

C++ Fokus på sproget #02

in [C++ Fokus](#) with 0 Comments

Arduinos sprog og IDE

Selve Arduino controlleren, som er en Atmega328, hvis det drejer sig om en UNO og en Atmega2560, hvis det drejer sig om en MEGA, skal fodres med noget kode for at fungere. Koden skal på passende vis opbygges, så vores Arduino foretager de handlinger, som vi ønsker den skal gøre. Hvis vores Arduino direkte skal forstå vores kode, blive vi nødt til at skrive i maskinkode. Maskinkode er en række instrukser, som controlleren kan forstå. For de fleste er maskinkode ganske og aldeles uforståeligt og uanvendeligt som arbejdssprog. Skulle man have lyst til at se, hvorledes det ser ud, kan man få syn for sagen på dette [link](#).

Heldigvis er der en genvej til controlleren, idet man kan skrive sin applikation i et højniveau sprog, der minder noget mere om vores almindelige sprog. Denne højniveau-kode bliver så efterfølgende automatisk transformeret til maskinkode. En 'applikation' hedder i Arduino-verdenen en 'sketch'. Processen med at omforme sketchen til maskinkode kaldes kompilering. Når det er sket, kan sketchen uploades til Arduinoen og vi er i gang. Alle der har arbejdet med Arduino kender det IDE (Integrated Development Environment), som frit kan downloades. Sådant et typisk IDE er en slags programmørernes schweizerkniv: Det indeholder en editor, en compiler og en debugger, så man kan teste sin kode undervejs. Skal man skrive meget kode, er det en meget stor hjælp, hvis ens IDE har auto completion, – kommer med kvalificerede gæt på, hvad du ønsker at skrive. Det lever Arduinos indbyggede IDE ikke op til, men da de fleste arbejder i dette IDE, er det udgangspunktet for kurset.

Der bliver på et tidspunkt afholdt en workshop i klubben, hvor man kan få hjælp til at installere et andet IDE. Det kan være lidt kompliceret at installere, – til gengæld er det gratis, hvis man kun anvender det til private formål.

'editor', 'compiler', 'debugger', 'auto completion' – programmeringens verden er en engelsksproget verden. Det kan man lige så godt indrette sig på først som sidst. Uanset om man prøver at fordanske sin sketch, vil væsentlige begreber og reservede ord ALTID være på engelsk. Så det eneste man får ud af at mixe, er noget kode, som bliver klunget og svær at læse.

Og husk: Din sketch skal bruges af din Arduino, men LÆSES af mennesker. Derfor skal du skrive den så letlæselig som muligt.

Det såkaldte højniveausprog, som man koder en Arduino i, er i det store og hele identisk med sproget C++. Og det er en gave til os programmører, for hvis man ikke kan få hjælp til dette eller hint problem med sin sketch i Arduino community, så findes der et endnu større community, der beskæftiger sig med C++. Faktisk søger jeg oftest mine løsninger hos C++ folket inden jeg søger på Arduinos forum.

Så i den sammenhæng er en letlæselig kode skrevet på engelsk. Jeg læste for kort siden på et forum et spørgsmål fra en tysker, der søgte hjælp til et problem i C++. I stedet for hjælp fik han en spand skidt i hovedet, for han havde navngivet på tysk, og det gjorde det uoverskueligt for ikke-tysksprogede at komme ind til kernen af, hvad der var på spil.

Skriv engelsk!

- og NU skal der skrives noget kode. Se næste post.