# Repeater Controller

*Author: Mark Walker*
*Circuit Graphics, Inc.*
*Hillsboro, OR*
*email: mmwatcgi@aracnet.com*

To get into truly REMOTE control, try sitting 7000 feet up a remote mountain, with commercial power running miles above ground and then more miles under ground to get to the site. Know that at any time day or night, stormy or fair you could be called upon to work reliably to help save the lives of those lost hunters, climbers or hikers, or those hurt in natural disaster, floods, fires or wind. Know that even if the commercial power is lost, you must keep on working reliably.

This is the job of those radio repeaters, usually installed and maintained by radio amateurs to fulfill the public service needs of their communities.

Here in the Northwest it is not unusual for this situation to be repeated many places, with service areas of some of these repeaters extending for hundreds of miles. What happens if the electricity is lost, and how does the repeater inform it's users that the AC power is down and that the batteries are still OK? Or not?

This project is a simple repeater controller that listens for activity from the receiver and turns on the transmitter to repeat the incoming signal; controls the system in case of a "stuck" transmitter microphone switch; add the system call sign to the transmitter audio every ten minutes of use for proper identification; and in this case, indicates the loss of AC power by adding a beep to the end of every transmission and adds the battery voltage and local temperature to the end of the identification information.

This makes the battery condition available to every user. This encourages reduced usage when on battery power so emergency information is not lost due to the surprise shutdown of power due to a discharged battery.

Since battery power is always limited, this circuit should add as little drain as possible to the system, saving all available energy to carry out the required radio communications. Therefore it is implemented by using interrupts, the watch dog timer, a power saving regulator, a trick to reduce drain for the temperature sensor, an extremely low current drain rail-to-rail op-amp and the ability to run its timers while sleeping at its low power state.

The following discussion refers to the schematic.

An interrupt is used to inform this baby A/D PICmicro™ that a carrier (a radio signal) has been detected at the receiver, allowing it to wake up, turn on the transmitter and tend to business. This uses the interrupt available on GP2.

The watch dog timer is used otherwise to wake up the PIC12C671 and have it run its timers for identification and for carrier time out. This system also adds an initial delay to the first input signal before activating the transmitter, known in Ham circles as "anti-kerchunk" time, and it delays turning off the transmitter for a bit after the receiver goes quiet to see if a response is coming.

The power saving regulator (National's LP2951) is used to maintain the processor power with little extra current drain, while allowing the processor's voltage, and therefore the reference voltage to the A/D to be adjusted for accurate measurements.

The low-current drain dual op-amp (Maxim,s MAX407) pulls only 2.4 µA and drives its outputs rail to rail to maximize the use of the PIC12C671's A/D by driving it over its entire range. This range translates into measuring the battery voltage between 10 and 15 volts, and the temperature between -40°C and +60°C.

The temperature sensor (National's LM338) was selected because of its easy implementation and accurate results. It does however draw noticeable current and therefore should be turned off except when needed. To this end the temperature is measured only when the system transmitter is transmitting, with the sensor powered down at all other times.

The clever use of the watch dog timer as the timing clock, at first glance seems like a gross error, since the watch dog timer is quite temperature sensitive, and would therefore introduce great timing errors (up to 50%) over the operating temperatures. However, since this remote control circuit measures temperature, it can compensate itself for the bulk of those errors and come up with accuracy results that are more than acceptable for this use. The use of sleep mode, without losing the ability to run timers, (since no other timer runs in sleep mode with this chip) allows the freeing of the pins that would have been used by external oscillator connections, a trick made possible by the use of the PIC12CXXX family's built in 4 MHz clock. The use of the internal clock would normally take too much current, (well, to some 2 mA is a lot) but since the chip will spend most of its time asleep the need for a low speed (and therefore low current) oscillator is eliminated and in fact, probably allows an overall reduction in actual average current drain over the use of a slower clock (with the resultant loss of those pins for other functions). The six available I/O pins allow just enough I/O to do a complete full featured job of REMOTE control, not otherwise possible with an itty bitty 8-pin part.

The audio output on GP5 uses a software PWM generator to make tones for the end-of-carrier beep when on battery power, and for the CW coded identification and other information sent every ten minutes or so. When on AC power only the identification is sent, at a Morse code tone frequency of 750 Hz, and at a 15 WPM rate. When on battery power, all tones used change frequency with the battery voltage, so even if the users are not up on their Morse code they can still tell the general condition of the system battery. The use of PWM allows the generation of a nice sounding sine wave, not the raspy square wave generally made by simple microcontrollers, and allows the use of a simple RC filter to smooth the audio. The PWM capability is increased by using the 4 Mhz clock rate, and the resultant one million instructions a second made possible as explained above, which allows for much better audio with a much simpler filter than would be possible if a much slower clock rate had been necessary to conserve power.

The battery voltage measuring circuit actually measures the 5 volt range between 10 and 15 volts, using a clever but simple technique to get this range into the 0 to 5 volt input of the A/D. Basically, it divides the input voltage by three referenced to ground, then multiplies it by three referred to the 5 volt reference used by the A/D. So 15 volts divided by three gives 5 volts, referenced to 5 volts gives 0 volts difference, multiplied by three gives 0 volts difference, subtracted from 5 volts gives 5 volts into the A/D. Likewise, 10 volts divided by three gives 3.333 volts, referred to 5 volts gives 1.667 volts difference, multiplied by three gives 5 volts difference, subtracted from 5 volts gives 0 volts into the A/D. And a last example, a fully charged 12 volt battery should read 12.68 volts, so 12.68 volts divided by three gives 4.22667 volts, referred to 5 volts gives .77333 volts difference, times three gives 2.32 volts difference, subtracted from 5 volts gives 2.68 volts into the A/D. So for battery voltage all that must be done is to add 10 volts to the value read by the A/D, which simplifies to sending a leading one before the A/D value. The A/D reads in 20 mV increments (with the reference voltage of exactly 5.12 volts, not quite the value used here but close enough) and is calibrated by inputting 12.68 volts and setting the CAL pot in the power supply for the correct reading out of the controller.

(To initiate the CALibrate mode, a jumper is placed across the CAL pins connected to GP4, at power up, which causes the controller to continually measure and report the voltage and temperature values.)

The A/D value could be simply multiplied by two (RLF command) and the 16-bit answer used to get the output data from a lookup table (the A/D value for 2.68 volts is 134 decimal, times two equals 268), an approach ideal to BCD type readouts. However since we are sending code the original 8-bit value can be used to derive the Morse code from the lookup table, with returned values incrementing in 20 mV amounts. For greatest accuracy the battery voltage is measured after the end of each transmission, eliminating the errors caused by system wiring resistance voltage drop due to the transmitter current, and reducing the influence of any residual battery voltage errors caused from recent charging.

The temperature sensor and its amplifier use a similar technique to spread the desired temperature range over most of the 5 volt input range.

The AC-on monitor at GP3 will run on AC inputs from 5 volt RMS to actual line voltage (110), but in most cases would tap the 12 volt AC at the rectifier diodes used in the battery charger for the system. (Some system supplies have internal diode switchover circuitry and an AC power good output line which makes this even easier).

An excess of filtering is used on all inputs and outputs, to allow use in RF environments without ill effects. Some filtering is also added to each A/D input to smooth noise, and simplify the need for much averaging of these readings.

This project could lead to application notes discussing:

- Software PWM in a practical application AC line monitoring techniques.
- Shared usage of pins to allow powering down circuits or components, or for little used functions like initial calibration.
- Simple way to level shift into the A/D from higher voltages.
- Expanded usage of the WDT in other timers (running of timers while asleep) by correcting for temperature errors.

This project shows a few of the possibilities offered by this the smallest member of the A/D Microchip Family.

 