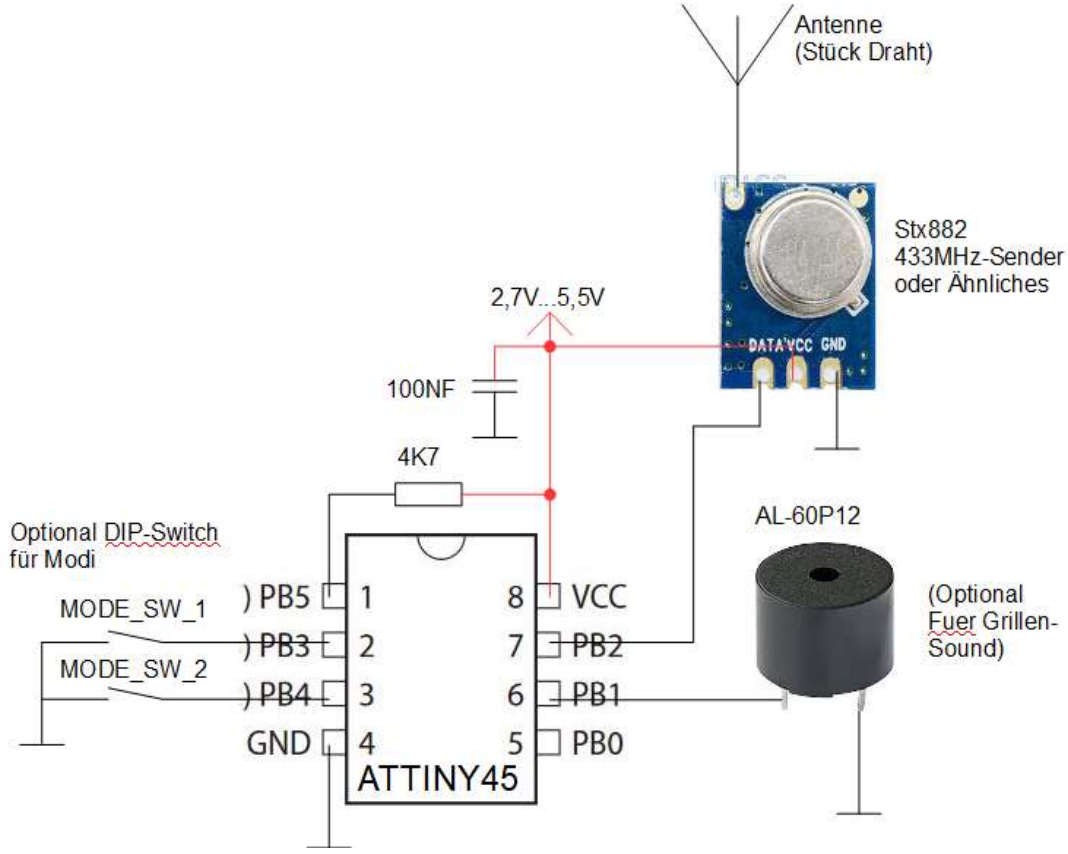
Grillen-Fuchs für 433MHz ISM-Band
JO-Discovery 04_01_2021

Bei dem 433MHz „Grillen-Fuchs“ handelt es sich um einen sehr simplen Fuchssender, welcher neben dem Funksignal zur Ortung, auch ein akustisches Signal von sich gibt (für den Nahbereich). Das akustische Signal ist dem Klang einer Grille nachempfunden. Hierbei wurde ein echtes „Grillen-Sample“ verwendet, wobei dieses Sample D/A gewandelt (per PWM) und auf einen Lautsprecher gegeben wird. Nach jedem Durchlauf (hier 10s) wird ein kurzer Grillensound abgespielt. Als Lautsprecher wird ein kleiner Signalgeber genutzt, dessen Spulenwiderstand so hoch ist, dass der ATTINY ihn direkt treiben kann. Wem das Signal zu leise ist, kann dieses jedoch nach Belieben mit einem Transistorverstärker verstärken. Wenn man die Grillen-Funktion nicht benutzen möchte, lässt man den PB1-Pin einfach offen.

Der ATTINY wird in den Pausen zwischen dem Senden immer in den Power-Down-Modus versetzt. Die Stromaufnahme der gesamten Schaltung beträgt dann etwa 5uA. Wenn im Code das #define PAUSE vergrößert wird, so kann der durchschnittliche Stromverbrauch so stark verringert werden, dass die ganze Schaltung mit einer kleinen Batterie über Wochen hinweg betrieben werden kann, was natürlich eher nicht nötig ist. Die Stromaufnahme während dem Senden ist vom Sender abhängig. Der Stx882 zieht bei 3,3V Versorgungsspannung etwa 34mA bei 15dBm Sendeleistung (ca. 30mW). Es ist hierbei (selbstverständlich) darauf zu achten, dass die landesspezifischen Bestimmungen eingehalten werden und dementsprechend die Sendeleistung nicht zu hoch sein darf. Die Reichweite kann mit 30mW leicht 1km betragen. Das ist aber sehr stark vom Standort und den Bedingungen abhängig.



Über die zwei Schalter SW1 und SW2 (Dip-Schalter) kann der Modus des Fuchssenders eingestellt werden. Folgende Modi sind in Firmware-Version 1.0 implementiert (können aber leicht modifiziert werden):

0 = Schalter geschlossen ; 1 = Schalter geöffnet

MODE_SW_1	MODE_SW_1	Bedeutung
0	0	1X 250ms senden und danach 10 Sekunden Pause
0	1	2X 250ms senden (mit 500ms Abstand) und danach 10 Sekunden Pause
1	0	3X 250ms senden (mit 500ms Abstand) und danach 10 Sekunden Pause
1	1	4X 250ms senden (mit 500ms Abstand) und danach 10 Sekunden Pause

Empfänger:

Grundsätzlich kann zum Empfangen jeder Funkscanner verwendet werden, sofern er die Empfangsfeldstärke irgendwie wiedergeben kann (sonst wird das Peilen schwer).

Sehr gut hat sich die Verwendung von einem SDR-Stick in Kombination mit einem Smartphone herausgestellt. Da hat man sogar eine Anzeige für die Empfangsfeldstärke. Dazu kann man noch eine kleine Richtantenne verwenden (zum Beispiel die für 433MHz sehr kompakte MOXXON-Antenne).

Getestet wurde das Ganze mit dem NooElec NESDR SMArTee v2 in Kombination mit einer selbstgebauten Moxxon-Antenne und einem Smartphone mit SDR-Touch für Android. Für die Feldstärkeanzeige muss man einen kleinen Betrag bezahlen, aber es lohnt sich.

```
/*
 * main.c
 * Grillen-Fuchs V1.1
 * Created: 04.01.2021 12:01:06
 * Author : jo D
 */

// Achtung! Darauf achten, dass F_CPU=8000000UL definiert ist und die FUSES entsprechend gesetzt sind.

#include <avr/io.h>
#include <avr/sleep.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <avr/power.h>
#include <avr/wdt.h>
#include <avr/pgmspace.h>

#define GRILLE_PIN PB1 // PWM Pin --> nicht aendern
#define SENDE_PIN PB2
#define MODE_SW_1 PB3
#define MODE_SW_2 PB4
#define FUCHS_SENDEDAUER 1 // definiert wie lange die Sende-Impulse sind. Sende-Impuls-Dauer = FUCHS_SENDEDAUER * 250ms
#define FUCHS_PAUSE 2 // definiert wie lange die kurze Pausen zwischen den Sende-Impulsen sind. Dauer = FUCHS_PAUSE * 250ms
#define GROSSE_PAUSE 40 // definiert wie lange die große Pause zwischen zwei Fuchs-Signalen ist. Dauer = GROSSE_PAUSE * 250ms

#define GRILLE_SOUND_SAMPLES 1576
#define GRILLE_SAMPLE_RATE 11025
const uint8_t GrilleSound[GRILLE_SOUND_SAMPLES] PROGMEM= {
         128, 126, 127, 126, 126, 128, 126, 128,
         128, 126, 128, 126, 127, 128, 126, 128,
         126, 127, 128, 124, 129, 126, 126, 129,
         124, 129, 126, 126, 129, 124, 129, 128,
         124, 131, 122, 127, 129, 122, 133, 124,
         126, 131, 120, 133, 126, 122, 135, 118,
         131, 129, 118, 137, 118, 127, 135, 115,
         139, 120, 122, 140, 111, 139, 126, 115,
         146, 109, 135, 133, 107, 150, 109, 129,
         142, 100, 153, 113, 120, 151, 94, 151,
         120, 109, 161, 91, 148, 131, 96, 170,
         91, 140, 146, 83, 177, 93, 128, 162,
         70, 181, 102, 109, 181, 59, 177, 120,
         85, 199, 56, 162, 146, 59, 212, 63,
         137, 177, 35, 216, 81, 104, 210, 21,
         207, 113, 67, 236, 21, 181, 151, 32,
         247, 39, 140, 197, 4, 242, 76, 85,
         243, 0, 212, 137, 21, 255, 10, 148,
         212, 0, 255, 80, 65, 255, 0, 179,
         168, 24, 221, 91, 91, 199, 67, 135,
         164, 74, 155, 135, 93, 159, 115, 115,
         146, 111, 129, 131, 122, 128, 124, 133,
         116, 129, 135, 109, 144, 122, 113, 153,
         104, 133, 142, 98, 155, 113, 120, 148,
         100, 146, 122, 118, 146, 107, 140, 128,
         115, 146, 111, 133, 135, 109, 146, 116,
         126, 140, 107, 142, 124, 118, 144, 109,
         139, 127, 115, 146, 107, 139, 129, 113,
         148, 109, 133, 135, 107, 150, 111, 129,
         140, 102, 151, 113, 124, 146, 100, 150,
         116, 118, 151, 98, 148, 124, 107, 159,
         96, 140, 139, 93, 164, 104, 120, 159,
         85, 155, 126, 100, 166, 94, 135, 144,
         93, 159, 109, 122, 148, 100, 148, 118,
         118, 146, 102, 146, 120, 113, 155, 93,
         148, 127, 100, 166, 91, 140, 139, 93,
         164, 102, 126, 150, 89, 162, 104, 128,
         146, 91, 170, 83, 159, 115, 111, 170,
         59, 201, 67, 150, 153, 52, 229, 30,
         185, 135, 52, 245, 8, 199, 133, 39,
         255, 0, 201, 148, 17, 255, 0, 188,
         174, 0, 255, 0, 164, 205, 0, 255,
         12, 133, 234, 0, 255, 34, 105, 255,
         0, 255, 54, 78, 255, 0, 255, 78,
         48, 255, 0, 255, 115, 10, 255, 0,
         221, 164, 0, 255, 0, 170, 218, 0,
         255, 12, 111, 255, 0, 255, 67, 54,
         255, 0, 249, 127, 0, 255, 0, 188,
         196, 0, 255, 15, 113, 255, 0, 255,
         89, 39, 255, 0, 197, 174, 0, 255,
         37, 107, 240, 0, 223, 128, 35, 253,
         35, 137, 196, 24, 201, 116, 76, 201,
         76, 127, 168, 76, 155, 133, 94, 161,
         111, 118, 148, 107, 133, 131, 118, 131,
         126, 126, 124, 131, 124, 124, 135, 118,
         129, 131, 116, 135, 126, 122, 135, 120,
         128, 129, 122, 129, 124, 129, 124, 129,
         126, 124, 133, 118, 133, 126, 122, 135,
         118, 133, 124, 126, 133, 118, 137, 120,
         129, 133, 116, 140, 118, 129, 135, 113,
```

142, 118, 129, 135, 115, 140, 118, 128,
135, 115, 137, 122, 124, 135, 116, 133,
124, 122, 135, 115, 135, 122, 122, 139,
107, 142, 116, 118, 148, 93, 153, 115,
109, 166, 72, 168, 111, 102, 181, 56,
183, 100, 107, 181, 50, 194, 89, 118,
177, 48, 201, 80, 128, 174, 50, 205,
72, 137, 164, 56, 208, 65, 150, 155,
59, 212, 54, 161, 150, 58, 223, 43,
170, 148, 52, 234, 34, 174, 150, 43,
243, 24, 179, 151, 35, 255, 15, 179,
159, 23, 255, 10, 174, 172, 6, 255,
12, 161, 188, 0, 255, 19, 146, 205,
0, 255, 30, 128, 220, 0, 255, 45,
107, 236, 0, 255, 65, 83, 255, 0,
249, 91, 56, 255, 0, 229, 122, 30,
255, 0, 199, 159, 0, 255, 0, 161,
201, 0, 255, 35, 109, 245, 0, 255,
93, 48, 255, 0, 201, 162, 0, 255,
26, 126, 223, 0, 236, 102, 59, 247,
21, 164, 172, 30, 220, 85, 100, 199,
52, 164, 142, 74, 183, 96, 120, 161,
89, 148, 129, 107, 148, 116, 126, 137,
118, 129, 129, 122, 129, 129, 122, 129,
127, 122, 133, 124, 126, 133, 120, 133,
126, 122, 135, 120, 129, 131, 118, 135,
122, 126, 135, 118, 133, 126, 122, 135,
118, 131, 128, 122, 133, 118, 131, 126,
122, 135, 118, 131, 129, 116, 139, 118,
126, 135, 115, 137, 124, 120, 139, 116,
131, 129, 116, 139, 118, 127, 133, 115,
137, 120, 124, 135, 115, 137, 122, 126,
133, 116, 135, 118, 129, 127, 120, 139,
109, 140, 120, 118, 150, 94, 151, 118,
109, 164, 81, 159, 122, 98, 177, 72,
161, 128, 87, 188, 67, 161, 135, 78,
197, 61, 159, 142, 70, 201, 65, 151,
150, 69, 197, 74, 144, 150, 76, 185,
81, 146, 139, 91, 175, 83, 155, 124,
105, 170, 76, 170, 107, 115, 170, 67,
185, 93, 122, 172, 56, 201, 81, 124,
181, 39, 216, 76, 120, 194, 24, 223,
78, 109, 207, 17, 223, 85, 98, 216,
13, 218, 94, 89, 220, 15, 212, 100,
83, 223, 13, 210, 105, 76, 231, 10,
209, 115, 61, 245, 2, 203, 133, 35,
255, 0, 183, 168, 4, 255, 17, 144,
210, 0, 255, 54, 93, 251, 0, 240,
109, 41, 255, 0, 186, 170, 4, 255,
37, 122, 220, 0, 229, 104, 65, 238,
26, 166, 162, 39, 216, 81, 111, 190,
56, 168, 131, 85, 179, 94, 127, 153,
93, 148, 128, 109, 148, 115, 126, 140,
113, 135, 129, 116, 139, 118, 126, 135,
115, 137, 124, 122, 139, 115, 133, 128,
118, 139, 118, 128, 131, 118, 135, 124,
126, 131, 122, 129, 126, 126, 127, 126,
127, 124, 129, 124, 128, 129, 122, 131,
124, 126, 129, 120, 133, 124, 127, 131,
120, 135, 122, 128, 131, 116, 137, 120,
126, 135, 115, 139, 120, 124, 139, 113,
137, 124, 118, 142, 113, 133, 131, 113,
144, 116, 124, 142, 107, 140, 127, 111,
150, 107, 131, 139, 102, 151, 113, 124,
146, 98, 151, 116, 120, 148, 98, 151,
113, 124, 146, 96, 157, 107, 128, 146,
93, 164, 98, 133, 144, 89, 174, 89,
140, 144, 81, 185, 78, 146, 146, 72,
197, 67, 150, 151, 61, 208, 61, 150,
157, 54, 216, 58, 150, 161, 47, 221,
52, 150, 164, 41, 227, 50, 150, 168,
35, 232, 48, 146, 175, 28, 236, 48,
140, 183, 23, 238, 52, 135, 190, 17,
238, 58, 126, 197, 13, 236, 65, 115,
207, 10, 232, 74, 104, 214, 10, 227,
85, 93, 223, 6, 221, 98, 78, 234,
4, 212, 115, 58, 249, 4, 197, 140,
34, 255, 12, 172, 172, 10, 255, 32,
137, 207, 0, 253, 65, 96, 234, 0,
227, 107, 56, 255, 2, 190, 153, 21,
255, 28, 142, 201, 0, 247, 72, 87,
240, 0, 212, 131, 35, 255, 21, 155,
186, 10, 240, 72, 100, 218, 21, 196,
126, 67, 214, 58, 148, 159, 65, 186,
96, 118, 164, 81, 159, 120, 109, 157,
98, 142, 126, 113, 148, 107, 139, 126,
118, 142, 109, 139, 124, 120, 140, 111,
139, 122, 122, 139, 111, 139, 122, 122,

```
        139, 113, 137, 124, 122, 137, 115, 135,
        124, 124, 135, 118, 133, 124, 126, 131,
        120, 133, 122, 127, 129, 120, 133, 122,
        127, 131, 120, 131, 124, 124, 131, 122,
        129, 128, 124, 131, 122, 128, 127, 124,
        129, 124, 127, 128, 126, 129, 124, 128,
        126, 126, 127, 126, 127, 128, 126, 127,
        126, 126, 131, 122, 129, 129, 120, 137,
        120, 126, 135, 113, 140, 120, 122, 142,
        105, 142, 122, 116, 148, 104, 142, 126,
        113, 150, 104, 142, 126, 115, 146, 105,
        142, 122, 122, 139, 113, 139, 118, 131,
        126, 124, 135, 111, 148, 105, 140, 129,
        102, 166, 83, 155, 128, 93, 185, 67,
        166, 126, 85, 197, 54, 174, 124, 80,
        205, 45, 181, 122, 80, 210, 37, 188,
        116, 81, 212, 32, 194, 111, 83, 212,
        30, 199, 105, 89, 210, 28, 203, 102,
        93, 208, 28, 207, 98, 94, 208, 26,
        209, 96, 94, 210, 24, 208, 96, 93,
        212, 23, 208, 100, 89, 214, 24, 205,
        104, 85, 216, 24, 199, 109, 80, 218,
        28, 192, 118, 74, 218, 34, 183, 129,
        67, 218, 43, 168, 144, 59, 214, 56,
        151, 157, 56, 209, 70, 135, 168, 54,
        196, 89, 116, 177, 59, 179, 111, 102,
        181, 70, 159, 129, 91, 177, 87, 139,
        146, 87, 166, 105, 120, 155, 91, 151,
        124, 107, 157, 100, 137, 139, 102, 151,
        113, 122, 146, 104, 142, 124, 115, 144,
        111, 133, 131, 116, 139, 120, 127, 129,
        122, 131, 124, 128, 124, 129, 124, 126,
        131, 118, 135, 122, 124, 137, 113, 139,
        122, 122, 139, 113, 139, 122, 124, 137,
        115, 137, 122, 126, 133, 118, 133, 122,
        127, 129, 122, 131, 124, 127, 128, 124,
        129, 124, 128, 127, 126, 128, 126, 126,
        127, 124, 128, 127, 124, 129, 124, 128,
        127, 124, 129, 124, 128, 128, 124, 129,
        124, 128, 127, 124, 129, 124, 127, 128,
        124, 129, 124, 129, 126, 126, 129, 124,
        129, 124, 128, 129, 122, 131, 124, 128,
        129, 122, 131, 124, 127, 129, 122, 131,
        122, 127, 129, 122, 131, 124, 128, 129,
        122, 131, 124, 126, 131, 120, 133, 124,
        126, 131, 120, 133, 124, 126, 131, 120,
        133, 122, 128, 129, 122, 131, 122, 128,
        127, 124, 129, 124, 128, 126, 128, 126,
        126, 127, 124, 129, 124, 127, 128, 0
};

void pwm_setup (void)
{
        // Set Timer 0 prescaler to clock
        TCCR0B |= (1 << CS00);

        // Set to 'Fast PWM' mode
        TCCR0A |= (1 << WGM01) | (1 << WGM00);

        // Clear OC0B output on compare match, upwards counting.
        TCCR0A |= (1 << COM0B1);
}

void pwm_write (uint8_t val)
{
        OCR0B = val;
}

void SendePin_init(void)
{
        DDRB |= (1 << SENDE_PIN);
        PORTB &= ~(1 << SENDE_PIN);
}

void ModeSw_Init(void)
{
        DDRB  &= ~ ( (1<<MODE_SW_1) | (1<<MODE_SW_2) ) ; // eingang
        PORTB &= ~ ( (1<<MODE_SW_1) | (1<<MODE_SW_2) ); // interner pull up wird deaktiviert um Strom zu sparen
}

uint8_t GetMode(void)
{
        uint8_t sw1 = 1;
        uint8_t sw2 = 1;
        uint8_t mode = 0;

        PORTB |= (1<<MODE_SW_1) | (1<<MODE_SW_2); // interner pull up wird aktiviert
        _delay_ms(20); // warten, bis sich der Wert eingependelt hat
```

```
        if ( !(PINB & (1<<MODE_SW_1)) ) { sw1 = 0; }
        if ( !(PINB & (1<<MODE_SW_2)) ) { sw2 = 0; }

        PORTB &= ~ ( (1<<MODE_SW_1) | (1<<MODE_SW_2) ); // interner pull up wird deaktiviert um Strom zu sparen
        mode = sw1+(sw2<<1)+1;
        return mode;
}

void PlayGrilleSound(void)
{uint16_t step = 0;

        // Ausgang aktivieren
        DDRB |= (1 << GRILLE_PIN);
        PORTB &= ~(1 << GRILLE_PIN);

        for (step=0; step<GRILLE_SOUND_SAMPLES; step++)
        {
                pwm_write(pgm_read_byte(&GrilleSound[step]));
                _delay_us(1000000/GRILLE_SAMPLE_RATE);
        }
        pwm_write(0);

        // Ausgang deaktivieren
        DDRB &= ~(1 << GRILLE_PIN);
        PORTB &= ~(1 << GRILLE_PIN);
}

void resetWatchDog ()
{
        MCUSR = 0;
        WDTCR = (1<< WDCE ) | (1<< WDE ) | (1<< WDIF );
        WDTCR = (1<< WDIE ) | (1<< WDP2 ); // 250ms second TimeOut
        wdt_reset ();
}

// Schlafen und mit Watchdog interrupt wieder aufwachen
void sleep(void) {

         set_sleep_mode ( SLEEP_MODE_PWR_DOWN ); // sleep mode Power Down setzen
         ADCSRA &= ~(1<<ADEN);                   // ADC aus
         power_all_disable ();                   // power off ADC, TIMER 1 and 2, Serial Interface
         cli();                                                                  // zur Vorsicht interrupts aus
         resetWatchDog ();                       // reset WatchDog
         sleep_enable ();                        // ermögliche sleep
         sei();                                                                  // interrupts ein
         sleep_cpu ();                           // system geht in den sleep mode
         sleep_disable ();                       // nachdem der interrupt geweckt hat, sleep_disable
         power_all_enable ();                    // power on ADC, TIMER 1 and 2, Serial Interface
}

// Schlafenszeit = sleeptime * 250ms
void sleep_time(uint16_t sleeptime)
{ uint16_t n = 0;
        for (n=0;n<sleeptime;n++)
        {
                sleep();
        }
}

void SenderAus(void)
{
        PORTB &= ~(1 << SENDE_PIN);
}

void SenderEin(void)
{
        PORTB |= (1 << SENDE_PIN);
}

void SendeFuchsSignal(uint8_t pattern)
{ uint8_t n =0;

        for (n=0;n<pattern;n++)
        {
                 SenderEin();
                 sleep_time(FUCHS_SENDEDAUER);
                 SenderAus();
                 sleep_time(FUCHS_PAUSE);
        }
}

int main(void)
{
        pwm_setup();
        SendePin_init();
        ModeSw_Init();
```

```
        while (1)
        {
                SendeFuchsSignal(GetMode());
                PlayGrilleSound();
                sleep_time(GROSSE_PAUSE);
        }
}

ISR(WDT_vect) {
        wdt_disable();  // bis zum nächsten mal wd disabled...
}
```