

Syntaxübersicht Arduino IDE

Variablendeklaration

//Datentyp Variable = Startwert;

```
byte x = 253;           // Byte -> 8 Bit, ganzzahlig, 0-255
int y = 3333;          // Integer -> 16 Bit, ganzzahlig, -32767 bis - 32768
long z = 100000;       // erweiterer Integer -> 16 Bit, ganzzahlig
float a = 3,14159;     // Gleitkommazahl -> 32 Bit
int array1[] = {1,2,3}; // Datenstruktur mit festgelegter Reihenfolge
char Zeichen = 'X';    // ASCII-Zeichen -> 1 Byte, -127 bis 127
char Wort[] = 'Hallo'; // String-Variable, Zeichenkette
```

Ansteuerung digitaler und analoger Ports

// pinMode(Pin, OUTPUT/ INPUT)

```
pinMode(CONTROLLINO_D0, OUTPUT); // Port D0 als Ausgang konfiguriert
pinMode(CONTROLLINO_A0, INPUT);  // Port A0 als Eingang konfiguriert
```

// digitalWrite(Pin, HIGH/LOW);

```
digitalWrite(CONTROLLINO_A0, HIGH); // Aktivierung interner Pullup Widerstand
digitalWrite(CONTROLLINO_D0, LOW);  // Ausgang wird Low gesetzt
```

// digitalRead(Pin);

```
x = digitalRead(CONTROLLINO_A0); // Zuweisung Zustand Port zu Variable
```

// analogWrite(Pin, Wert);

```
analogWrite(CONTROLLINO_D1, 255); // erzeugt ein PWM-Signal am PIN D1
```

// analogRead(Pin, Wert); // liest Int-Wert zwischen 0-1023

serielle Kommunikation

```
//Serial.begin(baudrate) // starte serielle Kom. , Angabe Datenrate  
  
Serial.begin(9600); // Baudrate 9600  
  
//Serial.print (data) // sendet Daten zum seriellen Port  
  
//Serial.read() // empfängt Daten der seriellen Schnittstelle.  
  
//Serial.flush() // löscht Daten aus seriellen Puffer
```

Kommunikation mit dem LCD

```
//lcd.setCursor(0, 0) // setzt den Cursor an Position 0, 0  
  
//lcd.print (16, 2) // Angabe Zeilen und Spalten LCD  
  
//lcd.begin(baudrate) // starte serielle Kom. , Angabe Datenrate
```

wichtige Bibliotheken

```
include <Controllino.h> // notwendig für Nutzung der Controllino Funktionen  
  
include <Servo.h> // notwendig zur Ansteuerung von Servomotoren  
  
include <LiquidCrystal.h> //notwendig zur Ansteuerung von LC-Displays
```

nützliche Funktionen

```
delay(ms) // Programm pausiert für die angegebene Zeit  
  
map(x, Untergrenze1, Obergrenze1, Untergrenze2, Obergrenze2)  
  
value = map(x, 0, 1023, 0, 180); // Skalierung des Int-Wertes x des Bereiches 1 auf  
// den Wertebereich des Bereiches 2  
  
random(min, max) // Erzeugung einer Zufallszahl innerhalb der  
// angegebenen Grenzen  
  
Objekt.attach(Ausgang); // Zuweisung PWM Ausgang zu Objekt
```