

```

; File: ATTiny13_SC_Doppler.asm
; *****
; * SC-DOPPLER STRG f. PA3BNX-SoundCardDoppler- *
; * Software (download + info www.pi4wag.nl) *
; * Dopplersignalauswertung per SoundKarte *
; * *
; *
; * Der hier vorgestellte PGM-Code veranlasst *
; * einen ATiny 13 4 Antennen anzusteuern und *
; * liefert den ReferenzSinus zur Peilwert- *
; * Ermittlung des empfangenen Dopplersinus *
; * zur Uebergabe an Stereo-Soundkarte *
; * *
; *
; * Projekt: Radio Direction Finding mit AVR's *
; * Nr002 *
; * Baugruppe: Antennen Steuerung *
; * Harttaster mit 4 Antennen un ATTiny13 *
; * *
; * Autor: Frank Scholz *
; * (C)2011 by Frank Scholz (DO2SFD) *
; * Version 1.0.04.2011 *
; * Beginn des Projekts: 30.04.2011 *
; * Letzte Aenderung : 30.04.2011 *
; * *
; *
; * ===== *
; * *
; * AVR Studio 4.18.700 *
; * GUI Version 4, 18, 0, 685 *
; * AVR Simulator 1, 0, 2, 1 *
; * ATTiny13 249 *
; * *
; * Operating System *
; * Major 5 *
; * Minor 1 *
; * PlatformID 2 *
; * Build 2600 *
; * Service Pack 2 *
; * *
; * Plugins: *
; * *
; * AvrPluginAvrAsmObject 1, 0, 0, 48 *
; * AvrPluginavrgccplugin 1, 0, 0, 11 *
; * Stk500Dll 1, 0, 1, 15 *
; * ----- *
; * ISP: AVRISP mkII *
; * *
; *
; * *****
;
.NOLIST
.INCLUDE "tn13def.inc" ;Headerdatei f.ATTINY13
.LIST
;
; =====
; H A R D W A R E I N F O R M A T I O N E N
; =====
;
; PB0 = Sinusausgang über RC-Tiefpass (PIN5)
; PB1 = Schaltausgang Antenne 1 (PIN6)
; PB2 = Schaltausgang Antenne 2 (PIN7)
; PB3 = Schaltausgang Antenne 3 (PIN2)
; PB4 = Schaltausgang Antenne 4 (PIN3)
; PB5 = Reseteingang 10k auf +UB (PIN1)
; UB 2,7-5,5V (PIN8)
; GND (PIN4)
; Antennen ueber 1k an Pindiodenschalter
; siehe Schaltung in Dokumentation
;
; =====
; P O R T S U N D P I N S
; =====

```

```

;
.EQU Antenne = PORTB
.EQU Analog = OCR0A
;
; =====
; K O N S T A N T E N   Z U M   E I N S T E L L E N
; =====
;
.EQU cCtc    = 0 ; Startwert für Timercounter=255-cCtc)
;
; =====
; F E S T E + A B G E L E I T E T E K O N S T A N T E N
; =====

.EQU cClock = 4800 ; Prozessortaktfrequenz in kHz

;
; =====
;   R E G I S T E R D E F I N I T I O N E N
; =====
;
.DEF rmp = R16 ; Vielzweckregister
.DEF ant = R22 ; Antennenausgabe
.DEF sin = R23 ; Sinusausgabe
.DEF tmr = R24 ; Timer- Voraufladung
; =====
;   S R A M   D E F I N I T I O N E N
; =====
;
.DSEG
.ORG 0X0060
; Format: Label: .BYTE N ; Reserviere N Bytes fuer ...
;
; =====
;   R E S E T   U N D   I N T   V E K T O R E N
; =====
;
.CSEG
.ORG $0000
    rjmp main ; Reset Vektor
    reti ; INT0-Interrupt, nicht verwendet
    reti ; PIN-Change-Interrupt
    rjmp TcntOvfl_Int ; TCO-Overflow-Interrupt
    reti ; EERDY-Interrupt, nicht benutzt
    reti ; ANACOMP-Interrupt, nicht benutzt
    reti ; TC0COMPA-Interrupt, nicht benutzt
    reti ; TC0COMPB-Interrupt, nicht benutzt
    reti ; WDT-Interrupt, nicht benutzt
    reti ; ADC-Interrupt, nicht benutzt
;
; =====
;   I N T E R R U P T   S E R V I C E
; =====
;
TcntOvfl_Int:
    ldi tmr,cCtc    ; Lade Register mit Timercounter Startwert
    out TCNT0,tmr   ; Lade Startwert in Timercounter
    out Antenne,ant ; Ausgabe letzter Tabellenwert an Antenneschaltport
    out Analog,sin  ; Ausgabe letzter Tabellenwert an OCOA
NXTTAB:
    lpm R0,Z+       ; ersten Wert aus Tabelle lesen
    mov ant,R0      ; Tabellenwert in Antennenregister schreiben
    tst ant         ; Teste ersten Tabellenwert auf 0
    brne NXTT2     ; wenn ja nochmal zurueck wenn nein weitermachen
    ldi ZH,HIGH(2*ATab) ; Z auf ATabelle
    ldi ZL,LOW(2*ATab)
    rjmp NXTTAB
NXTT2:
    lpm R0,Z+
    mov sin,R0
    reti
;
; =====

```

```

;   H A U P T P R O G R A M M   I N I T
;   =====
;
MAIN:
; Initiiere Stapel
    ldi rmp, LOW(RAMEND) ; Initiiere LSB Stapel
    out SPL,rmp
; Init Port B PINS B0...B4 als Ausgaenge
    ldi rmp,(1<<DDB0)|(1<<DDB1)|(1<<DDB2)|(1<<DDB3)|(1<<DDB4) ; Richtung Port B
    out DDRB,rmp
    ; Zeiger Z auf Tabellenanfang
    ldi ZH,HIGH(2*ATab) ; Z auf Sinus-Tabelle
    ldi ZL,LOW(2*ATab)

; Init Analogausgang an Port PB1
    ; Timer TC0 einstellen
    ldi rmp,0b10000011 ; Fast PWM,
    out TCCR0A,rmp ;
    ldi rmp,0b00000001 ; Fast PWM, Prescaler=1,
    out TCCR0B,rmp ; Timer starten
    ldi rmp,1<<TOIE0 ; Timer-Interrupt freigeben
    out TIMSK0,rmp
    ldi tmr,cCtc ; Startwert für Timercounter
    out TCNT0,tmr
;
    ldi rmp,1<<SE ; Enable sleep
    out MCUCR,rmp

    sei ; Interrupts freigeben
;
;   =====
;   P R O G R A M M - S C H L E I F E
;   =====
;
LOOP:
    sleep ; Schlafen legen
    nop ; Dummy fuer Aufwecken
    rjmp loop ; Zurueck nach Loop

;   =====
;
; Ende Quellcode
; Copyright (C)04/2011 by FSE Frank Scholz Electronic (DO2SFD)
;
;DATENTABELLE:

ATAB:
.DB 2,94 ;0..90 Grad
.DB 2,109
.DB 2,123
.DB 2,136
.DB 2,148
.DB 2,157
.DB 2,164
.DB 2,169
.DB 4,170 ;90..180 Grad
.DB 4,169
.DB 4,164
.DB 4,157
.DB 4,148
.DB 4,136
.DB 4,123
.DB 4,109
.DB 8,94 ;180..270 Grad
.DB 8,79
.DB 8,65
.DB 8,52
.DB 8,40
.DB 8,31
.DB 8,24
.DB 8,19
.DB 16,18 ;270..360 Grad
.DB 16,19

```

```
.DB 16,24  
.DB 16,31  
.DB 16,40  
.DB 16,52  
.DB 16,65  
.DB 16,79  
.DB $00,$00
```

```
;Ende Datentabelle
```

```
;  
;*****  
; ENDE ENDE ENDE ENDE ENDE ENDE ENDE ENDE ENDE ENDE ENDE ENDE  
;*****
```