NEURON® CHIP Quadrature Input Function Interface

August 1991

LONWORKS™ Engineering Bulletin

Introduction

The NEURON CHIP quadrature input function provides a simple means to process external data encoded in quadrature format.

Quadrature encoding is used in position sensing applications where only two external characteristics are needed to accurately determine the position of an object relative to its last position: magnitude and direction of change. The quadrature encoding allows both of the above attributes to be conveyed using only two signals, thereby simplifying the circuitry and eventual decoding that will be needed.

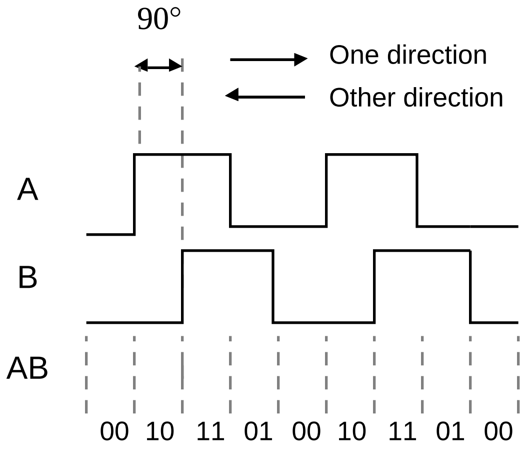

Figure 1. Ideal quadrature relationship.

A quadrature relationship between two signals, as the name implies, assumes a 90 degree phase difference between them, as shown in figure 1. Note that the resulting composite code (AB) follows the reflected (Gray) code sequence which has the property that only one bit is changed between consecutive values.

Position sensing can be divided into two broad classes: rotational and linear. Encoders for both classes exist in the market with a wide range of resolution and reliability. The resolution of a quadrature encoder refers to the number of different composite output code transitions it can generate for a given amount of movement (e.g., 16 counts per revolution for a rotational encoder).

Both mechanical and electronic (optical) encoders exist in the market allowing for flexibility and range in cost, reliability, accuracy, and size.

Figure 2 shows the contact pattern for a mechanical linear quadrature encoder.

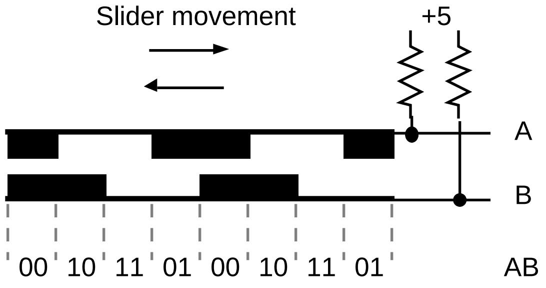

Figure 2. An example of a linear encoder's mechanical contact pattern. The common pin is not shown and is the slider which rides on the "rail" .

A rotational encoder can be realized by simply wrapping the pattern shown in figure 2 so that the two ends are connected and there is therefore no restriction imposed on the movement of the slider. The slider is then hinged on the central axis of the newly formed circle so that a more convenient rotating shaft could be used.

A rotational quadrature encoder, also referred to as a digital pot or a shaft encoder, is often used to replace variable resistor potentiometers or any other rotational dial input device in microprocessor-based systems. This provides an easy interface to the digital domain.

Linear quadrature decoders also simplify such interfaces by replacing linear slide variable resistor potentiometers or any other linear (straight-line) input device.

It is important to note that although these devices are called quadrature encoders, they do not necessarily produce perfect quadrature (90°) outputs at all times.

While at a constant non-zero speed the output of these encoders might look like the one shown in figure 1, it is generally accepted that the encoder is producing a quadrature output as long as the correct relative sequence of level transitions is obeyed and the interface circuitry's timing requirements are met. In other words, the duty cycle of the output waveforms need not be 50%.

An acceptable encoder output might look like the one shown in figure 3. Note that the same sequence of composite output codes is observed in both directions, as in figure 1.

The two characteristics mentioned earlier, namely magnitude and direction of movement, can be extracted from the quadrature output of an encoder in several ways.

The magnitude (i.e. angular or linear displacement) can be observed by simply counting the number of transitions on either output (A or B). The rate of change of magnitude (angular or linear velocity), could then be extracted by performing the same counting operation within a given period of time.

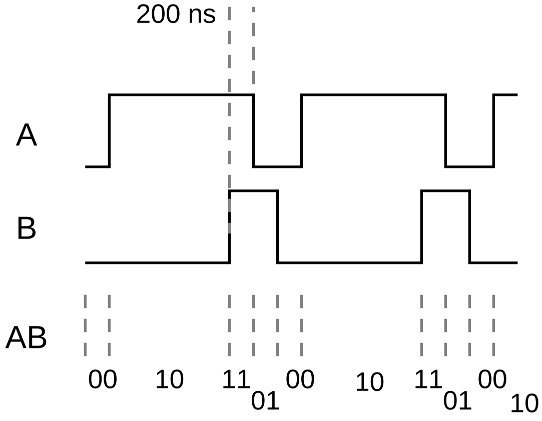

Figure 3. Another "quadrature" relationship.

The determination of direction from the quadrature signal is a bit more subtle. Two popular techniques are: comparing current composite output code to the previous one, and observing the level of one output (e.g., A) at the time when the other output (e.g., B) makes a transition.

Regardless of the technique used in interfacing a quadrature encoder to a system, some amount of circuitry and/or software is usually required on the designer's part. Using the NEURON CHIP quadrature input function block such a need is eliminated.

The quadrature encoder, as shown in figure 4, is connected directly to the NEURON CHIP, allowing the designer to concentrate on the intended application rather than on interface issues.

Usage

The quadrature input function of the NEURON CHIP can be invoked by using the following statement in a NEURON C application program:

```
pin  input  quadrature
io_object_name;
```

where:

`pin` is a NEURON CHIP I/O pin. Either IO_4 or IO_6 may be used. The second pin needed for the two quadrature inputs is assumed to be the next higher pin number (IO_5 or IO_7) on the NEURON CHIP.

`io_object_name` is a user-specified name for the quadrature device.

Refer to the NEURON C Programmer's Guide for a more detailed description of the syntax.

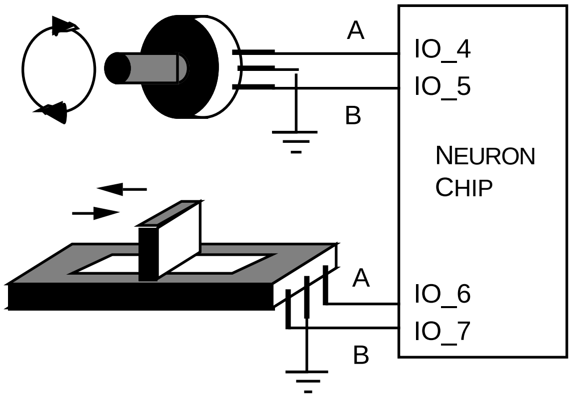

Figure 4. A typical encoder connection to the NEURON CHIP.

The following is an example for the above syntax:

```
#pragma enable_io_pullups
IO_4 input quadrature dial;
long dial_data;

when (io_update_occurs(dial))  {
     dial_data = input value;
...  //dial_data represents the change in offset of
     //the shaft encoder since last input.
}
```

The `pragma` compiler declaration enables the NEURON CHIP's on-chip pull-up resistors. This reduces the task of using a quadrature encoder to merely connecting the two encoder outputs to the two NEURON CHIP inputs (declared in your program) -- no additional components are required

Quadrature encoders normally provide a third pin designated as the common. In order to use the NEURON CHIP's built-in pull-up resistors, this pin must be connected to ground. Connecting the common pin of the encoder to +5 volts and using external pull-down resistors would also work and would cause the relative direction of the encoder to be reversed (A and B signals are inverted).

Another way of reversing the relative direction of the encoder is to invert the sign (take the two's complement) of the number returned by the `input_value` variable. This is generally the preferred way since it only requires a simple software modification and no hardware modification.

The NEURON CHIP makes use of its internal timer/counter resources, along with its built-in firmware, to decode quadrature signals. The time periods between transitions are measured by a counter which is controlled by additional logic responsible for determining the direction of count. The user can access the results of this measurement operation by using the `io_update_occurs()` function of the NEURON C.

The `io_update_occurs()` event evaluates to TRUE when the quadrature object specified (dial) has an updated value. At that point `input_value` contains the amount of change observed at the quadrature inputs (encoder position changes). This value is a signed long (16 bit) positive (negative) number representing the number of increments (decrements) at the quadrature input since the last evaluation of `io_update_occurs()`.

The number of times a quadrature input is evaluated is therefore not only dependent on the absolute speed at which the NEURON CHIP is running, but also on the size and structure of the application code. The task scheduler in the NEURON CHIP is responsible for executing all the tasks in the program including the `when (io_update_occurs())` statement mentioned above. Refer to the *NEURON C Programmer's Guide* for more information on the operation of the scheduler.

The `input_value` is in two's complement format. Therefore, the most significant bit (sign bit) represents the direction of the movement of the quadrature encoder.

The count returned through `input_value` is cumulative. That is, between two consecutive `io_update_occurs()` events, a finite movement in one direction followed by an equal amount of movement in the opposite direction would yield a total count of zero.

The NEURON CHIP firmware limits the maximum value of `input_value` to +16383 and -16384 in either direction. If the number of code changes observed at the quadrature inputs between consecutive `io_update_occurs()` events exceeds ±16K then the `input_value` for the second event will be the maximum in that direction (+16K or -16K).

The `io_changes()` event is not a good alternative for use with the quadrature input since at a constant encoder speed the same value is observed by the event and therefore no apparent "change" is observed.

With the NEURON CHIP running at 10 MHz, the input to the quadrature pins is sampled every 200 ns. Therefore, any external encoder output values at the quadrature inputs which occur faster than every 200 ns will not be recognized (Figure 3). As a consequence, the maximum frequency on either A or B inputs of the NEURON CHIP running at 10 MHz must not exceed 1.25 MHz. This should be more than enough for typical user-interface applications. The sampling rate scales at lower clock speeds.

The following NEURON C program illustrates a typical application of the quadrature input function. The variable count contains a cumulative updated `count` of encoder movements.

```
#pragma enable_io_pullups

////////////////////User defined parameters////////////
#define lower_limit 0         //lower count limit
#define upper_limit 999       //upper count limit
#define shaft_direction 1     //direction of count (e.g.,
0=CW,1=CCW)

/////////////////// Declarations //////////////////////
IO_4 input quadrature shaft_encoder;  //shaft encoder input
signed long count;
signed long increment;

when (io_update_occurs(shaft_encoder){

   if (shaft_direction) count += input_value;
     else count -= input_value;     //inc/dec based on user
                                    //definition
   if (count<lower_limit) count=lower_limit;
   if (count>upper_limit) count=upper_limit;  //take care of
                                              //overflow
               //and underflow due to fast shaft rotation
}
when (reset)  {
       count = 0;
}
```

Encoder Sources

The following is a partial list of some of the manufacturers of quadrature encoders:

Bourns Inc. , Resistive components Group
(714) 781-5050
Mechanical encoders. Shaft angle.

Clarostat Manufacturing Co.
(800)-872-0042
Mechanical and optical encoders. Shaft angle.

Litton Encoders
(818)341-6161
Mechanical, optical, and magnetic encoders. shaft angle.

Photoswitch Div., Allen-Bradley Co.
(617)466-8000
linear and shaft encoders. mechanical and optical encoders.

U.S Digital Corp.
(213)594-0094
Optical shaft and linear encoders.

Echelon Corporation
4015 Miranda Avenue
Palo Alto, CA 94304
Telephone (415) 855-7400
Fax (415) 856-6153

Echelon Europe Ltd
105 Heath Street
London NW3 6SS
England
Telephone (071) 431-1600
Fax (071) 794-0532
International Telephone + 44 71 431-1600
International Fax + 44 71 794-0532

Part Number 005-0003-01

Echelon Japan K.K.
AIOS Gotanda Building #808
10-7, Higashi-Gotanda 1-chome,
Shinagawa-ku, Tokyo 141, Japan
Telephone (03) 3440-8638
Fax (03) 3440-8639