++ Fokus på sproget #05

in C++ Fokus with 0 Comments

Funktioner

Endelig! Nu er vi nået frem til det sjove. Funktionerne.

Funktioner er

- kernen i vores kode.
- der, hvor tingene sker.
- det sted i koden, hvor der bliver beregnet, talt op og givet ordrer.
- rigtig smarte, hvis man har kode, som skal bruges igen og igen.
- mestrer man at oprette og kode funktioner og kan skelne mellem de forskellige typer af funktioner, – ja så har man fat i noget meget centralt i sprogets struktur.

Når man opretter en ny Arduino sketch, er den på forhånd forsynet med to funktioner, som er navngivet henholdsvis 'setup' og 'loop'. Kigger vi nøjere på dem, kan vi se, at funktioner har deres egen syntaks. Der er regler for, hvorledes man erklærer dem og hvorledes man anvender parenteser. Det kommer vi tilbage til.

Lad os kigge på, hvorledes man bygger en sketch op med funktioner. Når man skal lave en ny sketch, skal man selvfølgelig gøre sig klart, hvilke opgaver, man ønsker udført. Ofte, og ikke mindst mens man stadig er newbie, kan det være en rigtig god ide at beskrive de ønskede hændelser i almindeligt, men kortfattet sprog. Noget af det nemmeste man kan få sin Arduino til, er at tænde eller slukke for den indbyggede LED. Lad os kigge på, hvorledes vi får den til gentagne gange at sende bogstavet a i morse. HW? som vi siger i cw-verdenen.

Den kortfattede beskrivelse af vores sketch kan derfor sammenfattes i en funktion, der gør følgende:

- 'send et a'

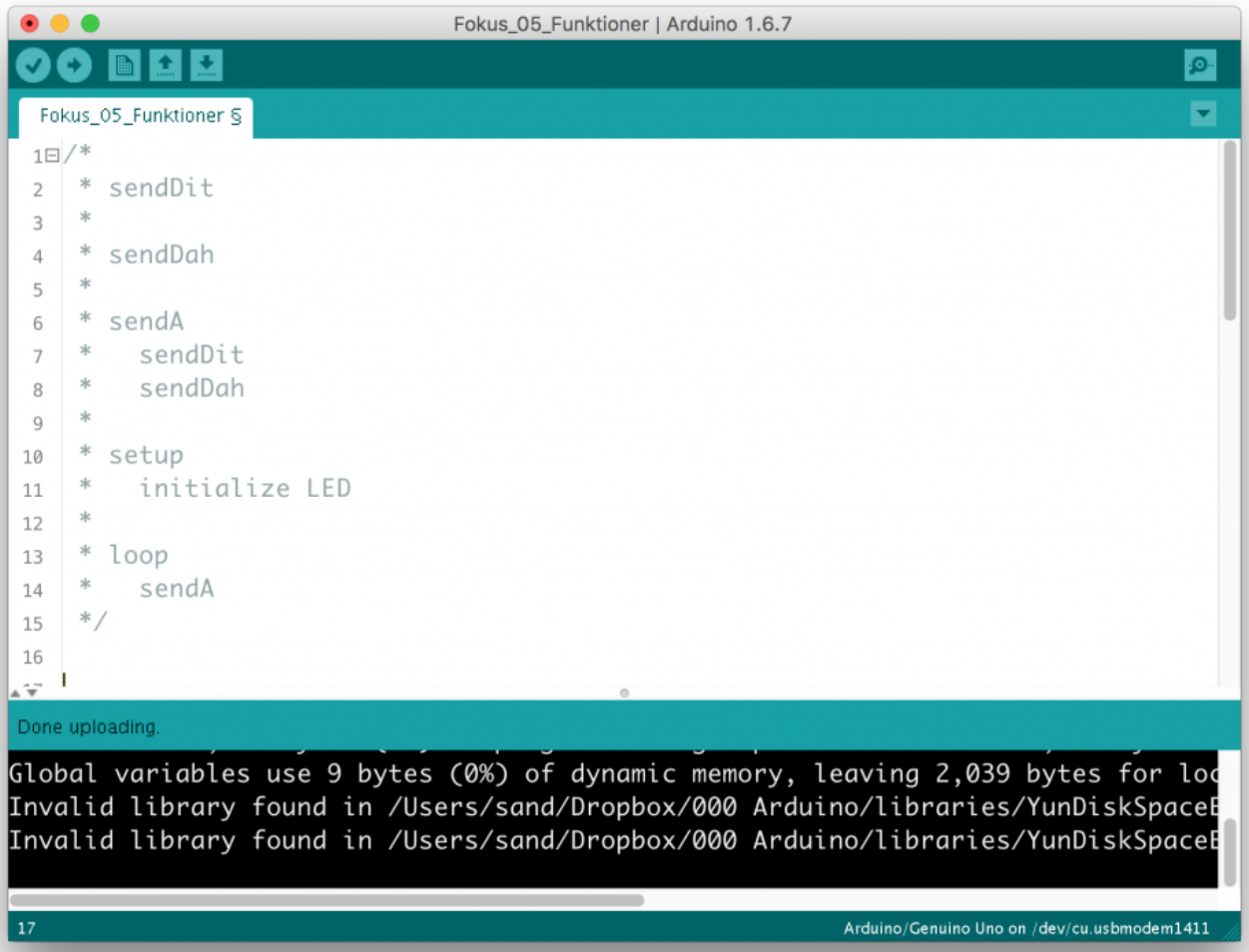
Selv uden at have bestået morseprøven kommer man hurtigt frem til, at vi får brug for to hjælpefunktoner:

- 'send PRIK'
- 'send STREG'

Sketchen kommer skal derfor indeholde disse tre funktioner, hvor den nederste **'send et a'** har brug for de to øverste:

- 'send PRIK'
- 'send STREG
- **'send et a'**

Allerede her kan vi se en af de store fordele ved funktioner: De kan bruges igen og igen. Når først vi har fyldt noget kode i vores funktioner, kan vi ved hjælp af 'send PRIK' og 'send STREG' hurtigt skrive kode, der sender et hvilken som helst tegn. For at sende en prik, skal den indbyggede LED lyse i et antal sekunder og derefter skal den være slukket i et antal sekunder. Hvis vi medtænker de to faste funktioner, kan vores pseudo-kode se således ud:



At skrive sådan kalder programmører at arbejde i 'pseudo-kode'. Man kan arbejde med blyant og papir, man kan bruge en tekstbehandler eller benytte lige præcis den skrivemetode, man bedst kan lide. Når det er små, ukomplicerede projekter, skriver jeg altid min pseudo-kode direkte i selve Arduinos IDE. Så har jeg min pseudo-kode lige ved hånden, når den skal skrives ud som rigtig kode.

Nu skal koden ud af kommentartegnene. Vi skal til at skrive rigtig kode

- Kunne man så ikke lige så godt have begyndt at skrive kode med det samme? Jo! er svaret, hvis man er en erfaren programmør, som er godt inde i sproget. Men som begynder kan man næsten ikke overdrive det at skrive pseudo-kode, fordi man derved tvinger sig selv til at tænke konkret over HVAD der skal gøres. Med andre ord er behovet for at skrive pseudo-kode omvendt proportionalt med ens færdigheder udi C++. Men uanset hvor kompetent man er, er det ALTID en god ide at lave skitser til sin sketch og tænke den igennem på et overordnet plan INDEN man koder løs. Alt for mange kaster sig ud i projekter, som ender med at være kodet hjertegribende dårligt og uoverskueligt, fordi man så at sige arbejder sig indefra og ud.

Bemærk to ting angående navngivning af funktioner.

For det første bestræber jeg mig på at indlede navnet med et udsagnsord. Det gør jeg for at fortælle, HVAD min funktion gør. Det fortæller meget mere, end hvis jeg blot havde kaldt dem ‘Dit’ og ‘Dah’. Det er en stor hjælp for andre og for mig selv, hvis jeg efter lang tid skal arbejde videre med den. Min kode bliver simpelthen lettere at læse.

For det andet indleder jeg navnet med små bogstaver, mens resten er CamelCase. Så ser man med det samme, at det drejer sig om en funktion.

Mine tre funktioner kommer derfor (med parenteser) til at hedde:

- sendDit()
- sendDah()
- sendA()

Inden man koder, skal man lige være klar på følgende:

- Den indbyggede LED sidder på pin 13.
- En streg er tre gange længden af en prik.
- Efter et tegn skal der være en pause af samme længde som en prik.
- Efter et bogstav skal der være en pause af samme længde som en streg.
- Hvis man fastsætter speed som words per minute WPM, kan man beregne længden af en prik som 1200/WPM.
- Man aktiverer en funktion, ved at foretage et kald til den. Man skriver dens navn efterfulgt af to parenteser.

Koden kommer derfor til at se således ud: