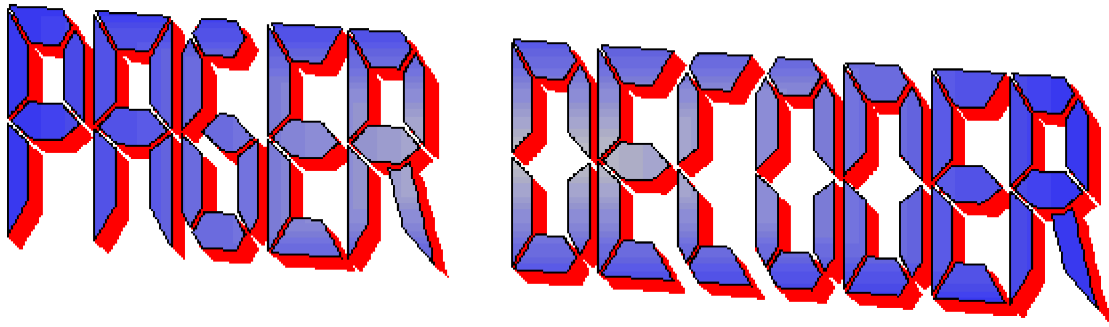




This project was done for my grade 11 Communications Tech course. I got the idea from an article in **Popular Electronics**. There were guidelines we had to follow, so this may not look quite like my other project pages. We chose our own projects, which were to consist of a proposal, technical report and a learning log. Those parts will be presented here. I admit that this proposal is not exactly my greatest literary achievement, but it served its purpose well.

Proposal

Communications Technology-Major Project Pager Decoder

For my major optional project, I have chosen to build a pager decoder. Pager transmissions are not coded. The reason it is called a decoder instead of just a receiver is because pager transmissions are done at non standard baud rates, often switching speed during the actual transmission. This alphanumeric transmission scheme is referred to as **POCSAG**. So, the decoder takes that signal and translates it into something that a PC can understand.

The project will consist of 3 parts; building, testing, then an explanation of pager transmission schemes.

Building

This is where the actual project is built. It will be assembled from a kit. I chose to purchase the kit instead of building it from scratch for a number of reasons. First, the decoder requires a custom programmed PIC. I don't have the equipment, and neither does CCH or JP2 to program PICs. I also don't know much...er...any assembly language. The second reason I am building from kit is because of the circuit board used. The decoder is built in a case no bigger than the casing of a standard 25 pin serial plug (which is the actual case). In order for it to fit, the board must be double sided, with traces about as wide as 22 AWG wire (really thin). There is no possible way that I can trace it out with a resist ink pen. I don't want to get into photo-sensitive boards and developing so the board that comes with the kit is the best choice. Third, the decoder needs a computer to work, and the computer needs software. I could write the software (in BASIC, with medium effort) if I had more time, but for now, I'll use the software that comes with the kit. The software is Windows based, meaning I can decode and log signals while I do other stuff. Besides, I just don't feel like hunting down all the parts, mostly because of the preprogrammed controller I need.

Testing

Testing shouldn't last long, no more than a week. The project is complicated and lots can go wrong, so testing is essential. Besides, the decoder is attached to the PC serial port. While it is true that the serial port is pretty hardy, it can still be damaged. I would hate to either fry an IO card on a school computer, or fry the IO controller on my motherboard. Also, since the traces on the board are so close together, solder bridges are easy to create. The testing will involve careful inspection of the board and parts placement. The power supply will also be checked to make sure that it is supplying the proper voltages to the proper components.

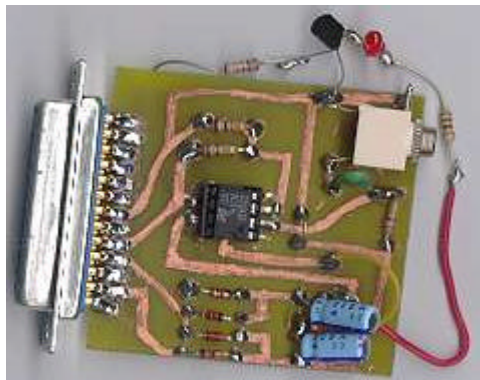
Pager Transmissions

This part will detail how pagers transmit information. There are many different transmission schemes, so there will be brief explanations, notes, etc. on each scheme.

I decided to get rid of the part about pager transmissions, and instead chose to build a different, much more simple decoder and then compare the two.

Once the project was built, I experimented and played around with it for a week or two, just to get a feel for the software. The simple Radio Monitor software is easy to install and configure (just copy the contents of the disk or ZIP file to the hard drive). The other, full fledged AccuPage software is more complicated, as it can also send pages (using the modem), selectively log entries, filter, etc. After that, I built the second decoder. The comparison is below.

Comparison

Decoder Kit	Other Design
	
Power: From serial port. Input: Buffered and filtered. Audio Output: Yes. Computer Output: Buffered and protected. Signal Processing: Op Amp (LF44CN) and PIC (PIC16C54). Visual Indicators: LED. Notes: None.	Power: From serial port. Input: Filtered (.1uf capacitor). Audio Output: No. Computer Output: No buffer. No protection. Signal Processing: LM741 Op Amp. Visual Indicators: LED (See Notes). Notes: Note the LED indicator. I added that as an after thought.

Comparison

As you can see from the chart above, the Decoder Kit has the advantage of more processing power, better filtering and being able to monitor the audio output of the scanner (this makes it easy to tell if you have tuned to the wrong frequency). Since the Kit also has more processing, it performs much better. In fact, I have yet to decode one single transmission with the 741 Op Amp Decoder. The Kit, on the other hand, decodes almost 100% of transmissions with no problems, even when the transmissions are at 2400 baud. The 741 circuit also loads down the processor, so the software cannot be run under Windows, which is a major annoyance. Since the Kit does a lot of its own processing, it places a much smaller load on the processor, allowing it to run under Windows. In fact, the AccuPage Radio Monitor software is a Windows program.

From the above comparison, I think it should be fairly obvious which decoder is better.....the Kit.

The next part of the project was the Technical Report. That is basically an outline of what was done, how long it took and any problems encountered.

Technical Report

Technical Report

Building

This went well. The Decoder Kit was built in about two weeks. That was longer than it should have taken, but there were many interruptions. There were no really big problems with the Kit. I did have a bit of trouble with missing parts because the Kit was always being transferred in and out of my bag, dragged around school, etc. The only really annoying thing about building the Kit was soldering the 25 pin serial plug. It was difficult to hold the thing still while soldering, as it was supporting the board (the school did not have one of those clamp-things.) I solved the problem by generously tinning the contacts, which created a very tight fit for the plug. I then forced it on and soldered the connections. After that, everything went fine.

The 741 Op Amp decoder was a different story. For that, I had to make the printed circuit board. I had most of the parts in my junk box. The board took about a half an hour to design, which is pretty long considering the simple circuit and how badly the traces are actually routed (take a look at the above pictures). It was not one of my best boards. Continuity checks showed that the board was perfectly workable. If you look closely at the picture, you can see some corrections on the board. Once the board was done and etched, the circuit was built. That went fast because it was such a small and simple circuit. It took two days but only because there was an incident involving a soldering iron, computer, wire, 120VAC and someone's hand.

Testing

This went fine for the Decoder Kit, except that I had some problems with the test computer (a P133....owned by the school, of course. You really didn't expect me to test it on *my* computer, did you?) locking up. The lock ups were caused by the network card messing with the IO card. Once these were fixed the thing worked perfectly. The only real problem was that I was running the Radio Monitor software from disk, which was much too slow to continuously update the log file. I ended up

playing with....er....testing the Kit for 2 weeks instead of one.

Here is a small sample of the log file:

```
04/05/97 16:21:48 B=P1200, A=0878583, F=1, msg = 234
04/05/97 16:21:48 B=P1200, A=1933832, F=0, msg = 85365711
04/05/97 16:21:49 B=P1200, A=0831168, F=0, msg =
04/05/97 16:21:49 B=P1200, A=0818745, F=0, msg = 5193524700
04/05/97 16:21:49 B=P1200, A=0942827, F=0, msg = 3534681
04/05/97 16:21:49 B=P1200, A=0016228, F=1, msg = New voice mail received.
04/05/97 16:21:49 B=P1200, A=1921002, F=0, msg = 7468293
04/05/97 16:21:49 B=P1200, A=0941437, F=0, msg = 675886991114314314314314
04/05/97 16:21:50 B=P1200, A=0830080, F=0, msg = -3914488
04/05/97 16:21:50 B=P1200, A=0868866, F=0, msg = 911
04/05/97 16:21:51 B=P1200, A=0019813, F=1, msg = 08 W704240796 CIBC EQUIP ACQ
04/05/97 16:21:52 B=P1200, A=0028893, F=1, msg = 09 W704240796 1 1 CIBC EQUIP
04/05/97 16:21:53 B=P1200, A=0020250, F=1, msg = MOVE YOUR TRUCK BEFORE 4:30
04/05/97 16:21:53 B=P1200, A=1901888, F=0, msg = 621584
04/05/97 16:21:53 B=P1200, A=1918430, F=0, msg = 4962900103
04/05/97 16:21:54 B=P1200, A=0016359, F=1, msg = 05 W704240717 1 1 CIBC EQUIP
04/05/97 16:21:54 B=P1200, A=1922062, F=0, msg = 4164586588
04/05/97 16:21:54 B=P1200, A=0886135, F=0, msg = 620900091
04/05/97 16:21:56 B=P512 , A=0017072, F=0, msg = 4165764960
04/05/97 16:21:56 B=P512 , A=0017185, F=0, msg = 3747426
04/05/97 16:21:56 B=P512 , A=0019459, F=3, msg = 8484
04/05/97 16:21:58 B=P512 , A=1653460, F=1, msg = ORD#378749 M3 O/N OPTICON SHP
04/05/97 16:21:58 B=P512 , A=0030100, F=0, msg = 9625
04/05/97 16:21:58 B=P512 , A=0031789, F=0, msg = 6258043
04/05/97 16:21:58 B=P512 , A=0008918, F=0, msg = 685-8333
04/05/97 16:21:59 B=P512 , A=0904728, F=0, msg = 520
04/05/97 16:21:59 B=P512 , A=0007633, F=0, msg = 5198235634
04/05/97 16:21:59 B=P512 , A=0031354, F=0, msg =
04/05/97 16:21:59 B=P512 , A=0908419, F=0, msg = 3291758
04/05/97 16:22:00 B=P512 , A=0913764, F=1, msg = CALL MIKE AT WORK 649-3410
04/05/97 16:22:00 B=P512 , A=0829042, F=0, msg = 21-411-467678268
04/05/97 16:22:00 B=P512 , A=0004804, F=0, msg = 18002634031
04/05/97 16:22:00 B=P512 , A=0008110, F=0, msg = 7874567
```

The "B=*some number*" is the baud rate (either 512, 1200 or 2400 in my area). The "A=*some number*" is the capcode, which is the number that identifies pagers. The "F=*some number*" refers to the type of page, either alphanumeric, numeric or beep. And finally, the "MSG=*some message*" is the message that was transmitted.

The 741 Op Amp decoder was another story. I had some problems getting the thing to work. I traced the problem to the fact that I was using the wrong input on the serial port and had not set up the software to work with the input. Once that was corrected the decoder worked.....kind of. This decoder just plain doesn't work well. After I corrected the problem with the serial input, I set it up to monitor the transmissions. In 10 minutes, it was only able to decode one transmission, and that was full of errors! In that time, the Decoder Kit would have decoded everything it was listening to, with very few errors. I tested this decoder 2 days, and had no luck decoding anything.

There is no log file since I didn't decode anything. Even if I had decoded something, the software doesn't generate a log file anyway.

The last part of the written project is the learning log. It is a simple, point form list of what we have learned, improved upon, or whatever.

Learning Log

Learning Log

- Learned about POCSAG and how it is transmitted and decoded
- Practiced soldering
- Learned about PICs (this was the first one I used in a project)
- Practiced making PCBs
- Learned more about how my scanner works (had to hunt down the discriminator for a better signal to the decoder)
- Practiced my public speaking skills (during the presentation)
- Practiced my Web design skills and learned about some of the new tags (only one or two of those new tags are used here)

Well, if you've made it this far, this is the end of the school part of the page. I have a few pager decoder links below. If you have a link that is not here, please [email Me](#) with the link.

Pager Decoder Links

[About the POCSAG Protocol](#)

[RadioPhone](#)

[Pager Surveillance System 8000](#)

[P_Base POCSAG Database](#)

[Cylex Inc. RM100 Page Decoder](#) (This is the decoder kit that I built)

[The Purple Pager](#)

[POCSAG-Decoder HomePage](#)

[POCSAG and Radio Paging](#)

[POCSAG Logging Receiver](#)

[tnn.comm.pager](#) (newsgroup)

[<- Back](#)[Forward ->](#)[Remote](#)[Mail Me](#)[Search](#)

[Back To Projects Page](#) | [Mail Me](#)