

Interfacing Micro-controllers with Incremental Shaft Encoders

Introduction

Incremental shaft encoders are popular low cost rotational sensors which convert rotary motion into a pair of digital outputs. As the sensor is rotated two pulse trains are emitted from the outputs with a 90° phase shift between them.



The faster the sensor is rotated the higher the frequency of the pulse train until some limiting rate where the sensor can no longer operate correctly. The maximum pulse frequency is a function of the technology used to construct the sensor and will be quoted in the manufacturers data. The hardware interface between the sensor and the rest of the system can affect the pulse shaping and mark/space ratio.

Although dedicated chips are available to decode the output of these sensors it is much more common (and cheaper) to connect the two pulse trains directly into a micro controller and use software or a combination of software and on-chip timer hardware to decode the data. Since these sensors are incremental the best that can be done is measure the total angle of rotation since the last reset (unless some additional reference signal is available). Despite the simplicity of these sensors and their common use the design of the interface software often causes problems. The solution presented here has been proven to work reliably and can be used in polled or interrupt driven systems.

Decoding Principle

If the sensor is operating normally the outputs will read:

A	B
1	0
1	1
0	1
0	0

for one direction and:

A	B
1	0
0	0
0	1
1	1

for the other direction.

The two outputs obviously exhibit all possible combinations of logical state during normal rotation in either direction. In order to generate directional information it is necessary to decode the present state and the previous state. If the bit pattern for the current state and the previous state are combined to generate a 4 bit word the results are:

old A	old B	A	B	dec
0	0	1	0	2
1	0	1	1	11
1	1	0	1	13
0	1	0	0	4

old A	old B	A	B	dec
1	1	1	0	14
1	0	0	0	8
0	0	0	1	1
0	1	1	1	7

The 'dec' column is the decimal value of the 4 bit binary word. During normal operation of the shaft encoder only 8 of the possible 16 values will occur.

The 4 bit value can be used to index into a look up table, the 4 entries for one direction of rotation are given the value +1, the 4 entries for the other direction of rotation the value -1 and the other 8 entries the value 0.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-1	+1	0	+1	0	0	-1	-1	0	0	+1	0	+1	-1	0

Every time the shaft encoder ports are sampled by the processor the index is calculated by shifting the 2 bit result from the previous encoder sample left two bits and ORing in the two new bits. The index is used to access a value of -1, 0 or +1 from the look up table and this value is added to the running total for the angle of rotation.

Depending on the hardware resources available in the processor it can be made to sample using interrupts generated by any change in the 2 encoder outputs or by regular sampling.

Interrupt driven sampling will usually require less processor time but requires 'any edge' interrupt generation on two inputs. A timer driven sampling system will be over-sampling for most of the time and may not make the most efficient use of processor time.

If the same state is read repeatedly or jitter causes the phase which has just changed to be read incorrectly then the only result is that a zero is added to the angle of rotation.

The decoding process is the same for either sampling method and is easy to code in C or assembler.

The C code fragment below assumes that the shaft encoder data is presented in the two ls bits of `port`.

```
extern signed short angle;          /* rotation since reset */
static unsigned char ab = 0;       /* the old value of the sensor ports */
const signed short table[] = {0,-1,1,0,1,0,0,-1,-1,0,0,1,0,1,-1,0};
                                /* increment of angle for the 16 possible bit codes */

ab = ab << 2;                     /* move the old data left two places */
ab |= (port & 0x3);               /* OR in the two new bits */
angle += table[(ab & 0xf)];       /* get the change from the 16 entry table */
```

`angle` is global so that it can be reset during program operation.

Summary

The code required to interface an incremental shaft encoder with a micro-controller is very simple, the same code can be used with interrupt driven or timed sampling systems.

If you need help with an application of this technique or any other micro-controller problem please contact me.

Michael Kellett,
Electronics and Software Consultant.

© August 2001

mk@mkesc.co.uk