Programmstruktur

Basis Programm Struktur

```
void setup ( ) {
  // einmal zu Beginn ausführen
}
void loop ( ) {
  // wiederholt ausführen
}
```

Kontroll-Strukturen

```
if (x < 5) { ... }   // mehrere Beding.
else if (x < 10) { ... } // && bzw. ||
else { ... }
while (x < 5) { ... }
do  { ... } while (x < 5); //mind. 1x
for (int i=0; i<10; i++) { ... }
break ;    // Schleife abbrechen
continue ; // nächste Wiederholung
switch (var) {
  case  1: ...  // wenn var = 1
    break;      // Ende des Falles
  case  2: ...  // wenn var = 2
    break;
  case  3:      // wenn var = 3
  case  4: ...  // oder var = 4
    break;
  default: ...  // alle übrigen Fälle
    break;
}
```

Funktions-Definitionen

```
<ret. Type> <name>(<params>) { ... }
z.B. int doppel(int x) {return x*2;}
return  x; // Datentyp wie <ret. Type>
return;    // bei void-Funktion
```

Operatoren

Allgemeine Operatoren

=	Zuweisung von rechts nach links		
+	plus	-	minus
*	mal	/	geteilt
%	Modulo (Rest der GZ-Division)		
==	Gleich wie	!=	Ungleich
<	Kleiner als	>	Größer als
<=	Kleiner oder gleich		
>=	Größer oder gleich		
&&	AND (log.)	||	OR (log.)
!	NOT (log.)	()	Klammerung

Zusammengesetzte Operatoren

++	Inkrement / hochzählen
--	Dekrement / runterzählen
+=	Addition mit Zuweisung
-=	Subtraktion mit Zuweisung
*=	Multiplkation mit Zuweisung
/=	Divisition mit Zuweisung
&=	Bitweise AND mit Zuweisung
|=	Bitweise OR mit Zuweisung
<<=	Links schieben mit Zuweisung
>>=	Rechts schieben mit Zuweisung
X++	Post-Inkrement ++X Prä-Inkrement
X--	Post-Dekrement --X Prä-Dekrement

Bitweise Operatoren

&	bitweise AND	|	bitweise OR
^	bitweise XOR	~	bitweise NOT
<<	shift links	>>	shift rechts

Pointer Zugriff (Zeiger)

&	Referenz: liefert einen Pointer
*	Dereferenz: Inhalt, auf den der Pointer zeigt

Standard - Funktionen

Digital I/O – ESP32 Pins34-39 nur In

```
pinMode (pin,
    [INPUT, OUTPUT, INPUT_PULLUP])
bool digitalRead (pin)
digitalWrite (pin, [HIGH, LOW])
```

Analog in – ESP32 Pins A0-A9

```
int analogRead (pin)
int analogReadMillivolts (pin)
analogReadResolution (bits)
//Auflösung der Wandlung einstellen
//Default: ESP32 12Bit, EPS32-S3 13Bit
```

PWM out –

```
analogWrite (pin, value) //kompatibel
                  // zu Uno, Nano, ESP8266
```

Advanced I/O

```
tone (pin, freq_Hz)  // Ton erzeugen
tone (pin, freq_Hz, duration_ms)
noTone (pin)      // Ton ausschalten
shiftOut (dataPin, clockPin,
  [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn (pin,
  [HIGH, LOW])  // Messe Dauer LOW-
   // oder HIGH-Teil eines Signals
```

Zeiten

```
delay (msec)
delayMicroseconds (usec)
unsigned long millis ()
unsigned long micros ()
```

Math

```
min (x, y) min (x, y) abs (x)
sin (rad)  cos (rad)  tan (rad)
sqrt (x)   pow (base, exponent)
int constrain(x, minval, maxval)
int map (val, x1, x2, y1, y2)
```

Random Numbers

```
randomSeed (seed); // long oder int
long random (max); // 0 bis max-1
long random (min, max);
```

Bits und Bytes

```
byte lowByte (x)   byte highByte (x)
bool bitRead (x, bitn)
bitWrite (x, bitn, bit)
bitSet (x, bitn)   bitClear (x,bitn)
byte bit (bitn) // bitn: 0=LSB 7=MSB
```

Typ Umwandlung

```
char (val)      byte (val)
int (val)       short (val)
long (val)      float (val)
```

Externe Interrupts

```
Interrupt-Nummer bestimmen mit
intNr = digitalPinToInterrupt(pin)
attachInterrupt (intNr, &func,
 [LOW, CHANGE, RISING, FALLING])
detachInterrupt (intNr)
interrupts () //Interrupts zulassen
noInterrupts ()//Interrupts sperren
```

Bibliotheken

Diese Bibliotheksfunktionen werden mit Name.Methode genutzt.

Serial - Komm. mit PC bzw. RX/TX

```
begin (long speed)//bis 921600Bit/s
begin (speed, config) //Config z.B.
  //SERIAL_7E1 oder SERIAL_8N1
end ( )
int available() //#Bytes angekommen
int read ()     // -1, wenn leer
int peek ()  // lesen ohne löschen
flush ()     // Sendepuffer leeren
print (data)   println (data)
write (byte)   write (char *str)
write (byte *data, size)
```

SoftwareSerial.h - an bel. Pin

```
SoftwareSerial (rxPin, txPin)
begin (long speed)//bis 115200Bit/s
listen ()// umschalten v. mehreren
isListening () // SoftwareSerial
available, read, write, print etc.
// Genauso wie bei Serial
```

EEPROM.h nicht-flüchtiger Speicher

```
byte read (addr)
write (addr, byte)
EEPROM[index] // Array-Zugriff
```

Servo.h – Bib. für Uno/ESP8266
ESP32Servo.h – Bib. für ESP32

```
attach (pin, [min_uS, max_uS])
write (angle)    // 0 bis 180°
writeMicroseconds (us)
//ca.1000-2000; 1500 ist die Mitte
int read ()      // 0 bis 180°
bool attached () // Kontrolle
detach ()        // Freigabe Pin
```

Variablen, Arrays und Definitionen

Datentypen

bool	true || false
char	-128 – 127,'a','?' (int8_t)
unsigned char	0 – 255 (uint8_t)
byte	0 – 255 (uint8_t)
short	-32768 – 32767 (int16_t)
unsigned short	0 – 65536 (uint16_t)
int, unsigned int	bei ESP32 mit 32 Bit wie (unsigned) long (int32_t, uint32_t)
long	-2147483648 – 2147483647 (2^{31}-1)
unsigned long	0 – 4294967295 (2^{32}-1)
float	±1.1755e-38 – ±3.4028e+38 (4Byte)
double	±5,0e-324 – ± 1,7e+308 (8Byte)
void	Kein Rückgabewert oder Parameter
long long oder **int64_t**	-2^{63} – 2^{63}-1
unsigned long long o. **uint64_t**	0 – 2^{64}-1

Strings

```
char str1[8] =
  {'A','r','d','u','i','n','o','\0'};
  // C-String endet mit NULL-Zeichen \0
char str2[8] = "Arduino";// \0 mitzählen
char str3[] = "Arduino";
String str4 = "Arduino"; // String-Klasse
String str5 = "Sensor-Wert: "
            + analogRead( A0 );
```

Numerische Konstanten

123	Normale Dezimalzahl
0b01111011	Binäre schreibweise
0x7B	Hexadezimalzahl (0 bis F)
0173	Oktalzahl – vermeiden!
123U	Erzwinge unsigned
123L	Erzwinge long
123UL	Erzwinge unsigned long
123.0	Kommazahl
1.23e6	Exponent-Darstellung $1.23*10^6$ = 1230000

Qualifier

static	Variable bleibt erhalten
volatile	im RAM (notwendig für ISR)
const	Nicht veränderbar (z.B.Pin)

Arrays

```
const int MyPins[]= { 2, 4, 8, 3, 6};
int MyInts[6];    // Array für 6 Werte
MyInts[0] = 42;   // Zuweisung an den
                  // ersten Wert
MyInts[6] = 21;   // Fehler! Index
                  // von 0 bis 5
```

Definitionen

```
#define Suche Ersatz // Suche ohne
// Leerzeichen, nach Ersatz kein ;
```