

Ansteuerung von bis zu 3 Modellbauservos über die TWI-Schnittstelle mit ATTiny 25 in C

Mit Hilfe eines kleinen ATTiny25 lassen sich mit wenigen externen Bauteilen bis zu 3 Servos ansteuern.

Als zusätzlich Bauteile werden nur der obligatorische Kondensator von 0.1uF benötigt.

Ggf. noch die Pullups für SDA/SCL sowie "Angstwiderstände" von ca. 470 Ohm in den Steuerleitungen der Servos, die bei einem Kurzschluss die Ausgänge des Controllers vor dem Exitus bewahren.

Die Pinbelegung des ATTiny 25:

reset	PB5	1	8	VCC	
Servo 2	PB3	2	7	PB2	SCL
Servo 3	PB4	3	6	PB1	Servo 1
	GND	4	5	PB0	SDA

Durch Umzug auf einen ATTiny24 könnten zusätzliche Servos angesteuert werden.

Wer aber den Aufwand der Migration scheut, der benutzt einfach mehrere ATTiny25's mit unterschiedlichen TWI-Adressen.

Ein Servo erwarten alle ca. 20ms einen Impuls der Dauer von ca. 0.9 bis 2.1 Millisekunden.

Die Länge des Impulses regelt die Stellung des Servoarmes.

Der ATTiny25 arbeitet mit dem 1MHz CPU-Takt des RC-Oszillators. Sein Timer0 gibt einen 20ms-Takt vor.

Der Timer1 läuft (mit Prescale 8) mit einem Grundtakt von 8us und regelt die Einstellung der Pulslänge.

Die von den Servos erwartete Pulsdauer liegt zwischen 0.9ms und 2.1ms, bei 1.5ms nimmt der Servo theoretisch die Mittelposition ein.

Für die Neutralposition bei 1.5ms ist ein Wert von $1500\text{us} / 8\text{us} = 187$ ins Compare-Register zu schreiben.

In der Praxis verhalten sich die Servos jedoch unterschiedlich, die erforderliche Impulsdauer für die Mittelposition weicht von der Theorie häufig deutlich ab.

Daher ist für jeden Servo die Neutralposition separat einstellbar.

Die gewünschte Soll-Servoposition wird mit Werten von -75 bis +75 bezogen auf die Ruheposition angegeben ($75 * 8\text{us} = 600\text{us} = 0.6\text{ms}$).

Dummerweise verfügt der ATTiny25 nur über 8Bit Timer/Counter.

Um die Position 2.1ms anzufahren, reicht der Zähler nicht aus, denn $2100\text{us} / 8\text{us} = 262,5$.

Daher wird bei längeren Impulszeiten das Register OCR1A zuerst mit dem Wert 255 und anschließend noch einmal mit der Differenz zum benötigten Wert geladen, damit Impulslängen $> 2.048\text{ms}$ erreicht werden können.

Ist eine geschmeidige Drehung von der aktuellen Position zur Sollposition gewünscht, so kann für jeden Servo eine (reduzierte) Dreh-Geschwindigkeit eingestellt werden.

Bei einem Speed-Wert von 0 wird unmittelbar und mit Höchstgeschwindigkeit die Zielposition angefahren.

Bei Werten von 1 ... 10 werden im Abstand von 1/25 Sekunde Zwischenpositionen adressiert, die sich gegenüber der jeweiligen Vorposition um jeweils $(1 \dots 10) * 8\text{us}$ Dauer unterscheiden.

Je höher der Wert, desto schneller dreht der Servo:

Um von der Position -75 nach +75 zu fahren, vergehen bei einem Speed-Wert von "1" $((150/25) / 1) = 6$ Sekunden.

Bei einem Speed-Wert von "3" sind es $((150/25) / 3) = 2$ Sekunde, bei einem Speed-Wert von "10" bleiben noch $((150/25) / 10) = \text{ca. } 0.6$ Sekunden.

Über die USI/TWI-Schnittstelle können die Servos gesteuert und ihre Grundeinstellungen konfiguriert werden.

An den USI-Slave ist seine Adresse gefolgt von einem Byte als Code für die auszuführende Aktion sowie ggf. nachfolgenden Parametern zu senden.

In den nachfolgenden Beispielen wird angenommen, dass der Slave auf die Adresse "80" programmiert ist (die Kommata zwischen den Werten werden selbstverständlich nicht gesendet !):

Setzen der Servoposition (jeweils für einen einzelnen Servo):

80, 1, 0 Adresse "80", Code "1" (= Servo 1), Parameter "0" (= Neutralposition + 0)
80, 2, 40 Adresse "80", Code "2" (= Servo 2), Parameter "40" (= Neutralposition + 40 * 8us)
80, 3,-50 Adresse "80", Code "3" (= Servo 3), Parameter "-50" (= Neutralposition - 50 * 8us)

Setzen der Servoposition (gleichzeitig für alle drei Servos):

80, 255, 0, 40, -50 Adresse "80", Code "255" (= alle Servos), nacheinander die Positionen der Servos
Sind weniger als 3 Servos beim Kompilieren definiert, dann werden auch nur soviele Parameter erwartet wie Servos angemeldet sind !

Setzen der Neutralpositionen (jeweils für einen einzelnen Servo):

80, 10, 187 Adresse "80", Code "10" (= Servo 1), "187" = Neutralposition bei $187 * 8\mu s = 1.5ms$
80, 20, 195 Adresse "80", Code "20" (= Servo 2), "195" = Neutralposition bei $195 * 8\mu s = 1.56ms$
80, 30, 170 Adresse "80", Code "30" (= Servo 3), "170" = Neutralposition bei $170 * 8\mu s = 1.36ms$

Setzen der Speed (jeweils für einen einzelnen Servo):

80, 11, 0 Adresse "80", Code "11" (= Servo 1), "0" = unmittelbare Ansteuerung der Sollposition
80, 21, 1 Adresse "80", Code "21" (= Servo 2), "1" = Ansteuerung der Sollposition in $1 * 8\mu s$ Schritten
80, 31, 5 Adresse "80", Code "31" (= Servo 3), "5" = Ansteuerung der Sollposition in $5 * 8\mu s$ Schritten

In den angeführten vier Fällen liefert das anschließende Auslesen von 6 Byte aus dem USI-Slave die aktuellen Daten zum vorher adressierten Servo zurück:

Die Nummer des Servos, seine Neutralposition, seinen Offset, den zugehöriger OCR1A-Wert (als 16Bit-Wert, high-Byte und low-Byte nacheinander), seinen Speed-Wert.

Werden die gesendeten Daten mit nachfolgenden 0xFF's zurückgeliefert, dann ist ein Fehler aufgetreten.

Schreiben der Neutralpositionen ins Eeprom (für alle Servos):

80, 131 Adresse "80", Code "131"

Schreiben der Offsets ins Eeprom (für alle Servos):

80, 133 Adresse "80", Code "133"

Bei einem Neustart fahren die Servos in ein Position, die aus Neutralposition + Offset berechnet wird.

Schreiben der Speed-Werte (Drehgeschwindigkeit) ins Eeprom (für alle Servos):

80, 135 Adresse "80", Code "135"

Jeweils nachdem Daten ins Eeprom geschrieben werden, können die drei Werte zur Kontrolle aus dem USI-Slave ausgelesen werden.

Werden die gesendeten Daten mit nachfolgenden 0xFF's zurückgeliefert, dann ist ein Fehler aufgetreten.

03.10.2012

mfg

Michael S.