Direkt Mode Servo handling

Servo2 Seite

Die Parameter:

1. Servo Modul Adresse
2. Servo Control Betriebsart („Normal Position“, „Training End Pos“, „Training End Pos (special)“, „Einstellen Max Geschwindigkeit“, „Reset Servo“)
3. Use fast Refresh – für Experten, schnellere Reaktion auf Schieberänderungen
4. Servo Position-Schieber
5. Enter-Taster
6. Pos 0 – Taste – stellt den Schieber auf 1 (0 geht im Legacy Mode nicht)
7. Pos255-Taste – stellt den Schieber auf 255
8. Servo Programmieren-Taster
9. Attiny Direkt Programmieren (nur zum Testen für experten)

**Servo Modul Adresse**

Adresse des Servo Moduls, das eingestellt werden soll - bis jetzt können leider noch keine LED-Strang-Kanäle angesprochen werden.

**Achtung** Bei der modifizierten Platine belegt jeder Servo eine RGB-Adresse. (Bei der originalen Platine hatte das gesamte Modul eine RGB-LED Adresse und die Servos wurden durch die R-G-B Farbkanäle angesprochen). Die Adressen bei der modifizierten Platine sind fortlaufend. D.h. wenn die erste Servo-Adresse 5 ist, dann haben die beiden anderen Servos die Adresse 6 und 7. Diese Adresse muß bei der Endlageneinstellung und in der Servo-Animation angegeben werden, damit der richtige Servo angesprochen wird.

Das mit den Kanälen ist leider etwas mißverständlich: Es gibt z.B. bei der ESP32-Platine 8 Kanäle für 8 LED Stränge. Und es gibt bei den Servo-Platinen Servo-Kanäle für die 3 Servos.

**Servo Control Betriebsart**

Bestimmt die aktuelle Betriebsart:

* "Normal Position",
* "Training End Pos",
* "Training End Pos (special for Experts only)",
* "Einstellen Max Geschwindigkeit",
* "Toggle Invers",
* "Reset Servo",
* "Factory Reset",
* "Reset last mem pos",
* "Tune Clock",
* "Einstellen Legacy Speed"

**"Normal Position"**

In dieser Betriebsart kann der Servo direkt über den Sero Positionsschieber gesteuert werden. Der Servo folgt dem Schieber direkt.

Achtung: Direkt nach dem Flashen des Attiny85 und im Normalmodus bewegt sich der Servo mit dem Schieberegler von 0 bis 255 nur etwa 30°. Die 30 Grad default sind eine Sicherheitsfunktion, damit man sich nicht gleich seine Mechanik beschädigt. Im Trainingsmodus kann der gesamte Bereich angefahren werden.

**"Training End Pos"**

In dieser Betriebsart werden die Endlagen eingestellt.

- Eine Schieberposition zwischen den aktuellen Endlagen suchen (siehe Achtungshinweis unten) - Mit dem Schieber aus dieser Position langsam in die eine Endlage fahren - Enter-Taste drücken - Mit dem Schieber in die andere Endlage fahren - Enter-Taste drücken - Fertig

In Normal\_Betriebsart, die Endlagen testen

**Achtung**: In der Trainingsbetriebsart kann **jede** Servo Position angefahren werden. Also auch Positionen, die für die Mechanik **gefährlich** sein können!! Der Attiny hat eine **Sicherheitsfunktion**. Da das Trainingstool nicht wissen kann, wo die aktuellen Endlagen des angeschlosenen Attinys sich befinden, muß der Schieber erst in eine Position zwischen den aktuellen Endlagen gebracht werden. Man bewegt also am besten den Schiber von der 0 Stellung aus langsam nach rechts, bis der Servo dem Schieber folgt. Ab jetzt kann man jede beliebige Endstellung anfahren und programmieren.

**"Training End Pos (special)"**

In dieser Betriebsart werden die Endlagen in einem erweiterten Bereich eingestellt.

- Mit dem Schieber aus der Mittelposition langsam in die eine Endlage fahren - Enter-Taste drücken - Mit dem Schieber in die andere Endlage fahren - Enter-Taste drücken - Fertig

In Normal\_Betriebsart, die Endlagen testen

Wenn du den Servo im Normalbereich mit den Endlagen trainierst, dann entspricht der Bereich von 0 bis 255 den PWM Werten von 1ms bis 2ms. Wenn du das selbe mit der Einstellung (spezial) tust, dann umfassen die Werte 0 bis 255 die PWM Werte von 0,5ms bis 2,5ms.

**Achtung**: In der Trainingsbetriebsart kann **jede** Servo Position angefahren werden. Also auch Positionen, die für die Mechanik **gefährlich** sein können!! Der Attiny hat eine **Sicherheitsfunktion**. Da das Trainingstool nicht wissen kann, wo die aktuellen Endlagen des angeschlosenen Attinys sich befinden, muß der Schieber erst in eine Position zwischen den aktuellen Endlagen gebracht werden. Man bewegt also am besten den Schiber von der 0 Stellung aus langsam nach rechts, bis der Servo dem Schieber folgt. Ab jetzt kann man jede beliebige Endstellung anfahren und programmieren.

**"Einstellen Max Geschwindigkeit"**

In dieser Betriebsart kann die „Höchstgeschwindigkeit“ eingestellt werden.

Hierzu eine kleine Erklärung, was ihr dort als „Höchstgeschwindigkeit“ einstellen könnt und wie dieser Wert wirkt. (die wirklich wirksame Geschwindigkeit macht die Servo Animation, bzw. die MLL-Hauptplatine!)

Einstellbar sind, als maximale Begrenzung, auch wieder die Werte von 0 bis 255

- Bei 0 (Null) ist die Begrenzung abgeschaltet. - Die Werte von 1 bis 255 stellen die maximal zulässige Schrittweite der Stellwerte (die gehen auch von 0 bis 255) [b]PRO ÜBERTRAGUNG[/b] von der Hauptplatine dar! - Wenn ich z.B. 10 einstelle, dann dürfen die Stellwerte immer nur 10 Stellwertpunkte pro Übertragung weiter springen, sonst werden sie begrenzt - Bei 20ms Übertragungstakt (also 50 Stellwerte pro Sekunde) kann ich damit z.B. die Gesamtstrecke des gesamten Stellwegs in einer halben Sekunde zurücklegen „1 / ( 50 / (255 / 10)) = ~0,5-Sek“ - Die maximale Begrenzung wäre der Wert 1, mit dem es 5 Sekunden dauert, bis der Stellweg zurückgelegt ist.„ 1 / ( 50 / (255 / 1)) = ~5-Sek“

Aber !!!ACHTUNG!!! Dieser Wert ist nur als **Obergrenze** gedacht (um mechanische Beschädigungen zu vermeiden) Wenn man zu stark begrenzt, dann werden die Werte von Harolds pyPG bzw. der Hauptplatine nicht mehr schnell genug ausgeführt sondern merkwürdig aussehend gedämpft.

PS: Der Wert 255 würde wirken, wie mit 0 abgeschaltet, da ja dann der Sprung von 255 Werten, als von Stellwert 0 zu Stellwert 255 ad hoc erlaubt ist.

**"Reset Servo"**

Der Servo wird wieder auf die Startwerte zurückgesetzt

**"Toogle Invers"**

Dreht die Drehrichtug des Servos um,

**"Factory reset"**

Der komplette Attiny wird zurück gesetzt,

**"Reset last mem pos"**

Um das schnelle ruckartige Bewegen des Servos zu reduzieren, speichert der Attiny die letzte Position des Servos. Wenn der Attiny wieder Strom bekommt, muß erst ein Stellwert mit dieser Position beim Attiny ankommen, bevor er den Servo bewegt. In bestimmten Fällen kann das zu einer Blockade führen, deshalb kann man mit diesem Befehl, den Speicher löschen.

**"Tune Clock",**

Hintergrund: Die DM-Servo Platine wird mit einem ATTiny85 ohne Quarz, nur mit dem internen Oszilator, betrieben. Dieses ist nur sehr maessig genau! Der Tiny wird zwar von Atmel/Micochip vorkalibriert, aber dennoch wurden bei eingestellten/gefuseten 16 MHz Werte zwischen 15,5 und 17,5 MHz gemessen. Zudem ist die Frequenz temperaturabhaengig. Diese Genauigkeit reicht, um das Signal eines WS2811 Chips, oder einer WS2812B GRB-LED, zu dekodieren. Auch reicht sie um am Ausgang ein WS2811 Signal weiterzugeben, das sich innerhalb der zulaessigen Toleranzen bewegt. Sie reicht allerdings nicht um bei mehreren hintereinander geschalteten ATTiny85 garantiert innerhalb der Toleranzen zu bleiben. Dass liegt daran, dass in diesem Fall der ATTiny85 nicht nur seine eigene Abweichung nicht kennt, sondern auch die des Vorgaengers und/oder Nachfolgers unbekannt ist.

Loesungsansatz: Es fiel auf, dass eine aufeinanderfolgende Kette von quarzlosen ATTiny85 dann funktionieren kann, wenn man sie nach Geschwindigkeit sortiert! Hierbei muss der schnellste an die erste Position und der langsamste an die letzte. Zudem duerfen zwei aufeinanderfolgende Tinys nicht mit ihren Geschwindigkeiten zu dicht zusammenliegen, so dass sie durch Schwankungen zweitweise die Rollen vertauschen. Dieses Prinzip des strikt absteigenden "Terrassen-Wasserfalls" kann durch die Benutzung einer von Atmel/Microchip vorgesehenen Kalibriermoeglichkeit (AVR053 Application Note 09/2016), auch ohne exakte Kalibrierung an einem externen Messgeraet, kuenstlich hervorgerufen werden.

Alle Trimwerte beziehen sich immer auf den von Atmel vorgetrimmten default Inhalt von OSCCAL!

Der trimbare Wertebereich geht von 0..35! Dies bezieht sich auf "default-OSCCAL + custom-Trim". Werte ab ca. +30 koennen auf "schnellen" Exemplaren schon kritisch sein und den Tiny in einen Bereich bringen, dass der WS2811 Empfang nicht mehr funktioniert!

Die eingestellten Werte werden persistent im EEPROM gespeichert und beim nächsten Tiny-booten, noch vor der ersten WS2811 Uebertragung, wiederhergestellt um zu verhindern, dass beim Start fehlerhafte Daten durch die WS2811 Kette laufen!

Kalibrierprozess fuer mehrere aufeinanderfolgende ATTiny85 bzw. DM-Servo Platinen:

1. Als erstes wird der LETZTE ATTiny85 der LETZTEN DM-Servo Platine auf den Wert 0 (s.o.) eingestellt. Hierzu wird voruebergehend die ERSTE Platine verwendet und der Tiny, noch erfolgreichem Test der Funktion und der WS2811 Weiterleitung, einen Platz weitergesteckt.
2. Als naechstes wird ein weiterer ATTiny85 in den ERSTEN Steckplatz, der nun dem bereits kalibrieten Chip vorhergeht, gesteckt und auch auf 0 eingestellt. Da die "0" des einen Tiny nichts mit der "0" des anderen Tiny zu tun hat und es zudem Schwankungen geben kann, durch die die beiden Tinys ihre Rollen staendig vertauschen, wird der zweite/letzte Tiny nun nicht mehr korrekt funktionieren. Auch die WS2811 Weiterleitung wird fehlerhafte Bits liefern. Nun erhoehen wir die Kalibrierzahl langsam um ca. 5-10 bis der nachfolgende Tiny und auch die WS2811 Weiterleitung wieder korrekt funktionieren. Je nach Schwankungsungenauigkeit kann die Zahl unterschiedlich sein, die dafuer benoetigt wird, dass ganz sicher der Vorgaenger etwas schneller laeuft als sein Nachfolger und die Rollen durch Schwankungen niemals getauscht werden! Ein Differenzwert, zwischen zwei Tinys, von mehr als 10 ist dafuer fast nie notwendig!
3. Fuer die Integration eines weiteren Tiny werden alle bisherigen Tinys einen Platz weiter gesteckt und der nun erste Tiny mit einem etwas groesseren Wert versehen, als der zweite Tiny in der Kette. Danach wieder Funktion der nachfolgenden Tinys und der WS2811 Weiterleitung püfen. Nach diesem Muster koennen auch weitere Tinys integriert werden!
4. Es ergeben sich Kalibrierwertketten wie z.B. (1):20-(2):10-(3):0 (drei DM-Servo) oder z.B. (1):30-(2):20-(3):10-(4):0 (vier DM-Servo)
5. Eine korrekte Funktion aller verketteten Tinys ist nun auch daran zu erkennen, dass sich bei gemeinsamem "power up" eine gleichmaessige Blink-Kaskade der Status-LEDs der DM-Servo Platinen ergibt!

**"Einstellen Legacy Speed"**

Mit dieser Einstellung kann der DM-Servo das Verhalten der Servos der bisherigen Servo-Platine 510 simulieren. Damit können die bisherigen Servo-Macros Servo2- Servo5 auch mit den DM-Servos verwendet werden.

Diese Funktion MUSS als letzte Einstellung durchgeführt werden.

Zuerst müssen die Endlagen, wie oben beschrieben, eingestellt werden.

Zum Schluß **"Einstellen Legacy Speed"** und mit dem Schieber die gewünschte Geschwindigkeit einstellen und „Enter“ betätigen. Die Betriebsart wechselt dann automatisch zu „Normal“ zurück. Jetzt kann man die Geschwindigkeit mit dem Pos0 und Pos255 Taster testen. Den Vorgang wiederholen, bis die Geschwindigkeit stimmt.

Danach kann der Servo mit den Macros Servo2 – Servo5 gesteuert werden.

Dazu muß beim LED-Kanal IMMER „Rot“ ausgewählt werden. Der Servo wird durch die Servo-Adresse bestimmt.

ACHTUNG: Bei der Platine 510 konnten 3 Servos angeschlossen werden Die 3 Servos hatten dieselbe LED Adresse, aber wurde durch die LED-Kanäle Rot, Grün, Blau unterschieden.

Bein den DM-Servos hat JEDER Servo eine eigene LED-Adresse die fortlaufend bei der Adresse des ersten Servos beginnt.

Beispiel:

Bisher: Platine 510 ist an Servoadresse 5

Servo1: 5-rot

Servo2: 5-grün

Servo3: 5-blau

Bei DM-Servo in legacy mode:

Servo1: 5-rot

Servo2: 6-rot

Servo3: 7-rot

Im Unterschied zur 510 Platine kann der Bereich von 0 bis 255 genutzt werden (obwohl das Programm meckert, daß das nicht geht)

Hintergrund:

Der Legacy Speed ist, wie in der alten 510er Lösung, in einer maximalen Veraenderung von 1/8 us (Mikrosekunde) pro 20ms definiert.

BEISPIEL: Ein mittlerer Wert von 128 ergibt eine Laufzeit von Stellung 0 (1ms) zu 255 (2ms) "8000 1/8us / 50 / 128" = 1.25 Sec.

FULL-LEGACY: Ist ein Legacy-Speed programmiert, sowie die Endlagen eingestellt, kann das Servo alleine durch Stellwerte im Bereich von 1..255 im zweiten Farbkanal bewegt werden. Der erste und der letzte Farbkanal muessen hierbei 0 sein!

ACHTUNG!: Auch die maximale Geschwindigkeit von 255 ist vergleichsweise LANGSAM und fuer schnelle Animationen, wie z.B. Nachschwingen, NICHT geeignet! Hier ist "max speed", definiert in Schrittweite, als Limit gegen Zerstörung, zu verwenden!

**Servo Position-Schieber**

Der Servo-Positionsschieber erlaubt die direkte Steuerung des Servos. Es werde direkt die eingestellten Werte an den Servo gesendet.

**Enter-Taste**

Mit der Enter-Taste werden Endlagen bestätigt

**Servo Programmieren-Taste**

Mit der Servo Programmieren Taste, wird ein Dialog zum Programmieren der Firmware des Attinys aufgerufen. Es öffnet sich das „Optionen“-Fenster des Patterconfigurators für den „Servo2“. Durch Drücken von „Prog Servo2“ wird ein Auswahldialog aufgerufen, in dem man die Firmware-Datei (.hex) auswählen kann. Wenn die Firmware einen fertigen Stand erreicht hat, wird dieser Dialog entfernt und die Programmierung startet direkt.

**Attiny direkt Programmieren**

Dies ist eine Funktion, die noch im Test ist, und erst später freigegeben wird.

**Endpositionen einstellen**

Der Einstellvorgang läuft folgendermaßen ab:

1. Die Adresse des Servomoduls einstellen
2. Servo Control Betriebsart „Normal Position“
3. Die Servo Position in die Mitte stellen.
4. Die Servo Control Betriebsart auf: „Training End Pos“ einstellen
5. Eine Schieberposition zwischen den aktuellen Endlagen suchen (siehe Achtungshinweis unten)
6. Mit dem Schieber aus dieser Position langsam zur gewünschten Endposition fahren. Der Servo sollte dieser Bewegung folgen.
7. Wenn die erste Endposition erreicht ist, „Enter drücken“
8. zur zweiten Endposition fahren.
9. Nochmal „Enter“ drücken
10. Betriebsart wieder auf „Normal“ stellen.

**Achtung**: In der Trainingsbetriebsart kann **jede** Servo Position angefahren werden. Also auch Positionen, die für die Mechanik **gefährlich** sein können!! Der Attiny hat eine **Sicherheitsfunktion**. Da das Trainingstool nicht wissen kann, wo die aktuellen Endlagen des angeschlosenen Attinys sich befinden, muß der Schieber erst in eine Position zwischen den aktuellen Endlagen gebracht werden. Man bewegt also am besten den Schiber von der 0 Stellung aus langsam nach rechts, bis der Servo dem Schieber folgt. Ab jetzt kann man jede beliebige Endstellung anfahren und programmieren.

**Achtung**: Wenn du den Servo im Normalbereich mit den Endlagen trainierst, dann entspricht der Bereich von 0 bis 255 den PWM Werten von 1ms bis 2ms. Wenn du das selbe mit der Einstellung (Training End Pos spezial) tust, dann umfassen die Werte 0 bis 255 die PWM Werte von 0,5ms bis 2,5ms. Damit ist ein größerer Stellumfang möglich. Es kann aber sein, daß nicht alle Servos diese unterstützen. Wenn der Einstellbereich mit den Endlagen bei „Training End Pos“ nicht ausreicht, kann man „Training End Pos spezial“ ausprobieren.

Anmerkung: In den ersten 10 Sekuden nach dem Booten findet eine Bewegung nur dann statt, wenn der Wert, der als letzter Positioneswert gespeichert wurde "ueberstrichen" wird, oder wenn vorher noch keine Position gespeichert wurd. (auch nach Reset der letzten Position) Nach den 10 Sekunden wird JEDER Wert akzeptiert und ausgefuehrt.

**Hinweise aus der Diskussion**

Hier sind einige Hinweise aus der Diskussion im Forum zusammengefasst:

- Genau genommen ist der Bereich von 0 bis 255, beim Direct Mode Servo, eine Art „Zahlenwerte Matroschka“! (die russische „Puppe in der Puppe“) Wenn du den Servo im Normalbereich mit den Endlagen trainierst, dann entspricht der Bereich von 0 bis 255 den PWM Werten von 1ms bis 2ms. Wenn du das selbe mit der Einstellung (spezial) tust, dann umfassen die Werte 0 bis 255 die PWM Werte von 0,5ms bis 2,5ms. Doch egal, welche Millisekunden Werte du für die Endlagen festlegst, der gewählte Bereich hat vom Python Programm Generator aus wieder den Bereich von 0 bis 255. Das wäre auch dann der Fall, wenn der Bereich zwischen den gewählten Endlagen sehr schmal ist. Damit erhält man in dem schmalen Bereich eine sehr hohe Genauigkeit!

- Ein DM Servo belegt den Adressraum einer RGB LED mit drei Mal 8 Bit bzw. Werten von 0..255! Im ersten Kanal kommt eine Steuerinformation und eine CRC Prüfsumme, im zweiten Kanal ein Stellwert von 0.255 und der dritte Kanal ist bei der Std. MLL derzeit nicht genutzt. Außerdem hat die Std. MLL die ersten beiden Kanäle technisch vertauscht. - Nach meinen Messungen und meiner Erfahrung haben SG90 Servos, bei der Servo Norm-PWM von 1-2ms, nur 90 Grad Stellwinkel! Die 170 Grad Aussage kommt dadurch zustande, dass fast alle eben nicht 1-2ms PWMs machen, sondern mehr! Das gibt es beim DM Servo auch und heißt „Training Pos Mode (spezial)“ Warnung: (gerade auch sehr kraftvolle) Servos können sich, wenn sie an den mechanischen Anschlag gefahren werden, selbst beschädigen!

Wenn der Attiny direkt an einem ARDUINO angeschlossen ist, verhält er sich komisch.

Antwort: Eine direkte Verbindung Attiny zum ARDUINO ist nicht zu empfehlen. Es sollte immer ein WS2811 oder WS2812 dazwischen liegen. Tatsächlich haben die Tinys hinter einer echten physikalischen WS2811 oder WS2812B LED eine ganz passable Reichweite! Die nächste WS2811, hinter dem Tiny, kann auch ganz ordentlich weg sein (z.B. 70cm kein Problem), aber ATMega238 auf ATTiny85 ist irgendwie seltsam. Es wundert mich ein wenig, denn ATTiny auf ATTiny, also zwei meiner Direct Mode Servo Platinen, vertragen durchaus einen halben Meter dazwischen!

Blinkcodes der blauen LED auf der 512 Platine:

Blinkcodes der Status-LED:

1. Die Prioritaet hat immer die WS2811 Bus Verarbeitung und nicht die Status-LED! Bei einem ungleichmaessigen WS2811 Signal, oder Stoerungen auf dem WS2811 Bus, kann es zu ungleichmaessigen Blinkmustern kommen! Am besten funktioniert die Status-LED bei einem gleichmaessigen zyklischen Refresh des WS2811 Busses!

2. Nach dem (Re)Boot leuchtet die LED fuer 5 Sekunden dauerhaft. Diese Zeit gilt fuer einen power up in einen aktiven WS2811 Bus. Diese Zeit verlaengert sich auf unbestimmt, wenn der WS2811 Bus inaktiv ist, bis zu einem Zeitpunkt, zu dem der WS2811 Bus aktiv geschaltet wird.

3. Nach den 5 Sekunden boot blinkt die LED in einem Wechsel von 1 Sekunde aus, zu 1 Sekunde ein, also einer Zykluszeit von 2 Sekunden, wenn der WS2811 Bus aktiv ist (LED in dieser Phase abschaltbar). Ist bzw. wird der WS2811 Bus inaktiv wird dies mit einem schnellen Blinken mit einer Zykluszeit von 200 ms ein zu 200 ms aus, also 400 ms, bzw. einer knappen halbe Sekunde gesamt, signalisiert. Wird der WS2811 Bus wieder aktiv, wird der 2 Sekunden Zyklus wieder aufgenommen.

4. Status-LED waehrend des Programmierens der Endlagen

4.1 Waehrend des Einstellens der Endlagen blinkt die LED schnell mit einem Zyklus von 100 ms

4.2 So lange man sich ausserhalb des bislang programmierten (und damit erlaubten) Bereichs befindet, blinkt die LED sehr schnell mit einem Zyklus von 20 ms

4.3 Wenn man das zweite Mal das ENTER-Bit auslöst und damit die Endlagen fest programmiert, blinkt die LED kurz fünffach mit 20 ms, gefolgt von Aus-Phasen von 300 ms, fuer eine Gesamtzeit von 3 Sekunden.

Danach ist das Programmieren abgeschlossen und es wird wieder der Blinkzyklus von 2. aufgenommen!

5. Status-LED als Bestaetigung des Programmierens der Maximal-Geschwindigkeit und des Endlagen-Invertierens

5.1 Zur Bestaetigung des Programmierens erfolgt ein Blinken wie bei 4.3

6. Status-LED als Bestaetigung eines Konfig-RESET bzw. factory default etc.

6.1 Nach dem Ausloesen eines factory default fuer das aktuelle Servo blinkt die LED kurz mit 50 ms, gefolgt von jeweils Aus-Phasen von 750 ms, fuer eine Gesamtzeit von 2 Sekunden.

6.2 Nach dem Ausloesen eines factory default fuer das ALLE Servos blinkt die LED kurz doppelt mit 50 ms, gefolgt von Aus-Phasen von 650 ms, fuer eine Gesamtzeit von 2 Sekunden.

6.3 Nach dem Zuruecksetzen der gespeicherten letzten Position blinkt die LED kurz dreifach mit 50 ms, gefolgt von Aus-Phasen von 550 ms, fuer eine Gesamtzeit von 2 Sekunden.

Danach wird jeweils wieder der Blinkzyklus von 2. aufgenommen!

7. Direktes Hochladen der Firmware:

7.1 Nach Starten des Bootloaders: Wenn alles OK ist, dann blinkt die Status-LED langsam, so lange auf dem Bus ein gültiges WS2811 Signal liegt. Ohne Signal blinkt die Status-LED nicht sondern behält ihren Zustand.

7.2 Wenn der Bootloader das File über den WS2811 Bus annimmt macht sich das an einem schnell flackernden Blinken der Status LED-bemerkbar.

7.3 Wenn das Laden, über den Bootloader, erfolgreich abgeschlossen wurde (CRC16 Prüfsumme OK), dann wird die Servo Firmware gestartet und die Status-LED bleibt für 5 Sekunden an, bevor si ewieder in das langsame Blinken übergeht. Wenn es negativ verläuft fehlt die 5-Sekunden an Phase und es wird gleich langsam geblinkt.