

PULSDETEKTOR BT1e



(Revision 8, vollständig überarbeitet Ausgabe)

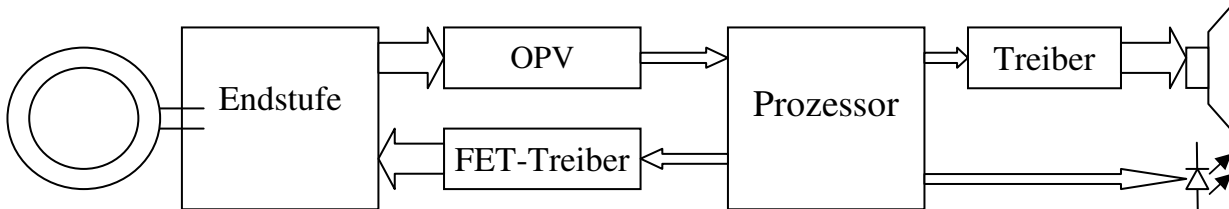
BT1e

Der Pulsdetektor BT1e ist eine vollständig überarbeitete Version des BT1 bei dem alle bisherigen Verbesserungen eingearbeitet wurden. Die wesentlichsten Verbesserungen betreffen die Treiberstufe für den Leistungstransistor und die Integration einer ISP - Schnittstelle.

BT1 ist aus zwei Gründen entstanden. Zum einen aus der intensiveren Beschäftigung mit den Möglichkeiten der digitalen Signalverarbeitung mit AVR Prozessoren und zum anderen aus der Idee einen Pulsdetektor in einem robusten Rohrkörper mit handelsüblichem Durchmesser (z.B.: 40mm Abwasserrohr aus dem Baumarkt) unterzubringen. Der Einbau in ein Rohr ist im Übrigen für den Aufbau eines Unterwasserdetektors sehr zweckmäßig ;-). Die vorliegende Schaltung war ursprünglich nur für Experimentierzwecke gedacht gefiel mir aber so gut, dass ich daraus einen kleinen Detektor gebaut habe. Geplant ist der Einsatz des BT1e als Handdetektor um den Aushub und das Grabungsloch zu durchsuchen, da für diese Aufgabe ein großer Detektor zu unhandlich und eine große Spule für die Positionierung des Fundstückes zu ungenau ist. Für diese Aufgabe hatte ich auch ansatzweise eine Metallunterscheidung (LED) in den BT1 eingebaut. Dies wurde zugunsten der Empfindlichkeit beim BT1e wieder aufgegeben. Die LED ist hier nur eine optische Anzeige parallel zum Speaker. Vom BT1 gibt es inzwischen auch eine Unterwasserversion den BT1uw. Dieser unterscheidet sich eigentlich nur durch die druckwasserfeste Bauform des Gehäuses.

Der Bauteilaufwand und die Dimensionierung des BT1 sollten möglichst gering ausfallen. Das wurde erreicht indem die Aufgabe des Pulsgenerators, Tongenerators und Differenzverstärkers durch den Prozessor übernommen wird. Als Prozessor wurde der ATtiny12L gewählt. Dieser verfügt über den nötigen Comperator, eine kleine Bauform (8Pin DIP) sowie einen internen Taktgeber. Damit fällt auch der Schaltungsaufwand für den Prozessortakt weg. Im Übrigen sollte alles auch so preiswert wie möglich ausfallen ☺ und bei einem Lieferanten beschaffbar sein (deshalb auch ein IRF9640 da der MPT2P50 nicht so gut verfügbar ist).

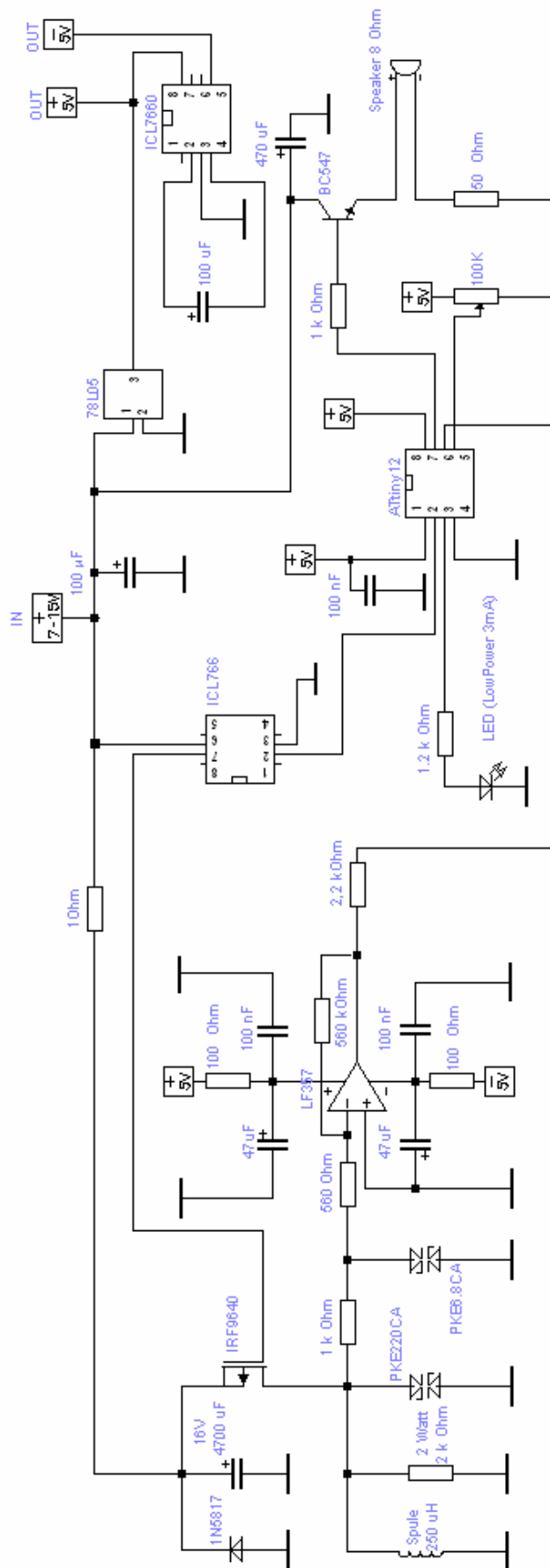
An dieser Stelle möchte ich ausdrücklich auf www.pulsdetektor.de verweisen, dort lauert der ansteckende Pulsdetektorvirus. Und ich muss Silentwater und Chris danken. Ohne die beiden hätte ich als blutiger Amateur das Handtuch werfen müssen. Wenn in diesem Dokument Fehler enthalten sind (was recht wahrscheinlich ist) bitte diese im Forum von www.pulsdetektor.de diskutieren.



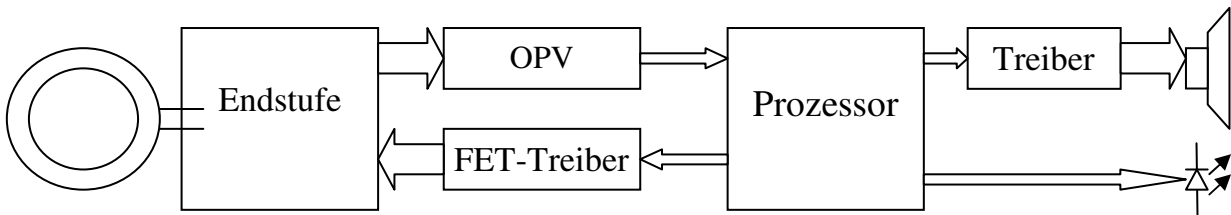
Technische Daten

Betriebsspannung :	7 – 15V / DC
Betriebsstrom :	60 - 120 mA (Software/Spannungsabhängig)
Frequenz :	300-500 Hz (Softwareabhängig)
Pulsdauer :	10 – 200 μ S (Softwareeinstellung)
Akku / Batterie :	6-12 Nc/Ni-MH, 9Volt Block, 5-10 Alcaline
Reichweite:	bis 100cm (Objekt- und Spulenabhängig)

Schaltung

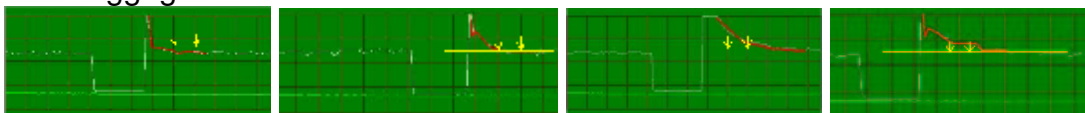


Funktionsweise



Im Pulsdetektor BT1e wird die gesamte Steuerung von einem ATtiny12L Mikrocontroller der Firma Atmel übernommen. Das gesamte Timing kann im Programm eingestellt werden (PulsDauer, HalbZyklus, Delay). Der Prozessor generiert in einem Abstand von ca. 1,7 – 3,5 mS ($2 \times \text{HalbZyklus} + \text{PulsDauer} + \text{Delay}$) ein Pulssignal von 10 bis 200 μS Breite ($\text{PulsDauer} = 1..20$). Das Pulssignal liegt an PB3 (PIN2 ATtiny12) an und wird über einen FET-Treiber (ICL7667) an den Leistungstransistor (IRF9640 / MPT2P50) gegeben. Die Treiberstufe ist notwendig, da die Ausgänge des ATtiny12L mit max. 5V 3mA HI / 20 mA LO belastet werden können und damit den MOSFET nicht direkt treiben können. Der Leistungstransistor schaltet den Primärimpuls der aus einem 4700 bis 10.000 μF Elko gespeist wird. Die Spule erzeugt damit einen elektromagnetischen Impuls. Das nach dem Abschalten des Impulses auftretende Sekundärsignal (bei Metall in Spulennähe verändert sich der Verlauf der abfallenden Flanke) wird vom Operationsverstärker (LF357 / CA3130 / LM318) verstärkt und an den Comperatoreingang PB1 (PIN6 ATtiny12L) des Mikrocontroller gegeben. Der Comperator arbeitet kontinuierlich und vergleicht die Spannung zwischen PB0 und PB1 (PIN5/6 ATtiny12L). Die Referenzspannung wird über den Poti mit Schalter (Spannungsteiler über Masse und + 5 V, Abgreifpunkt an PIN5) an PB0 geführt. Die softwareseitige Auswertung des Comperators wird nach einer Pause (Delay) durchgeführt. Liegt zu diesem Austastzeitpunkt (optimal bei 15 bis 20 μs nach Impulsende) das Signal an PB1 über der Referenzspannung an PB0, wird ein Impuls an den Speaker (PB2) gegeben. Der Speaker wird über einen Transistor (BC547) getrieben. Vergleiche dazu die Bilder vom Oszi. In der ursprünglichen Version des BT1 wurden weitere Samplepunkte zur Ermittlung der Breite des Sekundärsignals abgefragt. Damit konnte man kann also über die LED und den einen geänderten Ton auf Größe beziehungsweise Metallart schließen. Da der Signalpegel aber trotz Dämpfung in der Regel im weiteren Verlauf wieder etwas ansteigt führen mehrere Samplepunkte beim BT1 effektiv zu einer geringeren Empfindlichkeit. Hier gibt es aber bestimmt noch Lösungswege um diese Funktion bei maximaler Empfindlichkeit einzubauen.

Signalverlauf ohne Metall im Suchbereich und bei einem Euro im Suchbereich. Gelb die Referenzspannung vom Poti und die beiden Austastzeitpunkte des Comperators. Rot das vom Operationsverstärker aufbereitete Signal. Signalverlauf bei einem kleinen Schraubendreher (Eisen) und bei einem größeren Messinggegenstand.



Augenscheinlich ist die Verkürzung des SampleDelay ein Weg zu höherer Empfindlichkeit aber auch eine Verschiebung in den kritischen, steilen Bereich der fallenden Kurve. Das optimale Timing kann über die Software eingestellt werden und liegt bei 15 bis 20 μs . Dazu sollte aber ein Oszi benutzt werden. Das Timing ist in jedem Fall an die konkrete Spule (Induktivität) und die Versorgungsspannung (7-15V)

anzupassen. Das brutale Erhöhen der Versorgungsspannung auf 15V bringt weniger als zu vermuten ist. Bei einem 9V Block erreicht man auch schon recht beachtliche Ergebnisse bei weniger Platzbedarf und Gewicht für die Batterie.

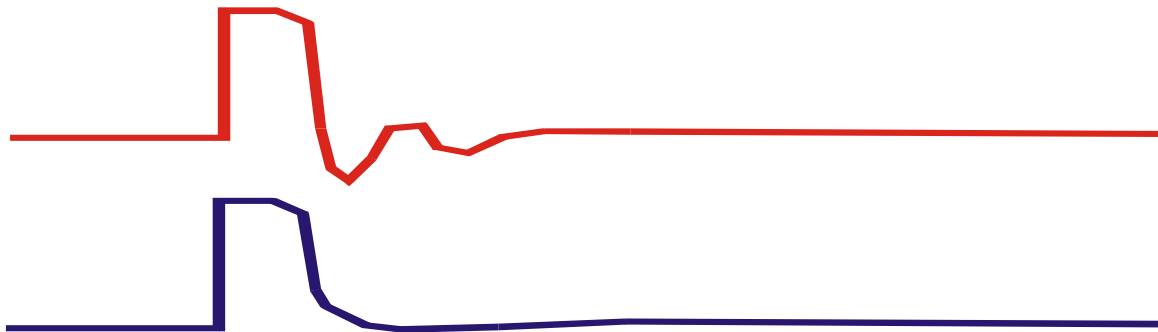
Die Halbzykluszeit der Hauptschleife kann durchaus verändert werden. Es ist dabei jedoch zu beachten, dass die Veränderung der Durchlaufzeit der Hauptschleife eine Tonveränderung bedeutet. Für die Treiberstufe (BC547 max. 100 mA Last) des Speakers (8 Ohm) kann bei Verwendung eines Kopfhörers (32 Ohm) der 50 Ohm Widerstand auf 22 Ohm reduziert werden. Die Versorgungsspannung darf nicht unter 7V liegen da sonst der Spannungsregler 78L05 keine saubere 5V Spannung mehr liefert. Besondere schaltungstechnische Maßnahmen um einen störungsfreien Power On Reset zu gewährleisten sind nicht nötig, da der ATtiny12 über einen verzögerten Power On Reset Zyklus verfügt.

Handhabung des BT1e

Bei Inbetriebnahme darf kein Metall im Suchbereich der Spule sein. Am Potentiometer mit Schalter einschalten und den Potentiometer voll aufdrehen. Danach langsam zurückregeln bis LED und Speaker aus sind (hier liegt die Kunst den empfindlichsten Bereich zu treffen). Betriebsbereit!

Kalibrierung mit Oszi (empfohlen)

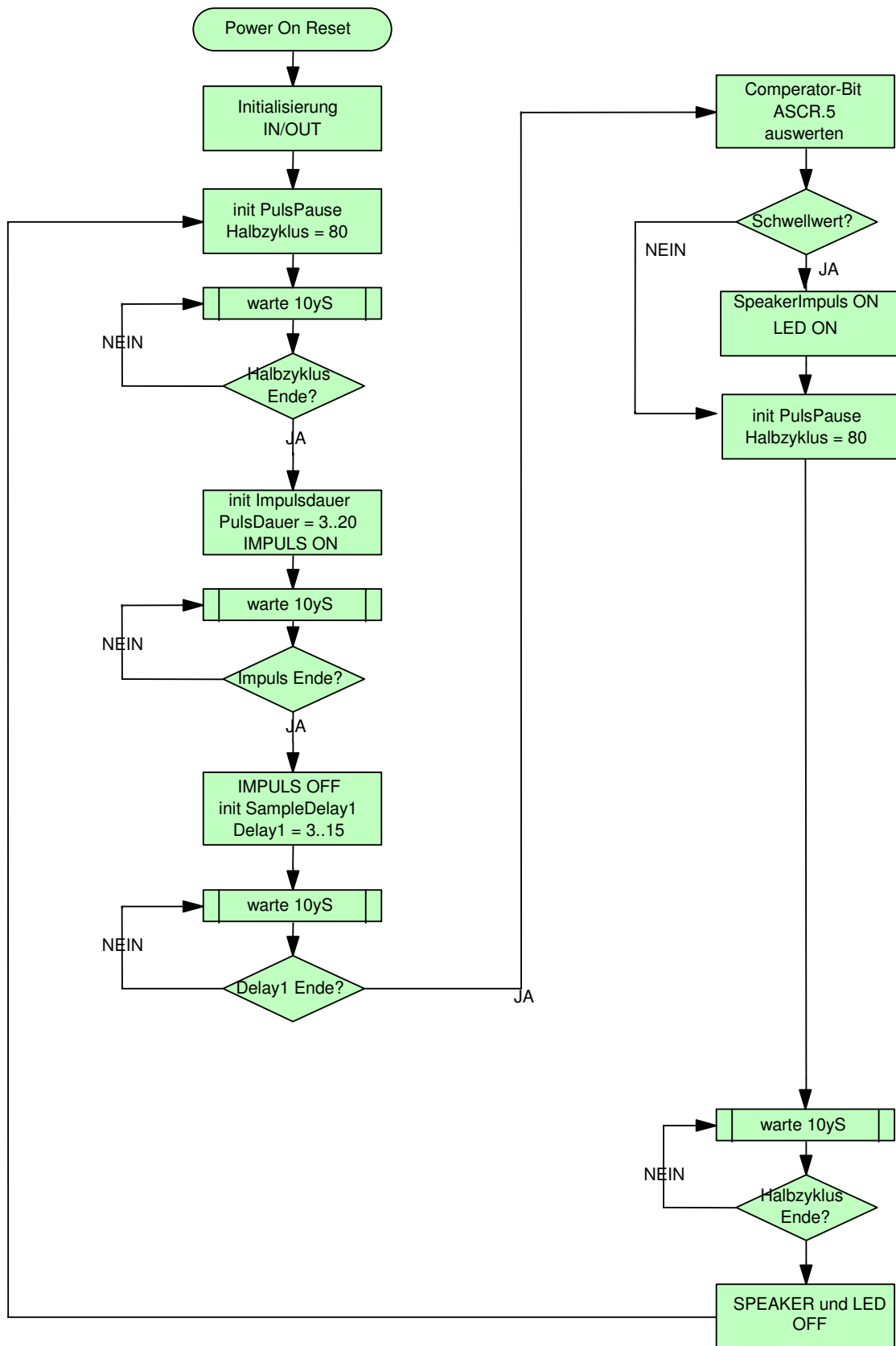
Dazu sollte die vorgesehene Spule und die geplante Spannungsversorgung eingebaut werden. Es darf kein Metall im Suchbereich der Spule sein. Als Dämpfungswiderstand ist ein Trimpoti 2K2 einzubauen. Mit dem Oszi das Signal an PIN6 des OPV zu überwachen. Den Dämpfungswiderstand langsam reduzieren bis keine Über- oder Unterschwinger (rot) mehr zu sehen sind. Danach den Trimpoti nicht mehr verändern und den Dämpfungswiderstand messen. Den Trimpoti gegen einen Festwertwiderstand 2Watt nahe dem gemessenen Wert ersetzen.



Kalibrierung ohne Oszi

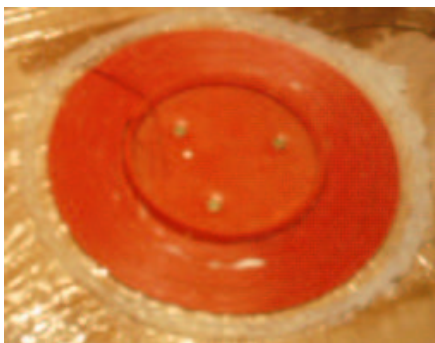
Dazu sollte die vorgesehene Spule und die geplante Spannungsversorgung eingebaut werden. Es darf kein Metall im Suchbereich der Spule sein. Als Dämpfungswiderstand ist ein Trimpoti 2K2 einzubauen. Der Drehpoti mit Schalter ist einzuschalten und in Mittelstellung zu bringen. Den Dämpfungswiderstand langsam reduzieren bis der Ton im Speaker gerade aus geht. Danach den Trimpoti nicht mehr verändern und den Dämpfungswiderstand messen. Den Trimpoti gegen einen Festwertwiderstand 2Watt nahe dem gemessenen Wert ersetzen.

SiSy-Programmablaufplan



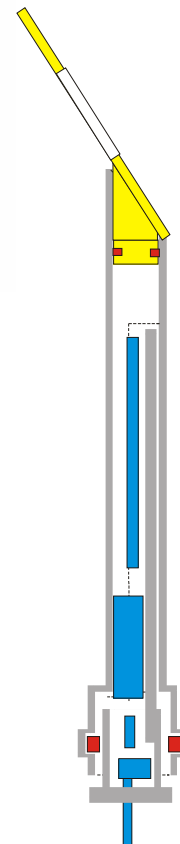
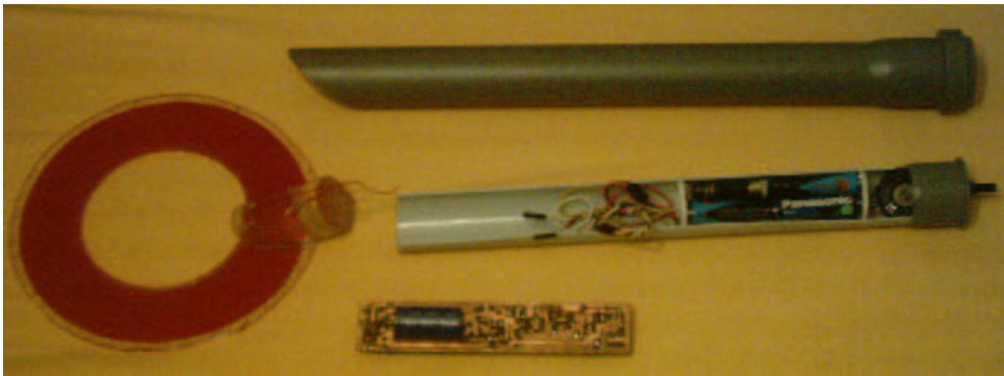
Spule (Planarspule)

- Unterlage mit Folie abdecken
- Eine Form für Innendurchmesser (100mm x 5mm hoch) fixieren (hier verschraubt).
- Spalt zwischen Unterlage Kernform abdichten (z.B. Silikon).
- Ca. 5cm um den Ring herum mit zähem Kleber gleichmäßig präparieren. Zum Beispiel mit UHU-Kraftkleber einpinseln und 3-5 Minuten zäh werden lassen.
- Spulendraht von innen nach außen unter leichtem Druck, spiralförmig und eng anliegend 24 Windungen um den Kern wickeln (der zähe Kleber hält die Litze in Position).
- In 5mm Abstand um die Spule herum eine 5 mm hohe Wulst aus Silikon ziehen.
- Spule mit Epoxidharz ausgießen.
- In einem Stück vom Rohrkörper ein ca. 7-8 cm langes Anschlussstück aus Epoxidharz gießen. Damit kann die Spule passgenau ins Rohr gesteckt werden und dieses verschließen.
- Nach dem Aushärten beide Teile mit 2 Komponenten Kleber verkleben.



Gehäuse

- 0,5 m Abwasserrohr mit 40mm Durchmesser abgeschrägt zuschneiden
- Elektroinstallationsrohr 40mm Durchmesser als Träger längs halbieren
- Träger in Verschlussklappe verkleben
- Batteriehalterung, Lautsprecher (max 35mm), LED und Poti mit Schalter verkabeln und einbauen (siehe Bild)
- Leiterplatte verbinden und im Träger fixieren
- bestückten Träger ins Rohr einführen
- Spule anschließen und mit der Spule die andere Seite vom Rohr verschließen
- Demontage in umgekehrter Reihenfolge ;-)



Stückliste

Widerstände:

1 1R0
1 50
2 100
1 220
1 560
2 1K
1 1K2
1 2K 2Watt
1 560K, bis 1M
1 100K Poti mit Schalter

Kondensatoren:

5 ELKO 100 μ F 16V
1 ELKO 470 μ F 16V
1 ELKO 4700-10000 μ F 16V
3 KERKO 100nF

Transistoren:

1 BC547

Dioden

1 1N5817
1 PKE220CA
1 PKE6,8CA
1 LED (max. 3 mA da diese direkt vom Prozessorport getrieben wird)

IC's

1 ATTiny12L
1 IRF9640, alternativ MPT2P50
1 LF357, alternativ CA3130, LM318 beide in der Schaltung getestet
1 78L05
1 ICL7660
1 ICL7667

Sonstiges

Leiterplatte
Speaker 28mm – 35mm Durchmesser 8Ohm

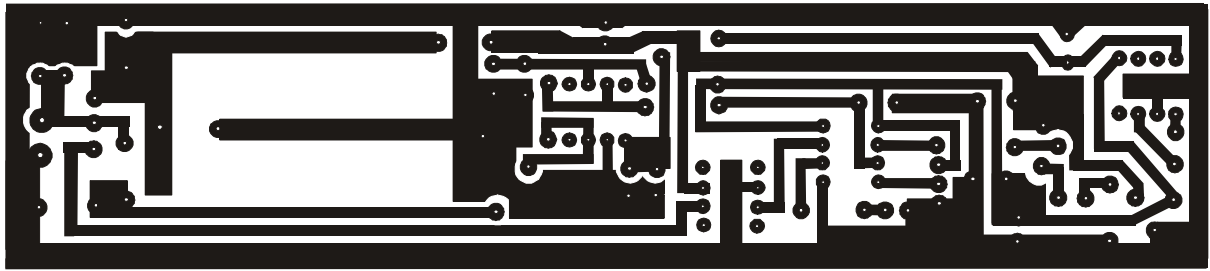
Spule

25m Schalllitze 0,2mm² PVC isoliert
Epoxydharz
IC-Sockel (min. 1 für den ATTiny12)
Konnektoren

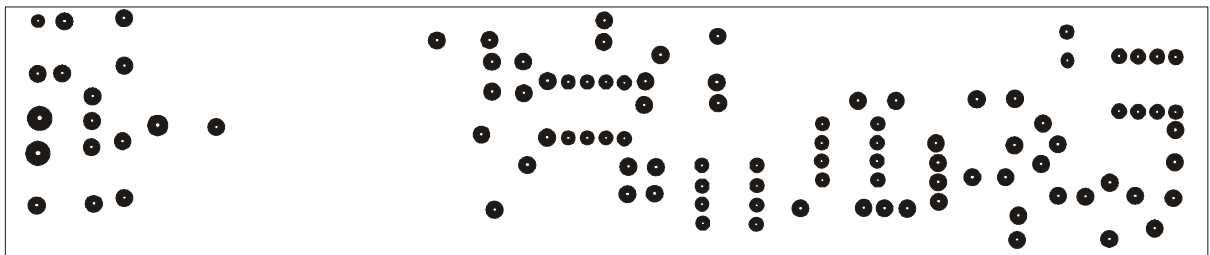
Rohrkörper

0,5 m KG Rohr
Verschluss

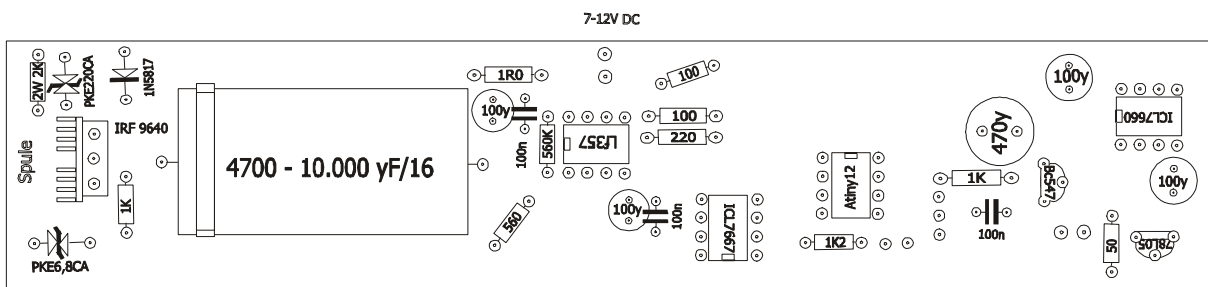
Layout



Bohrplan



Bestückungsplan



Leiterplatte



Quellcode

```

;-----
;* Titel           :Steuerung für Pulsdetektor BT1
;* Prozessor      :AT-Tiny12 1,2 MHz intern
;* Sprache        :Assembler
;* Datum          :02.03.2004
;* Version        :2.3
;* Autor          :böses Tier
;* Hardware       :für BT1 Layot-Version 1.3 mit FET-Treiber
;*               :PB0 an R220->LF357/6, PB1 an Poti,
;*               :PB2 an R1K->BC547->SPK, PB4 an R1K2->LED
;*               :PB3 an FET-TREIBER->IRF9640,
;* Werkzeuge      :SiSy 2.16 mit GNU Assembler/Linker 3.x, AVRDUDE 3.x
;-----
.equ             ACSR,0x08
.equ             PINB,0x16
.equ             DDRB,0x17
.equ             PORTB,0x18
;-----
.equ             PulsDauer,8      ;Pulsdauer..80 yS
.equ             HalbZyklus,80    ;Gesamtzyklus ca 1,7 mS 2 mal 800 yS
.equ             Delay,2          ;Sample-Delay 20yS plus Fine-Tuning
;-----
                rjmp      main    ;$000 Power On Reset handler
                reti
                reti
                reti
                reti
                reti
;-----
; Start, Power ON, Reset, I/O init bei ATiny12 kein SRAM und kein STACK
main:           nop
                cbi       DDRB,0      ; PB0 IN OPV comperator
                cpi       DDRB,1      ; PB1 IN poti comperator
                sbi       DDRB,2      ; PB2 OUT SPEAKER
                sbi       DDRB,3      ; PB3 OUT Endstufe
                sbi       DDRB,4      ; PB4 OUT LED
;-----
mainloop:      wdr
                ldi       r16,HalbZyklus ;wait ca. 800 yS
waitZyklus1:   rcall      wait10
                subi      r16,1
                brcc     waitZyklus1   ;Warteschleife
                ldi       r16,0b0001000 ;PB3=1
                out       PORTB,r16    ;IMPULS ON
                wdr
                ldi       r16,PulsDauer ;wait 80 yS IMPULS
waitPuls:     rcall      wait10
                subi      r16,1      ;Warteschleife
                brcc     waitPuls
                ldi       r16,0b0000000 ;PB3=0
                out       PORTB,r16    ;IMPULS OFF
                wdr
                ldi       r16,Delay    ; 1. sampledelay
SampleDelay:  rcall      wait10
                subi      r16,1
                brcc     SampleDelay   ;Warteschleife
                nop
                nop
                nop
                in        r24,ACSR     ;check COMPERATOR
                ldi       r16,0b0000000 ;alles aus oder ...
                sbrs     r24,5        ;wenn schwellwert skip next
                ldi       r16,0b0010100 ;PB2=1 Speakerimpuls Ton1
                out       PORTB,r16
                wdr
                ldi       r16,HalbZyklus ; HI für LED und SPK
waitZyklus2:  rcall      wait10
                subi      r16,1
                brcc     waitZyklus2
                ldi       r16,0b0000000 ;alle Lichtel aus ;- ) PB2-4
                out       PORTB,r16    ;LO für LED und SPK
skip:         rjmp      mainloop
;-----
wait10:       ldi       r25,2          ;warte 10 yS bei 1,2MHz
w10:          subi      r25,1
                brcc     w10
                ret
;-----

```

Hexdump

```
:1000000005C0189518951895189518950000B8987A
:100010007130BA9ABB9ABC9AA89500E520D00150DD
:10002000E8F708E008BBA89508E019D00150E8F708
:1000300000E008BBA89502E012D00150E8F70000EC
:100040000000000088B100E085FF04E108BBA8952E
:1000500000E505D00150E8F700E008BBDDCF92E0F5
:060060009150F0F7089535
:00000001FF
```

Messpunkte

