

```
// Single line comment
/* Multi-line comment
Skriv overordnet hvad de enkelte
programafsnit gør */
```

Program-Struktur

```
/* Header */
// Def af pins og variable mm.
#include
void setup() {} // do once
void loop() {} // do i loop
void name() {} // kaldes med "name();"
void myName(int y) {} // med parameter
/*kaldes med*/ myName(x);
-
int result = Sum_func(5,6);

int sum_func(int x, int y) {
  int z = x+y;
  return z;
}
```

Header

```
/* Skriv hvad, hvem - hvornår osv
*/
```

Pindefinitioner

```
uint8_t _blink = 13; // Bruger 1 byte
uint8_t sun = 5;
#define RESET 10
int buttonpin=2; // bruger 2 byte
```

def af Variable

```
int knap = 11;
uint8_t LED = 5;
const int rs = 12, en = 11;
float f = 3.6;
const byte upPin = 4; // read only
uint8_t myPins[] = {2,4,3,7};
boolean pinAState = 0; // false
```

```
#define BAUDRATE 9600
```

Include

```
// Before setup();
#include <LiquidCrystal.h>
```

void setup() {} // Udføres 1 gang

```
lcd.begin(20, 4);
pinMode (12, OUTPUT); // INPUT
pinMode(A0, INPUT);
( INPUT = 0, OUTPUT = 1 )
```

void loop() {} // Do forever

```
digitalWrite(outPin, HIGH); // LOW
val = digitalRead(inPin);
```

Datatyper

```
void // tom, kaldes uden data med
bool //8 bit Flag, 0 , 1, false, true
byte // 8 bit, 0 - 255
char // fx 'a', ( -128 to 127 )
unsigned char // brug byte i stedet
uint8_t // Unsigned, 0 to 255
int // 16 bit, -32,768 to 32,767
unsigned int // 0 to 65535 brug word!
word // unsigned 16 bit, 0 til 65535
long // -2,147,483,648 to 2,147,483,647
unsigned long // 0 to 4,294,967,295
float//32 bit -3.4028E+38 to 3.4028E+38
x = sizeof(myint) // returns 2 bytes.
```

```
volatile int = 3; //fx @ interrupts
volatile bool flag = 0; // 8 bit, global
```

Tal

```
HIGH, LOW, 1, 0, true, false
INPUT, OUTPUT
123 // decimal nummer
0b11000101 B11110000 // Binær nummer
0x7A // Hex tal
34.6 // decimaltal
```

```
2.5e5 // decimal, 250000
```

Algoritmer

```
if (digitalRead(13) == 0) x = 3;
if (x<5) {}
if (x<5) {} else {}
i>2 ? i=i+1 : i=i-1; // if ? Yes : No

if (x<5) {} elseif {} else {}
for ( int i=0; i<=255; i++) {}
while (x<6) {}// Bliv her så længe ...
while (!Serial.available()) { } // stay
here until
do {} while (x<6);
do { } while (!condition);~Repeat until
-
continue;// Go to next in do/for/while-
loop
if (statement) break; jump out of loop
return; // for voids
return x; //Return a parameter, no void

switch (myvar) {
  case 1:
  case 4 ... 9:
  // do stuf
  break; // jump to after ""
  case 2:
  case 3:
  case 'A' ... 'Z':
  // do stuf
  break;
  default:
  // do default stuf
  break;
}
```

```
case 35: Serial.print("0"); break;
// i 1 linje
```

Generelle operatorer

```
x = 5; // Tildel
x = y + z; //addition
x = y - 5; //subtraktion
x = y * 3; // multiplikation
x = y / 4; //division
```

Logiske Boolske operatorer

```
( true or false ) ( 1 or 0 )
( sammenlignings-operatorer )

if ( x == 7 ) {} // is equal to
if ( y != 9 ) {} // is not equal to
```

```
< // mindre end
> // Større end
>= // mindre end eller lig med
<= // større end eller lig med
&&, and // Logisk AND, "og samtidig"
||, or // or, " eller hvis "
&& og || har større prioritet end & og |
```

```
if( x > 0 || y < 7){ }
!, not // not, giver sandt eller falsk
if ( !x ){ } // sandt hvis x=0
```

Bitwise operatorer

```
// giver noget rod, hvis de bruges
direkte i Serial.print()

& // = bitand bitwise and,
x = 11 & 7; // -> x = 3
| // = bitor bitwise or,
x = 3 | 8; // -> x = 0000 1011
^ // = xor bitwise xor, 1 exor 1 = 0
x = x ^ B11111111; //Flip alle bit!
~ // = compl bitw. not Alle bit flippes
b = ~a; // virker på operand
x = x << 2; //bitshift 2 pladser left
x = x >> 1; //bitshift right
int b = a << 3; // skift alle bit 3 til
venstre, bit ud over venstre kant tabes
```

Operatorer

```
x++; // x = x+1;
x--; // x = x-1;
x+= 3; // x = x+3; Adder
x-= 3; // x = x-3; subtraher
```

```
x*= 3; // x = x*3; multiply
x/= 5; // x = x / 5; divide
x&= y; // x = x&y; bitwise and
x|= y; // x = x | y; bitwise or
x^= y; // x = x^y; Bitwise XOR, Toggle
bit
b = ~a; // bitwise not, invert all bit
uint8_t a = 103; // binary: 01100111
uint8_t b = ~a; // binary: 10011000
```

Arrays

```
uint8_t myInts[6]; // 6 elementer, 0-5
uint8_t myPins[] = {2,4,3,7};
-
uint8_t pinMatrix[3][3] = { // y,x
  {2, 3, 4},
  {5, 6, 7},
  {8, 9, 10}
}; // række,søjle, 0,0 = ØV.
-
char myArray[5] = {'1','2','3','4'};
int numb = atoi(myArray);
Serial.print(numb); // Returns 1234
-
char myArray[7];//the ASCII of the
integer
itoa(12596,myArray,10);//(integer, Array,
base)
```

Strings

```
char st[15];
char st1[6] = {'a','b','e','k','a','t'};
char st2[] = {'h','e','j','\0'}; // med
null terminering
char st3[] = "hej du";
char st4[12] = "Arduino";
x = myString.toInt(); // string to int
if (isDigit(myChar)) { }
myString += (char)inChar; // from
debugvindue
-
int a = 1234;
String myStr;
myStr = (String) a; //Converts integer
to string
-
char x[10] = "450";
int result = atoi(x);
Serial.println(result);
-
String str3 = String(stringTwo + "
more");// concatenating two strings
String str5 = String(analogRead(0),
DEC);// using an int and a base
String str8 = (String)(millis(), DEC);//
using a long and a base
```

Casting, converting types.

Der skal () om den type, der castes til, ellers vil compilere opfatte det som en function!

```
(char) (int) (long)
x = (byte) y; // Y: any type
x = (float) y; // Parantes om Casttype
x = (unsigned int) y;
-
int i;
float f = 3.6;
i = (int) f;
-
word z = (word)x;
-
int x = 5;
float y = 3.14;
float z = (float)x + y;
-
```

Qualifiers Ekstra info om variable

```
static int z; // Værdi kan kun ændres i
dens funktion
volatile byte x = 0; // x kan også ændres
i subs, fx ISR.
const float pi = 3.14;
```

Digital I/O INPUT/OUTPUT

```
pinMode(pin, INPUT); = pinMode(8,0);
pinMode(pin, OUTPUT); = pinMode(8,1);
```

```
digitalWrite(inputPin, 1); // -> internal pullup, 20 til 50 kOhm. pinMode(pin, INPUT_PULLUP); // -> def pin som input & internal pullup digitalWrite(pin, value); // 0, 1, LOW, HIGH int x = digitalRead(pin);
```

Analog I/O

pins A0 to A5, eller digital 14 to 19

```
int x = analogRead(pin); // 10 bit // kaldes 2 gange hvis ?? analogReference(INTERNAL); // 1,1 Volt //DEFAULT, EXTERNAL y = map(analogRead(A1), 0,1023,0,255); // omform fra range til range
```

PWM:

```
analogWrite(~pin, value); // 0 to 255
```

Flere I/O

SPI, fx 74HC595. Pins:

```
shiftOut ( dataPin, clockPin, bitOrder, value); byte incoming = shiftIn(dataPin, clockPin, bitOrder) // bitOrder = [MSBFIRST, LSBFIRST]
```

Timing

```
unsigned long x = millis(); // 49 dages overflow unsigned long x = micros(); // 70 min overflow delay(ms); delayMicroseconds(us);
```

Math

```
k = min(x,y); //k = mindste af x og y k = min(sensVal, 100); //k = max 100 k = max(x,y); // k = største af de to y = abs(x); // -> positiv x = constrain(y, minval, maxval ); // limits range of y to 10 -> 15 x = map(val, fromL, fromH, toL, toH); y = map(analogRead(A1), 0,1023,0,225); x = pow(base, exponent); y = round(x); // rund op eller ned
```

```
y = sqrt(x); // #include "math.h" x = sin(rad); cos(rad), tan(rad)
```

```
uint8_t x = 35; uint8_t tens = (x / 10); uint8_t ones = x % 10; // Modulo, Reminder, ones = 5. For int & float
```

Random numbers

```
randomSeed(seed); // long or int randomSeed(analogRead(0)); // randomize from noise int x = random(max); // max fx = 300 long x = random( min, max); // min er inklusiv, max er eksklusiv
```

Bits & bytes

```
( x er en SFR eller variabel ) byte x = lowByte(z); byte y = highByte(z); int v = word(x,y); bool y = bitRead(x, bit#); // I var bitWrite(x, bit#, value); bitSet(x, bit#); // 0 til 7 i var x bitClear(x, bit#); x |= bit(4); // OR bit 4 med 1 x ^= bit(4); // Toggle bit 4(Xor) i x if(bit_is_set(x,bit#)) { } // Hvis Bit i variabl x == 1 if(bit_is_clear(x,bit#)) { } // Hvis bit # i en variable == 0
```

Timersetup Mangler

Pin-interrupt

```
attachInterrupt(0, myFunction, [LOW,CHANGE,RISE,FALLING]); // Int0 = Pin 2, Int1 = Pin3 detachInterrupt(0) // 0 eller 1 noInterrupts(); // disable interrupts interrupts(); // enable interrupts igen myFunction() { }
```

Serial til debugvindue, pin 0&1, RX & TX

```
i setup(): Serial.begin(9600); // angiv Baudrate, 1200, 2400, 4800, 9600(default) Serial.end(); // luk objektet igen if ( Serial.available(>0) ) { // hvis sand x = Serial.read(); if ( x == 'V' ) { // ASCII if ((input >='0') && (input <='9')) return (input-'0'); // træk 30h fra - Serial.print(x); // Print omformer til menneske-læselig ASCII tekst. Converts ints, floats, etc. to strings and calls Serial.write() to send the string. - Serial.println(x); // + Ny linje serial.println(data, base); // Base: DEC, HEX, BIN
```

```
Serial.write(45); // As is. Værdien 45 while (!Serial.available()) { } // Wait for data to arrive while (Serial.available()) { Serial.read(); } // Tøm modtage-buffer Serial.flush(); // vent på, at sendebuffer er tøm Serial.println(78, DEC); // BIN, HEX Serial.println(1.2345, 2); //print 2 decimaler byte h = Serial.read(); byte l = Serial.read(); int y = word( h, l); - if (Serial.available() >= 2) { // når der er ankommet 2 bytes, for (int i = 0; i < 2; i++){ incoming[i] = Serial.read(); }
```

SoftwareSerial

```
Før setup: #include <SoftwareSerial.h> SoftwareSerial Navn(RxPin, TxPin); // opret og navngiv softserial-objekt
```

```
i setup() HC12.begin(9600); // initier object & angiv Baudrate, 1200, 2400, 4800, 9600
```

```
i loop() if ( HC12.available() > 0 ) { } if(HC12.available()){ } // hvis sand while (HC12.available()) { } // så længe der er data while (!HC12.available()) { } // Wait until data arriving x = HC12.Read(); - HC12.print(myData); // Omform først til menneske-læselig HC12.println(mydata); // + Linefeed HC12.write(myData); // Send "As is" HC12.write((byte)i); // løser 0-problem HC12.print(String(number)); // ??? - RS485.begin(9600); // init en port RS485.end(); // Stop port-objekt. Herefter kan et andet objekt begyndes portTwo.listen(); // Fokuser på portTwo @ flere softserial-objekter.
```

LCD

```
#include <LiquidCrystal.h> // før setup LiquidCrystal lcd(12, 11, 5, 4, 3, 2); //Pins : RS, En, D4 to D7. i setup(): lcd.begin(20, 4); // display størrelse i loop:
```

```
lcd.setCursor(x,y); // vandret ( 0- 19), nedad( 0 - 3 lcd.print("Hej"); lcd.print(x); lcd.clear(); lcd.cursor(); // Cursor on lcd.noCursor(); // Cursor off lcd.blink(); // Sort blinkende Karakter lcd.noBlink(); //
```

Wire til I2C, IIC

```
#include <Wire.h> // før setup Mangler !!
```

EEPROM (1024 Byte)

```
#include <EEPROM.h> // før setup x = EEPROM.read(adresse); // Læs EEPROM.write(adresse, value); // Skriv
```

#define

```
#define dusin 12 // define #define pin13_on bitSet(PORTB,5) #define pin13_off bitClear(PORTB,5) pin13_on; // do defined function. #define MYLED 8 digitalWrite(MYLED, HIGH); // #define DEBUG 1 // fjern "/" og gå i "debugmode" #ifdef DEBUG // Hvis DEBUG er sand: Serial.println("xp"); #endif
```

Include Libraries

```
#include <LiquidCrystal.h> #include <SoftwareSerial.h> #include <Wire.h> // I2C #include <EEPROM.h> #include "math.h"
```

Memory

```
Flash: 32 kB, ( 2k til bootloader ) SRAM: 2 kB EEPROM: 1 kB PROGEM // brug flashROM Mangler !!
```

Power

Der kan Max trækkes 20 mA pr pin, Max 200 mA total. Fra 5 Volt pin: Der kan max trækkes 500 mA

Biblioteker & timere bruger:

Timer0: Delay(), millis(), micros() og delayMicroseconds().
Timer1: Serv.
Timer2: tone.

Mere til historien:

Print flere ting i 1 linje:
Just convert output directly to a String:

```
Serial.println((String)"x:"+x" y:"+y);
```

// Spar på RAM. ??

```
Serial.println( F( "Hello World!" ) ) // teksten bliver gemt i Flash,
```

Shortcuts

Ctrl + r, Verify Sketch
Ctrl + U, Upload
Ctrl + T, Adjust, Allign og spaces
Ctrl + Shift + M, Open Serial Monitor