Microprocessor Interfacing Laboratory (MIL)

Using an MCU to Read Incremental Encoders

ICOM-5217 - Prof. Manuel Jiménez

(with contributions by Luis Francisco)

An incremental encoder is an electromechanical device that can be used to measure the movement, position, and displacement of a rotational or linearly moving mechanical component. In rotating parts, an incremental encoder can measure rotation direction and converts angular position and angular displacement into digital pulses. An incremental encoder has two outputs in quadrature, i.e., two outputs with a 90º phase shift between them. These outputs result from the optical or mechanical detection of two patterned tracks with a 90º geometric shift between them. Sample tracks and pulse trains are illustrated in Figure 1.

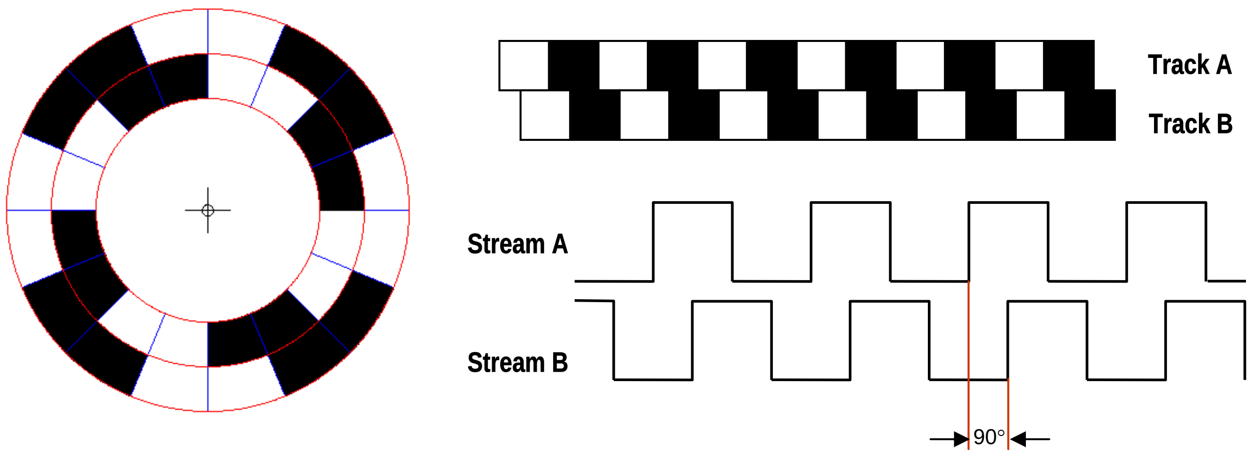

Figure 1: Output waveforms of an incremental encoder.

Detecting movement in either direction from the signal stream only requires determining the order of signal edges in streams A and B and encoding them with a binary code. The positions of the shaded regions generate a Gray binary code.

Figure 2 shows the waveform streams A and B labeled for rotation sequences in counter clockwise (CCW) and clockwise (CW) directions. Red and blue arrows denote the edge sequences for each direction.

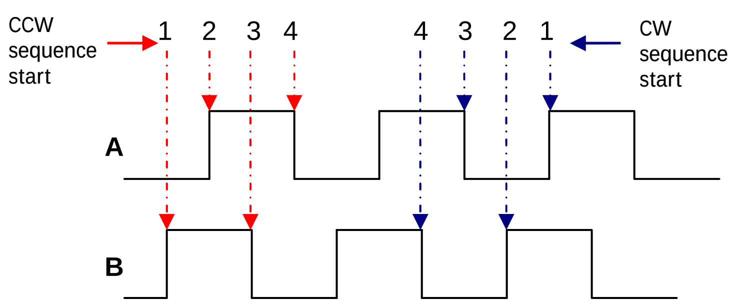

Figure 2: Rotation sequence for CCW and CW directions.

A sequence begins when signals A and B are in the same state. For example, in Figure 2, the CCW sequence begins at state 00 while the CW begins at 11. New sequence values are detected through signal edges in either of the streams. A sequence ends when its four states have been generated. At this point a new sequence begins by repeating the codes from the initial state. Measuring the time between state changes allows obtaining the rotation speed. Tables 1 and 2 below show the sequences for CCW and CW rotations.

A	B	Value
0	0	0
0	1	1
1	1	3
1	0	2

Table 1: CCW sequence values.

A	B	Value
1	1	3
0	1	1
0	0	0
1	0	2

Table 2: CW sequence values

To read the state values and detecting the sequence changes with and MCU it requires using two interrupt enabled I/O lines. The I/O lines levels indicate the state code and the interrupt capability allows detecting code changes. The interrupt needs to be configured so that it is triggered by any change in the encoder lines. This shall allow detecting both, the rising and falling edges of either signal. If the rotational speed were also of interest, a timer could be used to measure the time between edges. Multiplying the time between edges (t_{edge}) by the number of steps in the wheel yields the rotational period. For the sample wheel illustrated in Figure 1, two consecutive interrupts represent 1/16 of the wheel rotation, thus 16 times t_{edge} is one revolution.

To determine the rotation direction it is required to know the state of *A* and *B* before and after the occurrence of an edge. Combining the two two-bit codes, a *four-bit identifier* is obtained, which provides for any possible result in the sequence. Tables 3 and 4 list the values for the CCW and CW sequences. Note that each 4-bit code represents a state change in which only one bit changes before and after the edge. Recall that the encoder produces a Gray sequence, and therefore only one bit is allowed to change between consecutive states.

A_{old}	B_{old}	A_{new}	B_{new}	# (old:new)
0	0	0	1	1
0	1	1	1	7
1	1	1	0	14
1	0	0	0	8

Table 3: 4-bit codes for CCW sequence.

A_{old}	B_{old}	A_{new}	B_{new}	# (old:new)
1	1	0	1	13
0	1	0	0	4
0	0	1	0	2
1	0	1	1	11

Table 4: 4-bit codes for CW sequence.

To write a software function for determining the rotation direction and wheel position, we could use a lookup table (LUT). The 4-bit values resulting from he before and after codes could be used as to index the table, and the entries would be either +1, -1, or 0 for representing CW, CCW or no movement conditions, respectively. Table 4 shows such a LUT. Assuming the wheel started moving from a known “home” position (index hole and detector might be needed), adding the value fetched from the lookup table on each edge interrupt, we can have the absolute wheel position at any time.

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

Table 4: Lookup table to detect the direction of rotation and absolute position.