#### Rotaryswitch with interrupt & Table

/\* global variable \*/

Volatile uint8\_t EncCnt = 0;

/\* local variable \*/

static uint8\_t last = 0;

last = (last << 2) & 0x0F;

if(PHASE\_A) last |= 0x02;

if(PHASE\_B) last |= 0x01;

EncCnt += table[last];

|  |
| --- |
| Ones of Direction of rotation:  Ratched old(Bit 4 & 3) new(Bit 2 & 1)  A B A B  -------  --> 0 0 0 1 1  | > 0 1 1 1 7 Edge on A (\*)  | 1 1 1 0 14  | > 1 0 0 0 8 Edge on A (\*)  | 0 0 0 1 1  | > 0 1 1 1 7 Edge on A (\*)  | 1 1 1 0 14  | > 1 0 0 0 8 Edge on A (\*)  --< 0 0 0 1 1 |

|  |
| --- |
| Ones of Direction of rotation:  Ratched old(Bit 4 & 3) new(Bit 2 & 1)  A B A B  -------  --> 0 1 0 0 4  | > 0 0 1 0 2 Edge on A (\*)  | 1 0 1 1 11  | > 1 1 0 1 13 Edge on A (\*)  | 0 1 0 0 4  | > 0 0 1 0 2 Edge on A (\*)  | 1 0 1 1 11  | > 1 1 0 1 13 Edge on A (\*)  --< 0 1 0 0 4 |

\*= Value corresponds to an entry from the table

/\* This table is used to increment +2 \*/

const int8\_t table[16] =

{0,0,-1,0,0,0,0,1,1,0,0,0,0,-1,0,0};

/\* This table is used to increment +1 \*/

const int8\_t table[16] =

{0,0,-1,0,0,0,0,1,0,0,0,0,0,0,0,0};

4 bits are generated, these are composed of the old and the new state

0bxxxxABab (AB = old, ab = new)