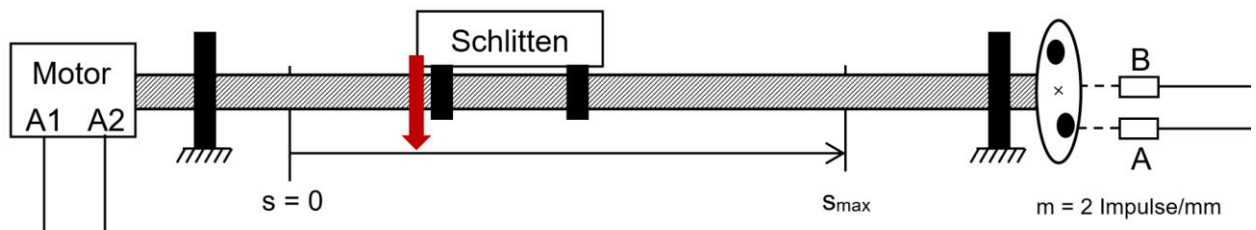


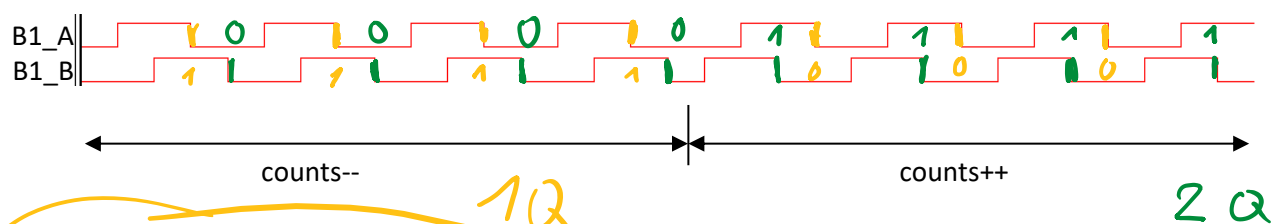
Bei einem inkrementalen Drehgeber kann durch die zusätzliche Auswertung der negativen Flanke von Signal A die Auflösung verdoppelt werden.

### Lageregelstrecke mit Inkrementalsensor

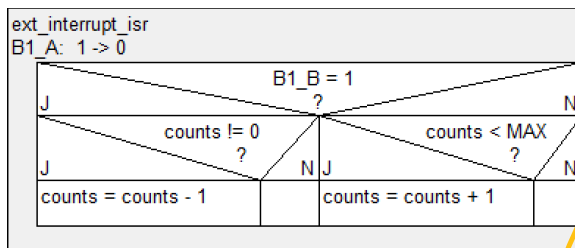


Der gezeigte Algorithmus verwendet den zweiten externen Interrupt für Signal B.

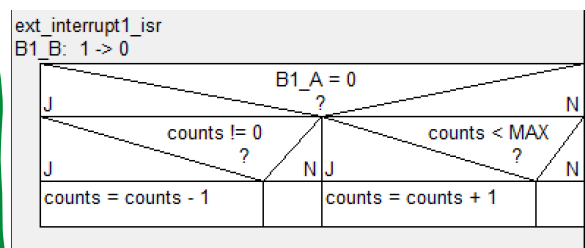
Impulsfolge bei Rechts-/Links Bewegungsumkehr



```
void ext_interrupt_isr(void)
{ // Interrupt bei fallender Flanke an A
  if (bit_read(Sensor,B1_B) == 1)
  {
    if (counts != 0) counts--;
  }
  else
  {
    if (counts < MAXCOUNTS) counts++;
  }
}
```



```
void ext_interrupt1_isr(void)
{ // Interrupt bei fallender Flanke an B
  if (bit_read(Sensor,B1_A) == 0)
  {
    if (counts != 0) counts--;
  }
  else
  {
    if (counts < MAXCOUNTS) counts++;
  }
}
```



### Arbeitsauftrag

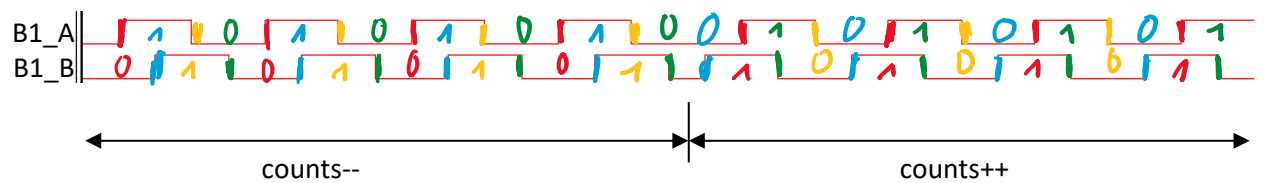
Ändern Sie das Programm für den Drehencoder so ab, dass pro Raststellung 2 counts gezählt werden.

#### 4 Quadranten Encoder

Die Auflösung kann nochmals verdoppelt werden, wenn nicht nur die negativen Flanken, sondern auch die positiven Flanken von A und B ausgewertet werden. In der Datei interrupt.h muss dafür Zweiflanken-Triggrerung für die externen Interrupts aktiviert werden:

```
// externer Interrupt Trigger
// #define _RISING_EDGE_TRIGGER_
// #define _FALLING_EDGE_TRIGGER_
#define _ANY_EDGE_TRIGGER_
```

Impulsfolge bei Rechts-/Links Bewegungsumkehr



```
void ext_interrupt_isr(void)
{
  if (bit_read(Sensor,B1_A) == 0)
  { //Interrupt bei fallender Flanke an A
    if (bit_read(Sensor,B1_B) == 1)
    {
      if (counts != 0) counts--;
    }
    else
    {
      if (counts < MAXCOUNTS ) counts++;
    }
  }
  else
  { //Interrupt bei steigender Flanke an A
    if (bit_read(Sensor,B1_B) == 0)
    {
      if (counts != 0) counts--;
    }
    else
    {
      if (counts < MAXCOUNTS ) counts++;
    }
  }
}
```

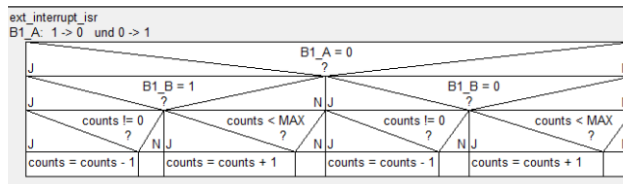
1Q (circled in yellow)

3Q (circled in red)

```
void ext_interrupt1_isr(void)
{
  if (bit_read(Sensor,B1_B) == 0)
  { //Interrupt bei fallender Flanke an B
    if (bit_read(Sensor,B1_A) == 0)
    {
      if (counts != 0) counts--;
    }
    else
    {
      if (counts < MAXCOUNTS ) counts++;
    }
  }
  else
  { //Interrupt bei steigender Flanke an B
    if (bit_read(Sensor,B1_A) == 1)
    {
      if (counts != 0) counts--;
    }
    else
    {
      if (counts < MAXCOUNTS ) counts++;
    }
  }
}
```

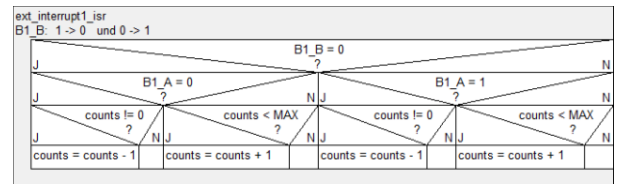
2Q (circled in green)

4Q (circled in blue)



1Q

3Q



2Q

4Q