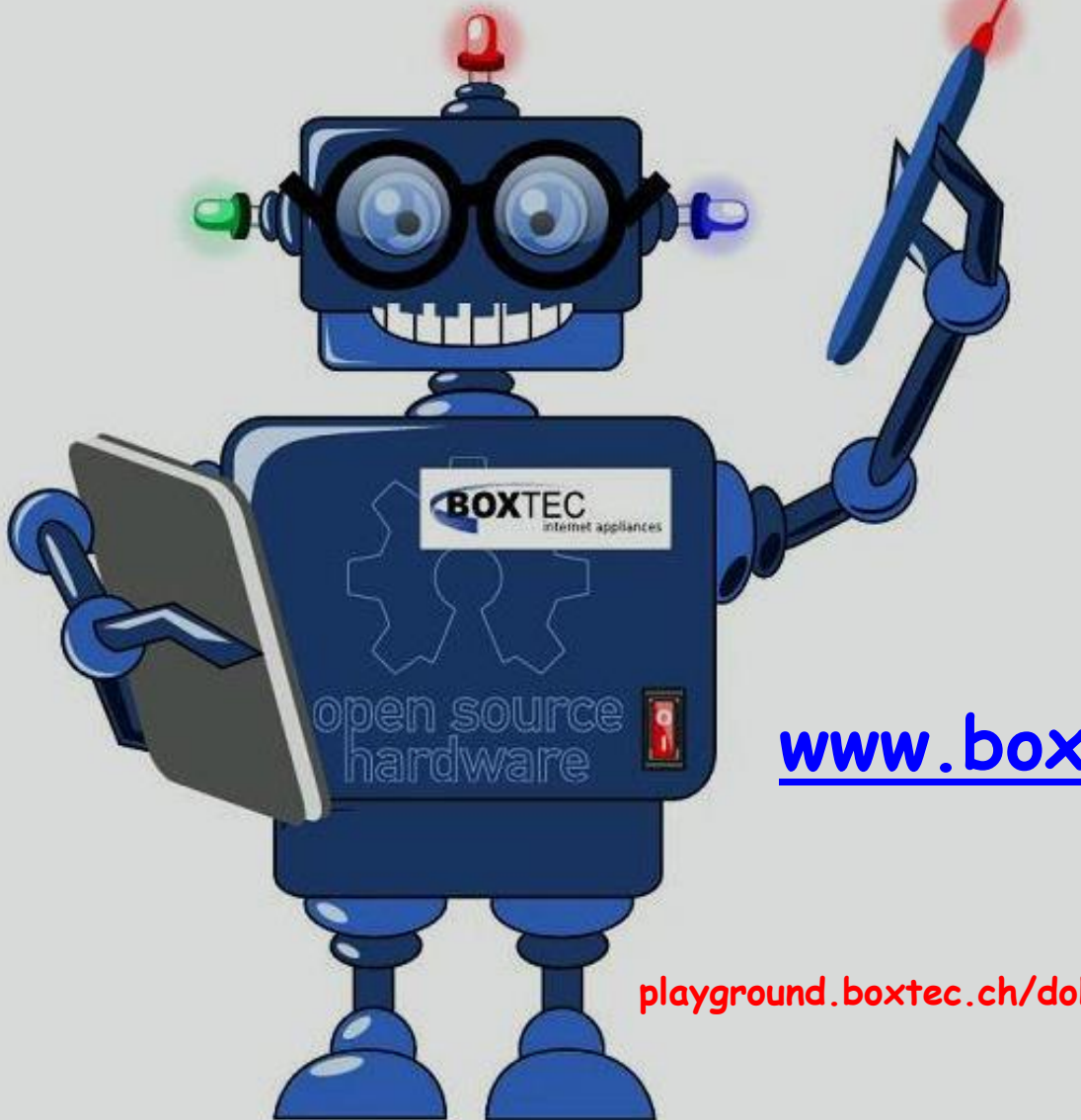


# MIKROKONTROLLER & I<sup>2</sup>C BUS



[www.boxtec.ch](http://www.boxtec.ch)

[playground.boxtec.ch/doku.php/tutorial](http://playground.boxtec.ch/doku.php/tutorial)

Board 1 und  
Servo 1

**Servo 1**



## Copyright

Sofern nicht anders angegeben, stehen die Inhalte dieser Dokumentation unter einer „Creative Commons - Namensnennung- NichtKommerziell-Weitergabe unter gleichen Bedingungen 3.0 DE Lizenz“



## Sicherheitshinweise

Lesen Sie diese Gebrauchsanleitung, bevor Sie diesen Bausatz in Betrieb nehmen und bewahren Sie diese an einem für alle Benutzer jederzeit zugänglichen Platz auf. Bei Schäden, die durch Nichtbeachtung dieser Bedienungsanleitung verursacht werden, erlischt die Gewährleistung/Garantie. Für Folgeschäden übernehmen wir keine Haftung! Bei allen Geräten, die zu ihrem Betrieb eine elektrische Spannung benötigen, müssen die gültigen VDE-Vorschriften beachtet werden. Besonders relevant sind für diesen Bausatz die VDE-Richtlinien VDE 0100, VDE 0550/0551, VDE 0700, VDE 0711 und VDE 0860. Bitte beachten Sie auch nachfolgende Sicherheitshinweise:

- Nehmen Sie diesen Bausatz nur dann in Betrieb, wenn er zuvor berührungssicher in ein Gehäuse eingebaut wurde. Erst danach darf dieser an eine Spannungsversorgung angeschlossen werden.
- Lassen Sie Geräte, die mit einer Versorgungsspannung größer als 24 V- betrieben werden, nur durch eine fachkundige Person anschließen.
- In Schulen, Ausbildungseinrichtungen, Hobby- und Selbsthilfewerkstätten ist das Betreiben dieser Baugruppe durch geschultes Personal verantwortlich zu überwachen.
- In einer Umgebung in der brennbare Gase, Dämpfe oder Stäube vorhanden sind oder vorhanden sein können, darf diese Baugruppe nicht betrieben werden.
- Im Falle einer Reparatur dieser Baugruppe, dürfen nur Original-Ersatzteile verwendet werden! Die Verwendung abweichender Ersatzteile kann zu ernsthaften Sach- und Personenschäden führen. Eine Reparatur des Gerätes darf nur von fachkundigen Personen durchgeführt werden.
- Spannungsführende Teile an dieser Baugruppe dürfen nur dann berührt werden (gilt auch für Werkzeuge, Messinstrumente o.ä.), wenn sichergestellt ist, dass die Baugruppe von der Versorgungsspannung getrennt wurde und elektrische Ladungen, die in den in der Baugruppe befindlichen Bauteilen gespeichert sind, vorher entladen wurden.
- Sind Messungen bei geöffnetem Gehäuse unumgänglich, muss ein Trenntrafo zur Spannungsversorgung verwendet werden
- Spannungsführende Kabel oder Leitungen, mit denen die Baugruppe verbunden ist, müssen immer auf Isolationsfehler oder Bruchstellen kontrolliert werden. Bei einem Fehlers muss das Gerät unverzüglich ausser Betrieb genommen werden, bis die defekte Leitung ausgewechselt worden ist.
- Es ist auf die genaue Einhaltung der genannten Kenndaten der Baugruppe und der in der Baugruppe verwendeten Bauteile zu achten. Gehen diese aus der beiliegenden Beschreibung nicht hervor, so ist eine fachkundige Person hinzuzuziehen

## Bestimmungsgemäße Verwendung

- Auf keinen Fall darf 230 V~ Netzspannung angeschlossen werden. Es besteht dann Lebensgefahr!
- Dieser Bausatz ist nur zum Einsatz unter Lern- und Laborbedingungen konzipiert worden. Er ist nicht geeignet, reale Steuerungsaufgaben jeglicher Art zu übernehmen. Ein anderer Einsatz als angegeben ist nicht zulässig!
- Der Bausatz ist nur für den Gebrauch in trockenen und sauberen Räumen bestimmt.
- Wird dieser Bausatz nicht bestimmungsgemäß eingesetzt kann er beschädigt werden, was mit Gefahren, wie z.B. Kurzschluss, Brand, elektrischer Schlag etc. verbunden ist. Der Bausatz darf nicht geändert bzw. umgebaut werden!
- Für alle Personen- und Sachschäden, die aus nicht bestimmungsgemäßer Verwendung entstehen, ist nicht der Hersteller, sondern der Betreiber verantwortlich. Bitte beachten Sie, dass Bedien- und /oder Anschlussfehler außerhalb unseres Einflussbereiches liegen. Verständlicherweise können wir für Schäden, die daraus entstehen, keinerlei Haftung übernehmen.
- Der Autor dieses Tutorials übernimmt keine Haftung für Schäden. Die Nutzung der Hard- und Software erfolgt auf eigenes Risiko.

# Servos 1

## Allgemeiner Aufbau

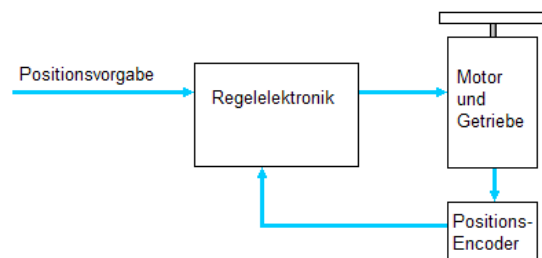
Wenn in einem Projekt eine bestimmte mechanische Arbeit verrichtet werden soll, z.B. eine Drehbewegung, dann arbeitet man gern mit einem Modellbauservo. Diese haben den Vorteil, dass sie leicht von einem Mikrocontroller angesteuert werden können. Die dazu notwendige Leistungselektronik und Regelelektronik ist bereits im Servo eingebaut.

Durch diese Elektronik erfolgt das Anfahren einer bestimmten Position, der Ausgleich von Schrittverlusten oder beim Einschalten findet er seine Neutralposition.

Servos gibt es in verschiedenen Preiskategorien, wobei der Unterschied in der Kraft der Motoren und in der Qualität des integrierten Getriebes liegt.

Servos (Servomotoren) sind kleine Getriebemotoren. Mithilfe eines Potentiometers, das an der Drehachse befestigt ist, wird intern die aktuelle Position bestimmt.

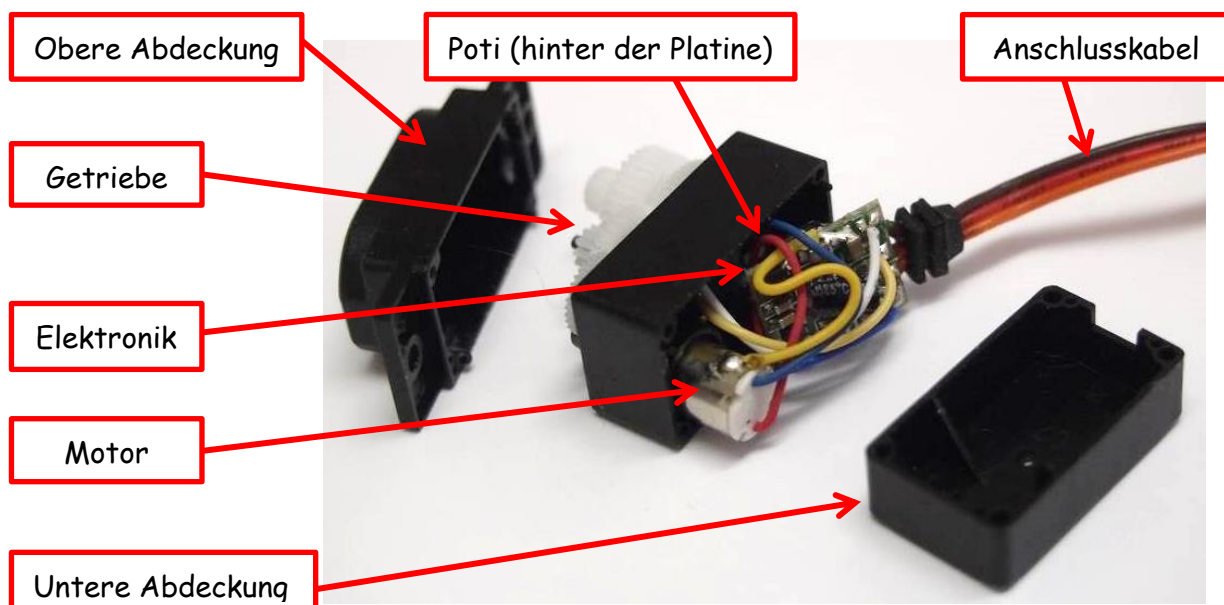
( Bild aus WIKI )



Im Grunde besteht ein Servo aus den folgenden Komponenten

- Elektronik für die Pulsauswertung
- Elektronik für eine Regelschleife
- Leistungselektronik zur Ansteuerung eines Motors
- der Motor samt Getriebe
- Positionsauswertung

Aus diesen Komponenten wird eine Regelschleife gebildet, so dass der Motor einem Positionssignal nachgeführt wird. Am Motor ist ein Getriebe angeflanscht, welches wiederum die Antriebsscheibe bewegt, an der die Bewegung mechanisch abgegriffen werden kann.

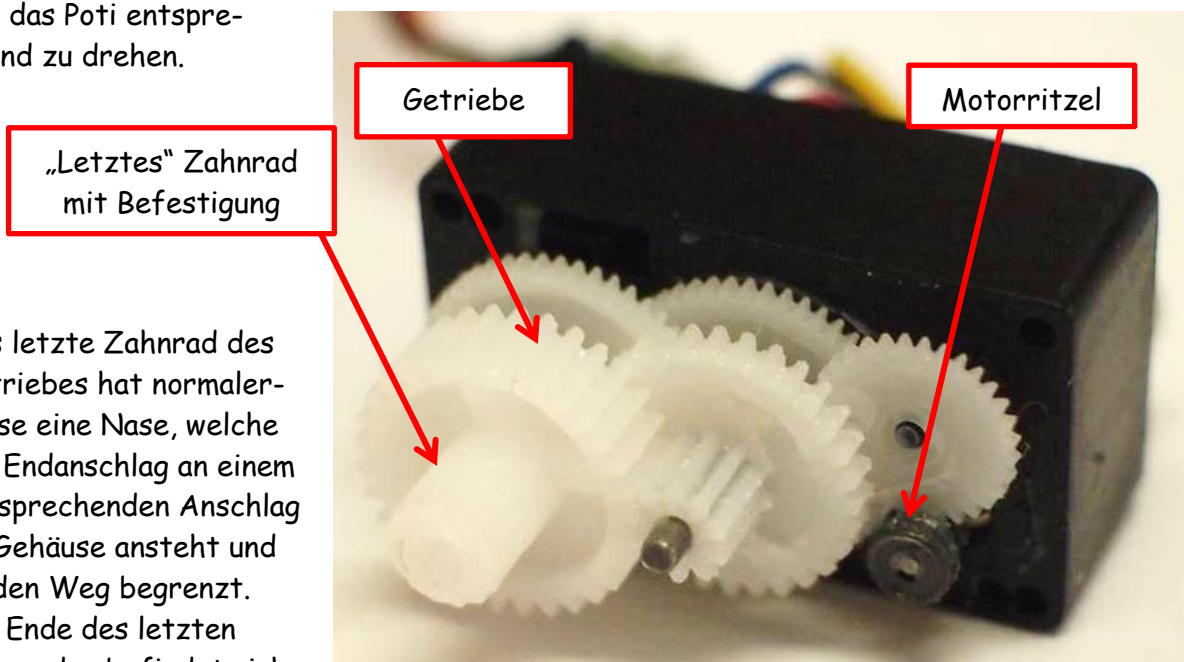


## Funktion der Regelschleife

Der Positionencoder, im Regelfall ein mechanisch mit dem Getriebe gekoppeltes Potentiometer, stellt die Positionsinformation der Regelelektronik zur Verfügung, die bei Abweichungen entsprechende Motorbewegungen veranlasst. Man gibt also einem Servo nicht eine Bewegung vor, sondern eine Position die es ansteuern soll. Die Regelelektronik fährt dann diese Position an und hält sie in weiterer Folge.

## Getriebe

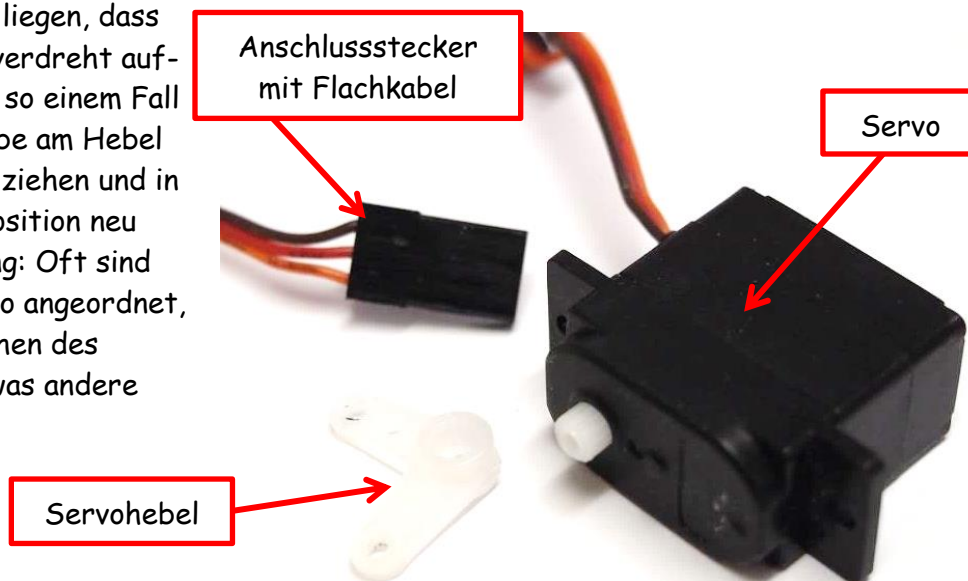
Zwischen dem Motor und dem letzten Zahnrad befindet sich das Getriebe. Das hat die Aufgabe, die Drehzahl des Motors auf eine 1/2 Umdrehung des letzten Zahnrades zu reduzieren und das Poti entsprechend zu drehen.



Das letzte Zahnrad des Getriebes hat normalerweise eine Nase, welche bei Endanschlag an einem entsprechenden Anschlag im Gehäuse ansteht und so den Weg begrenzt. Am Ende des letzten Zahnrades befindet sich

eine „Befestigung“. Die besteht meistens aus einer Verzahnung und einer Schraube zur Befestigung. Der Servohebel kann auf die Verzahnung aufgesteckt werden und wird mit einer Schraube befestigt.

Wenn ein Servo also bei einem Mittenimpuls den Hebel nicht in Mittelstellung fährt, dann kann es auch daran liegen, dass der Hebel bereits verdreht aufgesteckt wurde. In so einem Fall einfach die Schraube am Hebel lösen, den Hebel abziehen und in der gewünschten Position neu aufstecken. Achtung: Oft sind die Verzahnungen so angeordnet, dass sich beim Drehen des Hebels um 180° etwas andere Positionen ergeben.



## Anschluss

Gab es früher je nach Herstellerfirma unterschiedliche Stecksysteme, so hat sich im Laufe der Zeit der sog. Uni-Anschluss durchgesetzt. Unterschiede gibt es eigentlich nur noch in der Lage, Größe und Position diverser Verpolschutznasen an den Buchsen, die ein verpoltes Einstecken in den Fernsteuerungsempfänger verhindern sollen. Allerdings sind diese Einrichtungen meistens so "windig" ausgeführt, dass man die Buchse mit etwas "sanfter" Gewalt dann doch auch verkehrt herum an den entsprechenden Stecker anstecken kann. Vorausgesetzt am Empfänger ist überhaupt eine mechanische Blockade gegen Verpolung vorgesehen. Elektrisch ist der Uni-Stecker so aufgebaut, dass er das in der Elektronik übliche 2.54mm Rastermaß benutzt. Er passt also problemlos auf die in der Elektronik üblichen Steckerleisten mit genau demselben Rastermaß.

Dieser Stecker ist mit einem 3-poligen Flachband-Kabel mit der eigentlichen Servoelektronik verbunden. Gebräuchlich sind einige verschiedene Farbschemen bei diesen Kabeln:

- schwarz - rot - weiß (GND - Vcc - Signal)
- schwarz - rot - gelb
- braun - rot - orange
- schwarz - rot - blau

Getreu den in der Elektronik üblichen Gepflogenheiten ist schwarz immer Masse, rot immer die Versorgungsspannung und die dritte Leitung (weiß, gelb, orange, blau, ...) ist die Signalleitung, über die das Servo mit Pulsen versorgt wird, welche ihm die anzufahrende Position mitteilen. Wenigstens in einem Punkt sind sich aber alle Hersteller einig: Die Versorgungsspannung wird immer über die mittlere der 3 Adern des Flachbandkabels geführt, die auch immer rot ausgeführt wird.

## Stromversorgung

Servos werden an Vcc (+5V), GND (Masse) und Impulsleitung betrieben. Dabei muss Vcc und GND direkt bzw. so kurz wie möglich mit dem Netzteil verbunden werden. Die Impulsleitung wird direkt an einem  $\mu C$  betrieben. Manche Servos erzeugen kleine Störungen auf der Versorgungsspannung, die einen  $\mu C$  durchaus zum Abstürzen bringen können. Muss man Servos gemeinsam mit einem  $\mu C$  von derselben Stromquelle betreiben, so sollte man diese Störimpulse mit Kondensatoren unterdrücken. Am besten ist eine Mischung aus kleinen Kondensatoren ca. 100nF und etwas größeren ca. 10 - 1000  $\mu F$ .

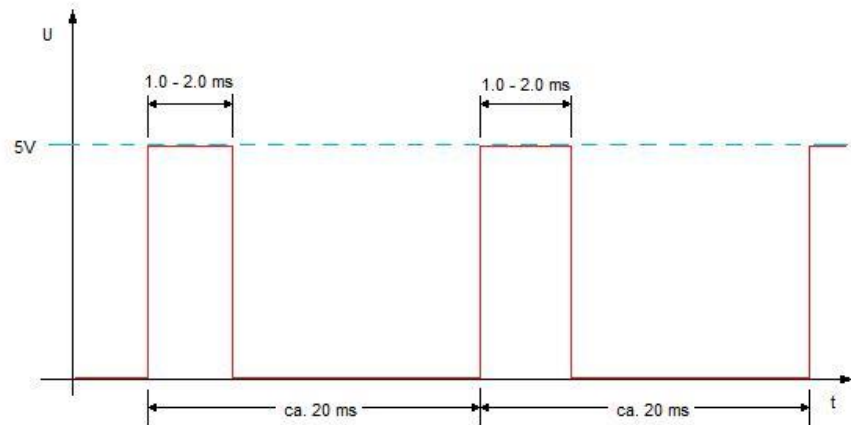
## Unterschiede

Leider gibt es noch relativ viele Unterschiede. Das sind z.B. links oder rechts drehend oder der wirkliche Drehbereich mit dem Anschlag. Der angegebene Drehbereich von 180° kann manchmal schlecht erreicht werden. Bei unterschiedlichen Typen, Größen oder Hersteller gibt es ebenfalls Unterschiede. Jeder Servo muss auf seine korrekten Daten getestet und angepasst werden.

## Signalaufbau

Das Signal, das an den Servo geschickt wird, hat eine Länge von ca. 20ms. Für das Servo wichtig ist die Impulsdauer in der ersten Phase eines Servosignals. Nominell ist dieser Impuls zwischen ca. 1ms und 2ms lang. Wobei das jeweils die Endstellungen des Servos sind, an denen es noch nicht mechanisch begrenzt wird. Eine Pulslänge von 1.5ms wäre dann Servomittelstellung.

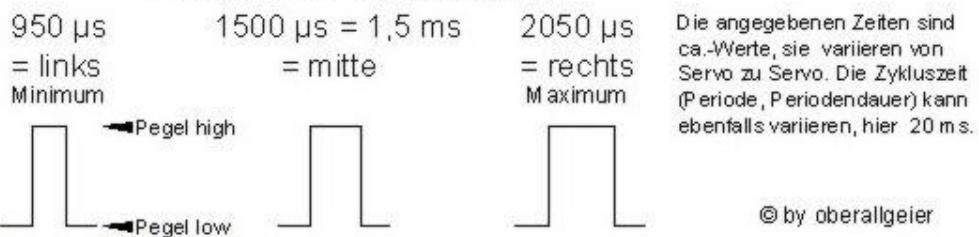
## Servo Impulsdiagramm



Im Internet habe ich diese sehr schöne Erklärung und Zeichnung von [oberallgeier](#) gefunden.



### Beispiel Pulslängen



In Abhängigkeit einer Impulslänge wird die Position eines Servos gezeigt. Dabei entstehen 3 Stellungen:

- 950  $\mu$ s - Stellung links
- 1500  $\mu$ s - Stellung mitte
- 2050  $\mu$ s - Stellung rechts

**Die Daten sind angenommen und können sich ändern. Bitte alle Angaben testen !  
Die Impulse müssen mehrfach gesendet werden.**

## Digital- und Analogservos

Der hauptsächliche Unterschied besteht darin, dass digitale Servos schneller und genauer sind und ihre Position besser halten können. Analoge Servos geben ihrem Motor alle 20ms ein Signal. Wirkt eine Kraft auf das Servo, braucht die Steuerelektronik bis zu 20ms, um dem Motor einen neuen Spannungsimpuls zu senden. Bei Digitalservos sendet die Steuerelektronik alle 400µs einen Impuls an den Motor. Der Motor bekommt die Spannungsimpulse viel schneller. Umso öfter ein Motor Spannung bekommt, umso mehr Leistung kann er verrichten. Man sollte aber auch bedenken, dass dies einen höheren Stromverbrauch zur Folge hat.

Als Beispiel habe ich wieder ein kleines Programm geschrieben. Dieses Programm ist sehr einfach und dient nur zur ersten Ansicht der Funktion.

```
/* ATB_B1_Servo_1.c Created: 12.11.2014 18:32:17 Author: AS */
#define F_CPU 16000000UL // Angabe Frequenz
#include <avr/io.h>
#include <util/delay.h>
int main (void)
{
  DDRA = (1<<PA1); // Angabe Ausgang
  while(1) // Wiederholung
  {
    PORTA |= (1<<PA1); // Ausgang einschalten
    _delay_us(1650); // Zeit Drehung
    // Begrenzung beachten - von ca. 1000us bis 2000us
    PORTA &= ~(1<<PA1); // Ausgang abschalten
    _delay_ms(18); // Wiederholt alle 18ms
  }
  return 0;
}
```

Innerhalb der **while(1)** Schleife wird der Ausgang **PA1** eingeschaltet und nach **1650 µs** wieder ausgeschaltet. Nach einer Pause von **18 ms** wiederholt sich der Vorgang. Die Information zur Position befindet sich in der Angabe **1650**. Dabei kann die Angabe **1000** z.B. **linker Anschlag** und die Angabe **2000** z.B. **rechter Anschlag** bedeuten. In Abhängig des verwendeten Servos können diese Angaben sehr stark schwanken.

### Bitte alle Angaben und Einstellungen testen!

Im ersten Programm haben wir uns allein auf die Einstellung einer Position beschränkt. Es geht aber auch komfortabler ohne Einsatz eines Timers. Man kann z.B. :

- Festlegung einer Startposition
- Festlegung einer Mittelposition
- Drehung mittels Taster Bedienung

Einstellungen / Ein- und Ausgänge

- Quarzfrequenz 16 000 000 Hz (16 MHz)
- Servoanschluss PA 1
- Taster PA3, PA5, PA7

```

/* ATB_B1_Servo_2.c Created: 15.11.2014 17:26:21 Author: AS */
#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

void waitus (uint16_t us)           // Unterprogramm Pause
{
    for(; us>0; us--)_delay_us(1);
}

int16_t stellung;
int16_t position;
int16_t positionmitt;
int16_t positionmax;
int16_t positionmin;

position = 1360;                    // Angabe start
positionmax = 1730;                 // Angabe max links
positionmin = 390;                  // Angabe max rechts
positionmitt = 1060;                // Angabe mitte

int main (void)
{
    DDRA = (1<<PA1);                // Ausgang und Servo
    while(1)
    {
        if (!(PINA & (1<<PA7)))      // berechnet max
        {
            _delay_ms(20);           // Tasterentprellung
            if (!(PINA & (1<<PA7)))
            {
                position = position + 5;
            }
        }
        if (!(PINA & (1<<PA3)))      // berechnet min
        {
            _delay_ms(20);           // Tasterentprellung
            if (!(PINA & (1<<PA3)))
            {
                position = position - 5;
            }
        }
        if (!(PINA & (1<<PA5)))      // setzt mitte
        {
            _delay_ms(20);           // Tasterentprellung
            if (!(PINA & (1<<PA5)))
            {
                position = positionmitt;
            }
        }
    }
}

```



```

    }
}
stellung = position;
if(stellung >= positionmax) // Begrenzung max
    stellung = positionmax;
if(stellung <= positionmin) // Begrenzung min
    stellung = positionmin;

PORTA |= (1<<PA1); // setzt Servo - ein
waitus(stellung); // Aufruf Unterprogramm
PORTA &= ~(1<<PA1); // setzt Servo - aus
_delay_ms(18); // ist nicht kritisch
}
return 0;
}

```

Das Programm in kleinen Schritten:

```

#define F_CPU 16000000UL
#include <avr/io.h>
#include <util/delay.h>

```

Angabe der Quarzfrequenz und der notwendigen Dateien

```

void waitus (uint16_t us) // Unterprogramm Pause
{
    for(; us>0; us--)_delay_us(1);
}

```

Da die Anweisung `_delay_us()` keine Variable zulässt, müssen wir mit einem kleinen Trick arbeiten. In unserem Unterprogramm wird `_delay_us(1)` so oft aufgerufen, wie wir für die Position des Servos benötigen. Dazu erfolgt die Übergabe der notwendigen Zeit mit der Variablen `us`. Das Unterprogramm wird erst verlassen, wenn die **for-Schleife** entsprechend oft durchlaufen wurde.

```

int16_t stellung;
int16_t position;
int16_t positionmitt;
int16_t positionmax;
int16_t positionmin;
position = 1360; // Angabe start
positionmax = 1730; // Angabe max links
positionmin = 390; // Angabe max rechts
positionmitt = 1060; // Angabe mitte

```

Definition der Variablen und Zuweisung der Werte

```

DDRA = (1<<PA1); // Ausgang und Servo

```

Angabe des Ports / Pins Servoanschluss

```

if (!(PINA & (1<<PA7))) // berechnet max
{
  _delay_ms(20); // Tasterentprellung
  if (!(PINA & (1<<PA7)))
  {
    position = position + 5;
  }
}

```

Tasterentprellung - einfach. Es wird PA7 abgefragt und negiert, 20ms gewartet und noch einmal abgefragt. Ist der Taster immer noch betätigt wird die Variable um den Wert 5 erhöht. Diese Abfrage wird noch für die Berechnung min und positionmitt verwendet.

```

stellung = position;
if(stellung >=positionmax) // Begrenzung max
  stellung = positionmax;
if(stellung <=positionmin) // Begrenzung min
  stellung = positionmin;

```

Übergabe der Variablen **position** an **stellung** und Begrenzung der min. und max. Werte zum Schutz des Servos

```

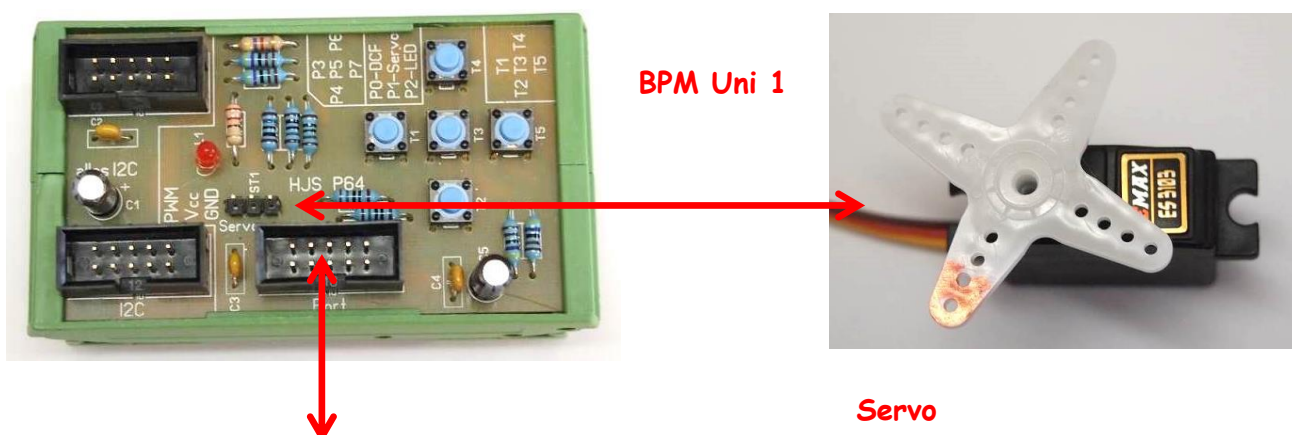
PORTA |= (1<<PA1); // setzt Servo - ein
waitus(stellung); // Aufruf Unterprogramm
PORTA &= ~(1<<PA1); // setzt Servo - aus
_delay_ms(18); // ist nicht kritisch

```

Einschalten PA1, Aufruf Unterprogramm und warten der angegebenen Zeit, PA1 ausschalten, 18ms warten und alles wiederholen.

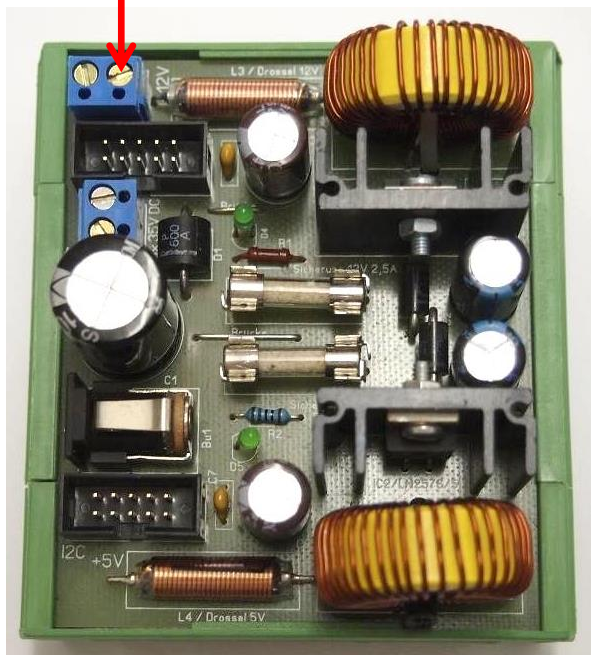
Innerhalb der Tasterabfrage ist eine Pause von 20ms. Diese wirkt sich nicht negativ auf die Einstellung des Servos aus.

So könnte ein Aufbau mit BM NT2, BM Board 1, BPM Uni 1 und einem Servo aussehen. Die Verbindung erfolgt mit 10-pol. Flachkabel. Bitte genau die Ports beachten.





**BM Board 1**



**BM NT 2**

**Beim Anschluss des Servo unbedingt die Kabelfarben bzw. Polung beachten. Auf meiner Platine habe ich sie extra vermerkt.**

Einige Teile des Textes wurden zur besseren Übersicht farblich gestaltet.  
Die Nutzung erfolgt auf eigenes Risiko.

Ich wünsche viel Spaß beim Bauen und programmieren  
Achim

[myroboter@web.de](mailto:myroboter@web.de)

Au dem Titelbild habe ich 3 Servos der Firma boxtec dargestellt.