Lärarhandledning - labb 2

Den här lärarhandledning ska ses som ett komplement till laborations instruktionerna. https://www.arduino.cc/reference/en/ är också en bra resurs. Där finns alla inbyggda funktioner, alla variabeltyper och alla kontrollord som behövs. Där finns även länkar till ordlistor och en lista på inbyggda bibliotek.

En rekommendation är att löda fast pinsen som följer med sensorn BMP280 för att vara säker på att pinsen har så bra kontakt med sensorn som det går. Det går att använda sensora utan fastlödda pins, men man behöver hålla den på plats så att pinsen har ordentlig kontakt med sensorn för att få rimliga värden. Se Appendix 2 för tips om lödning.

Om loopar

Om man vill repetera en bit kod utan att lägga den i `loop()` eller bara repetera den ett begränsat antal gånger eller tills ett visst kriterium uppfylls använder man en for- eller while-loop. Syntaxen för dessa är

- `for (start; kriterium; steg) {...}`

(notera semikolon mellan parametrarna i definitionen).

`start` genomförs först och är nästan alltid en variabeldeklaration.

`kriterium` testas innan varje varv i loopen, om det är sant utförs koden i steg och sedan körs koden i loopen. Ofta är detta en kontroll av hur många varv loopen har gjort.
steg är en bit kod som kör varje gång kriterium är sant. Oftast en ökning av variabeln som definierades i start.

Exempel:

```
for(int i = 0; i <= 10; i++) {
      Serial.println(i);
}
```

kommer skriva ut siffrorna 0 t.o.m. 10

Den andra typen av loop är en while-loop. Den används när något ska repeteras tills ett kriterium uppfylls snarare än ett bestämt antal gånger. Syntaxen är

- `while(kriterium) {...}`

E-post: cansat@au.se
Telefon: 070-000 90 56

där kriterium utvärderas innan varje loop och om det är sant så kör loopen, är det falskt så bryts den. Notera att det är lätt att skapa en while-loop som aldrig bryts.

Exempel:

```
int i = 0;
while(i <= 10) {
      i++;
      Serial.println(i);
}
```

kommer också skriva ut siffrorna 0 till 10, men här används ett externt kriterium istället för ett internt för loopen.

Det finns en tredje loop som är en variant av while-loopen. Den kallas do...while och skiljer sig från en vanlig while-loop i att den testar kriteriet efter koden kör istället för innan, så koden i loopen är garanterad att köra minst en gång. Syntaxen är

- `do {...} while(kriterium).`

Exempel 1

En if-sats kan byggas ut med en kod som utförs om kriteriet inte är sant. Syntaxen är

- `if (kriterium) {...} else {...}`

där koden inom de första klammerparanterserna endast körs om kriteriet är sant och koden inom de andra paret klammerparenteser endast körs om kriteriet är falskt.

Exempel 2

I den här labben kommer vi i vår `loop` behöver vi läsa in värdet från sensorn och ge det till ett par variabler. Vi ger de nya variablerna typen `float`, inte `int`, eftersom vår data utgörs decimaltal (floating point number).

Det är möjligt att göra beräkningar med variabler av olika format, t.ex. int + float. Det som händer är att den "lägsta" datatypen konverteras till den andra. I exemplet i förra meningen skulle heltalet konverteras till ett decimaltal innan beräkningen gjordes. Det är ändå bäst att använda samma typ av data i beräkningar eftersom det kan förekomma oväntade resultat (också känt som fel). Ett exempel på detta är om man skriver

```
float x = 1.2;
float y = 1.5;
int z = 0;
z = x + y;
Serial.print(z);
```

så kommer man se att z är sparat som 2 och inte 2.7 som var väntat. Eftersom detta inte producerar några felmeddelanden kan ett sådant fel vara svårt att hitta.

Ordlista

Funktion

Ett stycke kod som kan anropas för att köras flera gånger.

Global variabel

En variabel som finns och kan användas i hela programmet.

Lokal variabel

En variabel som endast finns i den funktion den skapas.

Loop

Ett stycke kod som upprepas ett antal gånger.

if-sats

Ett stycke kod som styr flödet i programmet. Utför ett visst stycke kod om ett villkor uppfylls.

Booleskt uttryck / Boolesk operator

En grupp jämförelseoperationer. Används för att styra programmet i loopar och if-satser. Se tabell nedan.

Operator	Betydelse	Exempel	Resultat
<	Mindre än	1 < 2 2 < 2	Sant Falskt
>	Större än	2 > 1 2 > 2	Sant Falskt
<=	Mindre eller lika med	1 <= 2 2 <= 2	Sant Sant
>=	Större eller lika med	1 >= 2 2 >= 2	Falskt Sant
==	Lika med	2 == 2	Sant
!=	Inte lika med	2 != 2	Falskt

float

Står för floating point variable. En datatyp som liknar decimaltal.

Bibliotek

En samling kod som kan importeras och sedan användas i programmet.

Funktioner som används

`static var`

Markerar variabeln som statisk. En statisk variabel kommer inte få ett nytt värde om programmet läser en nydefinition (definition som innehåller datatyp) av den medans variabeln redan har ett värde.

`analogWrite(pin, value)`

Skickar ström till den angivna pin-en. Ett värde på 255 ger maximal spänning (5V eller 3,3V beroende på kort), ett värde på 0 ger ingen ström.

`#include`

Nyckelord för att inkludera ett bibliotek. Använder inget semikolon efter.

`#define`

Nyckelord för att definiera variabler i ett bibliotek. Använder inget semikolon efter.

`pinMode(pin, mode)`

Sätter en viss pin i ett visst läge. Använda för att öppna pin-ar för att läsa eller skriva värden till dem.

`Serial.begin(speed)`

Starta serial monitor. Tillåter Arduinon att skicka meddelanden till datorn som kan läsas av användaren. Speed anger hur ofta uppdateringar sker, 9600 brukar vara lagom.

`Serial.print("meddelande")`

Skriv text till serial monitor utan att byta rad efter.

`Serial.println("meddelande")`

Skriv text till serial monitor. Byter rad efter utskriften.

`digitalRead(pin)`

Läser av spänningen på den angivna pin-en. Svarar med antingen HIGH eller LOW.