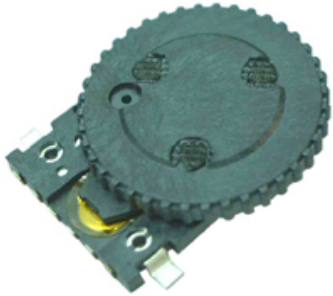


Digital pulsgivare

Inkrementella pulsgivare, vinkel



Lyxvarianten

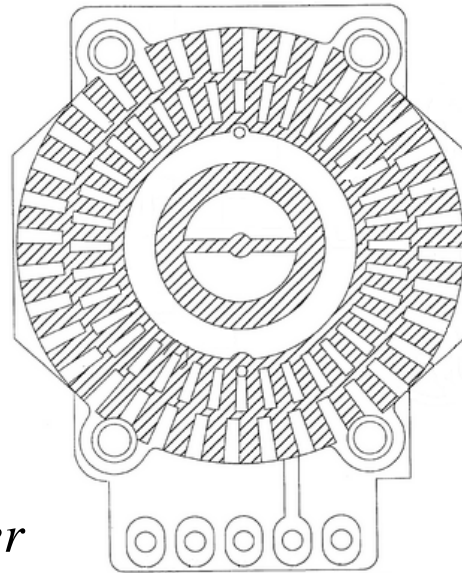


Lägsta pris



2st Fjädrande kontakter

- **Mekanisk avkänning**

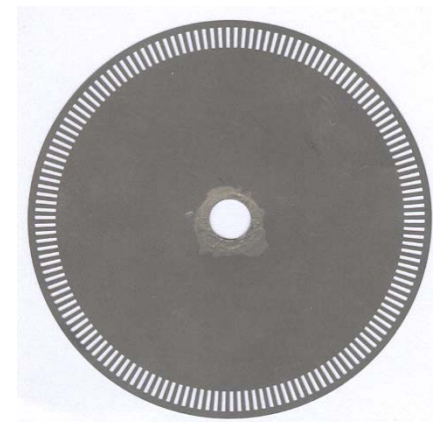


Kontaktmönster

- **Optisk avkänning**



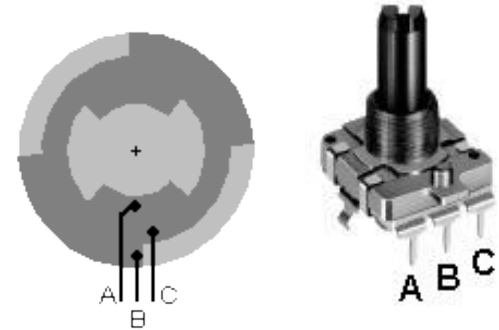
2st Optiska Läsafflar



Hål/Spalt-skiva

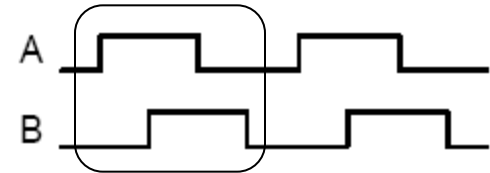
Pulsgivaren

Rotations-Pulsgivare (**RPG**) används ofta som digitala vinkelgivare i industrin, men de används numera även som inställningsrattar och vred i hemelektronik (Jog up/down). De senare typerna har mekaniska kontakter och mass-tillverkas till låga priser (det finns pulsgivare från c:a 20:-), så det finns all anledning att bekanta sig med givartypen.

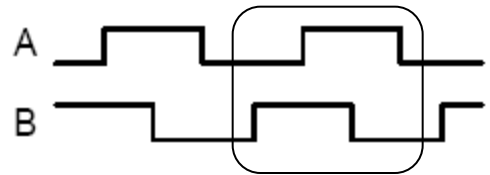


ett ”snäpp”

Medurs rotation



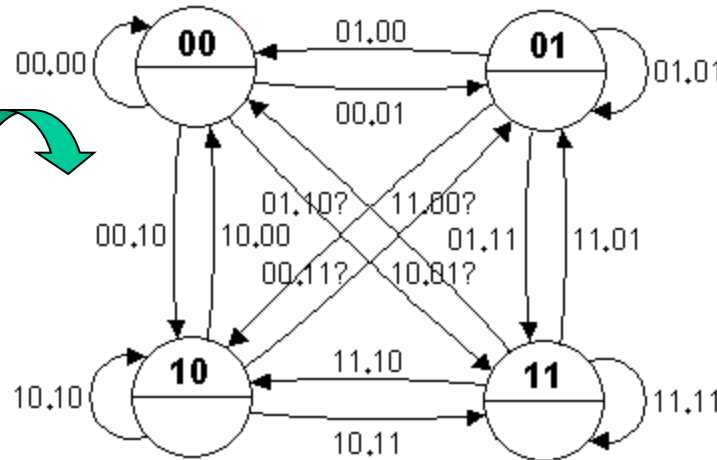
Moturs rotation



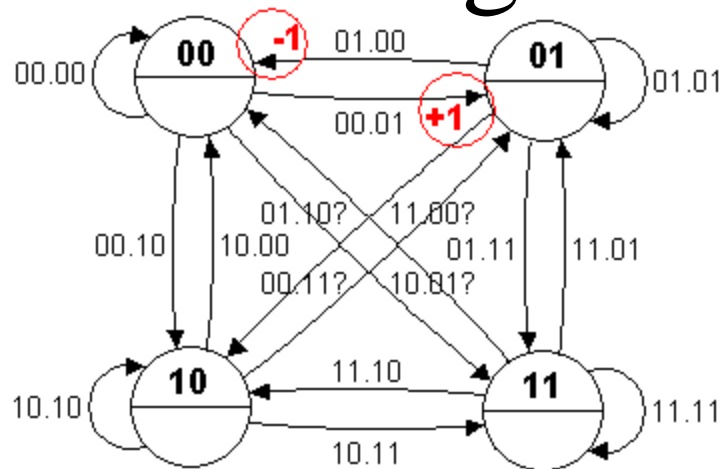
Gray-kod:

... 00 01 11 10 ...

För varje ”snäpp” med pulsgivarens axel så förflyttar man sig **ett varv** i tillståndsdigrammet.



Tillståndsdigrammet

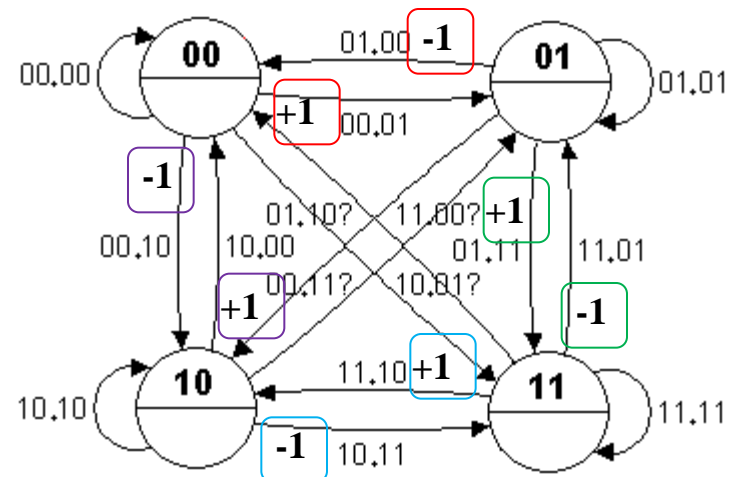
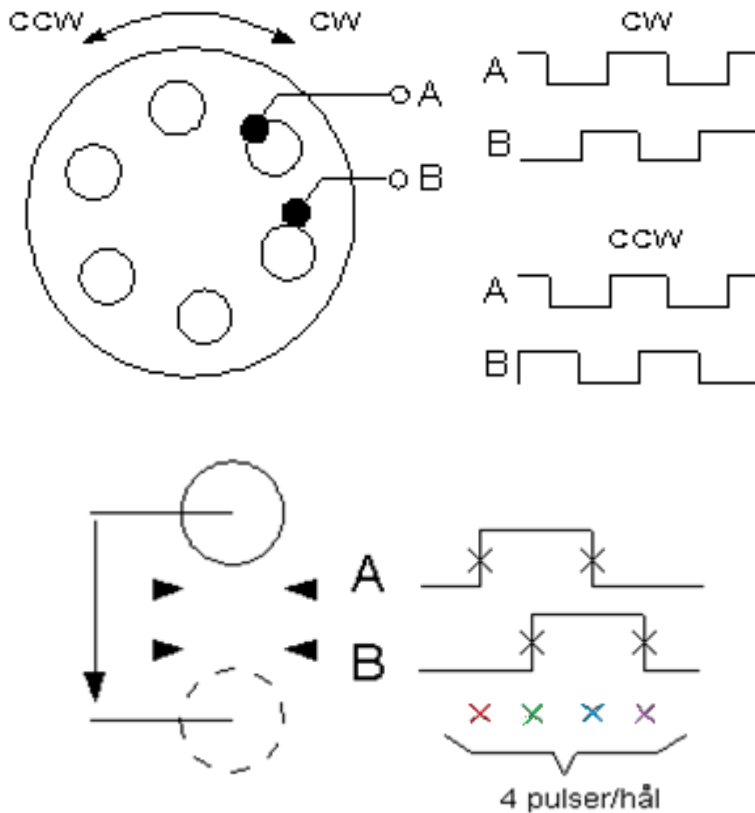


Givarens fyra kontakttillstånd kan ritas upp i ett tillståndsdigram. Mellan de fyra tillstånden finns det totalt 16 olika övergångar (pilar i diagrammet). De fyra diagonala tillstånden är egentligen "omöjliga" och kan bara uppstå av störningar, eller om man missat en avläsning.

- Man kan till exempel räkna upp antalet "snäpp" (+1) var gång man gått från 00→01 i tillståndsdigrammet, och ned (-1) vid 01→00.

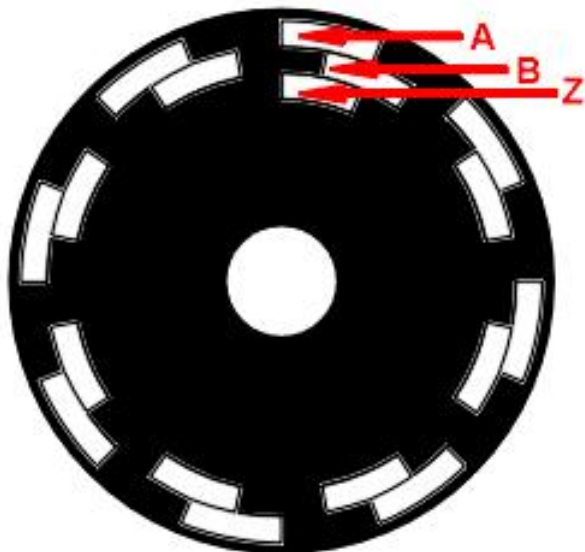
(Digital interpolering)

- Fyrfaldigt ökad upplösning är möjlig.



(Referenspuls)

Inkrementala givare bygger på att man *räknar* och följer med alla förändringar. Man måste då veta var man befinner sig från början?



Ett tredje givare Z ger en referenspuls en gång per varv.

William Sandqvist william@kth.se

(Binära konstanter)

Kompilatorn **Cc5x** tillåter binära konstanter (finns ej i ANSI-C). Man kan dessutom placera ut punkter för att markera vilka bitar som hör ihop och bildar grupper. Punkterna har *ingen betydelse* förutom att förtydliga koden.

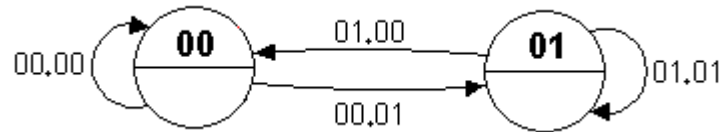
ex. old → new

`0b00.00` ⇒ 0

`0b00.01` ⇒ 1

`0b01.01` ⇒ 5

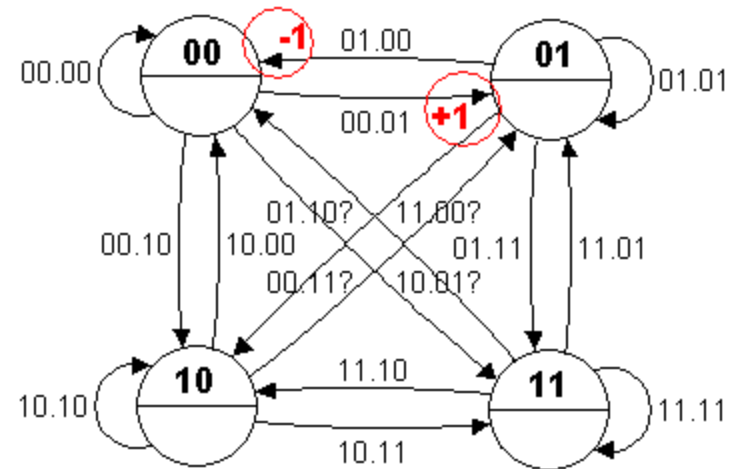
`0b01.00` ⇒ 4



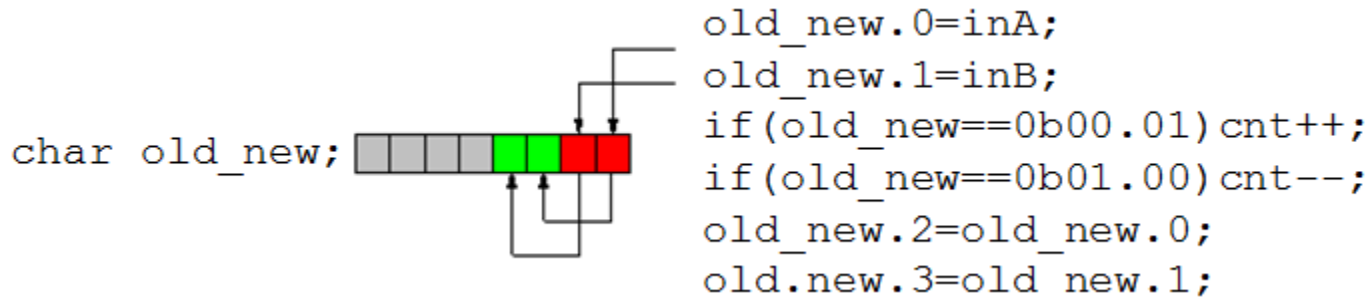
Så här kan man använda de binära konstanterna för att beteckna tillståndsövergångarna.

Räkna pulser

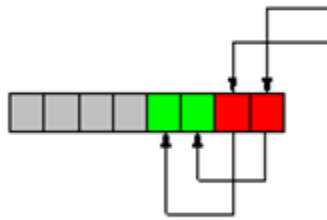
Man lagrar det föregående tillståndet för att jämföra detta med det nuvarande tillståndet. Varje pil i tillståndsdigrammet består av ett sådant tillståndspar **old.new**. Ett enkelt sätt att avläsa givaren är att räkna upp positionen vid pilen **00.01** och ned positionen vid pilen **01.00**.



Även om kontakten studsar blir "nettoresultatet" det rätta, eftersom man ju alltid måste gå den ena vägen *en* gång mer än den andra för att byta tillstånd.




```
char old_new;
```



```
old_new.0=inA;  
old_new.1=inB;  
if(old_new==0b00.01)cnt++;  
if(old_new==0b01.00)cnt--;  
old_new.2=old_new.0;  
old_new.3=old_new.1;
```

```
while(1)  
{
```

```
/* read encoder new value */
```

```
old_new.0 = A;
```

```
old_new.1 = B;
```

```
/* compare with old value */
```

```
if( old_new == 0b00.01 ) cnt--;
```

```
if( old_new == 0b01.00 ) cnt++;
```

```
/* replace old values with new values */
```

```
old_new.2 = old_new.0;
```

```
old_new.3 = old_new.1;
```

```
/* this part takes long time!
```

```
if(cnt != oldcnt) /* print value if changed ? */
```

```
    printf("Position: %d\r\n", cnt);
```

```
    oldcnt = cnt; /* update oldcnt missas! */
```

```
}
```

snabbt
6 μs

långsamt
*/

*Pulser under printf()
missas!*

William Sandqvist william@kth.se

Interrupt?

Medan processorn skriver ut positionen med `printf()` kan den *inte samtidigt* läsa av pulsgivaren – då kan den missa pulser!

”**Interrupt on change**”. PORTB har möjlighet att ge avbrott vid förändringar. Om det i stället är avbrottsrutinen som registrerar pulsgivaren så missar man inga pulser.

- `printf()` får nu *inte* använda ”bitbanging” – avbrotten skulle ”hacka” sönder seriekommunikationen.
- `printf()` *måste* använda den självgående EUSART-enheten som inte störs av interrupt.

Polling och Interrupt



Antag att Du sitter i en skön fåtölj och läser en bok. Plötsligt blir Du avbruten av att *telefonen ringer*, Du markerar med en blyertspenna var i boken Du befann dig och svarar.



Under samtalet ringer det på dörrklockan och Du ber den Du talar med i telefonen med att dröja kvar medan Du går till dörren.

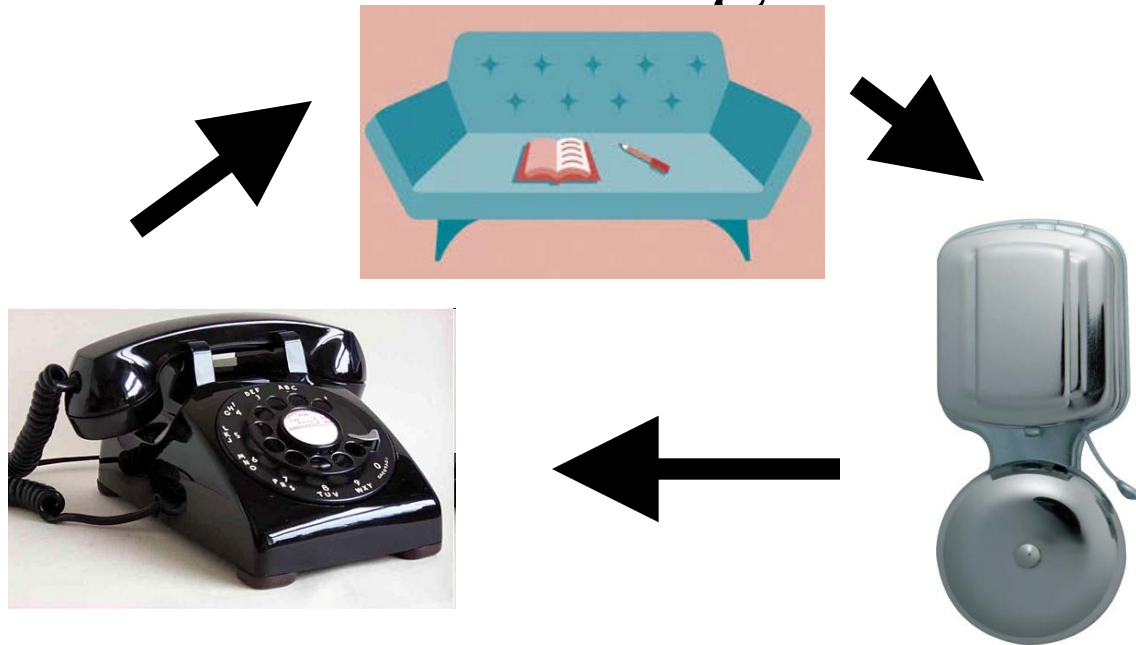


Interrupt

När Du är färdig med ärendet vid dörren *återupptar Du telefonsamtalet*. När Du efter ett tag har talat färdigt i telefonen och avslutat telefonsamtalet kan Du *återvända till fåtöljen* och fortsätta med att läsa den goda boken - vid blyertsmärket.



Polling



Om inte interrupt funnes vore man tvungen att rusa runt mellan dörren – står någon där? – telefonen – någon där? och soffan.

Det som kallas för **polling**.

Interruptmekanismer

Global och Local Enable



Vill Du inte bli störd kan Du sätta på dig öronproppar – Du har då omöjliggjort interrupt, *disable interrupt*.

Tar Du bort öronpropparna har Du åter möjliggjort interrupt, *enable interrupt*. Detta kallas för **Global Enable**.

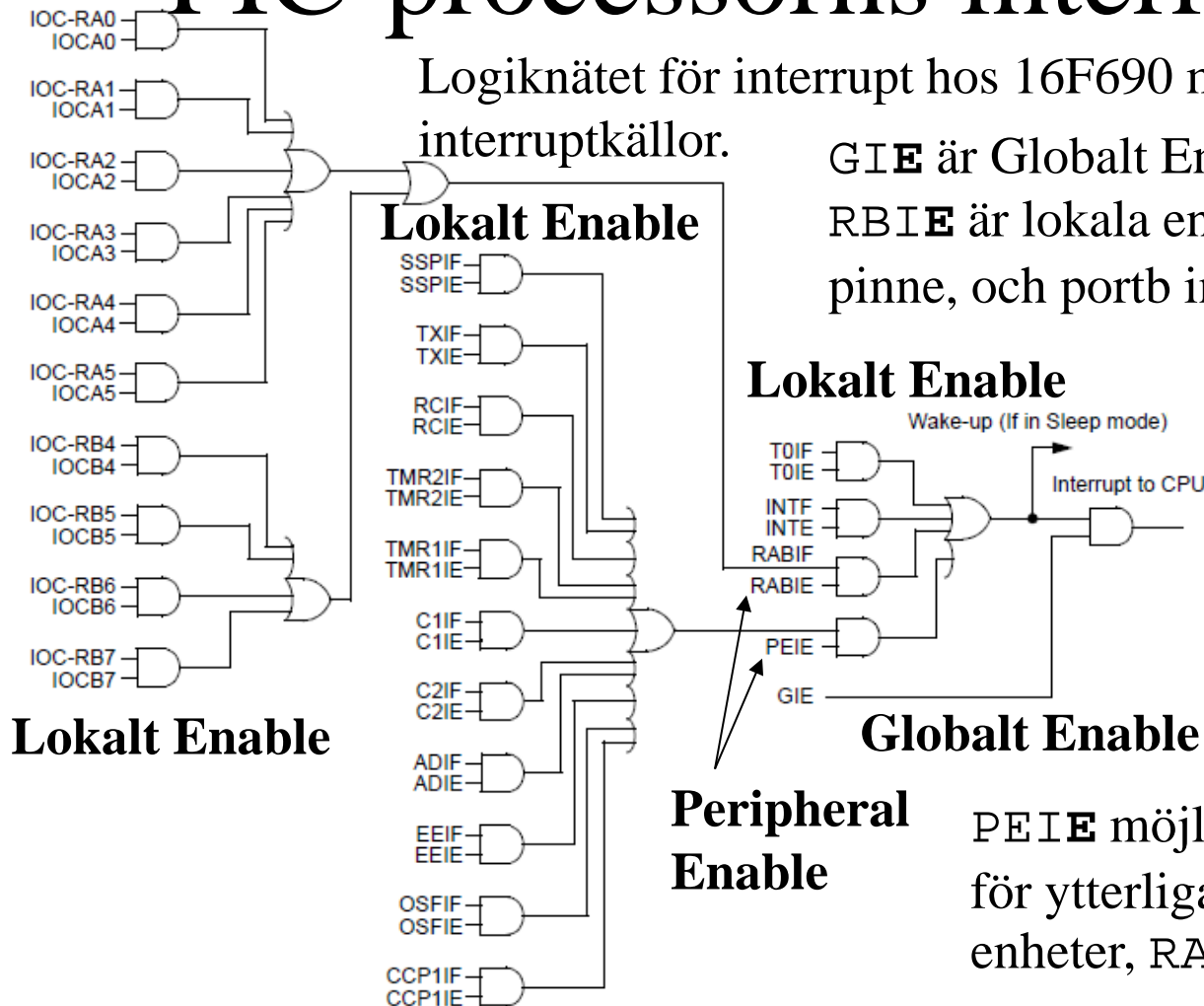
Du har även möjlighet på lokal nivå enabla/disabla interrupt, **Local Enable**. Du kan tex. disabla telefonen genom att dra ur jacket. Då hör Du fortfarande dörrklockan.



PIC-processornas interrupt

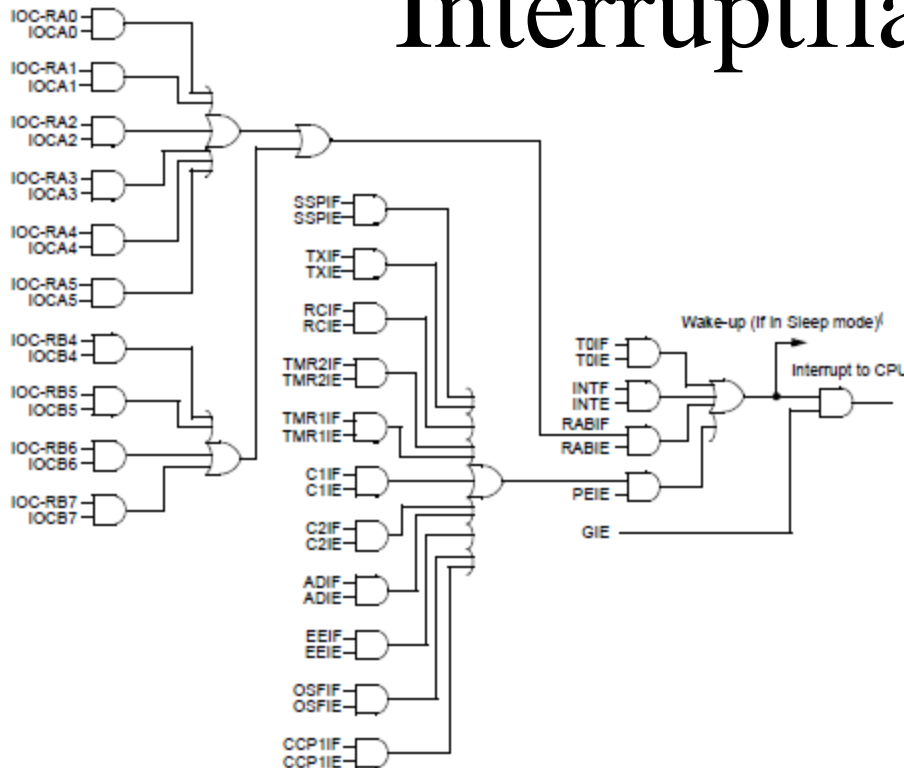
Logiknätet för interrupt hos 16F690 med 23 interruptkällor.

GIE är Globalt Enable. **TOIE**, **INTE**, **RBIE** är lokala enable för timer0, int-pinne, och portb interrupt.



PEIE möjliggör lokalt enable för ytterligare 11 periferi-enheter, **RABIE** för 10.

Interruptflaggor



TMR1IF TMR2IF
CCP1IF CMIF TXIF
RCIF EEIF TOIF
INTF RBIF är namnen
på några "flaggor" som
indikerar olika interrupt-
orsaker.

Om det finns en orsak, *och* källan är lokalt enablad (om det är en periferienhet – den även är perifert enablad) *och* globalt enable gäller – **då blir det Interrupt!**

Interruptflaggor

TABLE 14-6: SUMMARY OF INTERRUPT REGISTERS

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIE2	OSFIE	C2IE	C1IE	EEIE	—	—	—	—	0000 ----	0000 ----
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
PIR2	OSFIF	C2IF	C1IF	EEIF	—	—	—	—	0000 ----	0000 ----

Interruptrutin

Vid interrupt körs Interruptrutinen. Den ligger på fix plats i början av programminnet. Måste ligga först.

```
#pragma origin 4 Interruptrutinen startar alltid på adress 4!  
interrupt int_server( void )  
{  
  int_save_registers ← Makron för att spara register  
  /* interrupt routine */ annars återlämnar interrupt-  
  int_restore_registers ← rutinen förvanskade resultat  
  } till huvudprogrammet!
```

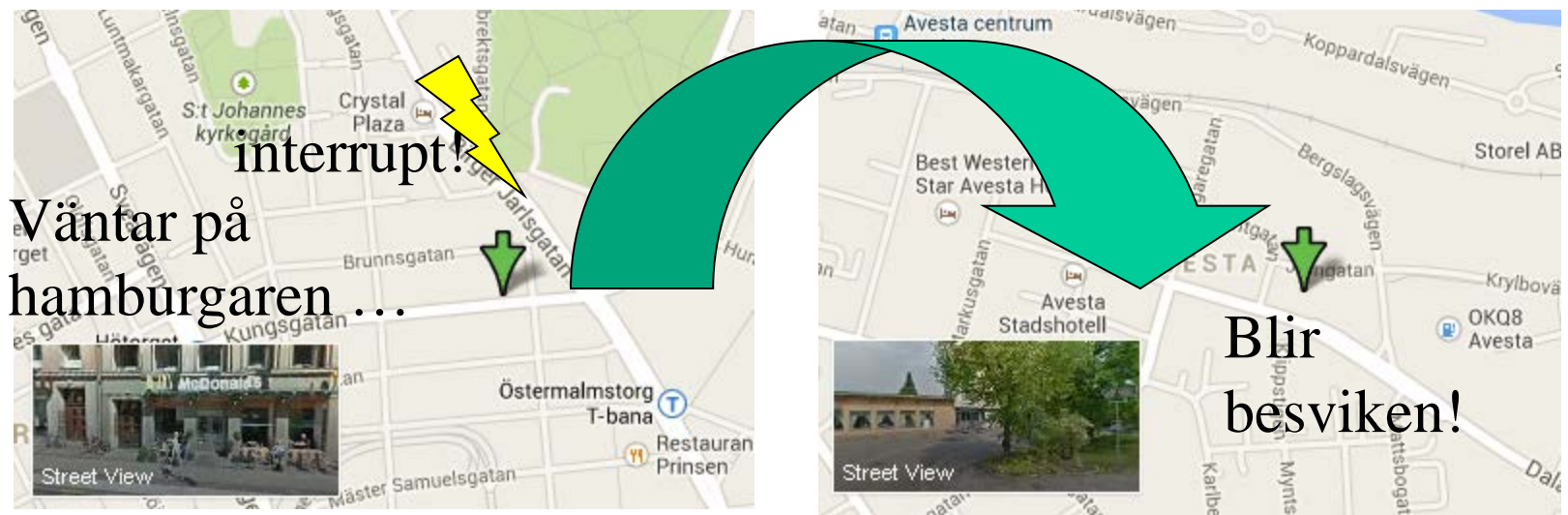
(Avsnitt 14.4 Context saving during interrupt.)

Context saving

Interrupt! Kontexten är viktig, **page**, **rambank** ...

Kungsgatan 4 – Stockholm

Kungsgatan 4 – Avesta



Cc5x sparar det viktigaste – och varnar om **mer** kan behöva sparas.

(Avsnitt 14.4 Context saving during interrupt.)

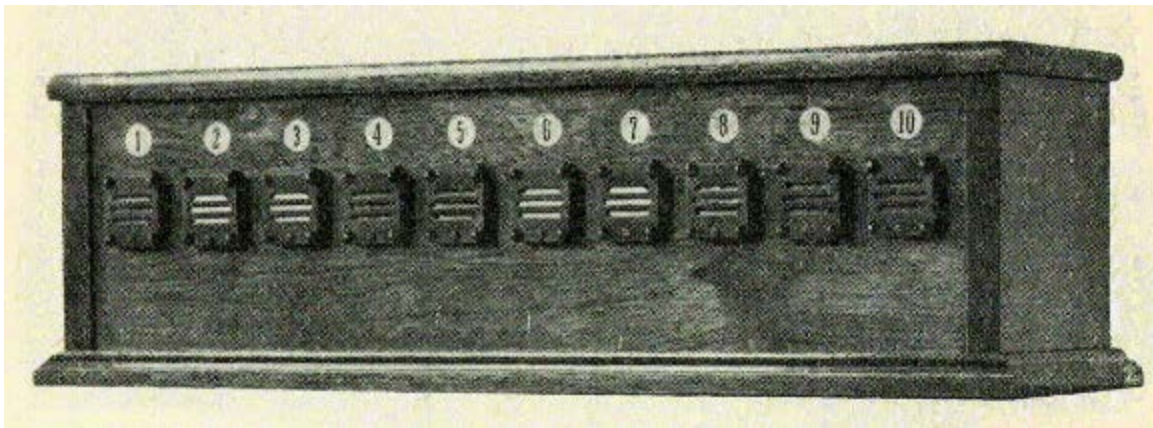
Återställ interruptflaggan

Interruptflaggorna visar vad som orsakat interruptet.

I interruptrutinen kontrollerar man flaggorna och gör vad som behöver göras.

Den interruptflagga som är 1 måste 0-ställas på slutet av interruptrutinen – annars fortsätter interruptet för evigt!

Klaffskåp med blänkare



Ett "klaffskåp med blänkare", en elektromekanisk signalanordning som förekom i början av 1900 talet i paradvåningar. Från tryckknappar i de olika rummen kunde man kalla på serveringspersonalen eller hembiträdet. Ringklockan i klaffskåpet ringde och motsvarande "blänkare" för respektive rum föll ned. När uppdraget utförts tryckte betjäningen på knappen under "blänkaren" så att den återställdes. - Är det någon härifrån Microchip fått idén till sitt interrupt?

William Sandqvist william@kth.se

RPG Interruptprogrammet

```
char old_new; /* global to store transitions */
int cnt;      /* global to store RPG count   */
```

- *Interruptrutinen måste ligga först*

```
#pragma origin 4 /* only place for interrupt routine */
interrupt int_server( void )
```

```
{
```

```
    int_save_registers
```

```
    old_new.0 = PORTA.5;
```

```
    old_new.1 = PORTA.4;
```

```
    if( old_new == 0b00.01 ) cnt ++;
```

```
    if( old_new == 0b01.00 ) cnt --;
```

```
    old_new.2 = old_new.0;
```

```
    old_new.3 = old_new.1;
```

```
    RABIF = 0; /* Reset flag before leaving */
```

```
    int_restore_registers
```

```
}
```

*körs varje gång något
ändrar sig på porta*



till huvudprogrammet

main() -programmet

- *main()* och andra funktioner följer sedan

```
void main( void)
{
    init();          /* init ports          */      Interrupt on
    RABIE    = 1;    /* local enable    */      change porta
    GIE      = 1;    /* global enable   */
    initserial();   /* init serial unit */
    new_old = 0; cnt = 0;

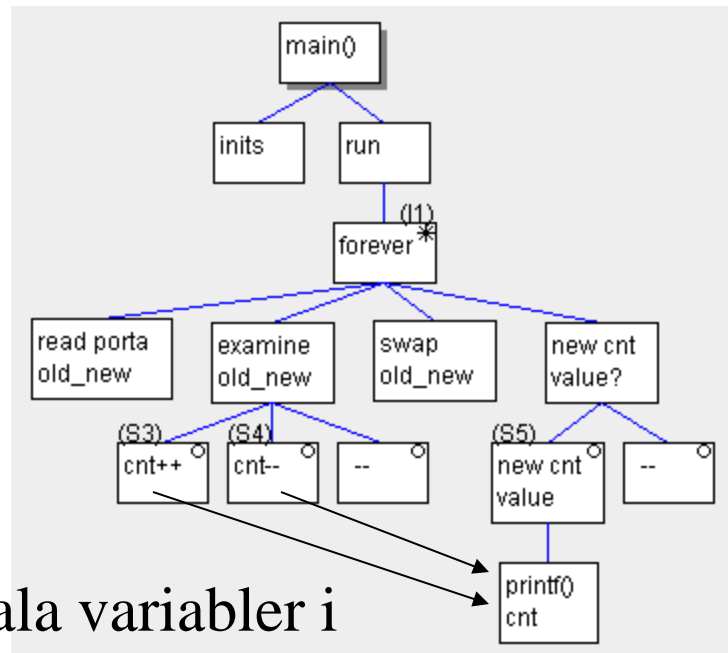
    while(1)
    {
        printf("Position: %d\r\n", cnt );
        delay10(100); /* print RPG-count every second */
    }
}
```

från ISR



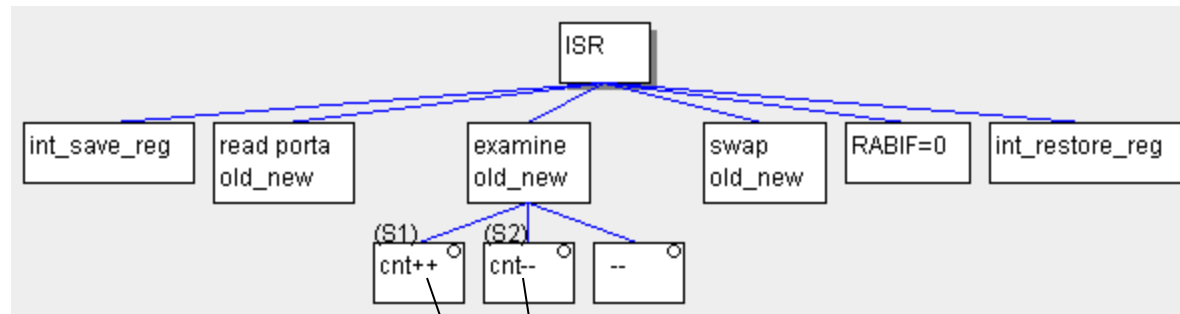
Ingen puls missas i cnt, och värdet skrivs ut utan störningar varje sekund

RPG *utan* interrupt

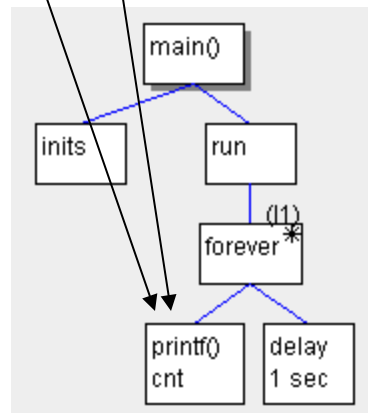


Lokala variabler i
`main()`.

RPG *med* interrupt



Globala variabler
utanför main().

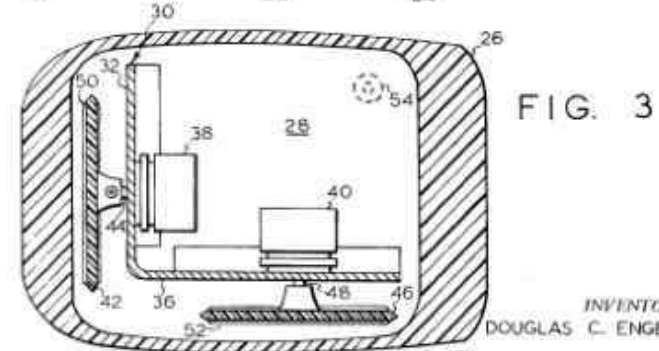
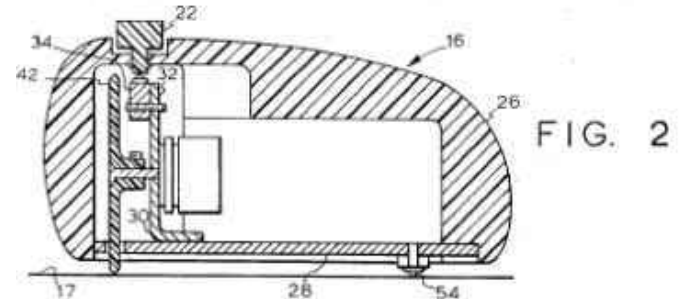
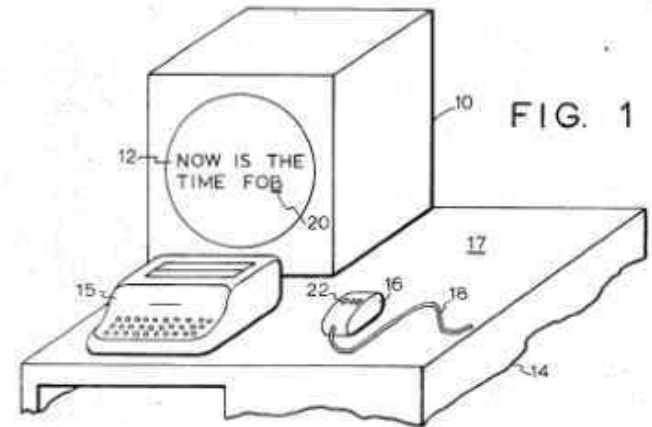
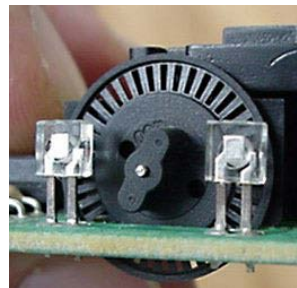
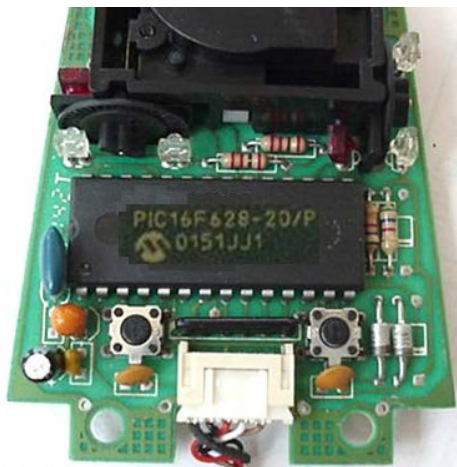


Datormusen...

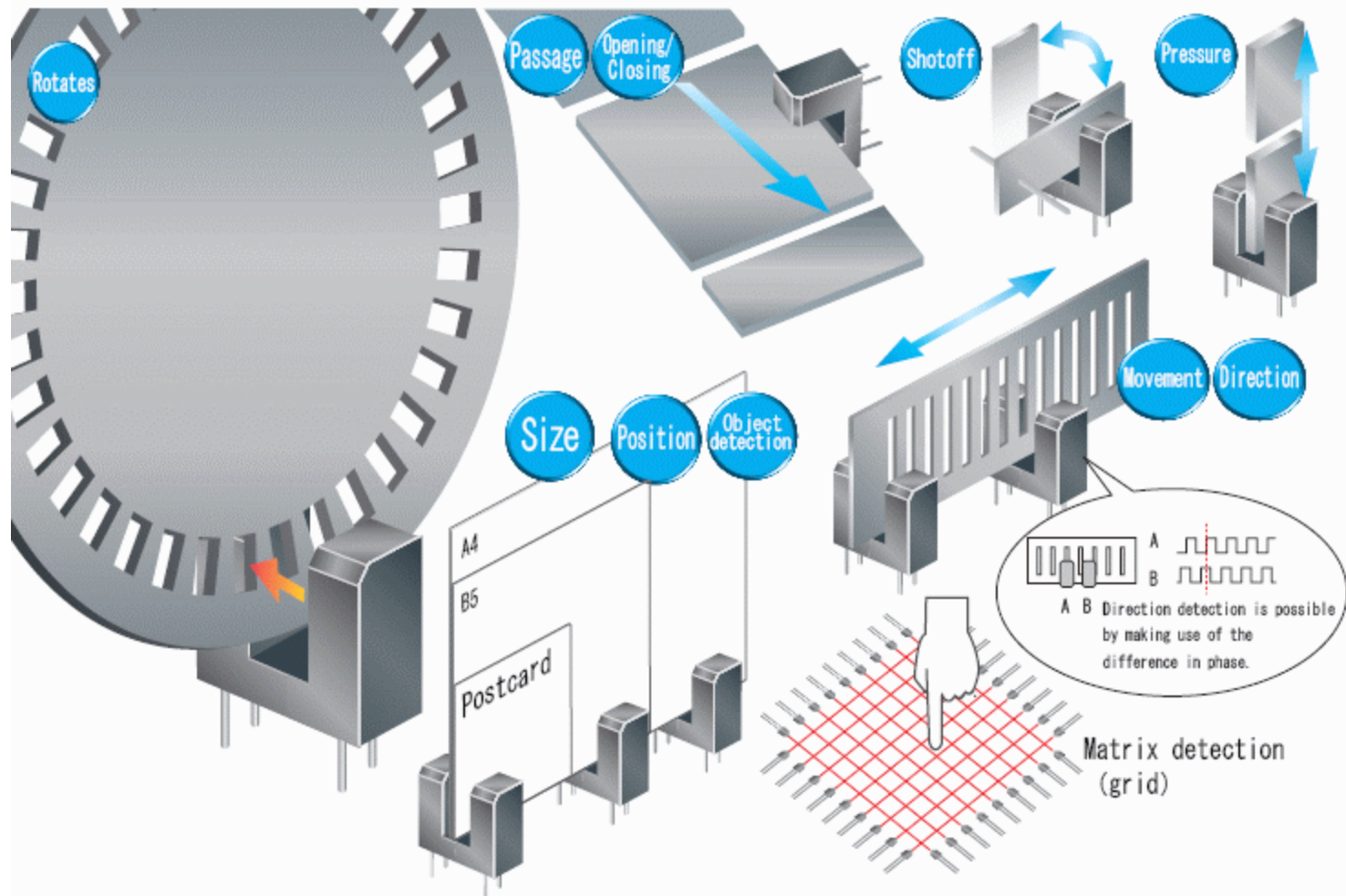
En datormus innehåller två pulsgivare, fast numera optiska.

PIC16F690 har "Interrupt on change" för fyra PORTB-pinnar och 6 PORTA-pinnar vilket räcker till fem pulsgivare!

- Så det kan mycket väl vara en PIC-processor som är chippet inuti musen!



INVENTOR
DOUGLAS C. ENGELBART
BY
Lindberg + Frickel



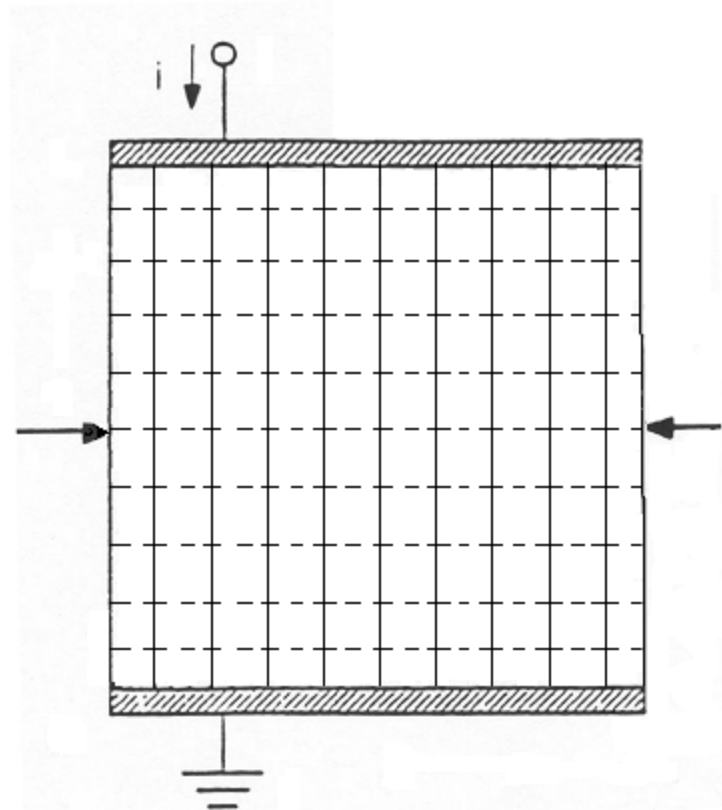
Några andra användningsområden ...

William Sandqvist william@kth.se

Magnetisk sensor

En platta genomflytes av en ström mellan två av sidorna. Strömbanorna blir parallella och laddningsfördelningen i skivan homogen.

De två elektroderna (pilarna) ligger längs samma spänningslinje, och det blir ingen resulterande spänningsskillnad dem emellan.

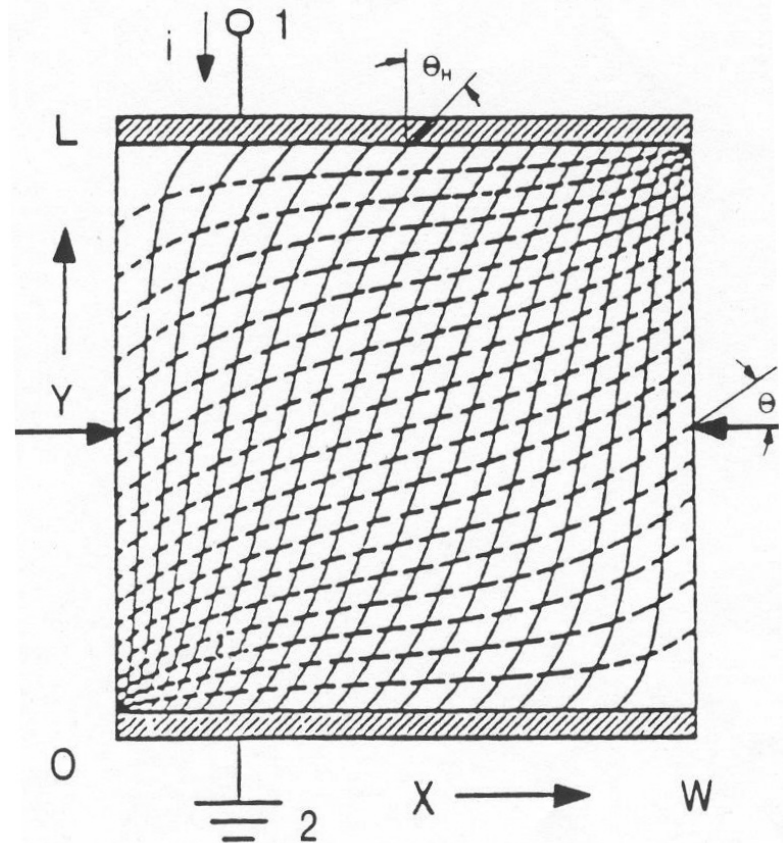


Magnetisk sensor

Nu tvingar ett magnetfält laddningarna ”ur kurs”. Strömbanorna böjer av, och laddningsfördelningen blir ojämn.

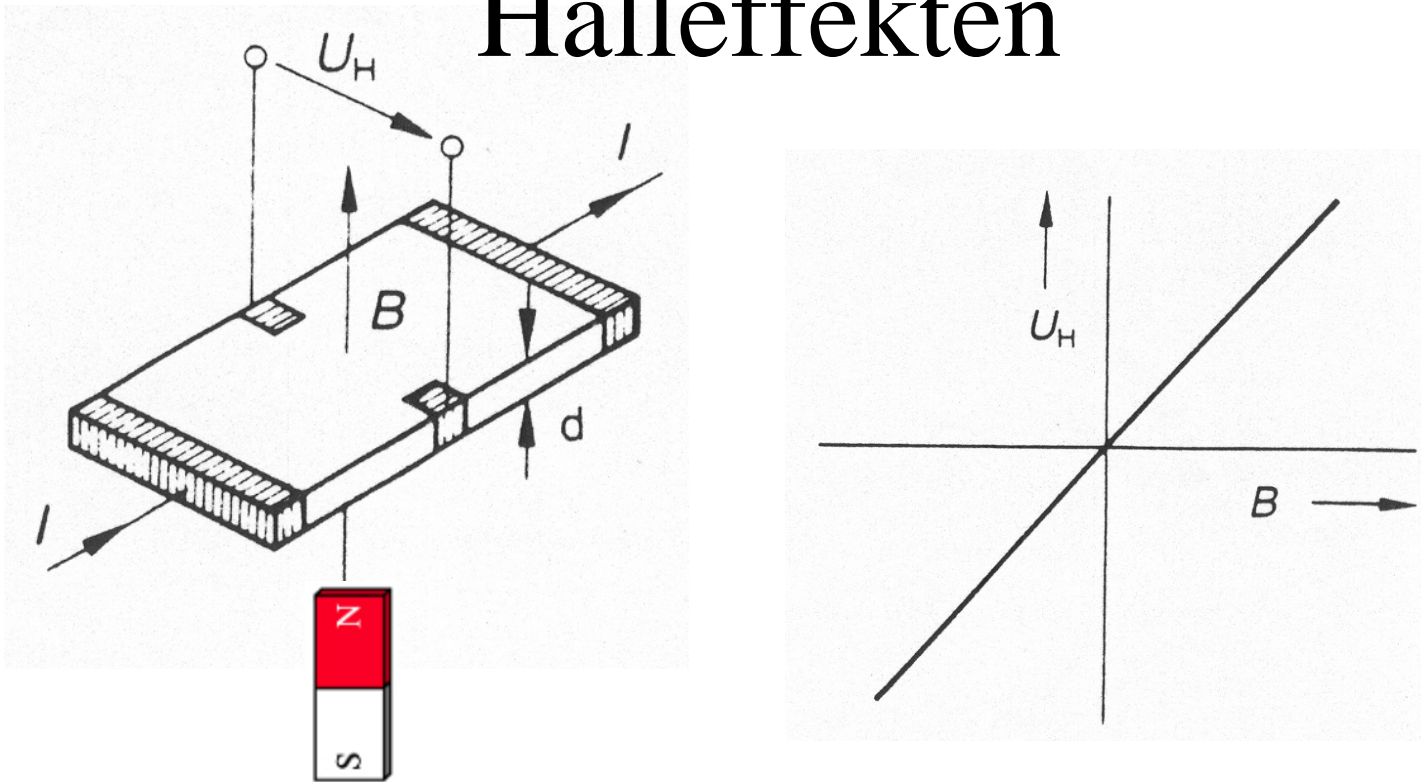


De två elektroderna (pilarna) ligger nu på olika spänningslinjer, och det uppkommer en spänningsskillnad.



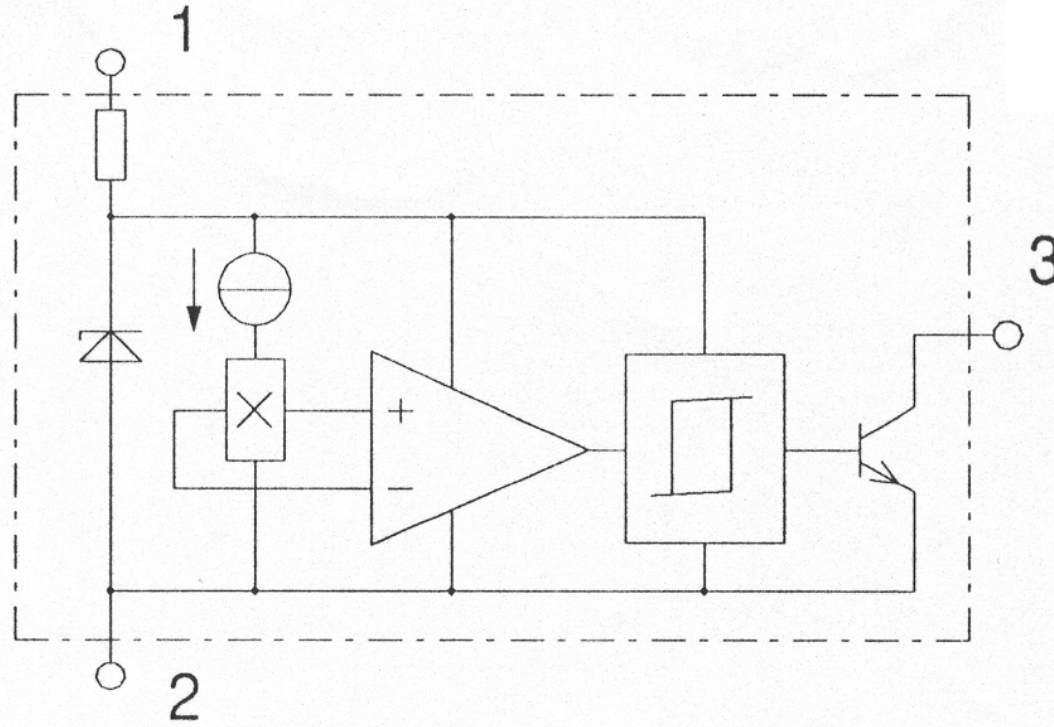
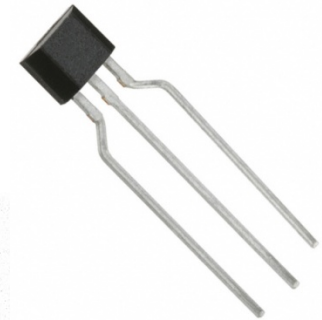
Ju starkare magnetfält, desto större spänning mellan pilarna!

Halleffekten



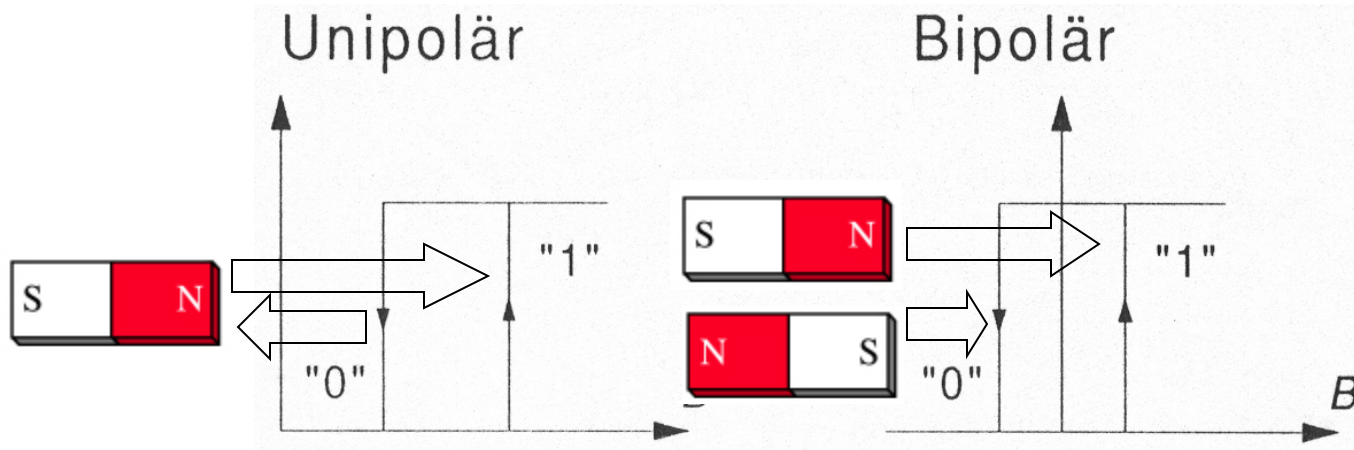
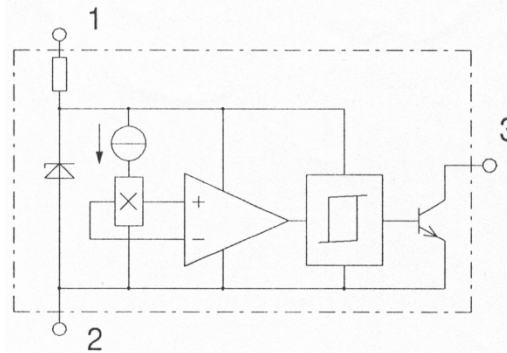
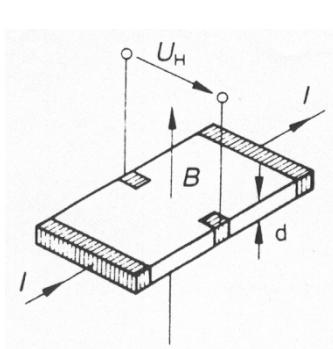
En svag Hallspänning U_H proportionell mot magnetfältets flödestäthet B , indikerar närvaron av magneten.

Halls witch

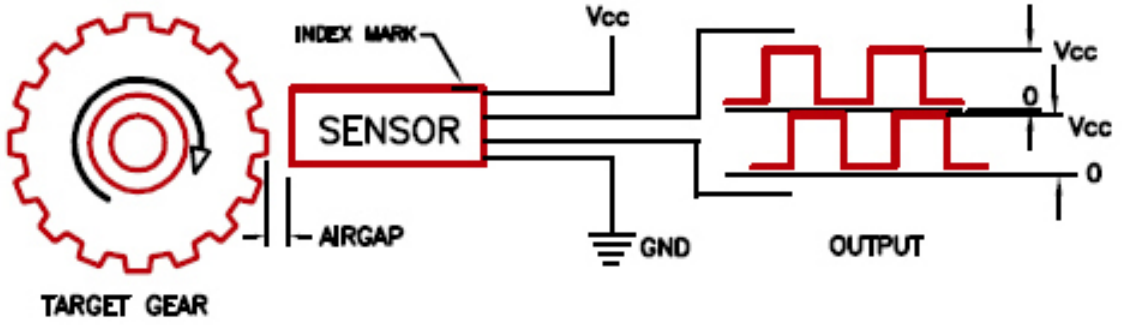
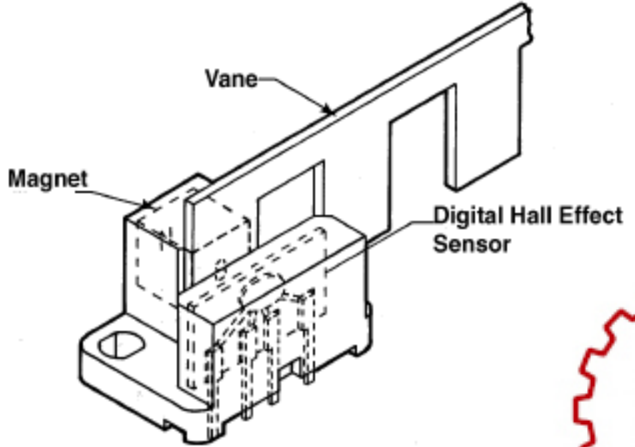
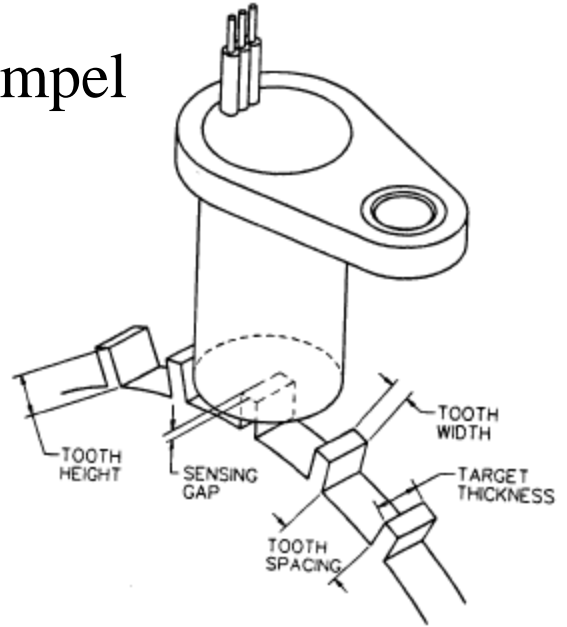
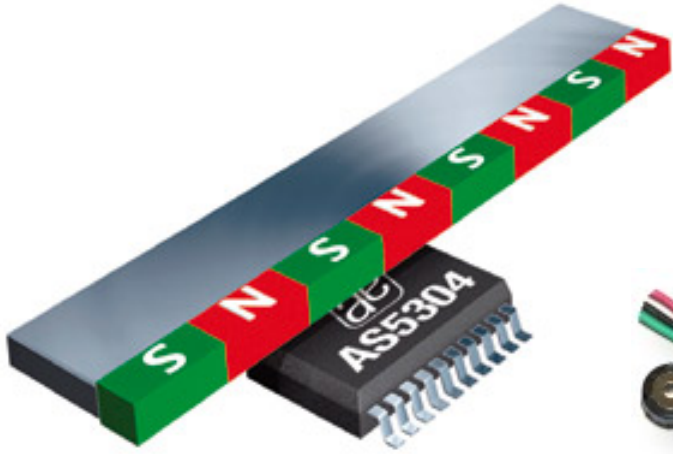


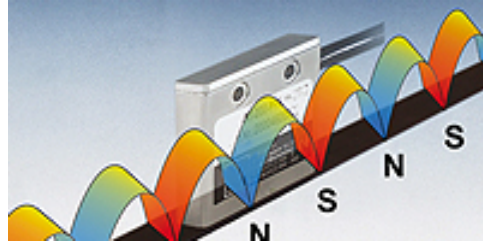
Strömregulator, Hallelement, förstärkare, Schmitt-trigger, drivsteg.

Hallgivare Unipolär/Bipolär



Några olika tillämpningsexempel





Hallswitchar över en utbredd magnettejp – du får se något sådant vid laborationen . . .

William Sandqvist william@kth.se