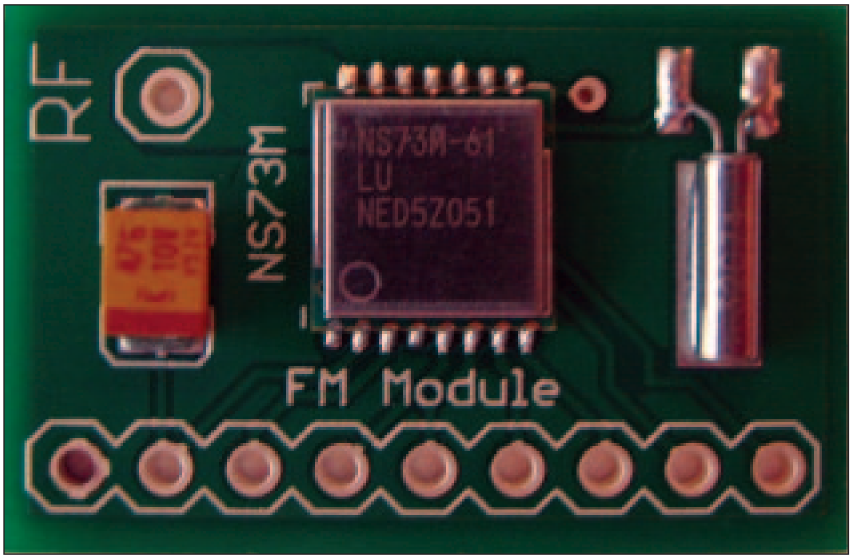
BUTTERFLY BROADCASTER

BY JAY CARTER

MP3 players are great when you are wearing earphones, but wouldn't it be nice to listen through your home or car sound system? Now you can! Just plug your player into this project and it will broadcast the music to any nearby FM radio.



■ PHOTO 1 (shown above). The FM radio transmitter on a chip, the NS73M, pre-assembled on a SparkFun Electronics breakout board.

The key component to this project is a thumbnail-sized FM radio transmitter on a chip — the NS73M — by the Niigata Seimitsu company. Just add a microcontroller and you can be on the air. No coils, no tricky alignment procedures, just pure digital bliss. Photo 1 shows the transmitter chip pre-installed on a small, experimenter friendly circuit board. This module is perfect for breadboarding. While many "wireless microphone" designs exist, low cost units often use a simple RF oscillator. The transmission frequency may be hard to set, and the units tend to drift off frequency easily. With the NS73M transmitter chip, the frequency is digitally selected with a microcontroller. The frequency is rock solid, being controlled by a crystal and a phase locked loop (PLL). Drifting frequencies are a problem of the past.

AVR Butterfly ATMega169 Processor

The ATMEL AVR Butterfly microcontroller board (shown in Photo 2) is perfect for this project. It includes an ATMega169 processor, six-character LCD display, five-position joystick, and a piezo-electric beeper, all put to good use in this project. The entire package is the size of a credit card, and costs about $21. Its built-in RS-232 serial port is used for programming, while its 4 MB DataFlash and thermistor go unused in this application.

You will use the joystick to select the frequency to transmit on, which is displayed on the LCD. This makes frequency selection trivial. The processor stores the frequency in its internal EEPROM, recalling it when the device is turned on in the future. For repeated use, just turn the device on and hit a pre-selected memory button on the radio. It can't get any easier than that!

Although the Butterfly board includes a three volt coin cell battery for normal operation, it is best to use an external battery or power supply. The battery will run down quickly if the transmitter is used extensively on its 2 mW RF output mode.

I²C Communications

Connecting the transmitter module to the Butterfly board is simple as both run on three volts and no level converts are required. The transmitter can communicate using either a two wire I²C protocol, or a three wire protocol. I chose the I²C interface, and tied the transmitter's clock and data lines to those on the Butterfly's USI port, with additional 4.7K pull-up resistors on each line. The transmitter's latch pin is tied to ground, while the "IIC" mode pin is tied to three volts. A short wire is attached to the antenna terminal. Power and audio round out the connections. The transmitter is designed for a maximum input signal amplitude of 200 mV RMS. For this reason, an attenuator with AC signal coupling is provided on the audio inputs to the transmitter.

The transmitter has a TEB terminal which is not tied to a Butterfly digital input in this configuration. It can be used to monitor the ability of the transmitter's phase lock loop to lock in on the desired transmit frequency using the currently selected transmitter "band." The transmitter has four possible bands used to cover the FM radio band from 87.5 to 108 MHz. In this configuration, the optimal band is selected based upon the frequency in use.

The NS73M transmits in stereo. It incorporates both pre-emphasis and a pilot tone without any additional external circuitry. The amplitude of the input signal required to give 100% modulation is software selectable, ranging from 100 mV RMS to 200 mV RMS. Typically, you'd just plug a patch cable from the MP3 player into the stereo audio input jack on the transmitter.

Basic Software

The software for the ATMega169 processor was written in Bascom-AVR Basic. As this dialect of Basic is designed for this family of processors, incorporates commands for I^2C communications, EEPROM access, and floating point math, all of the tools are present to bring this project together easily. You can focus your attention fully on understanding the datasheet for the transmitter software control, and not on language and hardware barriers to its implementation. Neither the transmitter chip's manufacturer's nor the reseller's websites, nor Goggle turned up any pre-existing software for interfacing with the chip in I^2C mode, or for implementing a user tunable interface. With no working examples to expand upon, close scrutiny of the datasheets and a little trial and error were in order. Before long, the transmitter was live with audio blaring from my nearby radio receiver.

The Basic code for this project is available on-line, and can be easily ported to other languages. Having a fully functional version to review will certainly pave the way for those wishing to follow. A pre-compiled hex file is also provided for those who do not have Bascom-AVR available to them.

The transmission frequency is determined by a 14 bit value uniquely determined for each frequency of interest. It is used in setting the transmitter's phase lock loop. Floating point math is a must for these calculations. Internally, the transmitter chip divides the FM radio band into four overlapping sub-bands, which require two additional bits when programming it. This data is dispersed across three of the 13, eight-bit registers in the NS73M. The Butterfly board has a 4 MBit DataFlash memory chip in addition to the 512 byte EEPROM within the processor itself. One could certainly calculate the frequency and band data for each frequency and store them in memory, recalling them as needed whenever the user selected a new frequency to transmit on. However, the floating point math instructions available — coupled with the speed of the processor — make it easy to calculate the values on-the-fly. No look-up tables were required.

In designing the software, I chose to use an index pointing into the FM radio frequency band as the key parameter. The FM band is divided into 206 steps, each 0.1 MHz, stepping upwards from a base frequency of 87.5 MHz. This pointer is a small integer, easily stored in a single byte. Each press of the joystick simply bumps this pointer up or down by one, with roll-over to wrap the frequency around at both the high and low ends of the FM band. The processor uses this pointer to calculate the actual frequency to be displayed on the LCD (1 = 87.5, 2 = 87.6, ... 206 = 108.0). The frequency is then plugged into the PLL setup equations to calculate the 14 bit value which generates that frequency.

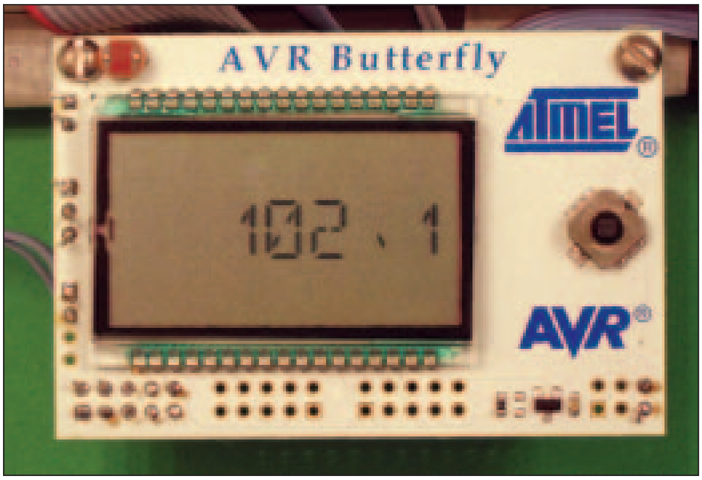
It is desirable to have the unit power-up on the same frequency it was last using. This is much more convenient than retuning the transmitter to a quiet spot on an FM radio each time it is used. By storing the frequency pointer, only a single byte of EEPROM memory is required for this purpose. The pointer method also saves you from storing over 600 bytes of look-up table information, had data tables been utilized.

The transmitter's datasheets elude to using the TEB signal to measuring the ability of the transmitter's PLL to lock in on the chosen frequency. If the PLL is having difficulty locking in and cannot obtain a stable transmission frequency, one bumps the internal, sub-band selection up or down until lock is obtained. I instead chose to measure the frequency range over which the PLL could obtain a lock while programmed for each of the four sub-bands. The program selects the optimal sub-band from this information and loads the correct sub-band selection whenever a new frequency is selected by the user.

Wireless Microphone

This project was designed with an MP3 player in

■ PHOTO 2. The AVR Butterfly demonstration and evaluation board, incorporating an ATMega129 processor, LCD, joystick, DataFlash memory chip, piezo-electric beeper, and coin battery.



mind. However, one could easily substitute a microphone and op-amp for the player, creating a wireless microphone. The small size, digital tuning, and crystal-controlled transmitting stability make this chip transmitter a natural for this purpose.

Audio Signal Generator

As small as it is, the Butterfly's processor is overkill for this application. With all that extra processing power available, I chose to add a simplistic audio signal generator to aid in testing the transmitter. Although one could use the D-to-A converter or PWM to generate a signal, I chose to generate a square wave by toggling a spare output bit high and low at a 1 kHz rate. A simple RC filter removed many of the high frequency components, essentially turning the square wave into an attenuated triangular wave. This signal was then fed into the audio input ports on the transmitter. Having a stable audio signal facilitated testing the rest of the project. Being able to create the signal from within the project itself was both fortuitous and convenient.

It is desirable to turn the audio signal generator on and off as needed. When on, a small amount of the 1 KHz signal can leak into the transmitter even when it is disconnected from the audio inputs. It should be off, therefore, when the unit is connected to a real audio source. Pushing the joystick inwards usually stores the new frequency in memory for later use. However, if the frequency is set to 108 MHz (the top of the FM radio band), the unit toggles the audio signal generator on and off when the joystick is pressed inwards.

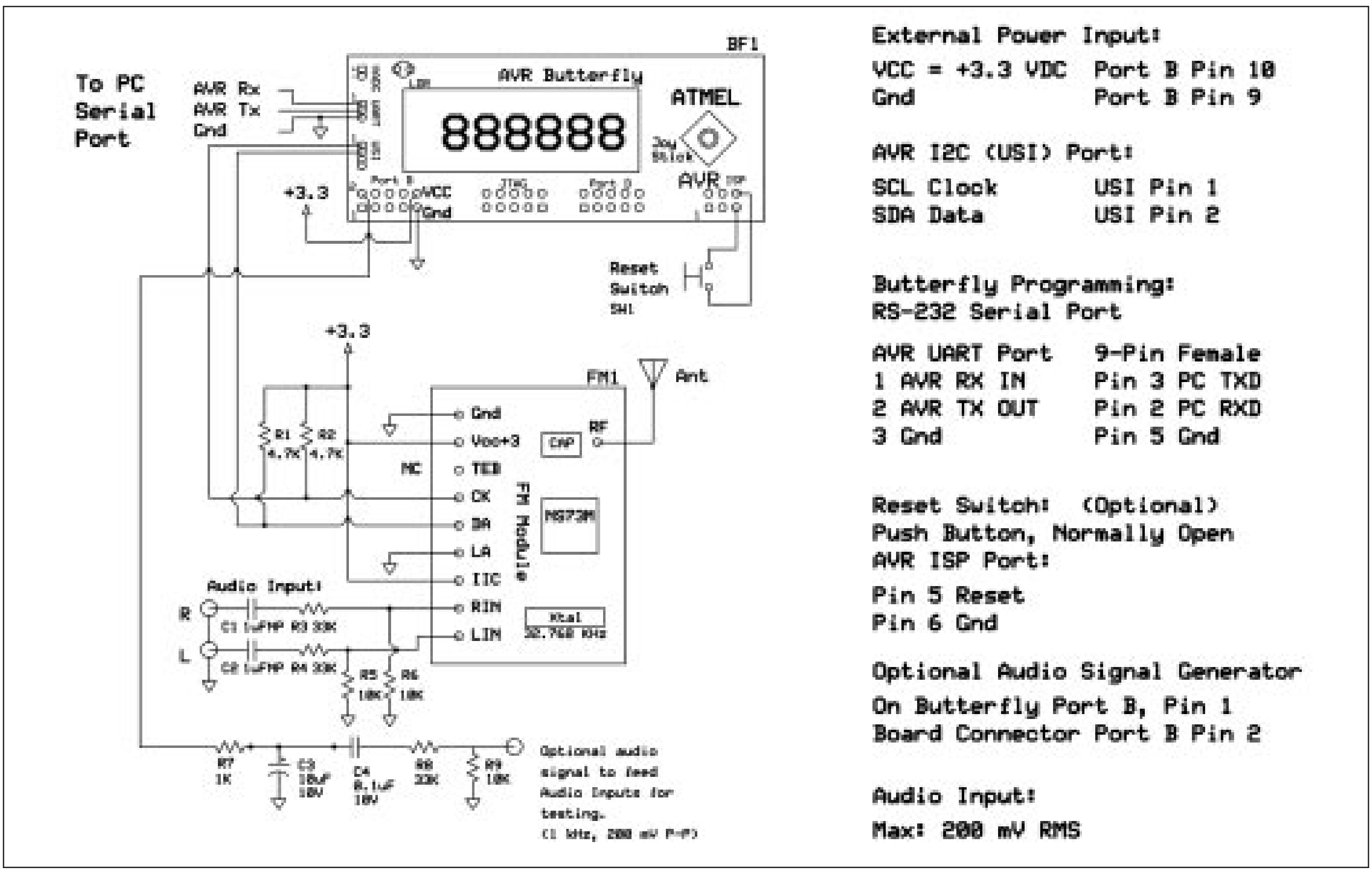
Construction Concepts

Figure 1 is the schematic for the project. Given the small number of components involved, one can lay this circuit out on a breadboard in a matter of minutes. I soldered a row of header pins to the FM transmitter module to allow me to insert it directly into a breadboard, although wires could be use as well. The audio input connections can be tied to whatever jack best fits your needs. A mini-stereo jack is ideal, but recall you will need a matching cable to plug into your MP3 player.

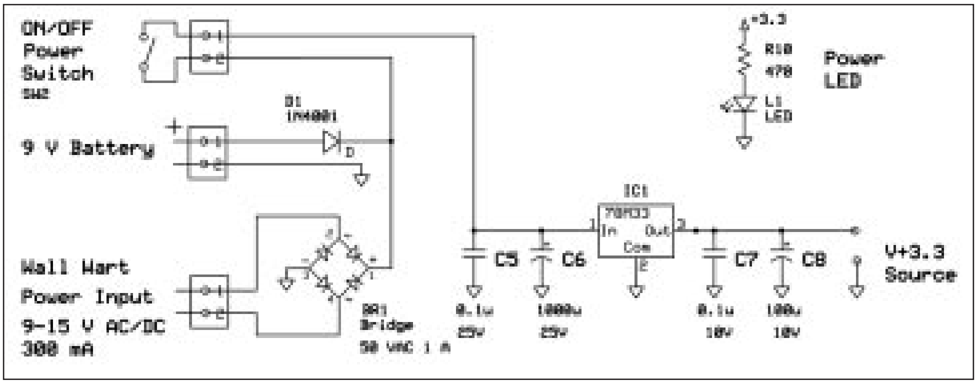
A short wire is soldered directly to the antenna pad on the module. The crystal and power supply filtering capacitor for the transmitter chip are already provided on the module. Be sure to order them separately if you order the chip without the board for your own design. There are two pull-up resistors for the I^2C clock and data lines.

The Butterfly is easily programmed using its on-board

■ FIGURE 1. The Butterfly Broadcaster schematic. Currently transmitting on 102.1 (MHz, FM radio band). An optional audio signal generator produces a 1 kHz tone. Following filtering and attenuation, it can be used as an audio source for setup and testing.



■ FIGURE 2. A generic 3.3V DC power supply for the Butterfly Broadcaster can use either a 9V battery or a wall wart source.



RS-232 serial port. The transmit, receive, and ground lines should go to a small, nine-pin, female RS-232 connector. This can then plug into a standard PC serial port. An optional push button reset switch is shown in the schematic. It is useful if you wish to experiment with the Butterfly and are frequently resetting the system. For a stand-alone transmitter, it is not required.

Power Supply

The Butterfly board includes a three volt coin cell. This works well for experimenting with the system, but is not designed to provide sustained power to the transmitter chip. For this reason, an external power supply is recommended. The Butterfly board is designed to use an external power supply from 3.1 to 4.5V DC. Do NOT connect it to a 5V power supply. Although the ATMega169 chip can be powered by 5V, the board's DataFlash chip and the NS73M FM transmitter chip are both designed for a maximum of 3.6V. The power supply, therefore, provides a regulated 3.3V for the circuit. You can feed the power supply with a 9V battery or a small 9-15V AC or DC wall wart. Diode D1 protects the circuit in the event that the battery gets connected backwards. The LED is a simple power-on indicator. A generic 3.3V DC power supply is illustrated in Figure 2.

Programming the Butterfly

The Butterfly board provides three separate connectors for programming it via an RS-232 serial port, a JTAG connection, or via an AVR ISP connection. The RS-232 serial port connection is simple, and does not require the use of any additional programmers. AVR Studio 4 is the PC software — available as a free download from the ATMEL website — which loads the program into the Butterfly. The Basic program is first complied generating a hex file, which is then downloaded into the Butterfly. This process is non-intuitive to those unaccustomed to the process, and warrants review. Download and install AVR Studio 4 following the prompts. Either hit the reset button on the Butterfly or cycle its power off and on. Doing so causes it to power-up running its internal "bootloader" program, awaiting the download from AVR Studio 4. Next, press the Butterfly joystick inwards, and hold it in this position while starting Tools/AVR Programmer in AVR Studio 4. A new AVRprog window will pop open, after which one can release the joystick button. On the new programming window, you can browse to the location of the hex file to download (BFFMTX.HEX). Then, hit the Flash Program button to initiate the download. Hit the Exit button to disconnect

PARTS LIST

ITEM	QTY	DESCRIPTION
❑ BF1	1	AVR Butterfly demo board
❑ FM1	1	FM radio transmitter module breakout board, with NS73M
❑ IC1	1	78M33, 3.3 V, 500 mA, fixed three terminal voltage regulator
❑ D1	1	1N4001 diode (50 V PIV, 1A)
❑ L1	1	Red LED
❑ BR1	1	50V 1A bridge rectifier
❑ R1, 2	2	4.7K, 1/8 W resistors
❑ R3, 4, 8	1	33K, 1/8 W resistors
❑ R5, 6, 9	1	10K, 1/8 W resistors
❑ R7	1	1K, 1/8 W resistor
❑ R10	1	470 ohms, 1/8 W resistor
❑ C1, 2	2	1 µF, 10V, non-polarized capacitors
❑ C3	1	10 µF, 10V capacitor
❑ C4, 7	2	0.1 µF, 10 V capacitors
❑ C5	1	0.1 µF, 25 V capacitor
❑ C6	1	1000 µF, 25 V capacitor
❑ C8	1	100 µF, 10V capacitor
❑ SW1	1	SPST mini-toggle switch, power on/off
❑ SW2	1	Mini-push button switch, uP reset

MISCELLANEOUS

❑	1	DB-9 mini female RS-232 connector
❑	1	Mini-stereo jack
❑	1	9V battery
❑	1	9V battery clip
❑	1	Wall wart power supply 9-12V AC or DC, 300 mA
❑	1	Case and mounting hardware Wire, solder, etc.

Digi-Key Corp — **www.Digikey.com**
SparkFun Electronics — **www.Sparkfun.com**

the Butterfly from AVR Studio 4 when the download and verify are completed. Finally, push the joystick upwards to start the program running. This is the process used when downloading any program to the Butterfly.

Additional Programs

The versatility of the Butterfly allows one to modify the provided transmitter program to incorporate revisions, modifications, and additional features. The provided transmitter program uses just a small portion of the memory available. When not being used as an MP3 broadcaster, you can also use it as a general-purpose learning platform for microcontrollers and programming.

The Butterfly board comes with a preloaded program which allows you to display scrolling messages on the LCD, play simple music through its piezo-electric beeper speaker, display the temperature using its on-board thermistor, provide a clock/calendar display, or measure and display an external 0-5 volt DC signal (voltmeter). This program is erased when loading the FM transmitter program.

You can download this original program from the ATMEL website and download it to the Butterfly using the above technique, restoring the Butterfly to its original state. You can also use AVR Studio 4 to enter and download assembly language programs to the Butterfly. Best of all, however, Bascom-AVR has a free, demonstration version available. It is limited to 4 KB of code. You can write short programs in Basic, compile them, and download them as above.

Smaller is Better

Although the Butterfly board is only the size of a credit card, this project could be miniaturized further. The Butterfly platform is perfect for providing a low-cost, small integrated processing package with which to learn the nuances of the NS73M transmitter. Given the transmitter chip's size of 7 mm x 7 mm, and equally small processors, the size of the display and the battery capacity become the size-limiting factors. If one chose to incorporate this chip within an MP3 player, these two factors also vanish, and today's MP3 players suddenly become obsolete! Where is a venture capitalist when you need one? **NV**

About the Author

When not practicing emergency medicine, Jay can usually be found tinkering with chips and an oscilloscope, or on the air as KD8HKD.