ALGORITHMS FOR DECODING QUADRATURE ENCODED PULSES

Assoc. Prof. Ph.D. Ratcho Ivanov, rmi@vmei.acad.bg
Assist. Prof. Alexander Kerezov, alexand@nsifc.ifc.pi.cnr.it
Ivan Lesichkov, lesichkov@yahoo.com

TECHNICAL UNIVERSITY OF SOFIA,
Faculty of Electronics
Department of Electronics

Summary: This paper presents some algorithms for decoding Quadrature Encoded Pulses (QEP). It discusses advantages and disadvantages, of the algorithms. Some basic principles of QEP and characteristic parameters of optical sensors, which generate QEP, are also described. In addition, some programs written in C and testing vectors are shown. Details on how to set up the DSP controller TMS320F241 to decode QEP are also presented.

Optical encoders are used to get position and speed information from a rotating machine. These sensors generate so-called Quadrature Encoded Pulses (QEP). The task of the embedded microprocessor system is to decode and count the quadrature encoded input pulses.

The Quadrature Encoded Pulses are two sequences of pulses with a variable frequency, and a fixed phase shift of a quarter of a period (90 degrees). An example for QEP is shown in fig. 1.

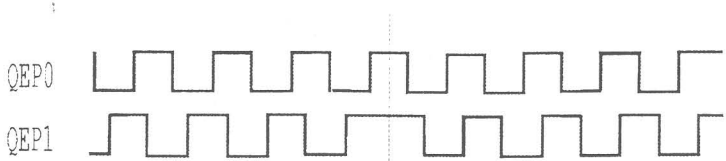

Fig. 1 An example for QEP

When an optical encoder on a motor shaft generates these pulses, the direction of rotation of the motor can be determined by detecting the leading sequence of the both pulse sets. The angular position and speed can be determined by the pulse count and pulse frequency.

Optical sensors are characterised with the following parameters, N [pulses per revolution] N = 2 000, 2 500 and s. o. If the sensor is rotating with speed equal to n [revolutions per second], it will generate $n.N$ [pulses per second]. Consequently, the signal will be with frequency fs (fs = n.N) and period Ts (Ts = 1/(n.N)).

When the object is performing a linear movement, there is an additional mechanical transfer coefficient K [meters per revolution]. So the overall transfer function of the system is $T = K/N$ [meters per pulse].

The block diagram is shown in fig.2.

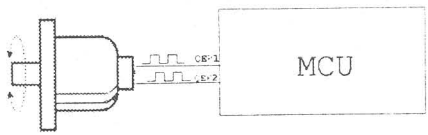

Fig. 2 Block diagram

There are various algorithms for decoding QEP. Some of them are discussed below.

First Algorithm

QEP1 is connected to one of the interrupt inputs of the microcontroller. The QEP2 is connected to one of the General Purpose Inputs.

The interrupt input is configured to make interrupt on rising (or falling) edge of the input sequences. Interrupt Service Routine is:

```
interrupt void IntX ()
{
     if (ReadBit (QEP2PortX, QEP2BitMask))
          Counter--;
     else
          Counter++;
}
```

Explanation of the algorithm is the follow:

When an interrupt is generated, the Interrupt Service Routine (ISR) checks the level of the QEP2 line. If it is high, the counter of the sensor is decremented. If it is low, the counter of the sensor is incremented.

The testing vectors are discussed below in detail.

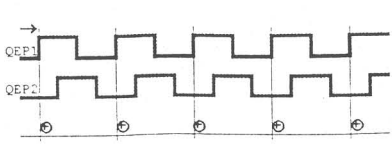

Fig. 3 When the sensor is rotated clockwise

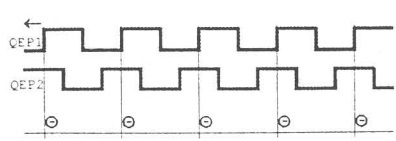

Fig. 4 When the sensor is rotated counter clockwise

Operation of the algorithm when the sensor is rotated clockwise is shown in figure 3. The Counter is incremented on every rising edge of QEP1 (because at this moment QEP2 is low).

Operation of the algorithm when the sensor is rotated counter clockwise is shown in figure 4. The Counter is decremented on every rising edge of QEP1 (because at this moment QEP2 is high).

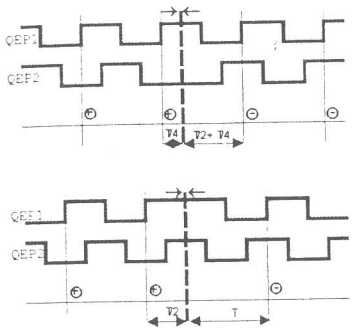

Fig. 5 Change of the direction (from 0 to T/2)

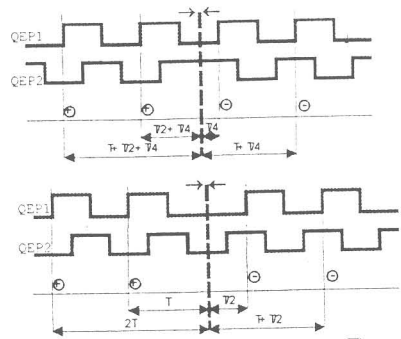

Fig. 6 Change of the direction (from T/2 to T)

Running of the algorithm when the sensor is rotated clockwise, and if the direction is changed, is shown in figure 5 and 6.

There are some special features. The change of the direction in the interval between 0 and T/2 is shown in figure 5. The absolute error in this case is:

$$\Delta X = T/4 - (T/2 + T/4) = -T/2 \quad \text{from 0 to T/4}$$
$$\Delta X = T/2 - T = -T/2 \quad \text{from T/4 to T/2}$$

Consequently, when change of the direction is in the interval between 0 and T/2, to the counter is added error $\Delta X = -T/2$.

The change of the direction in the interval between T/2 and T is shown in figure 6. The absolute error in this case is:

$$\Delta X = (T/2 + T/4) - T/4 = +T/2 \quad \text{from T/2 to 3T/4}$$
$$\Delta X = T - T/2 = +T/2 \quad \text{from 3T/4 to T}$$

Consequently, when change of the direction is in the interval between T/2 and T, to the counter is added error $\Delta X = +T/2$.

By analogy, when the sensor is rotating counter clockwise, and if the direction is changed, the error is:

$$\Delta X = (T/2 + T/4) - T/4 = +T/2 \quad \text{from } 0 \text{ to } T/4$$
$$\Delta X = T - T/2 = +T/2 \quad \text{from } T/4 \text{ to } T/2$$
$$\Delta X = T/4 - (T/2 + T/4) = -T/2 \quad \text{from } T/2 \text{ to } 3T/4$$
$$\Delta X = T/2 - T = -T/2 \quad \text{from } 3T/4 \text{ to } T$$

Second Algorithm

QEP1 is connected to one of the interrupt inputs of the microcontroller, which can make interrupt *on rising and on falling* edge of the input sequence. QEP2 is connected to one of the General Purpose Inputs.

Interrupt Service Routine is:

```
interrupt void IntX()
{
     BOOLEAN QEP1;

     QEP1 = ReadBit (QEP1PortX, QEP1BitMask);
     if (ReadBit (QEP2PortX, QEP2BitMask))
     {
          If QEP1
               Counter--;
          else
               Counter++;
     }
}
```

Explanation of the algorithm is the follow:

When an interrupt is generated, the Interrupt Service Routine (ISR) checks the level of QEP2 line. If QEP2 is high, then the ISR check the level of QEP1 line. If the QEP1 is high, the counter of sensor is decremented. If QEP1 is low, the counter of sensor is incremented.

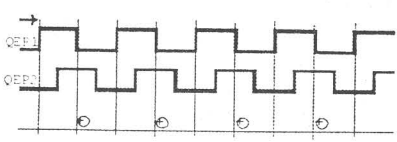

Fig. 7 When the sensor is rotated clockwise

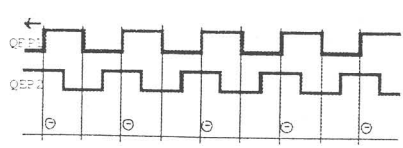

Fig. 8 When the sensor is rotated counter clockwise

Operation of the algorithm when the sensor is rotating clockwise is shown in figure 7. The Counter is incremented on every *falling* edge of QEP1 (because at this moment QEP2 is high).

Operation of the algorithm when the sensor is rotating counter clockwise is shown in figure 8. The Counter is decremented on every *rising* edge of QEP1 (because at this moment QEP2 is high).

Running of the algorithm when the sensor is rotating clockwise, and the direction is changed, is shown in figure 9 and 10.

Here, the absolute error is zero.

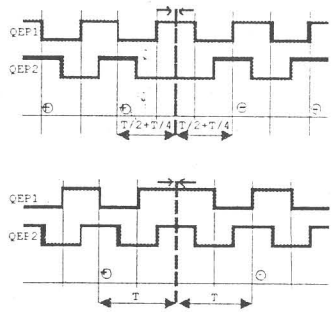

Fig. 9 Change of the direction (from 0 to T/2)

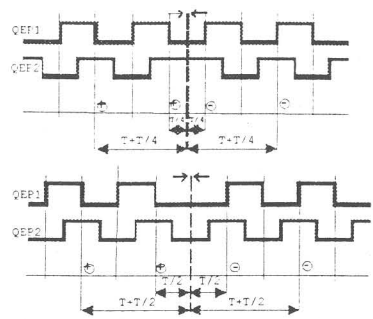

Fig. 10 Change of the direction (from T/2 to T)

Decoding QEP with DSP Controller TMS320F241

The Event Manager module of this family of signal processors has a quadrature encoder pulse (QEP) circuit. The QEP circuit, when enabled, decodes and counts the quadrature encoded input pulses on pins CAP1/QEP0 and CAP2/QEP1.

The two QEP input pins are shared between Capture Unit 1 and 2, and the QEP circuit. Proper configuration of CAPCON bits is required to enable the QEP circuit and disable capture units 1 and 2.

The counter of the QEP circuit is GP timer 2. The GP timer must be put in directional-up/down count mode with the QEP circuit as the clock source.

The direction detection logic of the QEP circuit in the EV2 module determines, which sequence is the leading sequence. Then, it generates a direction signal as the direction input to GP timer 2. The timer counts up, if CAP1/QEP0 input is the leading sequence, and counts down if CAP2/QEP1 is the leading sequence.

The QEP circuit counts both edges of the pulses of the two quadrature encoded inputs. Therefore, the frequency of the clock generated by the QEP logic to GP timer 2 is four times that of each input sequences. This quadrature clock is connected to the clock input of GP timer 2.

GP timer 2 always starts counting from its current value. A desired value can be loaded to GP timer 2's counter prior to enabling the QEP mode. When the QEP circuit is selected as the clock source, the timer ignores the TDIR and TCLKIN input pins.

Period, underflow, overflow, and compare interrupt flags for a GP timer with a QEP circuit clock, are generated on respective matches. (A peripheral interrupt request

can be generated by an interrupt flag, if the interrupt is unmasked.) An interrupt flag can generate a peripheral interrupt request, if the interrupt is unmasked.

To start the operation of the QEP circuit:

1) Load GP timer 2's counter, period, and compare registers with desired values, if necessary.

2) Configure T2CON to set GP timer 2 in directional-up/down mode with the QEP circuits as clock source, and enable the selected timer.

3) Configure CAPCON to enable the QEP circuit.

```
T2CNT  = 0x0000;     /* Configure counter register T2CNT     */
T2PER  = 0x00FF;     /* Configure period register T2PER      */
T2CON  = 0x9870;     /* Configure T2CON register (see p.2)   */
CAPCON = 0xE2F0;     /* Configure CAPCON register (see p.3)  */
```

Conclusions:

- The first algorithm can be used by any microcontroller. The second one can be used only by those microcontrollers, which have interrupt input, which can interrupt microcontroller *on rising and falling* edges of the input sequences.
- When the direction of rotation is changing frequently, the error will be accumulated and will be considerable, in the first algorithm. In the second one, the error is zero.
- In the second algorithm, the interrupts are twice more than the first one. Therefore, with the first algorithm, one can measure twice-higher frequency than the second one.
- Machines, where those sensors are installed, are on mechanical vibration. Let the sensor is stop just at the rising or at the falling edge of QEP1. If at this moment, the machine is shaken, the sensor will generate some pulses (it depends of the shock). The pulses with period longer then the ISR working time will be counted. The pulses with period shorter then the ISR working time will be ignored and an error would be appear.
- The suitable choice is TMS320F240, where the problems are solved in hardware and the Counter can make appropriative interrupts.

REFERENCES

[1] Г. Радулов " *Методи и средства за измерване и контрол* "МГУ – София, 1995

[2] *TMS320C24X DSP Controllers - Reference Set: Vol. 1*, Texas Instruments Inc., 1997.

[3] *TMS320C24X DSP Controllers - Reference Set: Vol. 2*, Texas Instruments Inc., 1997.

[4] Г. Михов " *Цифрови фазови синхронизатори за броячни групи*", ННПК ЕТ'95, сб. Докл. Т. III 197-202, Созопол, септември 1995
[5] G. Mihov, I. Tashev "*Industrial Controller For Discrete Manufacture*", NSC "Electronics 96", book I, 31-36, Sozopol, September, 1996
[6] Л. Иванова, Ст. Овчаров, А. Джонджоров, П. Бързаков, С. Калпакчиев, И. Захариев, К. Колев "*Лазарен интерферационен виброметър*", IV конгрес по механика, септември 1981, Варна, годишник на ВУЗ, Том 16, Кн. 3, "Техника", София 1981, стр. 61 - 67