Enable FEE driver

☐ Enable AJSM driver

Note :
** - Pins of these modules are muxed. Enable the corresponding pins in PINMUX Module.

File Edit View Tools Window Help

PI1 SPI2 SPI3 SPI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 HET2 I2C1 I2C2 EMAC DCC RTP DMM EMIF POM CRC ETPWM ECAP EQEP FEE AJSM

EQEP1 | EQEP2

General Configuration

Position Counter Mode: QUADRATURE_COUNT

External clock rate: RESOLUTION_2x

Select QDIR: CLOCKWISE

☐ Invert QEPxA Polarity
☐ Invert QEPxB Polarity
☐ Invert QEPxI Polarity
☐ Invert QEPxS 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 Shado

Compare Value: 0x00000000

Sync Pulse Width: 0x000 x 4 VCLK4

Position Counter Configuration

Counter Init Index Event: RISING_EDGE

Counter Init Strobe Event: DIRECTON_DEPENDENT

Position Counter Reset On: MAX_POSITION

Counter Latch Index Event: RISING_EDGE

Counter Latch Strobe Event: RISING_EDGE

Max Position Count: 0xFFFFFFFF

☐ Init Counter on Index Event
☐ Init Counter on Strobe Event
☐ Enable SW Inititialization

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_1

Unit Pos Event Prescaler: PS_1

Cap Timer Pos Mode: ON_UNIT_TIMOUT_EVENT

☐ Init Counter on Strobe Event

Unit Init Period: 0x00000FFF

Watchdog Configuration

Watchdog Timer Value: 0x0000

2. CCS

```c
16 /*  @fn  void main(void)
17 *    @brief Application main function
18 *    @note This function is empty by default.
19 *
20 *    This function is called after startup.
21 *    The user can use this function to implement the application.
22 */
23
24 /* USER CODE BEGIN (2) */
25 #define UNIT_POSITION_X 60U
26 /* USER CODE END */
27
28 void main(void)
29 {
30 /* USER CODE BEGIN (3) */
31
32     uint16 deltaT = 0U;
33     float velocity = 0U;
34     /* EQEP initialization based on GUI Configuration. */
35     QEPInit();
36
37     /* Enable Position Counter */
38     eqepEnableCounter(eqepREG1);
39
40     /* Enable Unit Timer. */
41     eqepEnableUnitTimer(eqepREG1);
42
43     /* Enable capture timer and capture period latch. */
44     eqepEnableCapture(eqepREG1);
45
46     while(1)
47     {
48         /* Status flag is set to indicate that a new value is latched in the QCPRD register. */
49         if((eqepREG1->QEPSTS & 0x80U) !=0U)
50         {
51             /* Elapsed time between unit position events */
52             deltaT = eqepREG1->QCPRD;
53
54             /* Calculate Velocity from deltaT and the value of the unit position. */
55             /* The value of Unit Position is a sample value and should be changed by the User as per the actual value in the
56             velocity = (float)(UNIT_POSITION_X/deltaT);
57             printf("velocity = %f\n", velocity);
58
59             /* Clear the Status flag. */
60             eqepREG1->QEPSTS |= 0x80U;
61         }
62     }
63 /* USER CODE END */
64 }
65
66 /* USER CODE BEGIN (4) */
67 /* USER CODE END */
68
```

3. Result

손으로 느리게 돌리면 velocity가 0이 나온다. 빠르게 돌려야 값이 나온다.

느리게 도는건 제대로 인식 못하는듯 하다

중간에 velocity = 4294967296 나온건 clockwise로 해놓고 반시계로 돌리니 에러값이 나온듯 하다.

```
Console ☆                                              ▣ ▣ ▣ | ☞ ▣ ▾ ▣ ▾ ▭ ▯
eqepTest:CIO
[CortexR5] velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 15.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 6.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 15.000000
velocity = 0.000000
velocity = 0.000000
velocity = 4294967296.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 5.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 8.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
velocity = 0.000000
```