

# Einfuehrung in Robotik und das Zusammenspiel von Software, Elektronik und Mechanik

2 SemWstd. Sommersemester 2021

Univ.-Lekt. Daniel Schatzmayr

[daniel.schatzmayr@uni-ak.ac.at](mailto:daniel.schatzmayr@uni-ak.ac.at)

[robotic@uni-ak.ac.at](mailto:robotic@uni-ak.ac.at)

Arduinos und Arduino IDE

HelloWorld

Analoge Sensoren

Servos

Lichtscanner Example

# Arduino

- Viele verschiedene Boards
  - Development Boards (Gross und Klein)
- Arduino IDE
  - Compile and Upload
- Arduino framework/libraries
  - Hardware configuration and access abstraction
- Arduino examples
- Arduino universe

# Programmierung

- source code --(compile&link)--> machine code
- upload tools/tool chains --> binary upload
- debugging tools -----> execution inspection

# C/C++

- Keywords
- Variablen
- Syntax
- Rechenoperation
- Vergleichsoperation
- Schleifen
- Funktionen

# C/C++ Keywords

<code>alignas (seit C++11)</code>	<code>enum</code>	<code>return</code>
<code>alignof (seit C++11)</code>	<code>explicit</code>	<code>short</code>
<code>and</code>	<code>export</code>	<code>signed</code>
<code>and_eq</code>	<code>extern</code>	<code>sizeof</code>
<code>asm</code>	<code>false</code>	<code>static</code>
<code>auto(1)</code>	<code>float</code>	<code>static_assert(seit C++11)</code>
<code>bitand</code>	<code>for</code>	<code>static_cast</code>
<code>bitor</code>	<code>friend</code>	<code>struct</code>
<code>bool</code>	<code>goto</code>	<code>switch</code>
<code>break</code>	<code>if</code>	<code>template</code>
<code>case</code>	<code>inline</code>	<code>this</code>
<code>catch</code>	<code>int</code>	<code>thread_local(seit C++11)</code>
<code>char</code>	<code>long</code>	<code>throw</code>
<code>char16_t(seit C++11)</code>	<code>mutable</code>	<code>true</code>
<code>char32_t(seit C++11)</code>	<code>namespace</code>	<code>try</code>
<code>class</code>	<code>new</code>	<code>typedef</code>
<code>compl</code>	<code>noexcept(seit C++11)</code>	<code>typeid</code>
<code>const</code>	<code>not</code>	<code>typename</code>
<code>constexpr(seit C++11)</code>	<code>not_eq</code>	<code>union</code>
<code>const_cast</code>	<code>nullptr (seit C++11)</code>	<code>unsigned</code>
<code>continue</code>	<code>operator</code>	<code>using(1)</code>
<code>decltype(seit C++11)</code>	<code>or</code>	<code>virtual</code>
<code>default(1)</code>	<code>or_eq</code>	<code>void</code>
<code>delete(1)</code>	<code>private</code>	<code>volatile</code>
<code>do</code>	<code>protected</code>	<code>wchar_t</code>
<code>double</code>	<code>public</code>	<code>while</code>
<code>dynamic_cast</code>	<code>register</code>	<code>xor</code>
<code>else</code>	<code>reinterpret_cast</code>	<code>xor_eq</code>

Example C++11

Keywords sind reservierte Schluesselwoerter

# C/C++ Variablen

Int Integer (Ganzzahlen) ...-1,0,1,2,3...255...  
signed und unsigned int

Float Gleitkommazahlen 3.14159

Bool Booleanzahle ... true (1), false (0)

Char/Byte (0-255 bzw 0x00-0xFF)

Selbstdefinierte Variablen

zB: typedefs, structs und class/objects

Variablen koennen Einzeln oder als Felder/Arrays[] definiert werden

# C/C++ Syntax

## Kommentare

// Kommentar Textzeile

/\* Kommentar Text Block

Wird nicht kompiliert \*/

## Statementsyntax

```
int a = 2;  
int sensorValue = analogRead(sensorPin);  
int c = a + sensorValue;  
  
Serial.print("Der Wert von Variable c ist: ");  
Serial.print(c);
```



# C/C++ Operatoren

- Assignment operator (=)
- Arithmetic operators ( +, -, \*, /, % )
- Compound assignment (+=, -=, \*=, /=, %=, >>=, <<=, &=, ^=, |=)
- Relational and comparison operators ( ==, !=, >, <, >=, <= )
- Logical operators ( !, &&, || )
- Bitwise operators ( &, |, ^, ~, <<, >> )
- Conditional ternary operator ( ? )

Level	Precedence group	Operator	Description	Grouping
1	Scope	::	scope qualifier	Left-to-right
2	Postfix (unary)	++ --	postfix increment / decrement	Left-to-right
		()	functional forms	
		[]	subscript	
		. ->	member access	
3	Prefix (unary)	++ --	prefix increment / decrement	Right-to-left
		~ !	bitwise NOT / logical NOT	
		+ -	unary prefix	
		& *	reference / dereference	
		new delete	allocation / deallocation	
		sizeof	parameter pack	
		(type)	C-style type-casting	
4	Pointer-to-member	.* ->*	access pointer	Left-to-right
5	Arithmetic: scaling	* / %	multiply, divide, modulo	Left-to-right
6	Arithmetic: addition	+ -	addition, subtraction	Left-to-right
7	Bitwise shift	<< >>	shift left, shift right	Left-to-right
8	Relational	< > <= >=	comparison operators	Left-to-right
9	Equality	== !=	equality / inequality	Left-to-right
10	And	&	bitwise AND	Left-to-right
11	Exclusive or	^	bitwise XOR	Left-to-right
12	Inclusive or		bitwise OR	Left-to-right
13	Conjunction	&&	logical AND	Left-to-right
14	Disjunction		logical OR	Left-to-right
15	Assignment-level expressions	= *= /= %= += -=>>= <<= &= ^=  =	assignment / compound assignment	Right-to-left
		?:	conditional operator	
16	Sequencing	,	comma separator	Left-to-right

# C/C++ Vergleiche

## If - Statement:

```
int a = 2;
int sensorValue = analogRead(sensorPin);
int c = a + sensorValue;

if(c > 100)
{
    Serial.print(" c ist grosser als 100: ");
} else
{
    Serial.print(" c ist kleiner oder gleich 100");
}
```

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

&& OPERATOR (and)		
a	b	a && b
true	true	true
true	false	false
false	true	false
false	false	false

OPERATOR (or)		
a	b	a    b
true	true	true
true	false	true
false	true	true
false	false	false

# C/C++ Schleifen

## For und while - Statement:

```
int a = 0;

for(int i = 0; i < 100; i++)
{
    a = a + i;
}
```

```
Serial.print(" a ist: ");
Serial.print(a);
```

```
a = 0;

while( a < 1000)
{
    a = a + a;
}
```

# C/C++ Funktionen

## Funktions-definition:

```
int HelloAddition(int a, int b)
{
    int c = a + b;
    return c;
}
```

## Funktions-aufruf:

```
int a = 2;
int sensorValue = analogRead(sensorPin);
int c = HelloAddition(a, sensorValue) ;
```

# Arduino Programm-Flow

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

- <https://www.arduino.cc/reference/en/>

# Arduino Programm-Flow

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per
  // second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);          // delay in between reads for stability
}
```

# Arduino Programm-Example Button

```
void setup()
{
    // initialize the LED pin as an output:
    pinMode(13, OUTPUT);
    // initialize the pushbutton pin as an input:
    pinMode(2, INPUT);
}

void loop()
{
    // read the state of the pushbutton value:
    int buttonState = digitalRead(2);

    // check if the pushbutton is pressed.
    if (buttonState == HIGH) {
        // turn LED on:
        digitalWrite(ledPin, HIGH);
    } else {
        // turn LED off:
        digitalWrite(ledPin, LOW);
    }
}
```



```
#include <Servo.h>
Servo myservo; // create servo object to control a servo

void setup() {
  Serial.begin(9600);
  Serial.print("Hallo liebe Studierende");

  analogReference(EXTERNAL); // Referenzspannung
  pinMode(A0, INPUT);

  myservo.attach(2); // attaches the servo on pin 2
  myservo.write(90); // start position
}

void loop() {
  int A0_Val = analogRead(A0);
  int val = map(A0_Val, 270, 600, 0, 180);

  if(val <= 0) val = 0;
  if(val >= 180) val = 180;

  Serial.println(val);
  myservo.write(val);

  delay(10);
}
```

```
int ScanHighest()
{
    int maxVal = 0, maxPos = 0;
    myservo.write(0); // start position
    delay(200); // bissi warten

    for(int i = 0; i <= 180;i++) // scan 0 - 180 grad
    {
        myservo.write(i); // servo positionieren
        delay(10); // bissi warten zwischen den messungen

        int val = analogRead(A0);

        if(maxVal < val)
        {
            maxVal = val;
            maxPos = i;
        }
    }

    return maxPos;
}
```

# Licht Scanner Beispiel

```
#include <Servo.h>
Servo myservo;  // create servo object to control a servo

void setup() {
  analogReference(EXTERNAL); // Referenzspannung
  pinMode(A0,INPUT);

  myservo.attach(2);  // attaches the servo on pin 2
  myservo.write(90);  // start position
}

void loop()
{
  int hi = ScanHighest();

  myservo.write(hi);

  delay(2000);
}
```

# Einfuehrung in Robotik und das Zusammenspiel von Software, Elektronik und Mechanik

Univ.-Lekt. Daniel Schatzmayr

[daniel.schatzmayr@uni-ak.ac.at](mailto:daniel.schatzmayr@uni-ak.ac.at)

[robotic@uni-ak.ac.at](mailto:robotic@uni-ak.ac.at)

