AMRX-15x0 series
Ultra-slim Scrollwheel with 5-way Switch
Using Quadrature Outputs

Application Note 5322

Description

Avago Technologies' AMRX-15x0 provides an integrated solution for scrolling, directional navigation and push button selection in a compact and ultra-slim package. With ease of use in mind, AMRX-15x0 is ideal for scrolling of menus in new handheld electronic devices, such as in mobile devices, music players, cameras, and entertainment consoles. Based on Avago Technologies' reflective optical technology, the motion sensor is non-contact and ensures reliable operations.

The signal outputs from the scrollwheel are quadrature outputs. Quadrature signals consist of two square waves 90° out of phase. They are used to determine the speed or velocity of the movement (or rotation), together with the direction of movement (or rotation).

The scrolling direction, either clockwise (CW) or counter-clockwise (CCW) on the dial of AMRX-15x0, which corresponds to UP or DOWN scroll on menu list on the consumer products can be determined by comparing whether the quadrature output signals, Channel A is leading or lagging Channel B.

Meanwhile, the velocity (in terms of revolution per second) can be determined by the dividing the total count (or pulses) per second by the number of pulses per revolution (output resolution of the motion sensor). A pulse is also known as a cycle and they are sometimes used interchangeably.

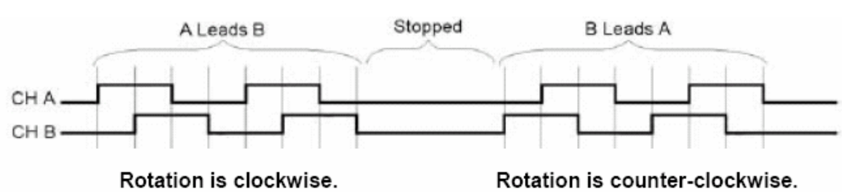


Figure 1. Leading channel at different rotational directions.

Direction and rotational position detection

The default phase difference of 90° in quadrature signals is used for detecting the direction of rotation. For example, if Channel A signal leads Channel B signal, the scroll wheel is rotating in CW direction. Conversely if Channel B signal leads Channel A signal, scroll wheel is rotating in CCW direction as illustrated in Figure 1.

The equation 1 below is used to determine the rotational position:

$$\textit{Rotation Degree}, \theta = \frac{Count}{\eta \times N} \times 360^{\circ} \qquad \text{[eq. 1]}$$

where,

Count = total number of detected pulses per revolution

η = 1 or 2 or 4 dependent on decoding mode;

N = Resolution or number of count within one revolution

The decoding modes, namely 1x, 2x or 4x decoding, which will be elaborated in the later section, is a technique to leverage the characteristics of quadrature signal.

For scrolling operation, though it may not necessary to translate quadrature signal to mechanical degree, the following example is used to illustrate the use of equation 1:

Calculation example Equation 1 when 1x decoding is used.

Say,

Resolution of AMRX-1500 = 45 cycles per revolution

1x decoding mode is used

Number of counts detected per revolution (or one complete turn of scroll wheel)

Therefore,

N = 45 counts (as 1x decoding is used)

η = 1 (as 1x decoding is used)

Count = 10 counts/pulses detected

Using Eq. 1,

$$\theta = \frac{Count}{\eta \times N} \times 360^\circ = \frac{10}{1 \times 45} \times 360^\circ = 80^\circ$$

From the calculation result, it can be deduced that when there is a movement or rotation on the scroll wheel, whereby 10 pulses were detected on its output signals, the scroll wheel is actually being rotated 80° from it's original position. The direction of rotation can be determined using the leading channel characteristics from Figure 1.

Different Decoding Modes

As discussed earlier, quadrature signal has 1x, 2x and 4x decoding modes. In 1x decoding mode, signal from one of the channels is taken directly (by rising edge or falling edge only). Hence, scrollwheel with 45 cycles per revolution (CPR) has 45 pulses per revolution or 45 counts per revolution as its resolution. In 2x decoding mode, additional resolution can be decoded by detection of rising edge and falling edge, from one of the channels (usually CH A). In 4x decoding mode, highest resolution is achieved by detection of rising edge and falling edge from both channels. Table 1 illustrates the resolution achievable through different decoding modes. Figure 3 shows the processed signal in 2x and 4x decoding modes. The PosCnt signal is the digitally processed signal whereby the previous state and the present state is compared to get the direction information. For example, for the first count, state changes from 11 to 01 and from Figure 3, this is a count up and therefore represented by +1. UPDWN signal derived from PosCnt is a control signal for direction where a high can represent UP direction while a low represent DOWN direction of scroll on consumer products or vice versa.

Table 1. Resolution with different decoding modes

Decoding Mode	Initial Resolution	Resolution after decoding
1X	45 CPR	45 PPR (no decoding required)
2X	45 CPR	90 PPR
4X	45 CPR	180 PPR

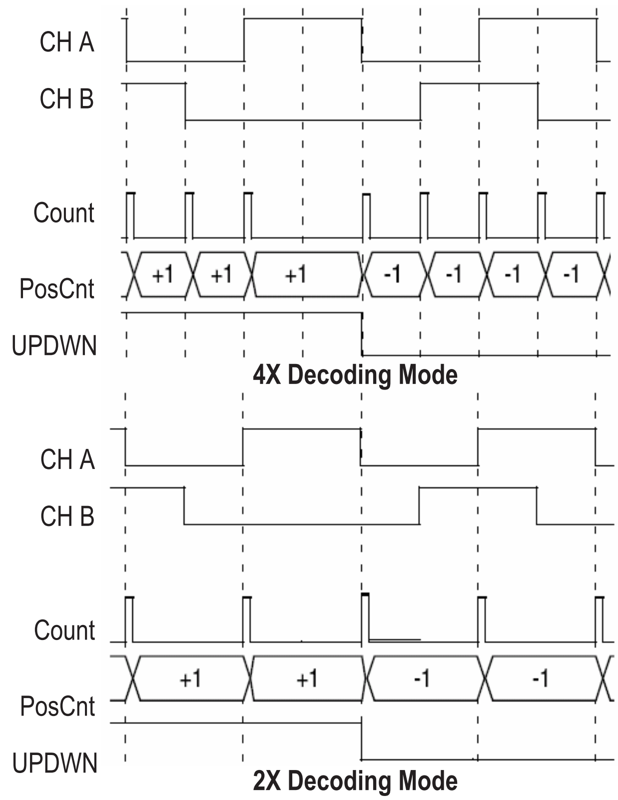
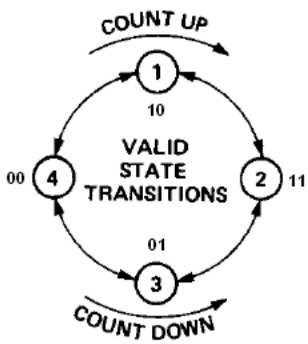


Figure 2. Quadrature signals in 2x and 4x decoding modes

Implementation of quadrature decoder

The quadrature decoder decodes the incoming signals into count information. This circuitry multiplies the resolution of the input signals by a factor of four (4X decoding). When using scrollwheel for motion sensing, the user benefits from the increased resolution by being able to provide better system control. The quadrature decoder samples the outputs of the CHA and CHB. Based on the previous binary state of the two signals and the present state, it outputs a count signal and a direction signal to the position counter. Figure 3 shows the quadrature states and the valid state transitions. Channel A leading channel B results in counting up. Channel B leading channel A results in counting down.

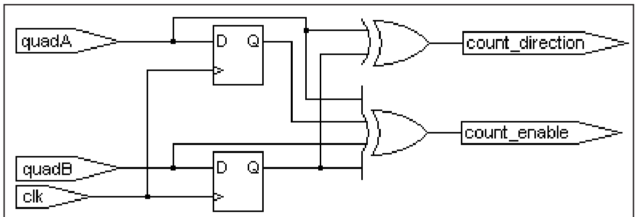


CHA	CHB	STATE
1	0	1
1	1	2
0	1	3
0	0	4

Figure 3. Quadrature signals in 2x and 4x decoding modes

The state diagram in Figure 3 can be represented (and decoded) by a simple Finite State Machine (FSM). For example, the change from state 00 to state 01 should increment the position counter, and the change from 11 to 01 should decrement the counter. A possible implementation could be to memorize the previous CH A and CH B values (quadA_d, quadB_d) with a pair of FlipFlops, and decode count_direction from the (quadA, quadB, quadA_d, quadB_B) truth table. This is illustrated in Figure 4.

Similarly, firmware implementation using microcontroller or microprocessor is possible using the similar approach.




```
module quad(clk, quadA, quadB, count);
input clk, quadA, quadB;
output [7:0] count;
reg quadA_d, quadB_d;
always @(posedge clk) quadA_d <= quadA;
always @(posedge clk) quadB_d <= quadB;

wire count_enable = quadA ^ quadA_d ^ quadB ^ quadB_d;
wire count_direction = quadA ^ quadB_d;

reg [7:0] count;
always @(posedge clk)
begin
 if(count_enable)
 begin
  if (count_direction) count<=count+1; else count<=count-1;
 end
end
endmodule
```

Figure 4. Example of a simplified quadrature decoder using Verilog coding

For product information and a complete list of distributors, please go to our web site: **www.avagotech.com**

AV02-0245EN - April 26, 2007