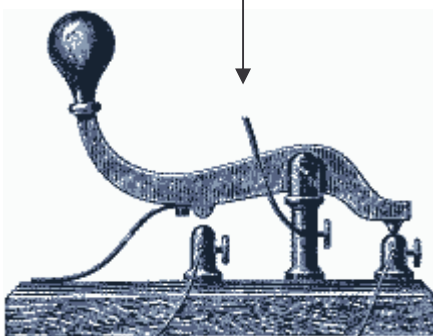
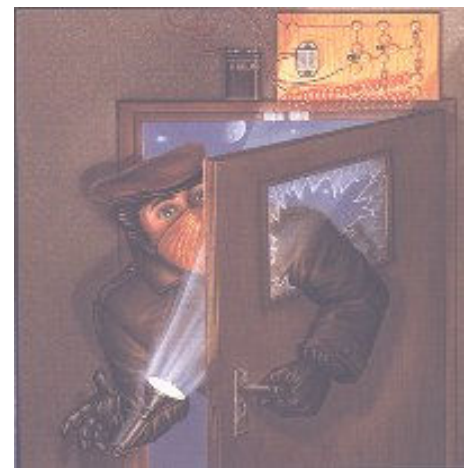
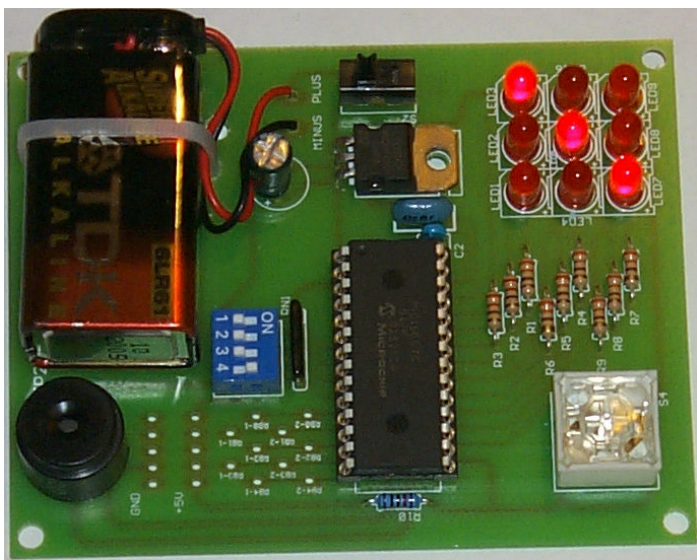


# Deutscher Amateur Radio Club

## Ortsverband Amberg U01



### Ferienprogramm 2004



## Vorwort

Hallo liebe Teilnehmer des Ferienprogramms 2004,

Ich darf mich kurz vorstellen. Mein Name ist...

Ja, wer bin ich eigentlich?

Oberflächlich betrachtet bestehe ich aus einer Platine, auf die eine Menge von elektronischen Bauteilen aufgelötet werden. Das große schwarze Ding ist ein integrierter Schaltkreis (IC), der Prozessor, so zu sagen mein Gehirn. In ihm ist alles gespeichert, was ich für Euch tun soll. Mein Wissen habe ich nicht, wie Ihr, in der Schule erlernt. Mein Programmierer hat all sein Können verwendet, um das Programm zu schreiben. Der gelbe "Knopf" mit den 3 Beinchen, gleich neben dem Prozessor, gibt den Takt vor. Der Resonator arbeitet ähnlich wie die Feder in einer Uhr. Somit weiß der Prozessor, was er zu tun hat, um 8:00 Uhr, um 10:00 Uhr, usw. Frühstück kennt er allerdings nicht. Der Elektronik reicht eine Batterie als Kraftquelle, mehr braucht sie nicht. Die roten Knöpfe, 9 an der Zahl, sind Leuchtdioden. Sie geben Licht ab, wenn Strom durch sie fließt, ähnlich wie Glühlämpchen in der Taschenlampe.

Damit der Prozessor weiß, was er für Euch tun soll, müsst Ihr die 4 Schalter, wie in der Tabelle auf Seite 12 beschrieben, einstellen. Zum Starten bzw. Stoppen drückt Ihr die Taste (S4). Wie Ihr seht, habe ich gewisse Ähnlichkeiten mit einem PC. Mein Programm ist allerdings fest im Speicher "eingebrannt". Der Bildschirm hat nur 3 mal 3 Pixel Auflösung und ist einfarbig rot. Die Tastatur ist auch übersichtlicher. Eine Taste genügt mir. So, jetzt wisst Ihr schon eine Menge über mich. Nach dem Zusammenbau stellt bitte am "Mäuseklavier" (der 4-fach Schalterblock) alle Schalter auf OFF (AUS) und lasst Euch überraschen! Es fehlt nur noch eines: Ich habe noch keinen Namen. Vielleicht fällt Euch etwas Passendes ein. Ich würde mich über einen Vorschlag freuen!

Den Namen könnt Ihr unter [www.amateurfunk-amberg.de](http://www.amateurfunk-amberg.de) in das Gästebuch eintragen.

## Sicherheitshinweise

- Diese elektronische Baugruppe ist nur für den Betrieb mit einer 9V Blockbatterie vorgesehen.
- Diesen Bausatz keinen hohen Temperaturen, starken Vibrationen oder Feuchtigkeit aussetzen.
- Das Gerät ist für den Gebrauch in trockenen und sauberen Räumen bestimmt
- Das Gerät ist von Blumenvasen, Badewannen, Waschtischen und allen Flüssigkeiten fernzuhalten.
- Die Baugruppe ist vor Feuchtigkeit, Spritzwasser und Hitzeeinwirkung zu schützen
- Das Gerät darf nicht in Verbindung mit leicht entflammbaren und brennbaren Flüssigkeiten verwendet werden.
- Die Baugruppe darf nur unter Aufsicht eines fachkundigen Erwachsenen oder eines Fachmanns in Betrieb genommen werden.
- Eine Reparatur des Gerätes darf nur vom Fachmann durchgeführt werden.
- Nach dem Gebrauch ist das Gerät abzuschalten.
- Die Baugruppe darf nicht auf leitfähigen Oberflächen (z.B. Metall) betrieben werden.

## Technische Daten

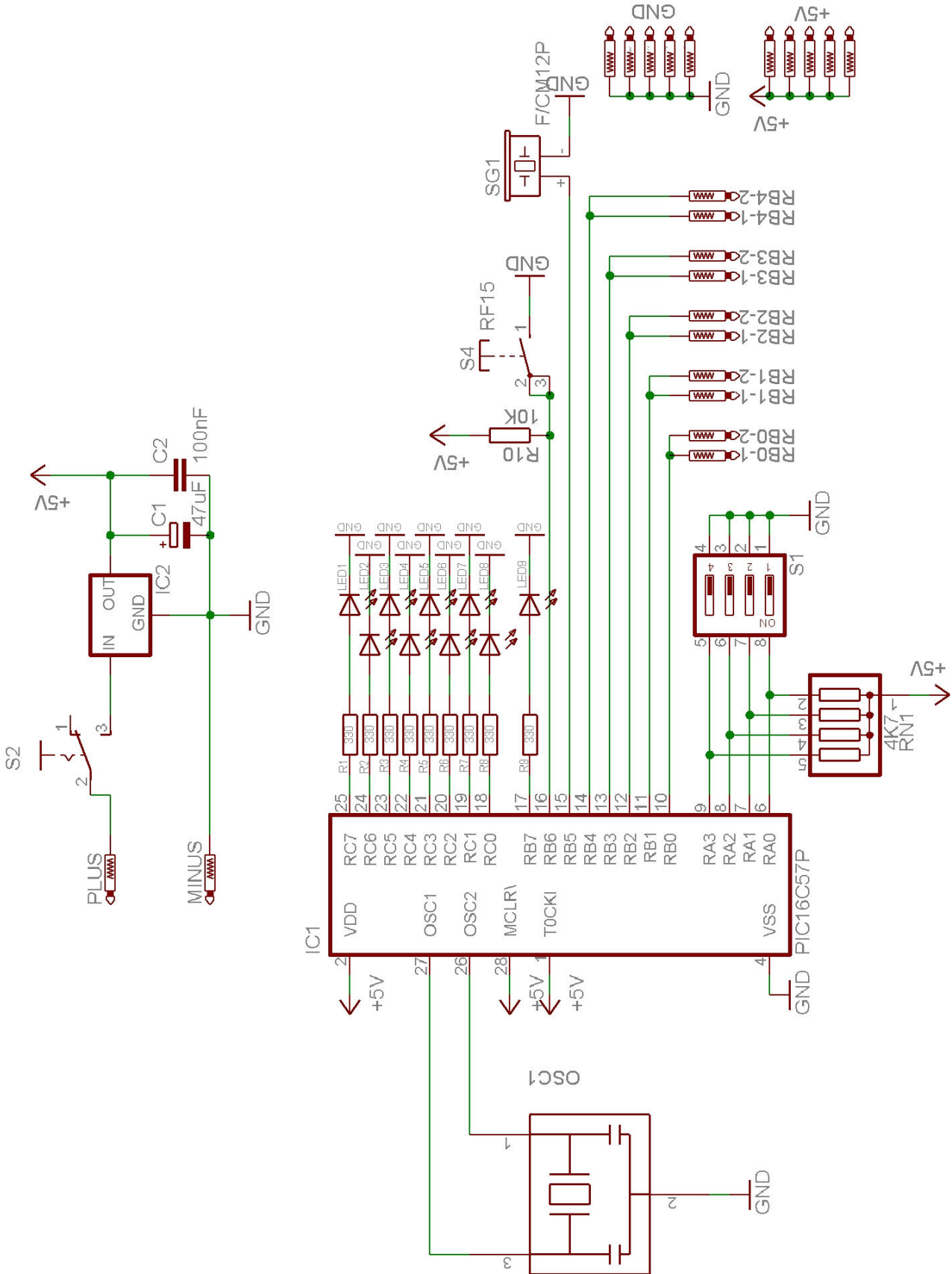
Betriebsspannung	9V
Ruhestrom	30mA
Abmessungen	80 x 100 mm

# Inhaltsverzeichnis

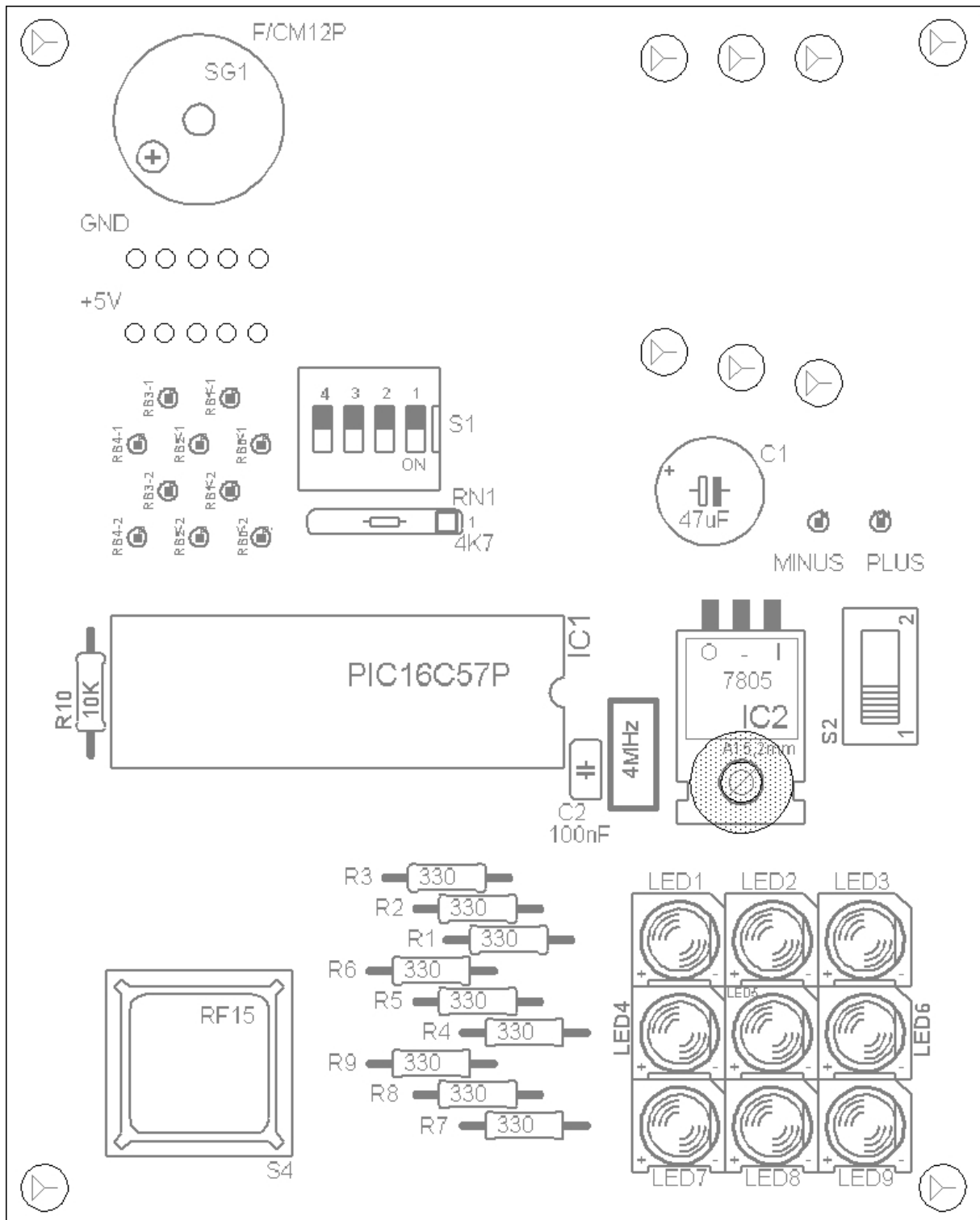
	Seite
Titelseite	1
Vorwort	2
Sicherheitshinweise, Technische Daten	3
Inhaltsverzeichnis	4
Schaltplan	6
Bestückungsplan	5
Platinenlayout	7
Stückliste	8
Funktionsübersicht	9
Weitere Experimente und Funktionen	10
Einfacher Türkontakt	10
Lichtsensor für Wecker-Funktion	11
Grundlagen	12
Hex-Code	12
Leuchtdioden	13
Widerstand	14
Fotowiderstand	15
Kondensatoren	16
Mikrocontroller	17
Das Morsealphabet	19
Quellangaben und interessante Links	20
Quellcode *	21

\*( falls nicht im Druck enthalten gibt es den Quellcode unter [www.amateurfunk-amberg.de](http://www.amateurfunk-amberg.de) )

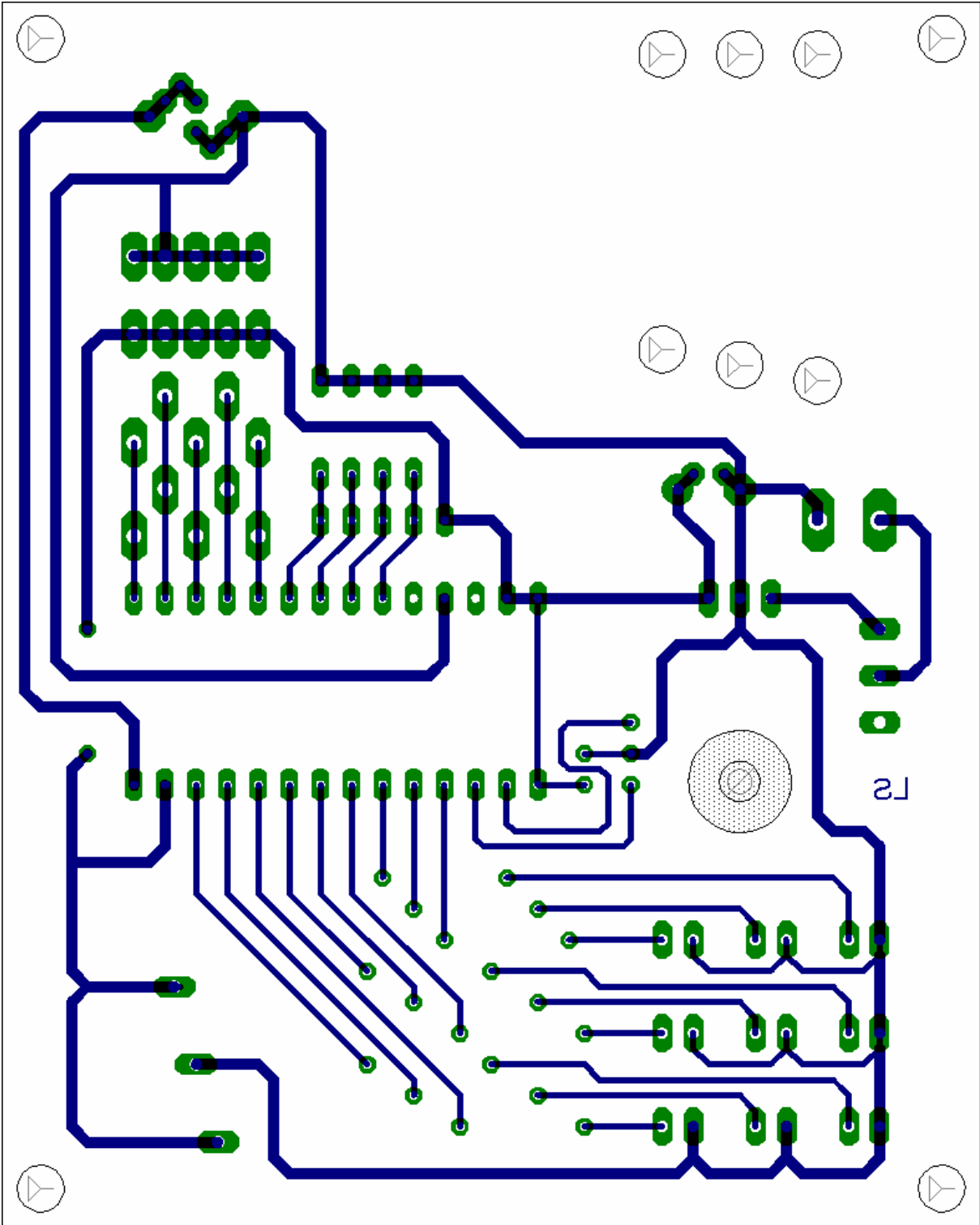
# Schaltplan



# Bestückungsplan



# Platinenlayout



Platinengröße 80mmx100mm

## Stückliste

Widerstand	R1 ... R9	330 Ohm
Widerstand	R10	10 kOhm
Widerstand-Netzwerk	RN1	10 kOhm
Kondensator	C1	47uF/16V
Kondensator	C2	100nF
Leuchtdiode	LED1 ... LED 9	rot 2mA
DIP-Schalter	S1	4-fach
Schalter	S2	1xUM
Kurzhub Taster	S4	Rafi
Summer	SG1	Piezo
Resonator	4MHz	4MHz
Prozessor	IC1	PIC16C57C
IC-Fassung	28pol.	28pol.
Spannungsregler	IC2	7805
Batterieclip	9V	9V
Kabelbinder	2x	



## Funktionsübersicht

Die Auswahl der Programme erfolgt mit dem Schalter S1. Das gewünschte Programm wird im BCD-Code ausgewählt (siehe Grundlagen BCD-Code).

### Programm 0

Beim Programm 0 wird an den Leuchtdioden das Wort HALLO ausgegeben. Mit jedem neuen Buchstaben ist zugleich der Morsecode der Buchstaben zu hören.

### Programm 1

Alarmanlage mit Leuchteffekt und Sirene auf Tastendruck

### Programm 2

Morsen mit der Taste

### Programm 3

Lauflicht Variante 1

### Programm 4

Lauflicht Variante 2

### Programm 5

Lauflicht Variante 3

### Programm 6

Würfel - damit spielt man Würfelspiele mit Turbo-Geschwindigkeit

### Programm 7

Spiel: Fang die Leuchtdiode (mit Tonausgabe)

Es muss die Leuchtdiode in der Mitte gefangen werden. Sobald diese aufleuchtet, sofort die Taste drücken. Leuchtet nun die mittlere LED auf so ertönt das Morsezeichen „S“ wie SUPER. Leuchtet eine andere LED, so ertönt ein „O“ wie Oh / schade.

### Programm 8

Lauflicht Variante 4

### Programm 9

Melodie auf Tastendruck

### Programm 10

Melodie mit LDR (siehe Lichtsensor für die Wecker-Funktion Seite 11)

### Programm 11

Alarmanlage mit Leuchteffekt und Sirene auf externen Schaltkontakt (siehe auch Türkontakt Seite 10)

## Weitere Experimente und Funktionen

Hier sollen nur ein paar Anregungen gezeigt werden, wie die Schaltung erweitert werden kann. Viel Spaß beim Nachbauen. Die zugehörigen Bauteile können bei Reichelt ([www.reichelt.de](http://www.reichelt.de)) oder Conrad Elektronik ([www.conrad.de](http://www.conrad.de)) bestellt werden.

### Einfacher Türkontakt-Schalter für die Alarmanlage (PRG 11)

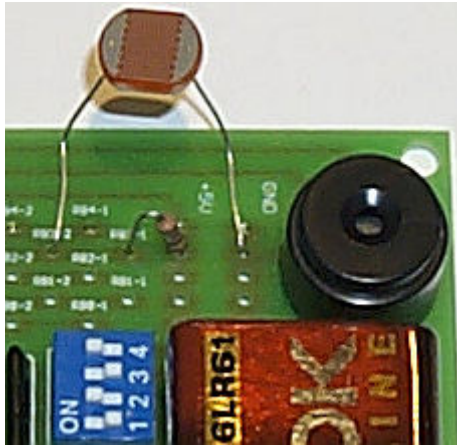
Benötigt werden ein Stück Aluminium-Folie, ein paar Meter dünnes Kabel und ein wenig Klebeband.

Die Drähte (2 Stück) müssen auf die passende Länge abgeschnitten werden, danach werden sie ca. 5 cm abisoliert und in ein Stück Alu-Folie gewickelt (siehe Foto). Diese beiden Streifen Alu-Folie werden dann mit einem Abstand von 2-5 cm und etwas Klebeband an einem Türrahmen befestigt. Ein weiteres Stück Alu-Folie wird nun so an der Tür befestigt, dass es die beiden anderen Alu-Streifen überbrückt, wenn die Türe geschlossen ist. Die beiden Kabelenden müssen dann nur noch, wie unten beschrieben, an der Platine befestigt werden.



Widerstand 10 KOhm von RB4-1 nach GND, Drähte vom Schaltkontakt auf +5V und RB4-2.

## Lichtsensord für die Wecker-Funktion (PRG 10)



Widerstand an RB3-1 und +5V / LDR an RB3-2 und GND

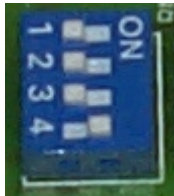
Bei Dunkelheit oder abgedecktem Fotowiderstand ertönt keine Melodie. Fällt Licht auf den Fotowiderstand, so wird eine Melodie abgespielt.

## Grundlagen (Wer mehr wissen möchte...)

### Der Binär-Code

Gebräuchlicher Code in der Datenverarbeitung. Stellt eine Zuordnung der Codezeichen zu den 16 möglichen Dezimalziffern dar (bei 4 Schaltern). Bei jedem Binär-Code sind vier unterschiedliche Codezeichen (Kombinationen) erforderlich, um die Dezimalziffern 0 bis 15 darzustellen.

Der Binär-Code wird über den blauen 4fach-Schalter eingestellt und bestimmt, welches Programm ausgeführt werden soll.



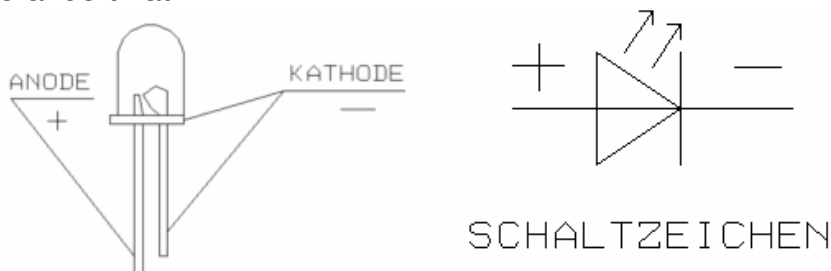
Beispiel für Programm 8 (SW1-3 OFF / SW4 ON)

HEX-CODE				
Programm	SW1	SW2	SW3	SW4
0	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF
2	OFF	ON	OFF	OFF
3	ON	ON	OFF	OFF
4	OFF	OFF	ON	OFF
5	ON	OFF	ON	OFF
6	OFF	ON	ON	OFF
7	ON	ON	ON	OFF
8	OFF	OFF	OFF	ON
9	ON	OFF	OFF	ON
10	OFF	ON	OFF	ON
11	ON	ON	OFF	ON
12	OFF	OFF	ON	ON
13	ON	OFF	ON	ON
14	OFF	ON	ON	ON
15	ON	ON	ON	ON

## Leuchtdioden

Die Glühbirne von Edison führte vor 120 Jahren zu einschneidenden Veränderungen. Ihre Nachfolgerin, die Leuchtdiode, ist gerade dabei, zumindest einen Teil der Beleuchtungswelt zu revolutionieren.

Leuchtdioden oder LED (Light Emitting Diodes) basieren auf Halbleiterverbindungen, die den Strom direkt in Licht umwandeln. Bezogen auf Größe, Effektivität, Haltbarkeit und Lebensdauer verhalten sich die Leuchtdioden zu konventionellen Glühlampen wie Halbleiterdioden zu Röhrendioden. Sie werden die Beleuchtungstechnik in ähnlicher Weise verändern, wie die Halbleitertechnologie schon die Elektronik verändert hat.



### Beispiel

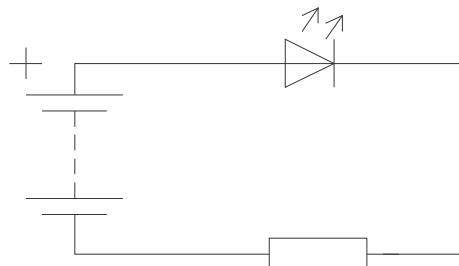
Liefert eine Spannungsquelle 9 Volt und ist der Bedarf der Leuchtdiode 2 Volt, so muss der Vorwiderstand eine Spannung von 7 Volt „vernichten“.

Doch welche Größe muss der Widerstand besitzen?

Das **Ohmsche Gesetz** hilft hier weiter:  $R = U / I$

**Die Rechnung:**

Die Spannung <b>U</b> ist bekannt:	7 Volt
Der Strom <b>I</b> ist bekannt:	0,02 Ampere
Der Widerstand <b>R</b> wird berechnet:	$7 / 0,02 = 350 \text{ Ohm}$



## Widerstand

Widerstand bedeutet umgangssprachlich, dass sich etwas entgegenstellt.

Eine ähnliche Bedeutung hat der Begriff Widerstand in der Elektrizität.

Unterschiedliche Materialien setzen dem Stromfluss unterschiedlichen Widerstand entgegen. Sie leiten den Strom unterschiedlich gut.

**Materialien mit geringem Widerstand, z. B. Metalle nennt man Leiter.**

Materialien mit sehr hohem Widerstand z. B. Porzellan, Gummi, Kunststoffe nennt man Nichtleiter (Isolatoren).

Jedes Gerät und jeder Verbraucher, der sich in einem Stromkreis befindet, stellt dem Strom einen Widerstand entgegen.

**Widerstand wird gemessen in Ohm ( $\Omega$ ). Der Formelbuchstabe ist R.**

Andere Einheiten sind

$k\Omega$  = kilo Ohm = 1000  $\Omega$


$M\Omega$  = Mega Ohm = 1 000 000  $\Omega$

## Widerstandstabelle


	1. Ring	2. Ring	3. Ring (Optional)		Multiplikator	Toleranz
schwarz	0	0	0	x	1	
braun	1	1	1	x	10	1%
rot	2	2	2	x	100	2%
orange	3	3	3	x	1k	
gelb	4	4	4	x	10k	
grün	5	5	5	x	100k	0,5%
blau	6	6	6	x	1M	
violett	7	7	7			
grau	8	8	8			
weiß	9	9	9			
gold				x	0,1	5%
silber				x	0,01	10%

Der Zahlenwert der ersten zwei oder drei Ringe ergibt mit dem Multiplikator den Widerstandswert.  
Beachte: bei zwei Zahlringen ergibt der selbe Multiplikator nur ein Zehntel des Wertes, wie bei drei Zahlringen!


**Beispiele:**



33k 1%



3,3k 5%



100R 5%

## Fotowiderstand

Fotowiderstand(LDR)



Ein Fotowiderstand ist ein Halbleiterwiderstand, der lichtabhängig ist. Er wird auch LDR (Light Dependent Resistor) genannt. Durch das Licht (Photonen) werden im Bauelement Kristallbindungen aufgelöst. Dadurch entstehen freie Ladungsträger, die dazu führen, dass der Widerstandswert sinkt (bei Lichteinfall).

Allerdings haben Fotowiderstände eine gewisse Trägheit, die ihre Einsatzfähigkeit etwas einschränkt

Bei einem LDR steigt die elektrische Leitfähigkeit mit zunehmender Beleuchtung, d. h. je stärker der Lichteinfall ist, desto kleiner ist der Widerstand, desto größer der Strom.

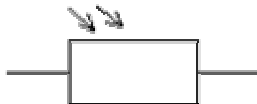
### Aufbau

Bei einem LDR sind zwei Kupferkämme auf einer isolierten Unterlage angebracht. Dazwischen liegt eine dünne Cadmiumsulfidschicht (CdS) in Form eines gewundenen Bandes.

Cadmiumsulfid ist ein Halbleitermaterial, bei dem die elektrische Leitfähigkeit von der einfallenden Lichtmenge abhängt. Je mehr Licht auf das CdS fällt, desto größer ist die Paarbildung (Rekombination zwischen Löchern und Elektronen), desto größer ist auch der elektrische Strom.

Weitere Halbleitermaterialien für Fotowiderstände sind z. B. CdSe, PbS, PbSe, CdTe, ZnO, Se, InSb, InAs, Germanium(Ge) oder Silizium(Si).

### Schaltzeichen



### Anwendung

Der Fotowiderstand befindet sich in Gleich- und Wechselstromkreisen im Einsatz. Er wird als Beleuchtungsstärkemesser, Dämmerungsschalter und als Sensor in Lichtschranken verwendet.

## Kondensatoren

Für Kondensatoren haben sich in den letzten Jahren verschiedene Normen für die Kennzeichnung des Wertes entwickelt, die vielfach zur Verwirrung führen.

Eine Methode, die gerne für Vielschicht-Kondensatoren (das sind die kleinen kissenförmigen Cs, meist in Braun oder Blau anzutreffen) benutzt wird, kennzeichnet die Kondensatoren als Potenz. Als Grundgröße, auch bei sehr hohen Werten, wird Picofarad (pF) benutzt. Der Code besteht aus 3 Ziffern, wobei die letzte Ziffer einfach die Anzahl der Nullen angibt:

100 = 10 und 0 Nullen = 10pF  
= 10 und 1 Null = 100pF  
= 10 und 2 Nullen = 10 00pF = 1nF  
= 10 und 3 Nullen = 10 000pF = 10nF  
= 10 und 4 Nullen = 10 0000pF = 100nF

In einem anderen Verfahren wird, genau wie bei den Widerständen, häufig der Dezimalbezeichner (z.B. p für Pico, n für Nano) als Trennzeichen benutzt:

1p5 = 1,5 pF  
2n2 = 2,2 nF



An Stelle des p für Picofarad findet man oft auch ein J. Das J gibt an, dass es sich um einen Kondensator mit 5% Toleranz handelt. 100J steht auf jeden Fall für 100pF +/- 5% und 150J steht für 150pF +/- 5%.

Weitere Bezeichner für die Toleranz sind:

B ±0,1pF J ±5%  
C ±0,25pF K ±10%  
D ±0,5pF M ±20%  
F ±1pF (wenn > 10pF dann ±1%) S -20...+50%  
G ±2pF (wenn > 10pF dann ±2%) Y 0...+100%  
H ±1,5pF Z -20...+80%



Einige davon sind aber so selten, dass ich sie noch nie gesehen habe. 5% ist eigentlich der üblichste Wert. Diese Bezeichner finden wir hauptsächlich bei Kondensatoren in Scheibenform. Folienkondensatoren haben in der Regel als Grundgröße meistens das Mikrofarad (µF).

0,22µF = 200nF

0,033µF = 33nF

0,0015µF = 1,5 nF



Beispiel Elektrolyt-Kondensator (meist nur kurz Elko genannt)



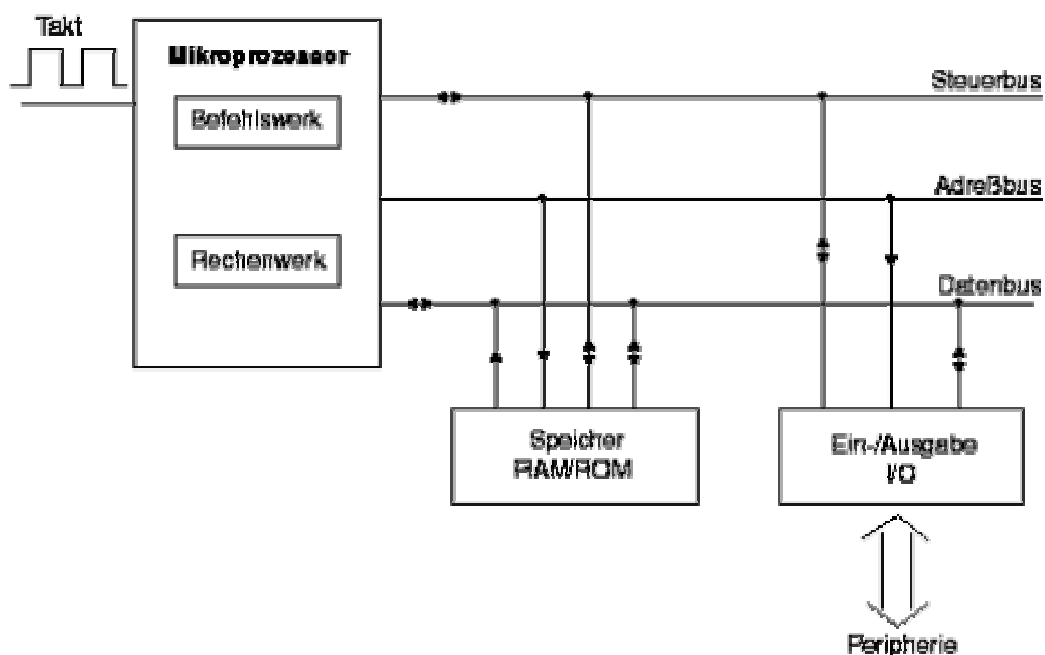
## Mikrocontroller

Der Mikrocontroller PIC 16C57C der Firma Microchip bildet das Herzstück unserer Schaltung. Dieser frei-programmierbare Mikrocontroller kann für viele verschiedene Aufgaben benutzt werden. In unserem Bauprojekt wurden ihm durch die Programmierung tolle Funktionen wie ein Würfel, Lauflicht, Alarmanlage, usw. beigebracht. Mehr Informationen zum Prozessor gibt es unter [www.microchip.com](http://www.microchip.com)

### Wie funktioniert nun ein solcher Mikrocontroller und wie ist er aufgebaut?

Den Grundstein für einen Mikrocontroller bildet der Mikrocomputer. Er besteht aus drei Baugruppen:

- Der **Zentraleinheit**, die unter dem Namen **CPU (Central Processing Unit)** bekannt ist,
- dem **Zentralspeicher**, der in **ROM (Read Only Memory)** und **RAM (Random Access Memory)** unterteilt ist,
- und den **Ein- /Ausgabe-Einheiten**, den sogenannten **Inputs und Outputs (I/O)**.



## Aufbau eines Mikrocontrollers

Verbunden sind diese drei Bausteine durch die drei Bussysteme **Datenbus**, **Adressbus** und **Steuerbus**.

Der **Datenbus** ist 8-Bit breit und transportiert reine Informationsdaten zwischen den Einheiten hin und her.

Der **Adressbus** ist meistens 16-Bit breit, über ihn gelangen die Speicherplatz-adressen zu den Speicher-Bausteinen.

Die Breite des **Steuerbusses** ist je nach CPU verschieden und liegt zwischen vier und zehn Leitungen. Er sorgt dafür, dass die einzelnen Bausteine getrennt voneinander gesteuert werden können.

Die Zentraleinheit setzt sich aus dem **Befehlswerk** und dem **Rechenwerk** zusammen.

Im **Befehlswerk**, der sogenannten **Control Unit (CU)**, befindet sich das Befehlsregister, das alle Befehle enthält, die die Zentraleinheit ausführen kann. Auch die Steuerung der Ein- und Ausgabe-Einheiten und des Steuerbusses gehört zu den Aufgaben des Befehlswerkes.

Das **Rechenwerk**, die **Arithmetic Logic Unit (ALU)**, führt alle logischen und mathematischen Operationen durch. Für diese Berechnungen sind die Operationen Addition und Subtraktion zuständig. Hinzu kommen noch die Operationen Multiplikation und Division, die zwar früher in Co-Prozessoren integriert waren, heutzutage aber in fast allen Zentraleinheiten zu finden sind. Das Rechenwerk bearbeitet zusätzlich die Flags (Ereignisspeicher) sowie unterschiedliche Registersätze, die nach einem bestimmten Zustand einer Funktion oder einer Berechnung gesetzt oder zurückgesetzt werden.

Die komplette Zentraleinheit wird in der Fachsprache als **Mikroprozessor** bezeichnet.

Jeder Mikroprozessor hat einen Eingang für ein rechteckförmiges Taktsignal, das in der Regel von einem Quarz erzeugt wird. Mit diesem Taktsignal werden alle Abläufe der Zentraleinheit und des gesamten Mikrocomputers gesteuert.

Wenn nun alle drei Baugruppen, die Zentraleinheit, Zentralspeicher und einige Peripheriebausteine in einem Gehäuse integriert sind, spricht man von einem **Mikrocontroller**, **Single Chip Computer** oder **Embedded Computer**.

## Das Morsealphabet

Menschen können sich über weite Entfernungen unterhalten, auch ohne Handy, Telefon, Funksprechgerät oder Computer, indem sie das Morsealphabet nutzen. Und das funktioniert so:

Das Morsealphabet wird mit Punkten und Strichen dargestellt. Ein Punkt bedeutet einen kurzen Ton, ein Strich einen langen Ton. Die „Unterhaltung“ kann entweder akustisch (hörbar, z. B. durch Klopfen) oder optisch (sichtbar, z. B. durch Aufleuchten eines Lichtsignals, wie einer Taschenlampe) erfolgen. Möchte man beispielsweise den Buchstaben „L“ akustisch morsen, so muss man einen kurzen Ton, einen langen Ton und wieder zwei kurze Töne hintereinander klopfen, was so ähnlich klingt wie: di-da-di-dit. Optisch wird ein „L“ gemorst, indem man ein Lichtsignal einmal kurz, einmal lang und wieder zweimal kurz hintereinander aufleuchten lässt. Probiert es mal aus!

A	● ———	S	● ● ●
Ä	● ——— ● ———	T	———
B	——— ● ● ●	U	● ● ———
C	——— ● ——— ●	Ü	● ● ——— ———
D	——— ● ● ●	V	● ● ● ———
E	●	W	● ——— ———
F	● ● ——— ●	X	——— ● ● ———
G	——— ——— ●	Y	——— ● ——— ———
H	● ● ● ●	Z	——— ——— ● ●
I	● ●	CH	——— ——— ——— ———
J	● ——— ——— ———	1	● ——— ——— ——— ———
K	——— ● ———	2	● ● ——— ——— ———
L	● ——— ● ●	3	● ● ● ——— ———
M	——— ———	4	● ● ● ● ———
N	——— ●	5	● ● ● ● ●
O	——— ——— ———	6	——— ● ● ● ●
Ö	——— ——— ——— ●	7	——— ——— ● ● ●
P	● ——— ——— ●	8	——— ——— ——— ● ●
Q	——— ——— ● ———	9	——— ——— ——— ——— ●
R	● ——— ●	0	——— ——— ——— ——— ———

## Quellangaben und interessante Links

### Vereine

Amateurfunk Amberg Ortsverband	<a href="http://www.amateurfunk-amberg.de">www.amateurfunk-amberg.de</a>
Deutscher Amateur Radio Club	<a href="http://www.darc.de">www.darc.de</a>
Arbeitskreis Amateurfunk u. Telek. in der Schule	<a href="http://www.aatis.de">www.aatis.de</a>

### Elektronische Bauelemente (Firmen in alphabetischer Reihenfolge)

Conrad Elektronik	<a href="http://www.conrad.de">www.conrad.de</a>
Pollin Electronic	<a href="http://www.pollin.de">www.pollin.de</a>
Reichelt Elektronik	<a href="http://www.reichelt.de">www.reichelt.de</a>
RS Components	<a href="http://www.rs-components.de">www.rs-components.de</a>
Schuricht	<a href="http://www.schuricht.de">www.schuricht.de</a>

## Quellcode

```
#PRAGMA DEVICE PIC16C57
#USE DELAY(CLOCK=4000000)
#FUSES NOWDT

#define LED_8  PIN_C0
#define LED_7  PIN_C1
#define LED_6  PIN_C2
#define LED_5  PIN_C3
#define LED_4  PIN_C4
#define LED_3  PIN_C5
#define LED_2  PIN_C6
#define LED_1  PIN_C7
#define LED_9  PIN_B7

#define SW_1  PIN_A0
#define SW_2  PIN_A1
#define SW_3  PIN_A2
#define SW_4  PIN_A3

#define TAST  PIN_B6
#define SND   PIN_B5

#separate main(void);
#separate LED_OFF(void);
#separate LED_ON(void);
#separate LED_H(void);
#separate LED_A(void);
#separate LED_L(void);
#separate LED_O(void);
#separate read_sw(void);
#separate wuerfel(void);
#separate game(void);
#separate sound_1(void);
#separate sound_2(void);
#separate morsen(void);
#separate play(int a,int b,int c);
#separate kurz_snd(void);
#separate ku(void);
#separate la(void);
#separate pause(void);
#separate pause_l(void);
#separate a(void);
#separate h(void);
#separate l(void);
#separate o(void);
#separate s(void);
#separate ll1(void);
#separate ll2(void);
#separate ll3(void);
#separate ll4(void);

char sw=0;
char help=0;

#BYTE RA=0x05
#BYTE RB=0x06
#BYTE RC=0x07

#separate main(void)
```

```

{
byte lauf=0;
LED_OFF(); OUTPUT_LOW(SND);

read_sw();

while(1)
{
switch(sw)
{
case 0:{          /* Hallo auf LED schreiben mit Morsezeichen */
while(1)
{ morsen(); }
}
case 1:{          /* Sirene auf Tastendruck */
while(1)
{
if(!input(TAST)) LED_ON(); else LED_OFF();
if(!input(TAST))
{
output_high(SND); delay_us(1000-lauf*2);
output_low(SND);  delay_us(1000-lauf*2);
++lauf;
}
restart_wdt(); ++help;
if(help==128)
{output_high(LED_5);delay_ms(10);output_low(LED_5);delay_ms(10);}
}
}
case 2:{          /* Morsen mit Taste */
while(1)
{
if(!input(TAST)) output_high(LED_5); else output_low(LED_5);
if(!input(TAST))
{
output_high(SND); delay_us(700);
output_low(SND);  delay_us(700);
}
restart_wdt();
}
}
case 3:{          /* Lauflicht 1 */
l11();
}
case 4:{          /* Lauflicht 2 */
l12();
}
case 5:{          /* Lauflicht 3 */
l13();
}
case 6:{          /* Würfel */
wuerfel();
}
case 7:{          /* Spiel */
game();
}
case 8:{          /* Lauflicht 4 */
l14();
}
case 9:{          /* Sound 1 */
sound_1();
}
case 10:{         /* Sound 2 */

```

```

        while(1)
        {
            if(!input(PIN_B3)) sound_2();
            restart_wdt();
        }
    }
    case 11:{ /* Alarmanlage mit Sirene über externen Kontakt */
        while(1)
        {
            if(!input(PIN_B4)) LED_ON(); else LED_OFF();
            if(!input(PIN_B4))
            {
                output_high(SND); delay_us(1000-lauf*2);
                output_low(SND); delay_us(1000-lauf*2);
                ++lauf;
            }
            restart_wdt();
            ++help;
            if(help==128)
{output_high(LED_5);delay_ms(10);output_low(LED_5);delay_ms(10);}
            }
        }
    }
}

#separate read_sw(void)
{
    sw=0;
    if(!input(SW_1)) sw|=1;    if (input(SW_1)) sw&=255-1;
    if(!input(SW_2)) sw|=2;    if (input(SW_2)) sw&=255-2;
    if(!input(SW_3)) sw|=4;    if (input(SW_3)) sw&=255-4;
    if(!input(SW_4)) sw|=8;    if (input(SW_4)) sw&=255-8;
}

#separate LED_OFF(void)
{
    OUTPUT_LOW(LED_1);
    OUTPUT_LOW(LED_2);
    OUTPUT_LOW(LED_3);
    OUTPUT_LOW(LED_4);
    OUTPUT_LOW(LED_5);
    OUTPUT_LOW(LED_6);
    OUTPUT_LOW(LED_7);
    OUTPUT_LOW(LED_8);
    OUTPUT_LOW(LED_9);
    restart_wdt();
}

#separate LED_ON(void)
{
    OUTPUT_HIGH(LED_1);
    OUTPUT_HIGH(LED_2);
    OUTPUT_HIGH(LED_3);
    OUTPUT_HIGH(LED_4);
    OUTPUT_HIGH(LED_5);
    OUTPUT_HIGH(LED_6);
    OUTPUT_HIGH(LED_7);
    OUTPUT_HIGH(LED_8);
    OUTPUT_HIGH(LED_9);
    restart_wdt();
}

```

```

#separate LED_H(void)
{
    OUTPUT_HIGH(LED_1);
    OUTPUT_LOW(LED_2);
    OUTPUT_HIGH(LED_3);
    OUTPUT_HIGH(LED_4);
    OUTPUT_HIGH(LED_5);
    OUTPUT_HIGH(LED_6);
    OUTPUT_HIGH(LED_7);
    OUTPUT_LOW(LED_8);
    OUTPUT_HIGH(LED_9);
    restart_wdt();
}

#separate LED_A(void)
{
    OUTPUT_HIGH(LED_1);
    OUTPUT_HIGH(LED_2);
    OUTPUT_HIGH(LED_3);
    OUTPUT_HIGH(LED_4);
    OUTPUT_HIGH(LED_5);
    OUTPUT_HIGH(LED_6);
    OUTPUT_HIGH(LED_7);
    OUTPUT_LOW(LED_8);
    OUTPUT_HIGH(LED_9);
    restart_wdt();
}

#separate LED_L(void)
{
    OUTPUT_HIGH(LED_1);
    OUTPUT_LOW(LED_2);
    OUTPUT_LOW(LED_3);
    OUTPUT_HIGH(LED_4);
    OUTPUT_LOW(LED_5);
    OUTPUT_LOW(LED_6);
    OUTPUT_HIGH(LED_7);
    OUTPUT_HIGH(LED_8);
    OUTPUT_LOW(LED_9);
    restart_wdt();
}

#separate LED_O(void)
{
    OUTPUT_HIGH(LED_1);
    OUTPUT_HIGH(LED_2);
    OUTPUT_HIGH(LED_3);
    OUTPUT_HIGH(LED_4);
    OUTPUT_LOW(LED_5);
    OUTPUT_HIGH(LED_6);
    OUTPUT_HIGH(LED_7);
    OUTPUT_HIGH(LED_8);
    OUTPUT_HIGH(LED_9);
    restart_wdt();
}

#separate wuerfel(void)
{
    char lauf=0;
    char helptwo=0;
    char help_cnt=0;
    char lauf_cnt=0;
}

```



```

LED_O();

while(1)
{
  restart_wdt();
  while(!INPUT(TAST))
  {
    output_low(LED_9); restart_wdt(); delay_ms(1);
    if(++help_cnt==40) help_cnt=0;
    if(++lauf_cnt==1) lauf_cnt=0;
    if (lauf_cnt==0) ++lauf;
    if(help_cnt==0) ++helptwo;
    if(lauf>6) lauf=1;

    switch(helptwo)
    {
      case(0):{ RC= 64; break; }
      case(1):{ RC= 4; break; }
      case(2):{ RC= 1; break; }
      case(3):{ RC= 16; break; }
      case(4):{ RC= 64; break; }
      case(5):{ RC= 4; break; }
      case(6):{ RC= 1; break; }
      case(7):{ RC= 16; break; }
      case(8):{ helptwo=0; break; }
    }
  }

  switch(lauf)
  {
    case(1):{ RC= 8; break; }
    case(2):{ RC= 34; break; }
    case(3):{ RC= 42; break; }
    case(4):{ RC=162; output_high(LED_9); break; }
    case(5):{ RC=170; output_high(LED_9); break; }
    case(6):{ RC=182; output_high(LED_9); break; }
  }
}

#SEPARATE kurz_snd(void)
{
  char x=0;

  while(x<150)
  {
    restart_wdt();
    OUTPUT_HIGH(SND); delay_us(350);
    OUTPUT_LOW(SND); delay_us(350);
    ++x;
  }
}

#separate ku(void)
{
  kurz_snd(); delay_ms(100); restart_wdt();
}

#separate la(void)
{
  kurz_snd(); kurz_snd(); kurz_snd(); delay_ms(100); restart_wdt();
}

```

```

#separate pause(void)
{
    delay_ms(100);restart_wdt();
}

#separate pause_l(void)
{
    delay_ms(400);
}

#separate a(void){ ku(); la(); pause_l();}
#separate h(void){ ku(); ku(); ku(); ku(); pause_l();}
#separate l(void){ ku(); la(); ku(); ku(); pause_l();}
#separate o(void){ la(); la(); la(); pause_l();}
#separate s(void){ ku(); ku(); ku(); pause_l();}

#separate ll1(void)
{
    while(1)
    {
        LED_0(); delay_ms(300);LED_OFF();
        output_high(LED_5); delay_ms(300);
        restart_wdt();
    }
}

#separate ll2(void)
{
    while(1)
    {
        output_high(LED_1); delay_ms(100);
        output_high(LED_2); delay_ms(100);
        output_high(LED_3); delay_ms(100);
        output_high(LED_4); delay_ms(100);
        output_high(LED_5); delay_ms(100);
        output_high(LED_6); delay_ms(100);
        output_high(LED_7); delay_ms(100);
        output_high(LED_8); delay_ms(100);
        output_high(LED_9); delay_ms(100);
        output_low(LED_9); delay_ms(100);
        output_low(LED_8); delay_ms(100);
        output_low(LED_7); delay_ms(100);
        output_low(LED_6); delay_ms(100);
        output_low(LED_5); delay_ms(100);
        output_low(LED_4); delay_ms(100);
        output_low(LED_3); delay_ms(100);
        output_low(LED_2); delay_ms(100);
        output_low(LED_1); delay_ms(100);
        restart_wdt();
    }
}

#separate ll3(void)
{
    while(1)
    {
        output_high(LED_1); delay_ms(50); LED_OFF();
        output_high(LED_2); delay_ms(50); LED_OFF();
        output_high(LED_3); delay_ms(50); LED_OFF();
        output_high(LED_6); delay_ms(50); LED_OFF();
        output_high(LED_9); delay_ms(50); LED_OFF();
        output_high(LED_8); delay_ms(50); LED_OFF();
        output_high(LED_7); delay_ms(50); LED_OFF();
    }
}

```

```

    output_high(LED_4); delay_ms(50); LED_OFF();
    restart_wdt();
}
}

#separate ll4(void)
{
while(1)
{
    RC=8; delay_ms(80);
    RC=73; delay_ms(80);
    RC=42; delay_ms(80);
    RC=28; delay_ms(80);
    RC=136; output_high(LED_9); delay_ms(80);
    LED_OFF();
    restart_wdt();
}
}

#separate game(void)
{
char helptwo=0;
char lauf=0;
char lauf1=0;
while(1)
{
    restart_wdt();
    ++lauf;
    if(lauf>14) lauf=1;

    output_LOW(LED_9);
    switch(lauf)
    {
        case(1):{ RC=128; break; }
        case(2):{ RC= 64; break; }
        case(3):{ RC= 32; break; }
        case(4):{ RC= 16; break; }
        case(5):{ RC=  8; break; }
        case(6):{ RC=  4; break; }
        case(7):{ RC=  2; break; }
        case(8):{ RC=  1; break; }
        case(9):{ RC=  0; output_high(LED_9); break; }
        case(10):{ RC=  4; break; }
        case(11):{ RC= 32; break; }
        case(12):{ RC=  8; break; }
        case(13):{ RC=  2; break; }
        case(14):{ RC= 16; break; }
    }

    for(lauf1=0;lauf1<=200;++lauf1)
    {
        restart_wdt(); delay_us(500);
        if(!INPUT(TAST))
        {
            if(lauf==5 || lauf==12)
            { ku();ku();ku(); }
            else
            { la();la();la(); }
            while(!INPUT(TAST))
            { restart_wdt(); }
        }
    }
}
}

```

```

}

#separate sound_1(void)
{
  unsigned int x=0;
  while(1)
  {
    play(19,51,100);
    play(15,17,100);
    play(12,76,100);
    play(9,1,250);
    play(7,90,250);
    play(9,75,150);
    play(11,20,70);
    play(12,76,100);
    play(11,20,200);
    play(10,1,70);
    play(11,20,100);
    play(12,76,250);
    delay_ms(1000);
    restart_wdt();
  }
}

#separate sound_2(void)
{
  unsigned int x=0;
  play(19,51,100);
  play(15,17,100);
  play(12,76,100);
  play(9,1,250);
  play(7,90,250);
  play(9,75,150);
  play(11,20,70);
  play(12,76,100);
  play(11,20,200);
  play(10,1,70);
  play(11,20,100);
  play(12,76,250);
  delay_ms(1000);
  restart_wdt();
}

#separate play(int a,int b,int c)
{
  unsigned int x=0,y=0;
  ++RC;
  if(RC>128) OUTPUT_LOW(LED_9); else OUTPUT_HIGH(LED_9);
  while(y<c)
  { OUTPUT_HIGH(SND);
    for(x=1;x<a;++x) delay_us(99);
    delay_us(b);
    OUTPUT_LOW(SND);
    for(x=1;x<a;++x) delay_us(99);
    delay_us(b);
    ++y;
    restart_wdt();
  }
  delay_ms(5);
  ++RC;
  if(RC>128) OUTPUT_LOW(LED_9); else OUTPUT_HIGH(LED_9);
}

```

```
#separate morsen(void)
{
  LED_H(); ku(); ku(); ku(); ku(); pause_1(); LED_OFF(); delay_ms(100);
  LED_A(); ku(); la(); pause_1(); LED_OFF(); delay_ms(100);
  LED_L(); ku(); la(); ku(); ku(); pause_1(); LED_OFF(); delay_ms(100);
  LED_L(); ku(); la(); ku(); ku(); pause_1(); LED_OFF(); delay_ms(100);
  LED_O(); la(); la(); la(); pause_1(); LED_OFF(); delay_ms(100);
  delay_ms(2000);
}
```