Cascade Caver

Newsletter of the Cascade Grotto of the National Speleological Society

July 1998 Volume 37 No. 5

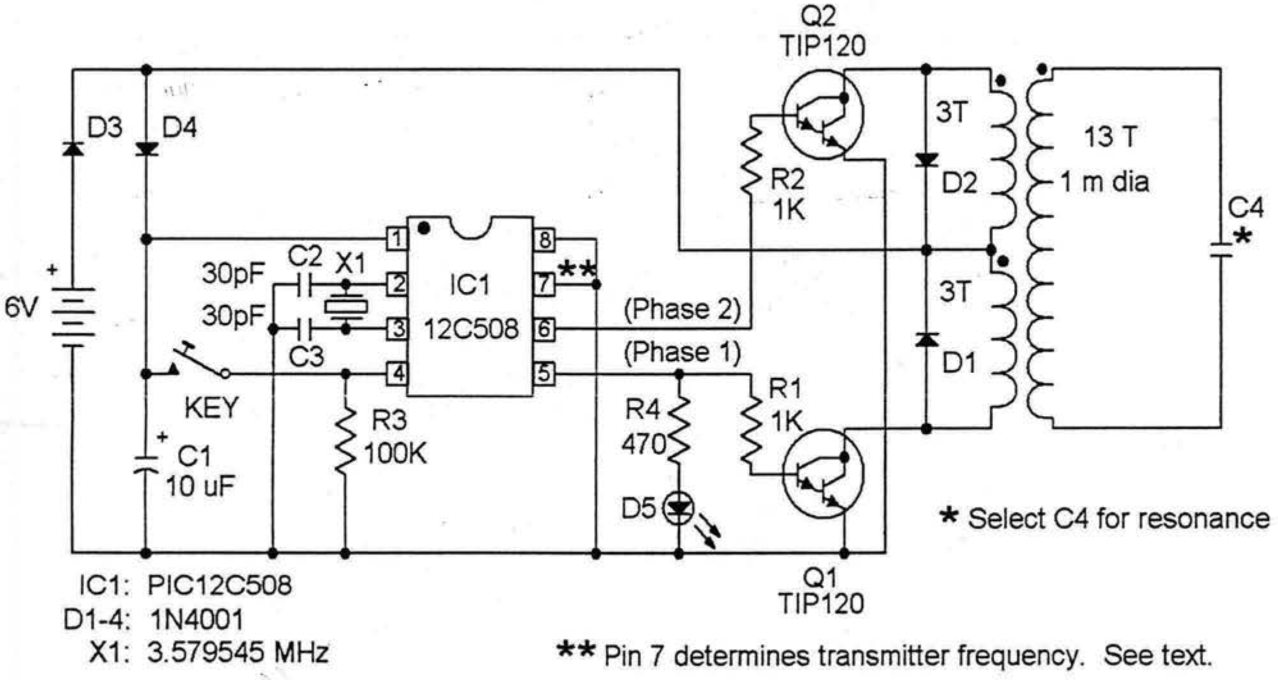


Figure 1. Schematic for a Simple Cave Radio Transmitter

Cascade Caver

ISSN 0008-7211

The *Cascade Caver* is published approximately 10 times a year by the Cascade Grotto, a local chapter of the National Speleological Society. Any material in this newsletter that is not copyrighted by an individual or another group may be copied or reprinted by internal organizations of the National Speleological Society provided that credit is given to the author and to the *Cascade Caver* and that a copy of the newsletter containing the material is sent to the Editor. Other organizations must contact the Editor.

Opinions expressed in the *Cascade Caver* are not necessarily those of the Cascade Grotto, the editors, or the NSS. Unsigned articles may be attributed to one of the editors.

All material to be published, subscription requests, renewals, address changes, and exchange publications should be sent to the Grotto address.

GROTTO MEMBERSHIP

Membership in the Cascade Grotto is $15.00 per year. Subscription to the *Cascade Caver* is free to regular members. Membership for each additional family member is $2.00 per year. Subscription to the *Cascade Caver* is $15.00 per year.

GROTTO ADDRESS

Cascade Grotto; PO Box 75663; Seattle, WA 98125-0663. This post office box should be used for both the grotto and for the *Cascade Caver.*

GROTTO OFFICERS

Chairman: Mark Sherman (206) 365-5386
Vice Chairman: Bruce Frank (425) 746-2928
Sec/Treasurer: Lane Holdcroft (206) 783-6534

OTHER POSITIONS

Trip Coord. Jim Harp (425) 745-1010
Librarian: Bill & Christine Bennett (425) 255-1466
Regional Rep. : Ben Tompkins (206) 546-8025
Editors Paul Ostby (425) 823-5107
email: postby@iname.com
* Mark Sherman (206) 365-5386
email: mas@tc.fluke.com
Steve Sprague (360) 387-3162
email: ssprague@whidbey.net
* Editor for the current issue.

MEETINGS

Regular grotto meetings are held monthly at 7:00 p.m. on the third Friday of each month at the University of Washington, Room 119, in Johnson Hall. Please see the map on the back cover of this issue.

UPCOMING EVENTS

Please notify Jim Harp at (425) 745-1010 of any trips.

July 17	Grotto Meeting 7:00 p.m.
July 18 - Aug 15	Kris Esterson <kae7077@garnet.acns.fsu.edu Ketchicave 1998. The expedition will be held on Kosciusko Island, SE Alaska
July 18-19	Wendel Pound PSG 253-863-1649 Three Sinks and Falls Creek. Camping and caving.
Aug 8-9	Wendel Pound PSG 253-863-1649 Ape cave clean up and general caving.
Aug. 21	Grotto Meeting 7:00 p.m.
Labor Day	Jennifer Dorman GSG 208-321-1239 Papoose Cave
Sept 4-7	Steve Knutsen WVG 503-695-2659 Labor day weekend in the Marble Mtns.
Sept 18	Grotto Meeting 7:00 p.m.
Sept 19	Jerry Thompson CG 360-653-7390 Windy Creek Cave

COVER

This month's cover is the schematic for Paul Ostby's cave radio. Please see the article on page 26 of this issue.

Late Dues

According to the Treasurer's records, the people listed below still have not paid their dues for 1998. Please contact Lane Holdcroft at (206) 783-6534 if you have any questions.

Bob Brown
Scott Davis
John Green
Greg Passmore
Jerry Thompson

Rod Crawford
Rob & Kris Fariss
Ed Lisowski
Tony Taylor
Jeff Wilson

The Walls Are Alive

By Charles Petit

Ed. Note: This article was taken from the February 9, 1998 issue of U.S. News & World Report.

For centuries and perhaps millenniums, a sulfurous cave near the small town of Tapijulapa in southern Mexico's state of Tabasco has been regarded by locals as the home of powerful spirits. Fertility rituals are still conducted there, the largely Mayan people feasting on tiny fish that thrive in milky-white waters flowing through the cavern's passages and out its entry.

Now it appears that the folklore is echoed eerily by scientific fact. Earlier this month, a 22-person expedition donned breathing masks for protection from acid fumes and ventured a mile deep inside the cavern, Cueva de Villa Luz, or Cave of the Lighted House, about 80 miles south of the Gulf of Mexico on the edge of the Chiapas highlands. It returned with an extraordinary confirmation that the walls of the limestone cavern are genuinely alive - lined, in many places, with a layer of microbial mucus as much as half an inch thick.

The researchers, nearly all from the United States, found the dark, damp world occupied by a profusion of larger creatures, too - close relatives of surface organisms. But anchoring it all are immense colonies of yet-to-be-identified microbes. The single-celled creatures appear to be expressions of a skein of life that continues, in the pores of near solid rock, down many miles. And its bizarre microbes, like other samples of life recently found in places traditionally thought to be too hostile for any organism, may give clues to how to find life on other planets.

Microbes have been found in other caves but not in such profusion. Most geology textbooks say non-biological processes form limestone caves. Weak carbonic acid created as rainwater reacts with carbon dioxide in the air and minerals - the story goes, has slowly etched vast, natural hollows. More recently, earth scientists have begun to suspect that living sculptors, microbes, made those immense chambers, acid drop by acid drop.

Proof for that new theory has been hard to come by, though. In most caves, the bacteria that did the job would have died long ago. Some vaguely microbial forms have been found in North America's deepest cave, Lechuguilla, in New Mexico's Carlsbad Caverns National Park. Its nearly 1,600-foot labyrinths contain strange, fossilized structures that resemble looped string cheese and may be microbial in origin. In Villa Luz, on the other hand, "we see the same things, but they are still alive," says Louise Hose, the expedition leader and a professor of geology at little (650-student) Westminster College in Fulton, Mo. Hose has been an avid speleologist, or caver, for 20 years. Reports that not only is Villa Luz teeming with life but its microorganisms drip sulfuric acid drew her like a magnet. In the cave, she says, she was seeing Lechuguilla as it may have been millions of years ago.

"Snot-tites"? The cave's microorganisms form viscous colonies that hang like slender, pale, glistening stalactites from ceilings and overhangs. The proper term for such things is microbial veil; the researchers dubbed them snot-tites. The walls crawl with spiders and other invertebrates adapted to the foul fumes. Near the entry and in fresh-air pockets, four kinds of bats, including vampires, hang overhead and flutter about, dropping guano that squirms with maggots. A lot of the travel inside the cave is through waist-deep water, on the bottom of which is black sulfurous mud a foot deep, overlain by inches of living slime. The small fish swim everywhere, in an abundance not seen in other caves. One of the explorers, Penelope Boston, a microbiologist at Complex Systems Research Inc., a Boulder, Colo., company that does government contract research, says she emerged from Villa Luz feeling almost as though she had not entered Earth but left it. "I have always been in love with space exploration," Boston says, "and this is the closest I am going to get."

While organic material falls in from above, the cave's primary food chain starts with the immense colonies of unusual microbes. They represent scores of species, most of them new to science. Cut off from sunlight, they cannot photo-synthesize. The energy that fuels the bacteria derives from hydrogen sulfide and other inorganic chemicals rising from deeper below. The drizzle of pure sulfuric acid is just one product of the microbes' stygian metabolism. "The cave is a spectacular example of an ecosystem driven not by light but by inorganic chemistry," says Norman Pace, a University of California microbiologist who plans to study many of the organisms the researchers found. "It is an interface of two worlds, the one on the surface that uses oxygen and eats organics, and the stuff down below."

The cave discovery also comes as evidence mounts that Earth's crust is shot through with microbes to great depths. In 1992, Cornell University astronomer Thomas Gold wrote a paper titled "The Deep, Hot Biosphere," which served as a spur for much of today's deep biology search. In it, he estimated that if all life inside the Earth were hauled to the surface from as much as 3 miles down, it might form a layer 5 feet thick. It would weigh more than all the life now on the surface. And that subsurface and barely explored world, scientists like Pace believe, is the main stage for the planet's bio-diversity. Genes of some of these extremophiles, as scientists call microorganisms living in extreme environments, suggest they represent dozens of kingdoms of life. Outwardly similar through a microscope, at the molecular level they are as different from one another as animals are from plants or fungi. They may contain enzymes or other biological products with medical or other uses beyond the ability of chemists to invent on their own.

Government backing for such research includes a new "LExEn" program, for Life in Extreme Environments, which the National Science Foundation set up to pave the way for a search for life on other planets. So far, it has granted money for researchers to examine organisms found in Antarctic ice, in boiling hot springs, in vents on the seabed, in rock drilled from volcanic formations more than 10,000 feet deep, and in the utter dark of caves such as Villa Luz. Scientists don't know what life on other planets is like, but they can practice looking for it by first seeing how cleverly life hides on Earth.

It will take months to years, and more expeditions, to begin understanding how the cave's many organisms interact and their exact relationships to life elsewhere. Hose expects to return in April, at the time of the next fertility ceremony. And, she adds, the cave yielded benefits to more than speleology and the science of microbial veils. After the last trip, "we all noticed that our complexions were clearer," Hose says, The air inside apparently acts like a beauty parlor acid peel.

A Microcontroller-Driven Cave Radio

By Paul Ostby

[This article, with some differences, has been awaiting publication in Speleonics.]

"You use a computer for the transmitter?" Lane Holdcroft asked this with some surprise. Lane is a fellow caver and a fellow electrical engineer.

I said "Yeah. It keeps the hardware simple."

Lane was not the first person to express surprise at my cave radio transmitter design. But a micro-controller does make the circuit simple. Look at the schematic in Figure 1 (see the cover) and you'll see what I mean.

Why use a microcontroller?

A microcontroller is a small self-contained computer. The circuit in Figure 1 uses a PIC-12C508 microcontroller from Microchip Technology. This part packs a lot in an 8-pin package: a crystal oscillator; a RISC CPU; 512 words of PROM; 25 bytes of RAM; and more. And the part costs less than US $2 apiece.

After it is programmed with the appropriate instructions, this 8-pin chip can take care of most of the tasks commonly needed for a beacon transmitter:

1) Divide the crystal frequency down to the chosen output frequency.
2) Allow continuous tone, auto-keyed, or hand-keyed (Morse code) modes of operation.
3) Shape the output waveforms for a switch-mode antenna driver.

The only other active devices are the two power transistors to drive the antenna coil.

Implementation details.

Pins 1 and 8 are Vdd (power) and ground, respectively. The microcontroller requires a supply voltage in the range of 2.5 to 5.5 volts. Diodes D3 and D4 protect from battery reversal and drop the voltage at pin 1 to something below 5.5V. If you use a battery voltage above 6.7V (e.g. a gel cell at 7.2V, or a 12-volt battery) then replace diode D4 with a 3V or 5V voltage regulator.

Pins 2 and 3 are the oscillator pins and connect to the quartz crystal. Crystals up to 4 MHz can be used.

The remaining pins are directly under program control. The program I've written uses pins 4 and 7 for inputs, and pins 5 and 6 for outputs.

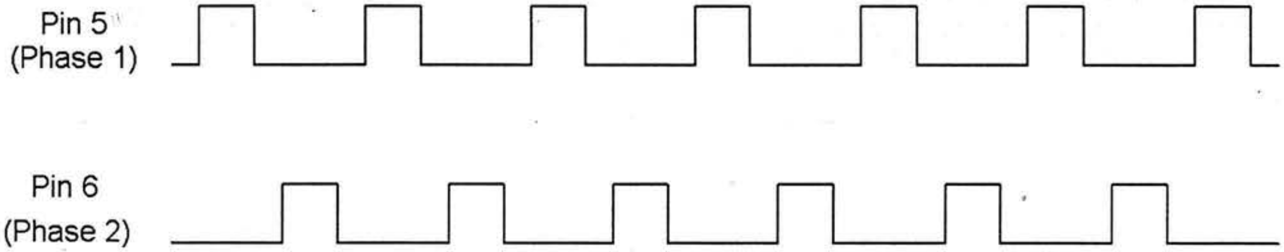


Figure 2. PIC12C508 Output Timing Diagram

Pin 4 can connect to ground, to Vdd, or to a telegraph key with a pull-down resistor. The software monitors the voltage on this pin to deter-mine which transmitter mode to use. While pin 4 stays at ground the transmitter is in auto-keyed mode and modulates the carrier at 1.7 Hz. If pin 4 is tied to Vdd the transmitter stays in continuous tone mode and the carrier is unmodulated.

If you use a telegraph key, the microcontroller starts up in auto-keyed mode. If you hold down the telegraph key for 5 seconds the transmitter switches to hand-keyed mode. In this mode the transmitter output is only enabled while the telegraph key is pressed. This allows the beacon operator to send Morse code. To return to auto-keyed mode simply hold down the telegraph key for another 5 seconds.

The software checks the voltage on pin 7 to deter-mine which output frequency to use. Four possible frequencies are available. The frequency is selected by connecting pin 7 to pin 1, 5, 6, or 8. For more details on how this is done see the software description below. The four available frequencies are: 873.91, 3276.8, 3495.6, and 3576.0 Hz. Other output frequencies are possible by changing the crystal frequency, or by changing the software.

Pins 5 and 6 provide two non-overlapped output phases. See the timing diagram in Figure 2. The two output phases work well with a center-tapped drive coil. If a single, non-center-tapped drive coil is used then you can leave pin 6 unconnected and get rid of R2, Q2, and D2.

An output duty cycle near 33% greatly reduces the third harmonic and makes the job of filtering easier. So the software tries to keep the output duty cycles as close as possible to 33%.

The transistors Q1 and Q2 are operated as switches. They are either saturated or off. The schematic calls for power transistors but in fact they generate very little heat during operation. You may be able to get by with a darlington in a smaller package.

The Software.

The program listing is too long to be included with this article. I have included some sections of the code to illustrate how the software works. The full source code is available via email from the author.

At startup the device must determine which output frequency to use. See the code fragment in Listing 1. Note that a semicolon (;) marks the start of a comment.

Listing 1 shows eleven instructions plus some comments. The purpose of this section of code is to set the "Frequency" variable to a number from 0 to 3, depending on which output frequency is chosen.

The code first sets pin 5 high and tests pin 7. If pin 7 is high at this point then the program sets bit 0 of Frequency (changing Frequency from 0 to 1.)

The program then resets pin 5 low, sets pin 6 high, and tests pin 7 again. If pin 7 is high at this point then the program sets bit 1 of Frequency.

Finally, pin 6 is brought back low.

The end result is this: If pin 7 is connected to ground (pin 8) then pin 7 always tests low and Frequency = 0. If pin 7 is connected to pin 5 then Frequency = 1. If pin 7 is connected to pin 6 then Frequency = 2. And if pin 7 is connected high (to pin 1) then pin 7 always tests high and both low-order bits of Frequency are set, so Frequency = 3.

The "Frequency" variable is later examined to determine which program loop to use in order to output the desired frequency.

Before looking at Listing 2 it will help to review the instruction timing for this chip. Each instruction cycle takes four oscillator cycles. So a 3.5795 MHz crystal lets the microcontroller execute about 894,880 instruction cycles per second. All instructions execute in one instruction cycle except program branches, which take two instruction cycles.

If you have a 3.5795 MHz crystal and you want an output frequency of 3,495.6 Hz, you need to divide the clock by 1024. 1024 clock cycles divided by four is 256 instruction cycles. Hence a program loop which takes 256 instruction cycles will execute 3,495.6 loops per second.

Listing 2 shows the program loop which outputs at 3,495.6 Hz. This loop always takes exactly 256 instruction cycles each time through the loop. Remember that the desired output duty cycle is 33%. So phase 1 goes high for 85 cycles (=256/3) then both phases are low for 43 instruction cycles. Then phase 2 goes high for 85 instruction cycles. Then both phases are low for another 43 cycles. Then the loop repeats. The total time taken is 256 instruction cycles (= 85 + 43 + 85 + 43). The output duty cycle is 33.2% (=85/256).

The fragment of code in Listing 2 is deceptively simple. Remember that the entire loop always takes exactly 256 instruction cycles to execute.

Each instruction has a comment at the end of the line documenting how many instruction cycles have gone by in the loop so far. The first instruction is a "btfsc" instruction and it takes up instruction cycle number 1. Similarly the second instruction is a "bsf" instruction and it accounts for instruction cycle number 2. The first instruction tests a bit flag to see whether the transmitter should be sending right now. If sending, the second instruction switches on the pin 5 (Phase1) output. If not sending then this output is left off.

The third line says "kill 84". "Kill" is not a single microcontroller instruction. It is a macro that I have defined earlier in the program. The "kill" macro is complicated, but think of it as expanding into the specified number of "nop" (no-operation) instructions. The sole purpose of the "kill" macro is to use up the specified number of instruction cycles without doing anything useful. So "kill 84" does nothing in particular, but it takes up instruction cycles numbered 3 through 86.

The fourth line says "clrf GPIO". This instruction resets the pin 5 output. This takes place during instruction cycle number 87. So the pin 5 output switches on (if enabled) in the middle of instruction cycle number 2 and switches off in the middle of instruction cycle number 87. Hence the pin 5 output stays on for 85 cycles (=87-2).

We'll gloss over the next few lines. They provide a delay of 43 cycles, then pin 6 output goes high for 85 instruction cycles.

Pin 6 is brought back low during instruction cycle number 215. Next comes a "call" instruction which calls a subroutine named "KeyCheck". The Key-Check subroutine is somewhat complicated but it is guaranteed to always take exactly 31 instruction cycles to execute, including the call and return time. This line in the program accounts for the 31 instruction cycles numbered from 216 to 246.

The KeyCheck subroutine checks the telegraph key and decides whether you are in continuous, auto-keyed, or hand-keyed mode. KeyCheck sets the OutputEnable bit in the Flags variable whenever it decides that the transmitter should be sending.

The KeyCheck subroutine is somewhat complicated because it must: de-bounce the Morse key; determine whether we're in continuous, auto-keyed, or hand-keyed mode; set the OutputEnable bit appropriately; and handle transitions from one mode to the next depending on how long the Morse key has been held down. And in all cases it must take exactly the same amount of time to execute. Use of a state table makes this subroutine easier to write and maintain.

Back in the main loop you'll see that after the call to KeyCheck we kill some more time until it is time to jump back to the start of the loop. The "goto" instruction at the end of the loop takes two instruction cycles and accounts for cycles numbered 255 and 256.

Other output frequencies.

Other frequencies are produced by program loops with a different number of instructions in them. For example the 873.91 Hz output is generated by a loop taking 1024 instruction cycles.

But what about other frequencies? My own radio-location transmitters have all used 3,276.8 Hz, a frequency suggested by Frank Reid (Reid 1985). Using this circuit with a 3.5795 MHz crystal the program loop should take 273.1 instruction cycles.

In this case we could use a loop which takes 273 instruction cycles to execute. But this gives us an output frequency of 3,278.0 Hz. This error of 1.2 Hz is too large for some of the extremely narrow-bandwidth receiver designs in use today. (See for example Pease 1997). How do we get closer to the right frequency?

One way is to choose a different crystal. 3.93216 MHz is a common and inexpensive crystal. Using this crystal and a program loop taking 300 instruction cycles we could get exactly 3,276.8 Hz for the output.

But there is another possibility. We cannot write a program loop with 273.1 instructions, but we can write a combination of loops which average out to 273.1 instruction cycles per loop. For example we could execute a 273-instruction loop nine times out of ten. Then the tenth time we execute a loop which takes 274 instruction cycles. Over the course of ten loops we will average 273.1 instruction cycles per loop. The output frequency will be within 0.1 Hz of our target (an error of 31 ppm). This is better than the tolerance of a cheap quartz crystal.

The software available from the author contains a program loop which averages 273.1 instructions. This is the loop used for the 3,276.8 Hz output.

This averaging technique allows us to divide the clock crystal by non-integer numbers to get the exact output frequency we want. We increase the harmonic content slightly using this technique. But harmonic filtering is usually not a problem for cave radio transmitters.

David Gibson suggests that 3,575.97 Hz might be a good choice for a radio-location beacon (Gibson 1996). This frequency is generated by a program loop which averages 250.25 instruction cycles.

Those familiar with direct digital synthesis (DDS) might suggest using phase accumulator techniques instead. The problem with using a phase accumulator in a microcontroller is that it can take 10 or more instruction cycles per accumulation loop. So the output jitter will be 10 or more instruction cycles instead of one instruction cycle. This will contribute substantial noise at undesired frequencies. But users of underground beacon transmitters seldom need to worry about radiated harmonics. So this may be a good option, especially for a transmitter which must be tunable over a broad range of frequencies.

Another option worth mentioning is to use a different microcontroller with a different number of clock cycles per instruction cycle. An obvious choice is the new AT90S1200 chip from Atmel. This US $3 chip boasts one instruction cycle per clock cycle. Using a color-burst crystal, this chip gives us 3,579,545 instruction cycles per second compared to 894,886 with the Microchip part. Using the AT90S1200, a 3.5795 MHz crystal, and a program loop taking 1001 instruction cycles we could get an output of 3,575.97 Hz without loop averaging.

Availability.

The source code is available via email at no charge. You can contact the author at:

postby@iname.com

In case of typographic errors or a change of email address, you should be able to contact me through the Speleonics list.

The microcontroller was available from Digi-Key for US $1.88 each at the time I wrote this in June 1997. The Digi-Key part number is PIC12C508-04P-ND. No doubt the part is available elsewhere for a similar price.

Simplifying further.

The circuit in Figure 1 can be simplified in some cases. As mentioned above the center-tapped antenna coil can be replaced with a single coil with some loss of signal strength. In this case you can get rid of R2, Q2, and D2. Just leave pin 6 of IC1 unconnected.

If you do not need or want to send Morse code, you can get rid of R3 and the telegraph key. Just strap pin 4 to pin 8 for auto-keyed mode. Or strap pin 4 to pin 1 for continuous mode.

You can get rid of R4 and D5 if you don't need a visual cue when the transmitter is running.

You can get rid of D4 if you always use 3 or 4 NiCd batteries (max 5.4V), or if you use only 3 alkaline batteries (max 4.95V). The only purpose for D4 is to drop the voltage to 5.5V or less.

You can replace Q1 and Q2 with power MOSFET devices such as IRLZ14 or NDP4050L. This will let you get rid of R1 and R2. Be sure to choose a logic-level MOSFET because the gate drive voltage will only be 4 or 5 volts.

Finally, you can even get rid of C2, C3, and X1 by using the internal 4 MHz RC oscillator. But note that this will require some program changes. Also note that the internal RC oscillator frequency may vary by more than 10% over the rated temperature and supply voltage ranges. So for precise frequencies you must use the external crystal oscillator with C2, C3, and X1

```
     ; Determine what frequency to use.  This will depend on
GP0.
     ; The following code will set the Frequency register to:
     ;      0 - if pin 7 is connected to ground.
     ;      1 - if pin 7 is connected to pin 5.
     ;      2 - if pin 7 is connected to pin 6.
     ;      3 - if pin 7 is connected to Vdd.
     clrf    Frequency          ; Zero frequency code for now.
     bsf     GPIO,Phase1        ; Bring pin 5 high.
     nop                        ; Wait for voltage to settle.
     btfsc   GPIO,FreqSelect
     bsf     Frequency,0        ; Pin 7 tied to pin 5 or Vdd.
     movlw   Phase2Mask         ; Bring pin 6 high, pin 5 low.
     movwf   GPIO
     nop                        ; Wait for voltages to settle.
     btfsc   GPIO,FreqSelect
     bsf     Frequency,1        ; Pin 7 tied to pin 6 or Vdd.
     clrf    GPIO               ; Reset outputs.
```

Listing 1. Startup code - Frequency determination.

```
F3496     ; Frequency = 2.  Use 3495.65 Hz.
     ; Instruction frequency / 256 = 3495.65 Hz.
     ; This loop must have 256 instructions.
     ; Each phase should be high for 85 instructions.
     ; (85 / 256 => 33.20% duty cycle.)
     ; The following sequence will be used:
     ;     Phase 1 on for 85 instructions.
     ;     Off for 43 instructions.
     ;     Phase 2 on for 85 instructions.
     ;     Off for 43 instructions.
     btfsc   Flags,OutputEnable      ; Instr cycle  1.
     bsf     GPIO,Phase1             ; Instr cycle  2.
     kill    84                      ; Instr cycles 3-86.
     clrf    GPIO                    ; Instr cycle  87.
     kill    41                      ; Instr cycles 88-128.
     btfsc   Flags,OutputEnable      ; Instr cycle  129.
     bsf     GPIO,Phase2             ; Instr cycle  130.
     kill    84                      ; Instr cycles 131-214.
     clrf    GPIO                    ; Instr cycle  215.
     call    KeyCheck                ; Instr cycles 216-246.
     kill    8                       ; Instr cycles 247-254.
     goto    F3496                   ; Instr cycles 255-256.
```

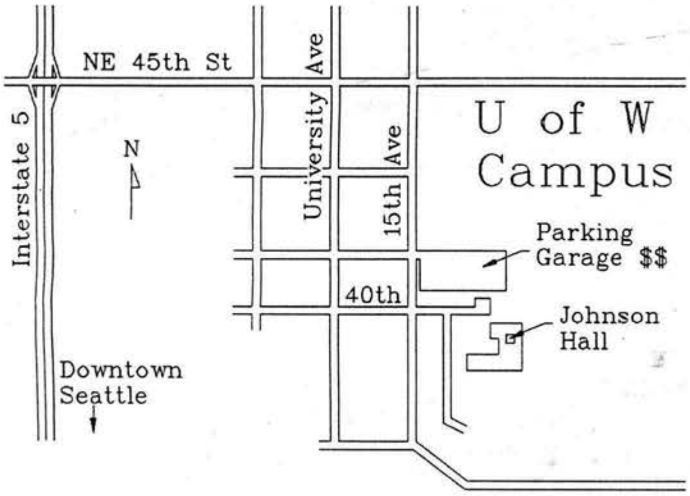
Listing 2. Program loop to output at 3,495.65 Hz.

References

Gibson, David (1996), *Choosing a Frequency for a Radio-Location Beacon*, JCREG 26, p 18.

Pease, Brian (1997), *The D-Q Beacon Receiver*, Speleonics 21, pp 10-19.

Reid, Frank (1985), *Thoughts toward designing an "International" cave radio: Choosing an operating frequency*, Speleonics 2, p. 11.



The Cascade Grotto meets at 7:00 p.m. on the third Friday of each month. **We are now meeting in room 119** in Johnson Hall on the University of Washington campus.

We look forward to seeing you at one of our meetings

Cascade Caver
PO Box 75663
Seattle, WA 98125-0663

Windy City Grotto
c/o Ralph Earlandson
802 S Highland Ave
Oak Park, IL 60304-1529

60304-1E