

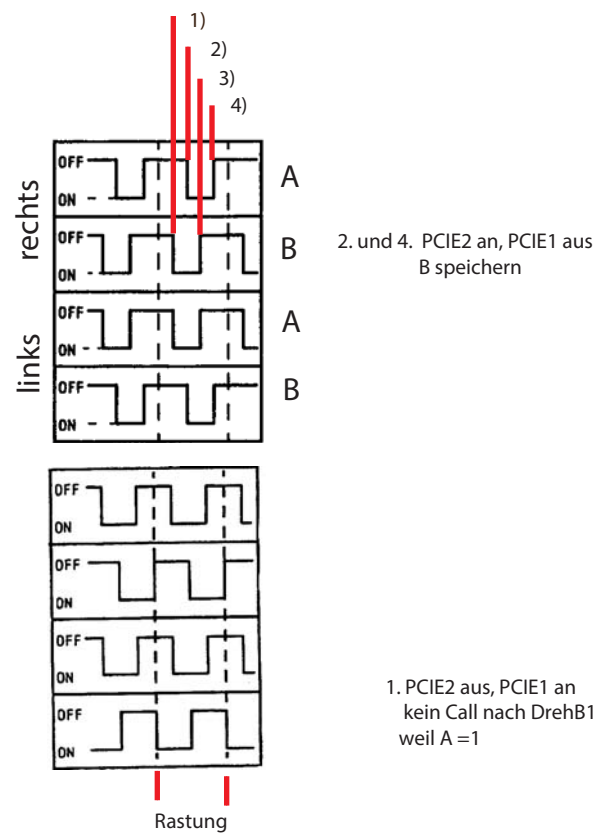
```
.def AL = r16
.def AH = r17
MPinB: .byte 1 ;Merker für Pin B Drehgeber
```

Initialisierung Ports

```
ldi AL, 0b11111110 ; 1= Ausgang
out DDRB, AL
ldi AL, 0x01 ;Bit 0 Pull up
out PORTB, AL
ldi AL, 0b11111110 ; 1= Ausgang
out DDRC, AL
ldi AL, 0x01 ;Bit 0 Pull up
out PORTC, AL

;PCI
ldi AL, 0b00000010 ;erstmal nur A vom Drehgeber
sts PCICR, AL
ldi AL, 1
sts PCMSK1, AL ;Maske auf Bit 0
sts PCMSK2, AL
```

```
.cseg
.ORG 0 ;0
JMP Reset
.ORG INT0addr; = 0x0002 ; External Interrupt Request 0
rjmp EXT_INT0 ;
.ORG INT1addr; = 0x0004 ; External Interrupt Request 1
reti
.ORG INT2addr; = 0x0006 ; External Interrupt Request 2
reti
.ORG PCI0addr; = 0x0008 ; Pin Change Interrupt Request 0
reti
.ORG PCI1addr; = 0x000a ; Pin Change Interrupt Request 1
rjmp DrehA
.ORG PCI2addr; = 0x000c ; Pin Change Interrupt Request 2
rjmp DrehB
.ORG PCI3addr; = 0x000e ; Pin Change Interrupt Request 3
reti
```



DrehA:

```
push AL
in AL, SREG
push AL

ldi AL, 0b00000100 ;PCIE2 - nur B vom Drehgeber auswerten
sts PCICR, AL ;- - - PCIE3 PCIE2 PCIE1 PCIE0 = PCICR
```

```
in AL, PINC ;lese Drehgeber B
sts MPinB, AL ;sichern
```

```
pop AL
out SREG, AL
pop AL
reti
```

DrehB:

```
push AL
in AL, SREG
push AL
push AH

ldi AL, 0b00000010 ;PCIE1 - nur A vom Drehgeber auswerten
sts PCICR, AL ;- - - PCIE3 PCIE2 PCIE1 PCIE0 = PCICR
```

```
in AL, PINB
in AH, PINC
```

```
sbrs AL, 0 ;Überspringe, wenn A vom Drehgeber = 1
rcall DrehB1 ;nur, wenn A vom Drehgeber = 0
```

```
pop AH
pop AL
out SREG, AL
pop AL
reti
```

DrehB1:

```
lds AL, MPinB
cbi PORTC, 6 ;Setze Richtung auf 0
sbrs AL, 0 ;Richtung vom Drehgeber
sbi PORTC, 6 ;Setze Richtung
```

```
pushw z
rcall Schritt
popw z
ret
```

Richtung in MPinB
Beispiel - hier durch eigenen
Programmteile ersetzen

Nach der ersten Flanke
wird umgeschaltet

