

## Trin for trin design proces for microcontollere.

I det følgende kommer et trin for trin forslag til hvordan man designer noget nyt med microcontollere. Der tages udgangspunkt i Pic serien af controllere fra Microchip men den beskrevne proces kan også bruges til andre fabrikater.

Dokumentet er tiltænkt som et supplement til den pic programmeringskursus der skal afholdes i Næstved afdelingen af EDR. Den vil blive opdateret med efterhånden som der kommer nye ting til som forfatteren ikke havde fantasi til at tænke på. (Jeg mener at når man fortæller folk hvordan de kan konditræne så skriver man vel ikke om at de skal huske at trække vejret. Sagt på en anden måde: Os der har prøvet det her før tager vi mange ting for givet som ikke er nødvendigvis givet for en nybegynder.)

### Trin 1

Hvad skal du bruge? Dvs. hvad er formålet med et nyt design? Skal du bruge et nyt varme styring til huset, eller er der tale om et synthese loader til en seriel loaded synthesizer? Hver applikation har sin udfordring. Det man skal gøre sig klart er at det der er meget vigtigt at tage stilling til, hvordan man vil "kommunikere" med sin opstilling. Eller sagt på en anden måde, hvilken måder man vil påvirke controlleren til at udføre forskellige opgaver NÅR den ER programmeret og fungerer i opstillingen. Det er nemlig ofte sådan at selve styringen er ret simpel men det man i fagsproget kalder "MMI" (Man Machine Interface), kommunikation mellem bruger og maskine er ret kompliceret. Der skal bruges knapper, en eller anden form for display, indikator lamper, og hvad man ellers kan finde på for at vise hvad maskinen har gang i. Her er et bud på en checkliste:

- Hvad skal du bruge den til?
- Hvor mange ind og udgange skal der være?
- Hvilken type ind og udgange skal du bruge? (analog, digital)
- Hvilken display type vil du bruge? (Alfanumerisk, 7 segment, lysdioder)
- Hvilken type brugerflade skal der være? (tastatur, RS232, (dvs. pc), ingen)
- Hvilken type funktioner skal der være en del af konstruktionen? (ur/kalender, logning)
- Hvilken måde vil man styre/overvåge ting? (Relæer, analoge følere, IIC bus, RS232, osv)

Det ovenstående skal ikke betragtes som en udtømmende liste men som inspiration til de spørgsmål man kan stille sig selv inden man går i gang.

## **Trin 2**

Tegn din forplade, tilslutninger, følere, styreenheder (relæer, motorer,). Skriv din brugervejledning til konstruktionen inden du gør noget som helst andet!!

Når du har tegnet din forplade så ved du også hvilken ting du skal bruge til betjening af systemet. Find tingene frem! Hav den i hånden! Se hvilke tilslutninger der er brug for at de kan fungere korrekt. Skaf et overblik over hvilken type ind og udgange du har brug for fra processoren for at kunne tilgodese alle de ind og udgange som konstruktionen/opgaven har brug for.

## **Trin 3**

Anskaf ALLE komponenter og byg konstruktionen. Det kan måske godt virke lidt forhastet men der er ikke andet at gøre end at gå i gang. Man kan alligevel ikke teste hverken HW eller SW uden at der er en "platform" man kan teste på. Det kan ligeså godt være den endelige konstruktion. I den forbindelse vil jeg gerne gøre lidt reklame for internettet. Her kan man finde løsninger på mange ting og det er en god ide at få checket hvordan andre har løst de udfordringer som dukker op i fbm. en ny konstruktion. Der er ingen grund til at genopfinde den "dybe tallerken".

## **Trin 4**

Test dit HW!! Brug al den tid der er nødvendigt for at sikre dig at HW'en fungerer. Skriv om nødvendigt små korte programmer hvor du kan teste at tingene fungerer som tiltænkt. Det er her man ofte opdager at der mangler noget, eller ting opfører sig en lille smule anderledes end det databladet antyder. Skriv en lille rutine der skriver "Hello" i den display du evt. måtte have valgt. Prøv at vise tallene du trykker på et tastatur hvis du har sådan en. Prøv at aflæse værdien af en føler hvis der er nogen. Tænd og sluk for relæer for at se at de virker. TEST ALT!!! Det er uhyre vigtigt at du er HELT sikker på at konstruktionen virker inden du går i gang med den helt store programudvikling. Der er ikke noget som er mere håbløst end at slås med et stykke program der ikke gør som man selv vil og fejlen egentligt befinder sig i HW'en.