

Fernseh-Simulator

Diese Steuerung mit dem **Microcontroller ATMEL 89C4051** ist für universelle Steueraufgaben bis **8 Bit** ausgelegt. Der große **Betriebsspannungsbereich von 9 - 35 Volt**, die **8 leistungsfähigen Ausgänge** und die **16 Programmspeicherplätze** machen diese Schaltung für viele Anwendungsbereiche einsetzbar. So können ganz einfach Ablaufsteuerungen, Lichtmuster, Effektschaltungen, Ampelsteuerungen und viele weitere Anwendungen programmiert werden. Mittels des DIP-Schalters sind 16 verschiedene Programme innerhalb nur eines Bausteines ohne Schaltungsänderung frei wählbar. Zusätzlich besteht über die beiden **Eingänge E1 und E2** die Möglichkeit, die Schaltung mit externen Funktionen (Dämmerungsschalter, Lichtschranke, Thermoschalter,) beliebig zu erweitern. Der hier abgebildete **Fernsehsimulator** ist nur ein spezieller - jedoch sehr praktischer - Anwendungsfall.



Microcontroller

Ein Microcontroller ist ein komplettes System, bestehend aus der CPU (Recheneinheit/Mikroprozessor), dem Programmspeicher (Flash oder Eprom) dem Arbeitsspeicher (RAM) und der Ein/Ausgabe auf einem Chip. Diese Bausteine werden in vielen Geräten als „Mini-PCs“ eingesetzt und steuern z.B. Heizungen, Drucker, Wecker, Garagentore,

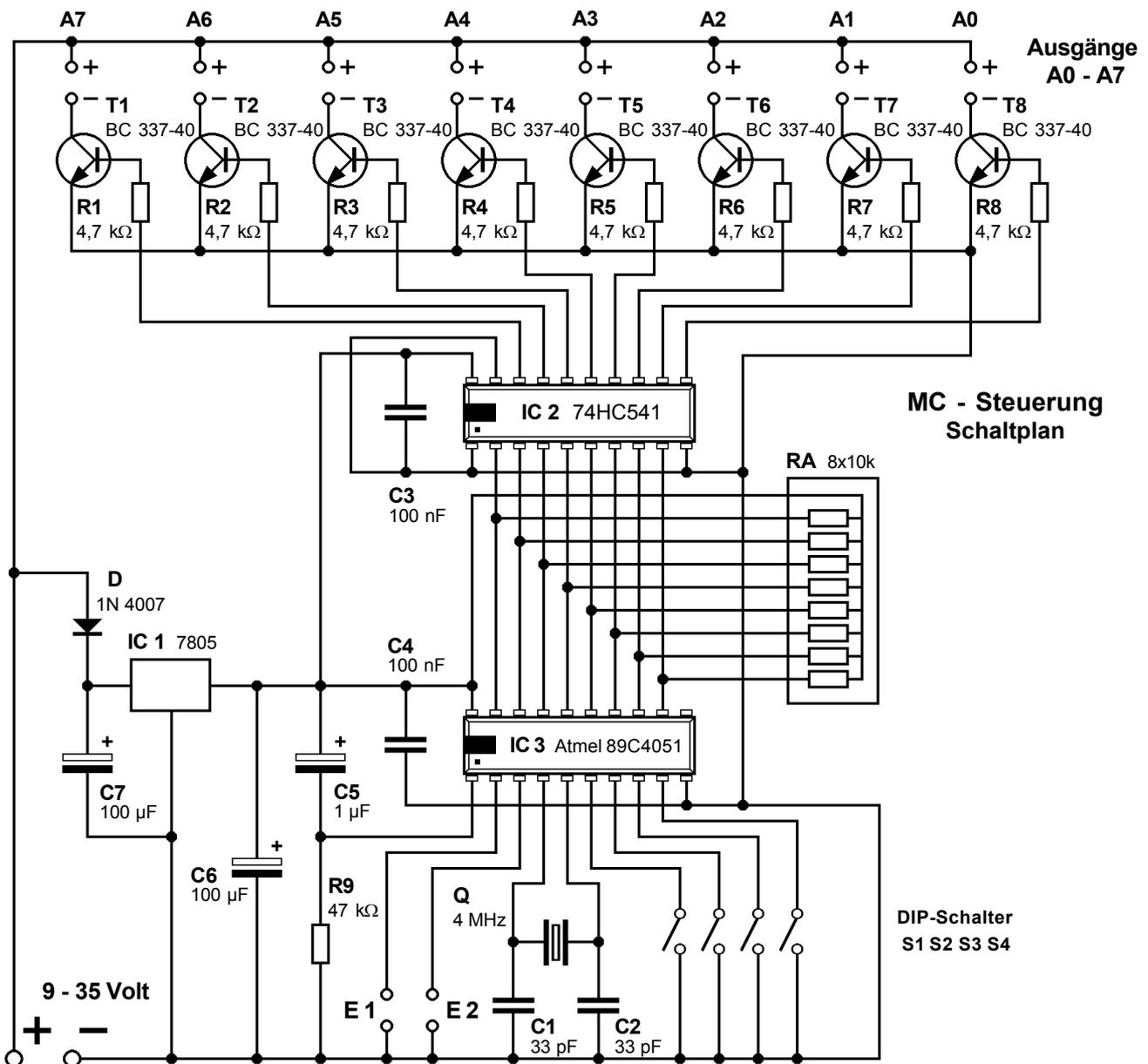
Daten Atmel 89C4051

Flash-Programmspeicher mit 4 kByte
128 Byte integriertes RAM
Taktfrequenz von 0 Hz bis 24 MHz
15 programmierbare Ein-/Ausgänge
zwei 16-Bit Timer
3,0 bis 6,0 Volt Betriebsspannung

Fernseh-Simulator

Mit dieser Schaltung können im Idealfall bis zu 5000 Leuchtdioden angesteuert werden. Für den Fernseh-simulator wird man in der Praxis ca. 25 bis 50 ultrahelle Leuchtdioden benötigen, wobei die Mischung dem Farbspektrum eines Fernsehbildes entsprechen sollte. Eine mögliche **Mischung** wäre z.B.:

LED: 18 x weiß, 10 x blau, 8 x gelb, 4 x rot, 4 x orange
Im Normalfall wird man eine elektronisch stabilisierte Stromversorgung mit 12 Volt, die für einen Strom von mindestens 250 mA ausgelegt sein soll, verwenden. Es werden so viele Leuchtdioden wie möglich in Reihe geschaltet und dann der Vorwiderstand berechnet. Dazu sollte man den Spannungsabfall der LEDs bei einem Strom von 20 mA kennen. Im Normalfall wird dieser beim Produkt angegeben. Er kann natürlich auch mit dem Messgerät ermittelt werden.



Schaltungsbeschreibung MC-Steuerung (Fernseh-Simulator)

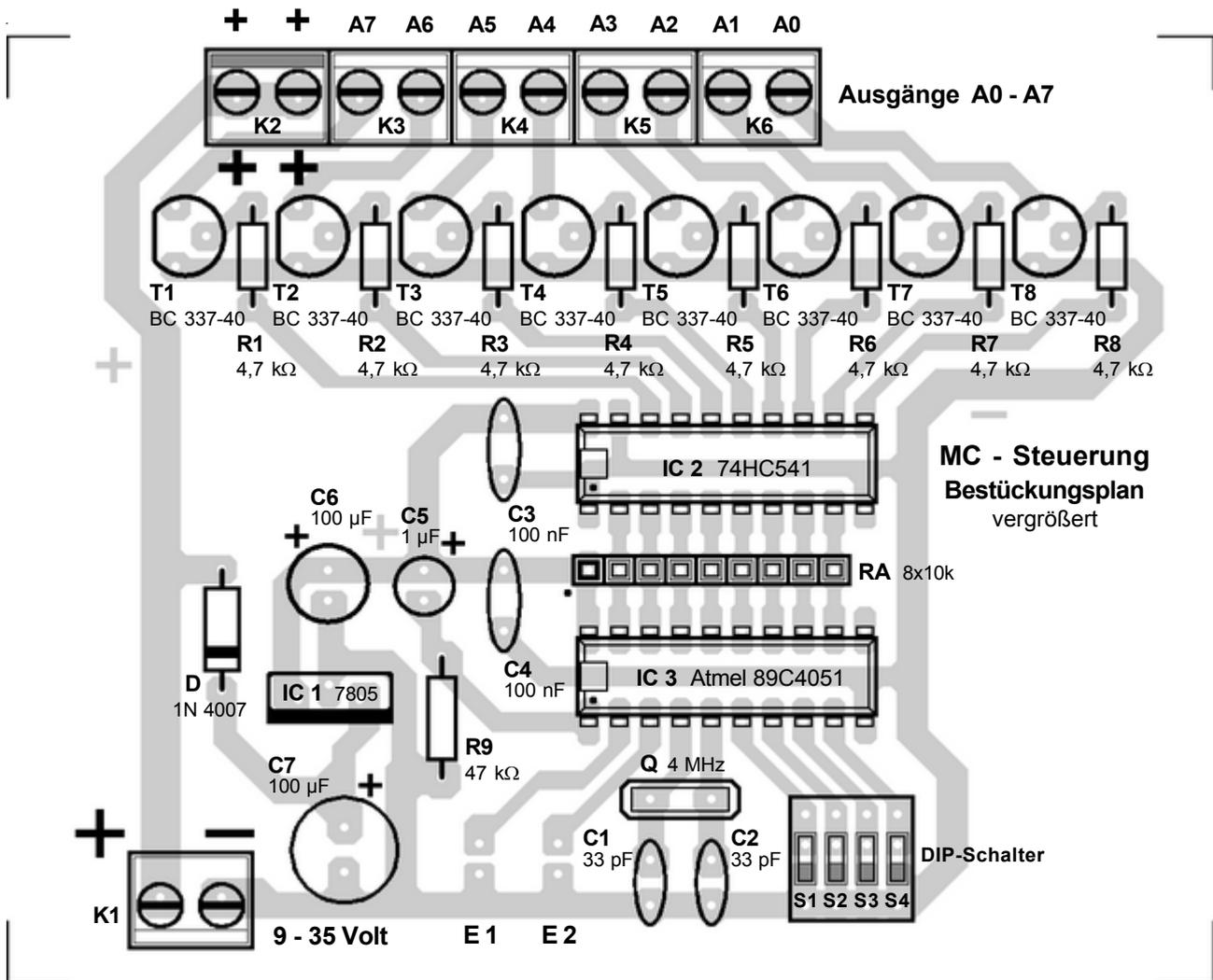
Diese Microcontroller-Steuerung ist grundsätzlich sehr einfach aufgebaut. Ihre **Intelligenz** liegt in der **Software**, die im Mikrocontroller gespeichert ist. Diese Software kann **nachträglich** nahezu **beliebig verändert** und **erweitert** werden, ohne die Hardware ändern zu müssen.

Der **Microcontroller ATMEL 89C4051** und der **Treiber-Baustein 74HC541** (74HCT541) werden über den **Spannungsregler LM 7805** mit einer stabilen Betriebsspannung von 5 Volt versorgt. Als Schutz gegen Verpolung dient die Diode D. Da die beiden Ausgänge des Microcontrollers Pin 12 (P1.0) und Pin 13 (P1.1) einen Pull-Up-Widerstand benötigen, wurde auf Grund des einfacheren Aufbaus hier das Widerstandsarray RA (8 x 10 kOhm) eingebaut.

Mit dem Treiber-Baustein 74HC541 können auch noch wesentlich stärkere Transistoren als die hier verwendeten Schalttransistoren **BC 337-40** angesteuert werden. Der Anzahl der LED, die mit dieser Schaltung gesteuert werden können, sind fast keine Grenzen gesetzt, **mehrere tausend Leuchtdioden** sind möglich.

Nach dem **Anschluss der Betriebsspannung** bewirken C5 und R9 einen kurzen **Reset** und sorgen somit für einen definierten Anfangszustand der Schaltung. Als **Taktgeber** für den Programmablauf fungiert ein **4 MHz - Quarz (Q)** mit den beiden 33 pF - Kondensatoren C1 und C2.

Die Keramik-Kondensatoren C3 und C4 (100 nF) sorgen für die Unterdrückung von kurzen Spannungsimpulsen.



Bauelemente - Stückliste

R1 - R8	4,7 kOhm	RA	Array 8 x 10 kOhm	S1 - S4	DIP-Schalter
R9	47 kOhm	D	1N 4007	K1 - K6	Klemmen
C1, C2	33 pF	T1 - T8	BC 337-40	IC-Sockel	20-polig
C3, C4	100 nF	IC1	LM 7805	IC-Sockel	20-polig
C5	1 µF/16 V	IC2	74HC541	Br.	Brücken
C6	100 µF/16 V	IC3	Atmel 89C4051	Platine	80 x 100 mm
C7	100 µF/35 V	Q	Quarz ... 4 MHz			

Bestückungsplan

Der Bestückungsplan ist vergrößert dargestellt, mit **Blick von oben** auf die **Bauelementeseite** bzw. auf die **Bestückungsseite** der Platine.

Die Bauelemente werden wie abgebildet in die Platine eingebaut. Bei folgenden Bauelementen ist auf die **Polarität** zu achten:

C5, C6, C7, D, T1 - T8, RA, IC1, IC2, IC3

Auch das Widerstands-Array RA ist gepolt in die Schaltung einzubauen. Gegebenenfalls kann der gemeinsame Anschluss mit dem Multimeter herausgemessen werden. Für IC2 (74HC541) und IC3 (ATMEL 89C4051) ist je eine 20-polige IC-Fassung vorgesehen. Der MC muss für die Programmierung entnommen werden können.

Kennzeichnung von IC1, IC2 und IC3:

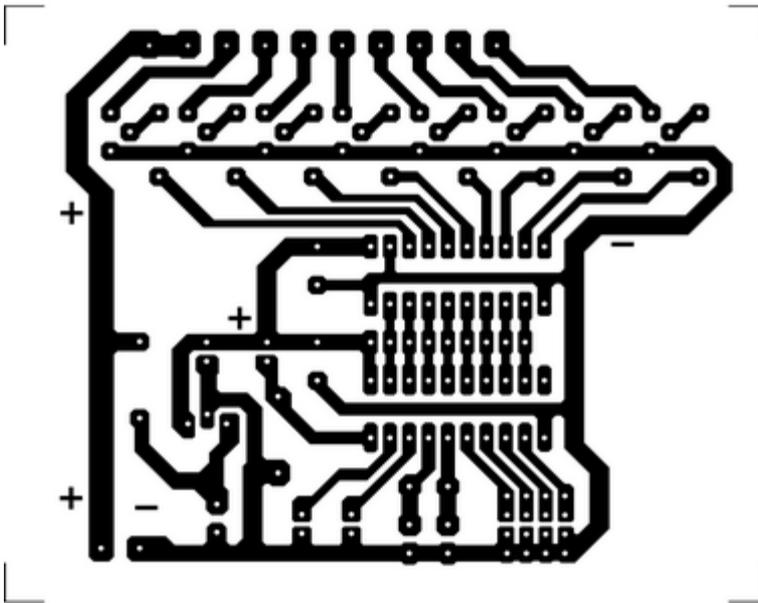
Der Spannungsregler IC1 verfügt zwecks besserer Kühlung über eine Metallseite (Kennzeichnung durch einen dicken Strich im Bestückungsplan).

Der Microcontroller ATMEL 89C4051 und der Teiber-Baustein 74HC541 werden laut Abbildung normal lesbar in die Schaltung eingebaut. **Pin 1** der ICs ist zusätzlich durch eine Kerbe oder einen Punkt gekennzeichnet.

Für die **Stromversorgung** (Plus-Pol) der angeschlossenen Leuchtdioden sind zwei PLUS-Pole herausgeführt. Bei höherer Strombelastung können somit bei Verwendung von Flachbandleitungen zwei Anschlüsse verwendet werden.

Platinenlayout der MC-Steuerung 1 zu 1 - Vorlage für die Fotomethode

Das **Layout der Schaltung** ist so abgebildet, als würde man von **oben** - also von der **Bestückungsseite** - auf die Schaltung blicken.



Das **Layout** kann mit einem guten Kopierer auf eine Kopierfolie, die dann **zum Belichten** verwendet wird, übertragen werden. Auch der Ausdruck mit einem Laserdrucker bringt gleich gute Ergebnisse.

In beiden Fällen ist zu beachten, dass **eine Kopie** oder **ein Ausdruck** normalerweise **nicht ausreicht**, da diese zumeist zu wenig lichtdicht sind.

Üblicherweise genügen jedoch **zwei Folien**, die genau **deckungsgleich** an den Rändern übereinandergeklebt werden.

Die **Belichtungszeit** ist abhängig vom Belichtungsgerät und auch von der Qualität der fotopositiv beschichteten Platine. Es empfiehlt sich, kleine Probestücke zu belichten und zu entwickeln. Gegebenenfalls ist die Belichtungszeit so lange anzupassen, bis reine Kupferflächen entstehen. Zeiten zwischen 2 - 3 Minuten sind üblich. Die Qualität der Platine ist bereits nach dem Entwickeln erkennbar.

Microcontroller - Steuerung mit ATMEL 89C4051 Software

Für die Programmierung des Microcontrollers wurden die Hardware und die Software von der Firma Batronix (www.batronix.de) verwendet, aber es sind auch viele andere Programmiergeräte möglich.

Beim hier vorgestellten Programm handelt es sich um ein Musterprogramm, das **16 voneinander unabhängige Teilprogramme** enthält.

Auf **Seite 5** ist die **Programmstruktur** mit den Schleifen „IF ... ELSE ... END IF“ und der Zeitschleife „Delay01“ dargestellt. Im Microcontroller ATMEL 89C4051 haben im 4 kByte großen Speicher etwa 20 DIN A4 - Seiten Platz. Daher ist hier nur die Struktur, die sehr einfach nachzuvollziehen ist, abgebildet.

Im Folgenden werden die wesentlichen Programmteile erklärt, sodass die Programmierung auf unterschiedlicher Programmiersoftware und Hardware durchgeführt werden kann.

INCLUDE 89C4051.mc

Mit diesem Befehl wird der Assembler-Software mitgeteilt, welcher Microcontroller eingesetzt wird.

MOV P1, #11111111b

MOV P3, #11111111b

Mit diesen Befehlen werden die Ausgänge an Port 1 und Port 3 auf high (H-Pegel) gestellt. **Port 1** sind die **8 Ausgänge** des MC (Pin 12 bis Pin 19), an denen die Konstantstromquellen angeschlossen sind.

Insbesondere wichtig ist es, dass Ausgänge an **Port 3**, an denen die **Schalter S1 - S4** (Pin 6 bis Pin 9) angeschlossen sind, auf high gestellt werden, da sonst die Schalter nicht aktiviert werden können.

Schleifenstruktur

```
IF BIT P3.2 THEN ..... ELSE ..... END IF
```

Mit den Schaltern S1 - S4 können über diese Schleifenstruktur die einzelnen Programme angewählt werden.

BIT P3.2 steht für Schalter S1 an Pin 6.

BIT P3.3 steht für Schalter S2 an Pin 7.

BIT P3.4 steht für Schalter S3 an Pin 8.

BIT P3.5 steht für Schalter S4 an Pin 9.

Auf diese Weise werden die einzelnen Programme über die Schalter im Binärcode aufgerufen. 4 Schalter ergeben somit **16 mögliche Programme**.

MOV P1, #10000000b 8 Bit an Port 1 (A0 - A7)

MOV R0, #5 Parameter für Verweilzeit

LCALL time01 Aufruf der Zeitschleife 1

LCALL time02 Aufruf der Zeitschleife 2

Diese 3 Zeilen sind für die Datenausgabe an A0 bis A7 (8 Bit) und deren Verweilzeit zuständig.

Zeile 1: Die **8 Stellen im Binärcode** bestimmen, welcher Ausgang angesteuert wird (1) und welcher nicht (0). Es dürfen weder 7 noch 9 Stellen sein.

Zeile 2: Hier wird die **Zeiteinheit** - in diesem Fall **0,1s bzw. 1,0s** - an den Parameter der Zeitschleife (R0) übergeben. Es dürfen **nur Werte zwischen 1 und 255** eingesetzt werden.

Zeile 3: Hier werden die **Zeitschleifen time01** bzw. **02** am Ende des Programms aufgerufen. Die Zeitschleife time01 beträgt etwa 0,1 Sekunden, die Schleife time02 etwa 1 Sekunde. Die Zeitschleifen werden mit dem eingegebenen Faktor multipliziert.

ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1	LCALL time01	
MOV P1, #00000000b	'Schalterstellung = 0011	MOV P1, #11000000b	
MOV R0, #20		MOV R0, #5	
LCALL time01		LCALL time01	
MOV P1, #11111110b		END IF	'Schleifenende von Abfrage P3.5
MOV R0, #1		END IF	'Schleifenende von Abfrage P3.4
LCALL time01		ELSE	'Wenn Bit P3.3 = 0 dann ... = Schalter 1
END IF	'Schleifenende von Abfrage P3.5	IF BIT P3.4 THEN	'Wenn Bit P3.4 = 1 dann ... = Schalter 0
END IF	'Schleifenende von Abfrage P3.4	IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0
ELSE	'Wenn Bit P3.3 = 0 dann ... = Schalter 1	MOV P1, #10000000b	'Schalterstellung = 1100
IF BIT P3.4 THEN	'Wenn Bit P3.4 = 1 dann ... = Schalter 0	MOV R0, #4	
IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0	LCALL time01	
MOV P1, #10101010b	'Schalterstellung = 0100	MOV P1, #01000000b	
MOV R0, #5		MOV R0, #4	
LCALL time01		LCALL time01	
MOV P1, #01010100b		ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1
MOV R0, #5		MOV P1, #10000000b	'Schalterstellung = 1101
LCALL time01		MOV R0, #5	
ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1	LCALL time01	
MOV P1, #00000001b	'Schalterstellung = 0101	MOV P1, #11000000b	
MOV R0, #5		MOV R0, #5	
LCALL time01		LCALL time01	
MOV P1, #11111110b		END IF	'Schleifenende von Abfrage P3.5
MOV R0, #5		ELSE	'Wenn Bit P3.4 = 0 dann ... = Schalter 1
LCALL time01		IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0
END IF	'Schleifenende von Abfrage P3.5	MOV P1, #10000000b	'Schalterstellung = 1110
ELSE	'Wenn Bit P3.4 = 0 dann ... = Schalter 1	MOV R0, #6	
IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0	LCALL time01	
MOV P1, #11110000b	'Schalterstellung = 0110	MOV P1, #11000000b	
MOV R0, #6		MOV R0, #6	
LCALL time01		LCALL time01	
MOV P1, #00001110b		ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1
MOV R0, #6		MOV P1, #10000000b	'Schalterstellung = 1111
LCALL time01		MOV R0, #4	
ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1	LCALL time01	
MOV P1, #00000001b	'Schalterstellung = 0111	MOV P1, #11000000b	
MOV R0, #15		MOV R0, #4	
LCALL time01		LCALL time01	
MOV P1, #00000000b		END IF	'Schleifenende von Abfrage P3.5
MOV R0, #7		END IF	'Schleifenende von Abfrage P3.4
LCALL time01		END IF	'Schleifenende von Abfrage P3.3
END IF	'Schleifenende von Abfrage P3.5	END IF	'Schleifenende von Abfrage P3.2
END IF	'Schleifenende von Abfrage P3.4	LJMP Simulator	'Endlosschleife
END IF	'Schleifenende von Abfrage P3.3		
ELSE	'Wenn Bit P3.2 = 0 dann ... = Schalter 1		
IF BIT P3.3 THEN	'Wenn Bit P3.3 = 1 dann ... = Schalter 0	time01:	
IF BIT P3.4 THEN	'Wenn Bit P3.4 = 1 dann ... = Schalter 0	st1: MOV R1, #020	'0,1 Sekunden
IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0	st2: MOV R2, #100	
MOV P1, #00000000b	'Schalterstellung = 1000	st3: DJNZ R2, st3	
MOV R0, #7		DJNZ R1, st2	
LCALL time01		DJNZ R0, st1	
MOV P1, #11111111b		RET	
MOV R0, #7			
LCALL time01		time02:	
ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1	st4: MOV R1, #200	'1,0 Sekunden
MOV P1, #10000000b	'Schalterstellung = 1001	st5: MOV R2, #100	
MOV R0, #4		st6: DJNZ R2, st6	
LCALL time01		DJNZ R1, st5	
MOV P1, #01000000b		DJNZ R0, st4	
MOV R0, #4		RET	
LCALL time01			
END IF	'Schleifenende von Abfrage P3.5		
ELSE	'Wenn Bit P3.4 = 0 dann ... = Schalter 1		
IF BIT P3.5 THEN	'Wenn Bit P3.5 = 1 dann ... = Schalter 0		
MOV P1, #11111111b	'Schalterstellung = 1010		
MOV R0, #1			
LCALL time01			
MOV P1, #00000000b			
MOV R0, #1			
LCALL time01			
ELSE	'Wenn Bit P3.5 = 0 dann ... = Schalter 1		
MOV P1, #10000000b	'Schalterstellung = 1011		
MOV R0, #5			