Direct Mode Servo

In pyMobaLedLib die Seite „Servo Test 2“ öffnen.



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:

1. „Normal Position“,
2. „Training End Pos“,
3. „Training End Pos (special for Experts only)“,
4. „Einstellen Max Geschwindigkeit“,
5. „Toggle Invers“,
6. „Reset Servo“,
7. „Factory Reset“,
8. „Reset last mem pos“,
9. „Tune Clock“,
10. „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.

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 öffnmet 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 diser Dialog entfernt und die Programmierung startet direkt.

Attiny direkt Programmieren

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

Tutorials

- Tutorial: Endposition einstellen
- Tutorial: Mehrere Attinys hintereinander abgleichen
- Tutorial: Einstellen Legacy Speed

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 zwischeneinander!

From:
https://wiki.mobaledlib.de/ - **MobaLedLib Wiki**

Permanent link:
https://wiki.mobaledlib.de/anleitungen/spezial/pyprogramgenerator/servo2_seite?rev=1738837001

Last update: **2025/02/06 10:16**