2004 Design Contest - Project Number A3713

"Hi-Fi" DTMF encoder with D class power amplifier

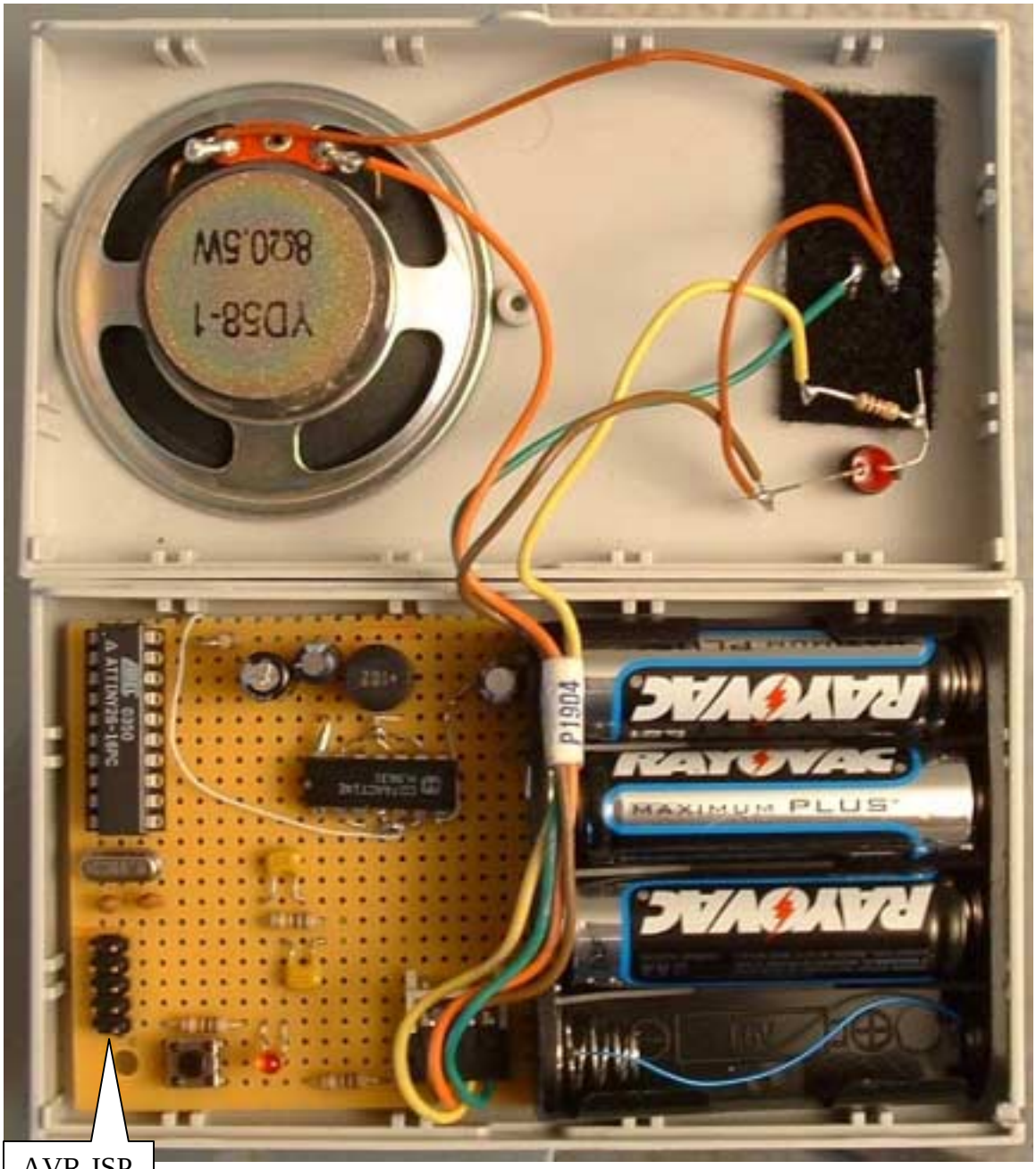
Description

High Fidelity AtTiny26 based DTMF generator with D class power amplifier and low Power consumption - a modern DTMF dialer implementation

This is a Reference Design for a DTMF generator using the High Speed PWM peripheral of the AVR series. The design has some key performance parameters:

*** High Purity DTMF Signal generator with the harmonics attenuated by 40db.**
*** Simple Low Pass Output filter due to the high PWM frequency - 500khz, and high sample frequency - 31.25Khz.**
*** Low power consumption- 1µA in standby mode and 25mA in active mode.**
*** DTMF encoder block for telephony applications or stand alone DTMF dialer.**
*** Small Footprint ~1024 bytes leaving the other 1024 bytes of the AtTiny 26 free for other applications.**

The prototype is enclosed in a small plastic box. It has just a push button to start the DTMF dialer, an LED to indicate when is active and a Speaker. There is no power switch as the CPU goes to sleep and the power consumption is only 1µA. Three AA batteries are used to power up the device and the batteries will last for few years.



Low Power Considerations

The CPU goes to Power Down sleep mode when not used – all the clock sources are stopped and the INT0 pin is used to wake up the CPU (a low level on INT0 pin will wake up the CPU). The power consumption is 1µA only including the 74ACT14N buffers.

Clock fuse used:
Ext. Crystal/Resonator High Freq ; Startup Time 16CK+64mS

In order to obtain such a low power performance the Brown Out detector and the Watchdog Timer are disabled. All the ports are configured like inputs in stand-by mode with the pull-up resistors activated.

In active mode the average current is only 25mA for the CPU and the D class power amplifier connected to 8-ohm speaker.

A power switch is not necessary and the three AA batteries used will last for several years. The dialer is working even with 2.7 volts power supply.

Software

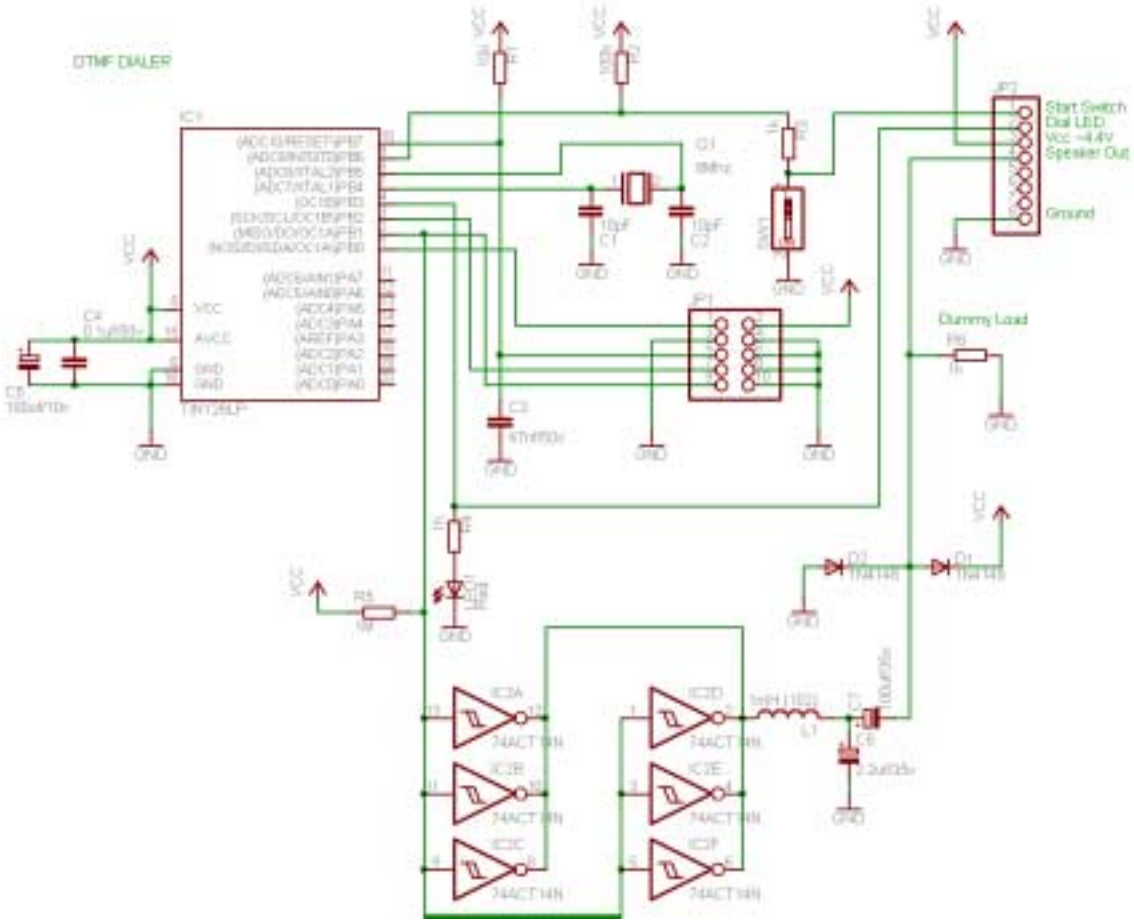
Timer 0 used to generate software interrupts @ 31.25Khz as sample frequency.
Timer 1 used to generate PWM signal @ 500khz.
Internal clock generator and PLL used to generate 128Mhz High-speed peripheral clock (64 Mhz according to the specifications).
8 Mhz external Crystal used as CPU and Timer0 Clock.
Small footprint ~ 1024 bytes – ½ of the AtTiny26 EEPROM.

All the software is written in C – GCC 3.3.2 from GNU – the WinAvr distribution.

```
SIGNAL (SIG_OVERFLOW0 ) /* interrupt service routine for Timer0 */

{
if (flag_delay)   /* delay ISR */
      {
      acc_delay--;
      if (acc_delay==0)
            flag_delay = 0;

      }


if (flag_pwm)           /* PWM isr */
      {
      volatile  uint8_t temp_val_hi;
      volatile  uint8_t temp_val_lo;

      /* read value from the program memory, 128 entries half sin table
*/
    /*  OCR1A = sin_table(acc_hi)+ 0.75*sin_table(acc_lo) + 128 */
      temp_val_hi = pgm_read_byte_near (sine_table + ((unsigned
char)(acc_hi>>8) & 0x7f));
      if((acc_hi>>8) & 0x80)
            temp_val_hi = 128-temp_val_hi;
      else
            temp_val_hi = 128+temp_val_hi;

      temp_val_lo = pgm_read_byte_near (sine_table + ((unsigned
char)(acc_lo>>8) & 0x7f));
      temp_val_lo = (unsigned char)(temp_val_lo>>1);
      temp_val_lo = temp_val_lo+(unsigned char)(temp_val_lo>>1);
```

```
        if((acc_lo>>8) & 0x80)
              OCR1A = temp_val_hi+temp_val_lo;
        else
              OCR1A = temp_val_hi-temp_val_lo;


        acc_lo = acc_lo + add_lo;      /* */
        acc_hi = acc_hi + add_hi;      /* */
        }
};
```

Schematic Diagram



PCB Layout

Resources

www.avrfreaks.com

www.atmel.com
AVR131: Using the AVR's High-speed PWM

8-bit Microcontroller with 2K Bytes Flash
ATtiny26
ATtiny26L

Sources

http://sourceforge.net/projects/winavr
Win AVR package

www.atmel.com
AVR studio 4

www.CadSoftUsa.com
Eagle 4.1 schematic and PCB editor

www.digikey.com
Electronic Components