

```

1 // Simple AM Radio Signal Generator :: Markus Gritsch
2 // http://www.youtube.com/watch?v=y1EKyQrFJ-o
3 //
4 //
5 //
6 //
7 //
8 //
9 // audio input
10 //
11 //
12 //
13 //
14 //
15 // fg < 7 Hz
16 //
17 //
18 //
19 //
20 // fg = 1 / ( 2 * pi * ( R1 || R2 ) * C1 ) < 7 Hz
21 // fres = 1 / ( 2 * pi * sqrt( L1 * C2 ) ) = 734 kHz
22
23 #define INPUT_PIN 0 // ADC input pin
24 #define TIMER_PIN 3 // PWM output pin, OC2B (PD3)
25 #define DEBUG_PIN 2 // to measure the sampling frequency
26 #define LED_PIN 13 // displays input overdrive
27
28 #define SHIFT_BY 3 // 2 ... 7 input attenuator
29 #define TIMER_TOP 20 // determines the carrier frequency
30 #define A_MAX TIMER_TOP / 4
31
32 void setup() {
33     pinMode( DEBUG_PIN, OUTPUT );
34     pinMode( TIMER_PIN, OUTPUT );
35     pinMode( LED_PIN, OUTPUT );
36
37     // set ADC prescaler to 16 to decrease conversion time (0b100)
38     ADCSRA = ( ADCSRA | _BV( ADPS2 ) ) & ~( _BV( ADPS1 ) | _BV( ADPS0 ) );
39
40     // non-inverting; fast PWM with TOP; no prescaling
41     TCCR2A = 0b10100011; // COM2A1 COM2A0 COM2B1 COM2B0 - - WGM21 WGM20
42     TCCR2B = 0b00001001; // FOC2A FOC2B - - WGM22 CS22 CS21 CS20
43
44     // 16E6 / ( OCR2A + 1 ) = 762 kHz @ TIMER_TOP = 20
45     OCR2A = TIMER_TOP; // = 727 kHz @ TIMER_TOP = 21
46     OCR2B = TIMER_TOP / 2; // maximum carrier amplitude at 50% duty cycle
47 }
48
49 void loop() {
50     // about 34 kHz sampling frequency
51     digitalWrite( DEBUG_PIN, HIGH );
52     int8_t value = ( analogRead( INPUT_PIN ) >> SHIFT_BY ) -
53         ( 1 << ( 9 - SHIFT_BY ) );
54     digitalWrite( DEBUG_PIN, LOW );
55
56     // clipping
57     if ( value < -A_MAX ) {
58         value = -A_MAX;
59         digitalWrite( LED_PIN, HIGH );
60     } else if ( value > A_MAX ) {
61         value = A_MAX;
62         digitalWrite( LED_PIN, HIGH );
63     } else {
64         digitalWrite( LED_PIN, LOW );
65     }
66
67     OCR2B = A_MAX + value;
68 }
end

```