Decodes & counts pulses function V. 1.0

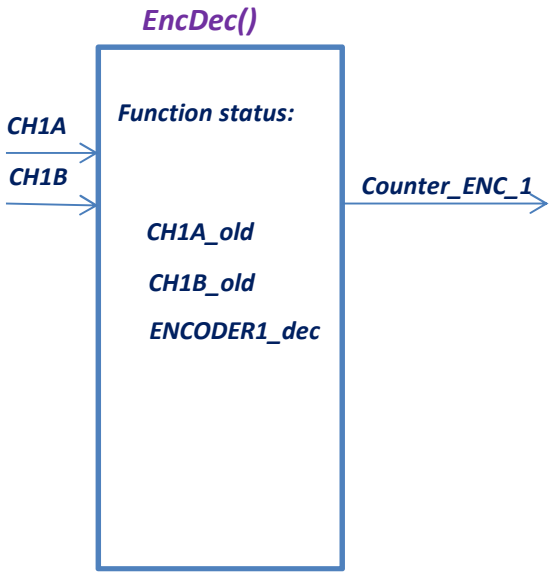
Descriptions

This function is a routine of "ENCODER QUADRATURE with X4 multiplication".
The mains benefit of this implementation are:

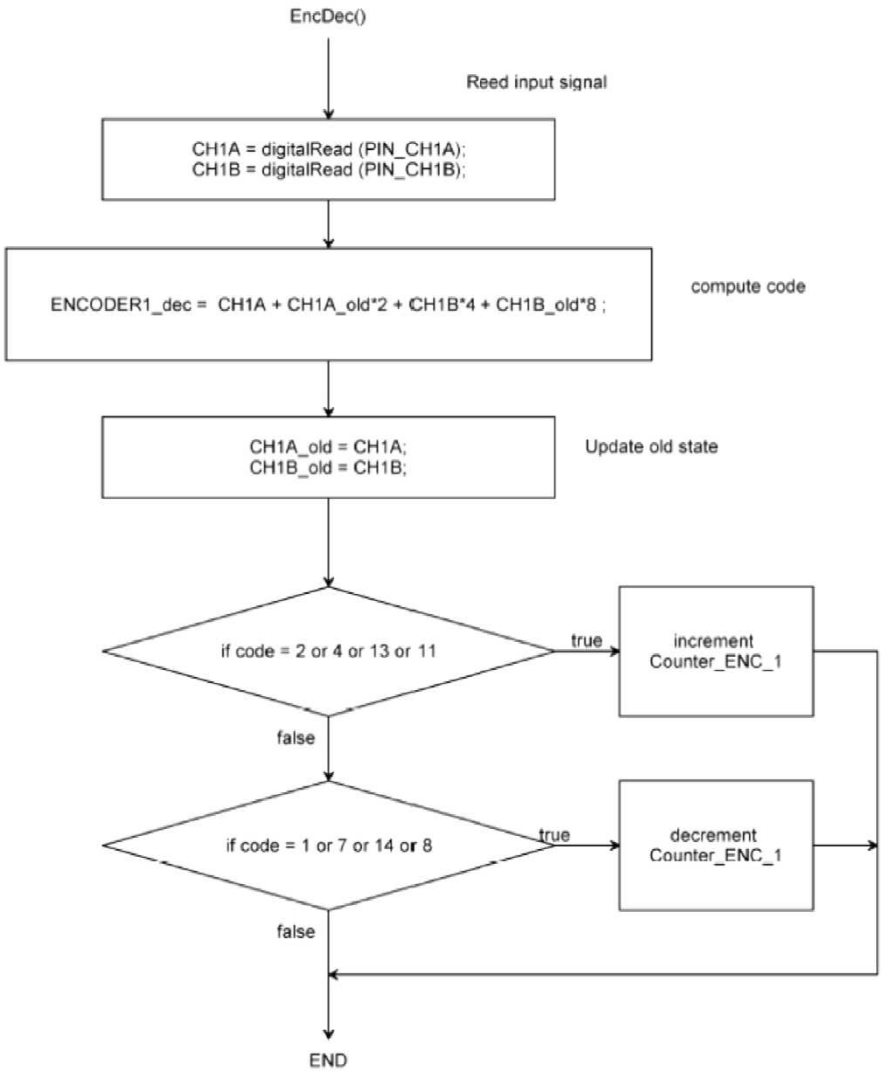
- ***Don't use dedicated interrupts. Are used only two digital input pins (PIN_CH1A = 12, PIN_CH1B = 11)***
- ***Shares and works under the interrupt of TIMER1 (in the case of ARDUINO MICRO).***
- ***It has a simple filter of the signals in quadrature.***
- ***It increase for four the resolution of the hall sensor.***
- ***It is very simple. Requires one line mathematical instruction for recognize the pulses and one operation of "Case" to count.***
- ***It is very fast (about 17 micro second in the case of ARDUINO MICRO).***

Syntax:	***EncDec()***
Input parameters:	***CH1A, CH1B***
Output parameters:	***Counter_ENC_1***
Status:	***CH1A_old, CH1B_old, ENCODER1_dec***
Calls:	

EncDec function




```
void EncDec ()

{
  CH1A = digitalRead (PIN_CH1A);          // Reads the state of input  channel A
  CH1B = digitalRead (PIN_CH1B);          // Reads the state of input  channel B

  ENCODER1_dec =  CH1A + CH1A_old*2 + CH1B*4 + CH1B_old*8 ;

  CH1A_old = CH1A;                        // Save as old value (channel A) for next interrupt
  CH1B_old = CH1B;                        // Save as old value (channel B) for next interrupt

   switch (ENCODER1_dec)

                        { // codes that increase
                         case 2: Counter_ENC_1++;
                        break;
                        case 4: Counter_ENC_1++;
                        break;
                        case 13:Counter_ENC_1++;
                        break;
                        case 11:Counter_ENC_1++;
                        break;

                        // codes that decrease
                        case 1: Counter_ENC_1--;
                         break;
                         case 7: Counter_ENC_1--;
                        break;
                        case 14:Counter_ENC_1--;
                        break;
                        case 8: Counter_ENC_1--;
                        break;
                         }
    }
```


EncDec()
Reed input signal
CH1A = digitalRead (PIN_CH1A);
CH1B = digitalRead (PIN_CH1B);
ENCODER1_dec = CH1A + CH1A_old*2 + CH1B*4 + CH1B_old*8 ;
compute code
CH1A_old = CH1A;
CH1B_old = CH1B;
Update old state
if code = 2 or 4 or 13 or 11
true
increment
Counter_ENC_1
false
if code = 1 or 7 or 14 or 8
true
decrement
Counter_ENC_1
false
END