

Bounce-busting Bullet Proof Rotary Encoder Code

ROTARY ENCODER, KY-040

No pullups and no capacitors!

Initial post by Peter Mather and further refinement by @apalert on TBS.

[My bounce-busting bullet proof Rotary Encoder code](#)

One thing I learned while testing is that it's very easy to exceed the speed these rotary encoders can cope with which causes missed pulses.

The data sheet for my encoder gives maximum rotary speed of 100 rpm. It has 20 indents per full turn so that calculates down to 33 clicks per second, a bit over 1 1/2 turns, 30mS per click. You can easily exceed that in short bursts. I setup a timing loop and with a finger twirl on the knob for 1/4 turn some pulses were as low as 6mS apart, 5 times the maximum speed recommended. Little wonder it missed a few pulses!

One tweak was to grab the direction data on the first clock pin interrupt but not finalize it until the data pin interrupt. This means the user can overshoot 49% of the way to the next indent and back off without a miscount.

Anyway, in case there is interest I'll attach the code:

```
'Code for rotary encoder interpretation by Malcolm Young
'Rotating the encoder prints incremental numbers as appropriate
' eg 0 1 2 3 4 3 2 1 0 -1 -2 etc as well as a direction flag
'in the form of 1 and -1. Developed on Picomite

OPTION EXPLICIT
OPTION DEFAULT NONE

dim float pre_action = 0 'temporarily holds data from new encoder movement
dim float dir_action = 0 'makes "public" the new directional data
dim float inc_action = 0 'makes "public" the new incremental data
dim float old_inc_action = 1 'used to decide if data has changed
dim float clock_enable = 1 'flag to control clock pin interrupt process
dim float reset_enable = 0 'flag to control data pin interrupt process
dim float clk = 20 'use to name clock pin
dim float dat = 19 'use to name data pin

setpin clk, intl , clock_interrupt,pullup 'clock pin (encoder pin a)
setpin dat, intb, reset_clock_enable,pullup 'data pin (encoder pin b)

print
print "inc_action", "dir_action"
do
  If inc_action <> old_inc_action then
    print inc_action,, dir_action
    'cls: text 96,60, str$(inc_action),RB,6 'LCD panel output
    old_inc_action = inc_action
  end if
loop
```

```
sub clock_interrupt
  if clock_enable = 1 then
    clock_enable=0 'prevents any further "noise" pulses being processed
    if pin(dat) = 1 then 'direction was clockwise
      pre_action = 1
    else 'direction was counter-clockwise
      pre_action = -1
    endif
    reset_enable=1 'makes reset_clock_enable ready to accept data
  endif
end sub

sub reset_clock_enable
  if reset_enable = 1 then
    reset_enable = 0 'prevents any further "noise" pulses being processed
    if pin(clk) = 0 then 'prevents processing if RE position reverted to origin
      dir_action = pre_action 'directional output
      inc_action=inc_action + pre_action 'incremental output
    endif
    clock_enable = 1 'enables data pin again, even if reverted to origin
  endif
end sub
```

From:
<https://fruitoftheshed.com/wiki/> - FotS

Permanent link:
https://fruitoftheshed.com/wiki/doku.php?id=mmbasic:bounce-busting_bullet_proof_rotary_encoder_code

Last update: 2024/03/16 22:13

