The Hyperlink class I wrote last week was reused, screen shot below: -

Week 26: Monday 22/01/2001 to Friday 26/01/2001

Experimented with serial communications using Microsoft Visual C++ 6.0, example code was found on the Internet, soon I was able to send and receive characters. I also discovered direct access to the hardware is not allowed under Windows NT. Interaction with the serial port was achieved through a file handle and various WIN32 communication API's. This method is Windows 95 compatible.

I decided to create a small windows based master program for the Access 4000 control panel, using my Modbus protocol. GenModbus workspace was created using MS Visual C++ MFC wizard.

Started adding code to initialize the communications, then function to read & receive a character.

Created a real time timer, which by default will be called every 50ms. This function will contain the protocol for transmitting queries and receiving responses. The first time the timer is called the Modbus query for read all controls is transmitted (1,1,0,0,0,10,3D,C6). The next time the timer is called the RX buffer is checked to see if a character has been received, if no character has been received nothing happens and the function returns. This can happen a max of three times (50ms x 3 = 150ms timeout) then the query frame will be transmitted again. One a character is received, the reset of the response frame is received and checked, if valid the received data is added to the

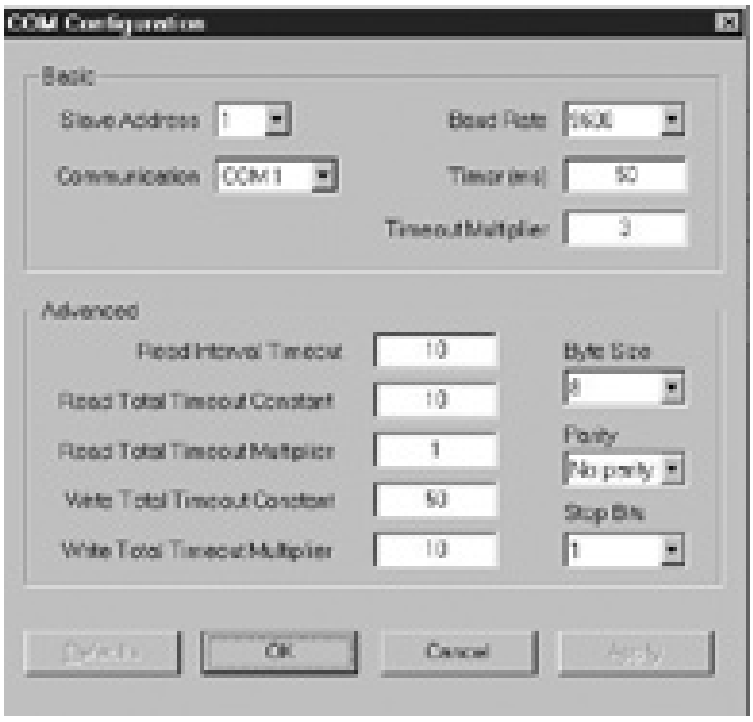
array table[n]. Next query for Status \ Alarms etc… Below shows the basic operation of the timer function: -

```
Void Timer();  /* Called every 50ms */
{
        switch(m_state)
        {
                case TX_CONTROLS:
                        Build and transmit Modbus query frame for reading all controls.
                        Limit = 0;
                        m_state = RX_CONTROLS;
                        Break;

                case RX_CONTROLS:
                        if (char has bee received)
                        {
                                Receive & decode Modbus response frame and if valid
                                add data to Table[n]

                                m_state = TX_ALARMS;
                        }
                        else
                        {
                                limit = limit + 1;
                                if ( limit > 3) m_state = RX_CONTROLS
                        }
                        Break;

                Case TX_ALARMS:
                        Build and transmit Modbus query frame for reading all Alarms.
                        Limit = 0;
                        m_state = RX_ ALARMS;
                        Break;

                case RX_ ALARMS:

                        if (char has bee received)
                        {
                                Receive & decode Modbus response frame and if valid
                                add data to Table[n]

                                m_state = TX_ MONITORING;
                        }
                        else
                        {
                                limit = limit + 1;
                                if ( limit > 3) m_state = RX_ ALARMS
                        }
                        Break;

                case TX_MONITORING:
                        ...
                case RX_ MONITORING:
                        ...
                case TX_CONFIG:
                        ...
                case RX_CONFIG:
                        if (char has bee received)
                        {
                                Receive & decode Modbus response frame and if valid
                                add data to Table[n]

                                m_state = TX_CONTROLS; /* Complete loop
                        }
                        ...
                        break;
        }
}
```

This meant that constant communication can be achieved, net step was to display the table[n] on the screen updating the display every time it's changed. Created a new muliti-doc template called "Address Allocation Table", this window displays the table in it raw form. See screen shot below: -

On Thursday 25th of January I attended an internal training course "Material Master Creation / Amendment – Cources for Engineers". Length 2 hours, which showed how to Create / modify Material Master records in SAP. Included a practical exercise: Fill out material master creation form for item 7 & 11d on drawing MGS3687A.

Added configuration dialogue to GenModbus.exe, see screen shot bellow: -



Started work on a mulitidocument view for viewing the panel configuration.

Week 27: Monday 29/01/2001 to Friday 02/02/2001

Monday to Wednesday off took three floating holidays.

Thursday, continued working on my windows based monitoring program for Access 4000 control panel using my Modbus protocol, which I developed and tested, several months ago. This program is for test proposes only and will properly not be released to the outside world, as the Modbus protocol on Access 4000 was designed for customers to develop there own programs easily.

The only test program FG Wilson currently has for testing the Modbus protocol on Access 4000 control panel is the dos program I developed many moons ago, this program is very low level and only an engineer could understand how to operated and understand what happening. Again this program was not designed for customer use, but customers in Hong Kong needed a test program and my small dos test program was sent.

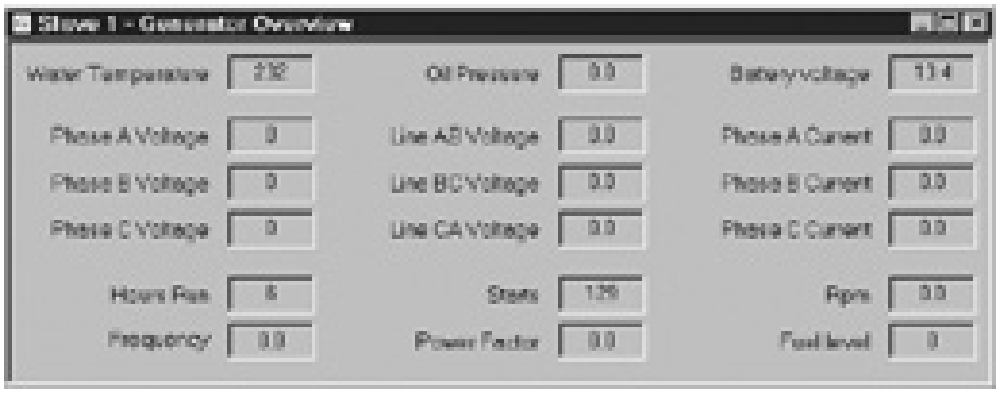
Once communication protocol was complete and tested using my mbpc.exe program running on a laptop (PC simulation of the Access 4000 Modbus protocol). Using RS232 link, the programs communications were checked, the address allocation table on the mbpc.exe have been set to pre-programmed values, so the first thing was to check that the windows application received the whole table correctly. After a view modifications the monitoring communication using a timer has been tested to work correctly. Also checked that the communication did not crash when communication was lost, the protocol worked as designed and continued communication as soon as it was reestablished.

Next added 4 additional views, for viewing the received information at a high level, hence any body can understand quickly and easily what happening: -

1) Generator Overview, e.g. Phase A,B,C Voltage, Line AB, BC,CA Voltage, etc…
2) Status / Alarms Overview, e.g. General Alarm, High Bat Voltage, Low Bat Voltage, etc…
3) Power Overview, e.g. Total kW, Total kVr, Total KVA, etc…
4) Panel Setpoints & Configuration, e.g. High frequency setpoint, Low frequency setpoint, etc…

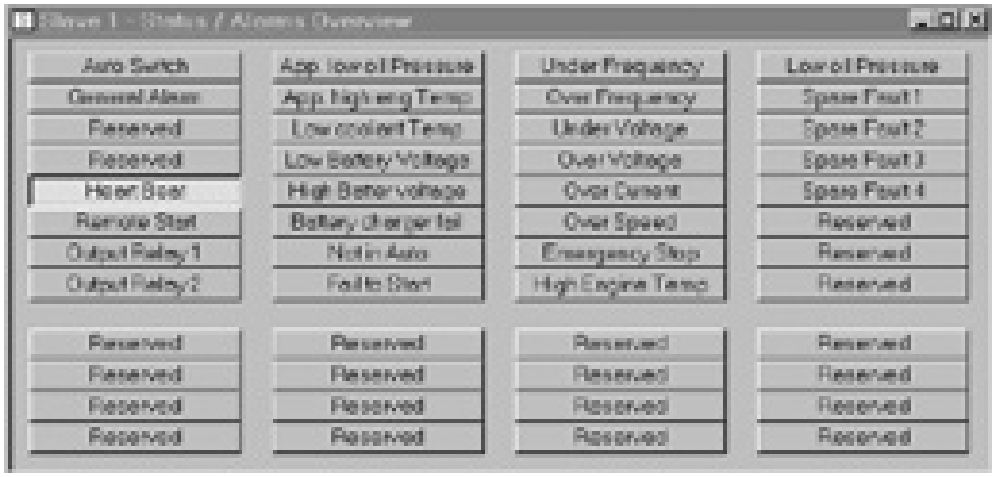
These four additional views take the data directly of the array Table[n], which is filled by the communication protocol.

Screen Shot of Generator Overview: -



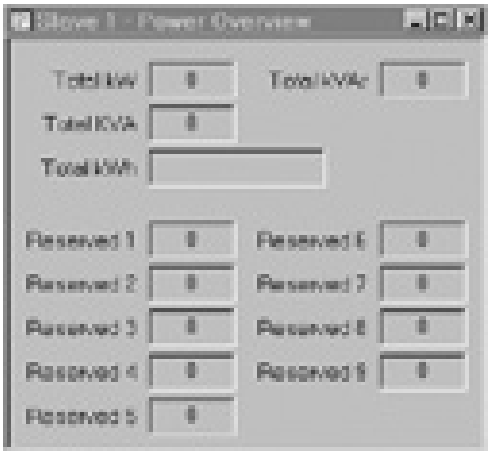
(The above screen shot was taken when connect to an actual Access 4000 control panel, but it was not connected to an generator, or simulator box at this stage, that will be tested next week)

Screen Shot of Status / Alarms Overview: -



(Digital signals, either 1 or 0. Highlighted = logic 1, e.g. “Heart Beat” was at logic 1 when this screen shot was taken)

Screen Shot of Power Overview: -

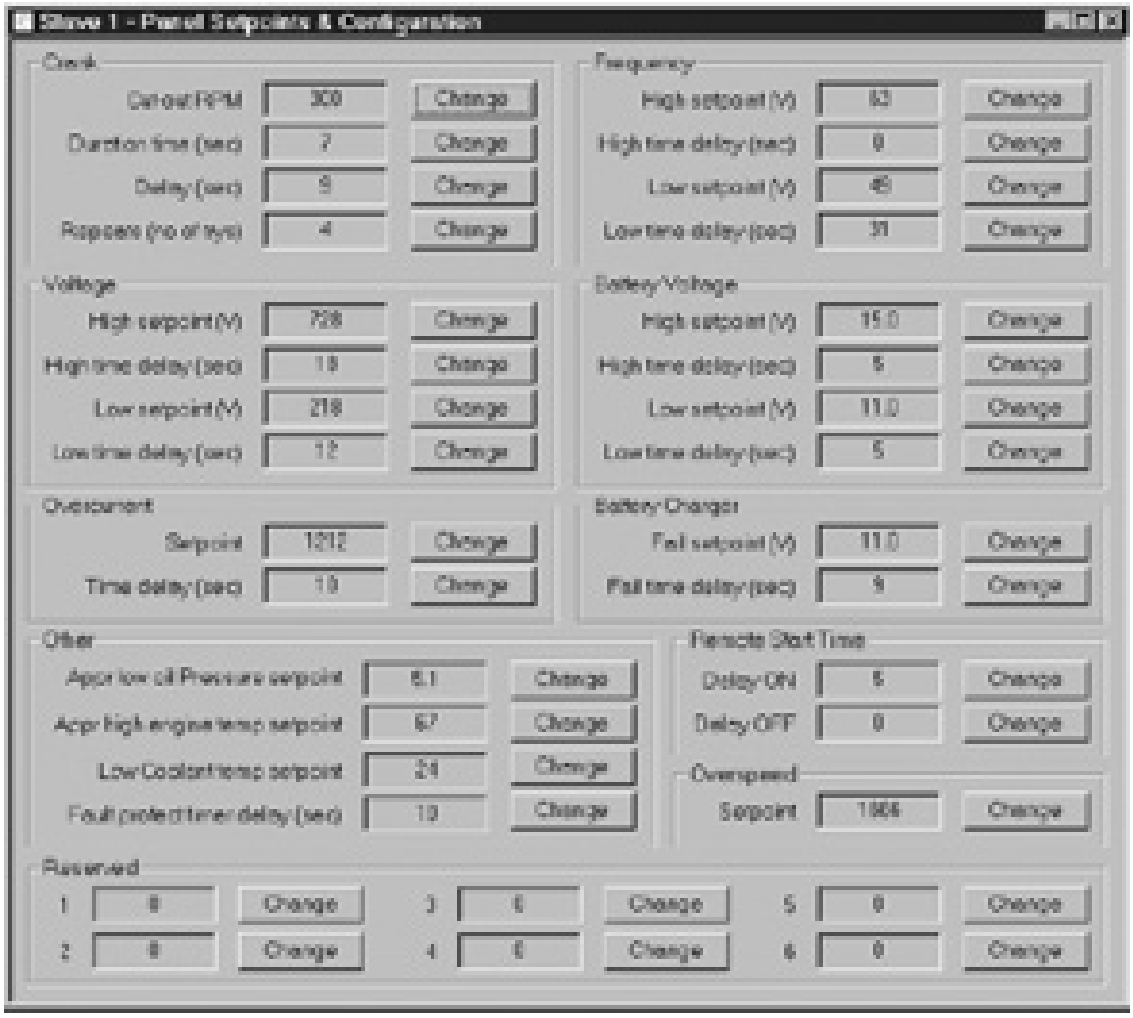


(All fields are double-byte numbers expect for “Total kWh” which is a 32-bit number, it is blank on the screen shot because this feature was disabled on the Access 4000 control panel)

The program can tell that the “Total kWh” feature is disabled when it’s high word is equal to FFFFh, if disabled the program displays nothing in the edit box.

The “Panel Setpoints & Configuration” view displays the current values of all the setpoints configurable in the access 4000 control panel, also a button [Change] beside every setpoint, when clicked a dialogue box appears which allow the user to change the value of that setpoint.

Screen shot of Panel Setpoints & Configuration shown below: -

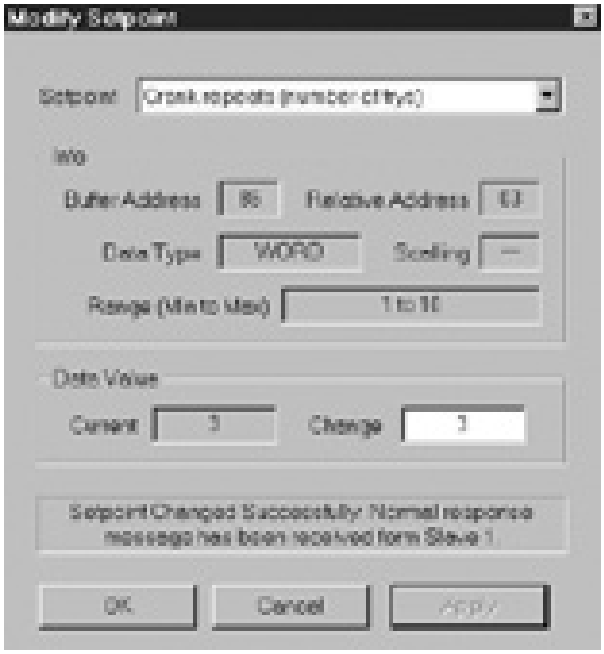


Added Modify setpoint Dialogue, communication is stopped when the dialogue is opened and started again when closed. If the setpoint value is change and OK or APPLY are hit, the program communicates with the Access 4000 control panel and requests that the setpoint should be changed to the new value.

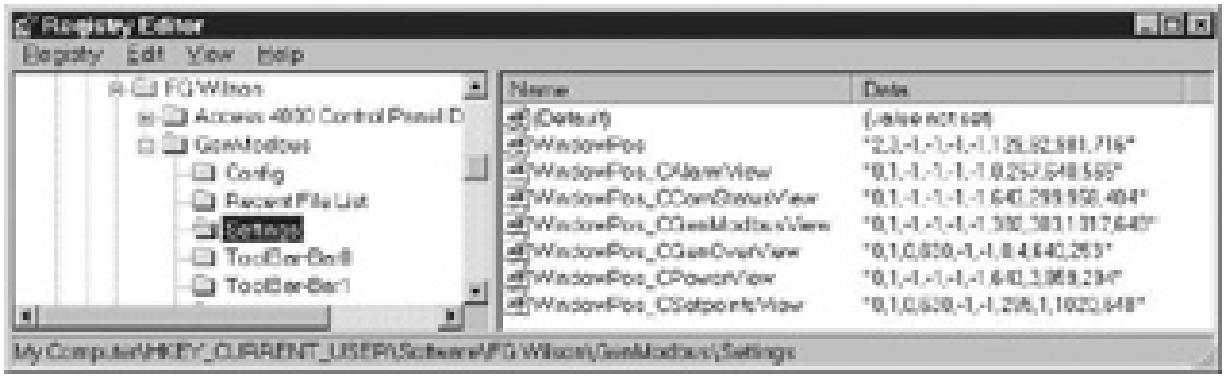
If timeout occurs the program tries again, this will happen a maximum of 5 times. After 5 times an error message will be displayed in the dialogue box, and had been pressed to dialogue will not close. If setpoint was updated successfully, a message saying so is also displayed and if [OK] was pressed the dialogue will close.

The dialogue uses a combo box to select setpoints, user can stay in the dialogue and change a number of setpoints by selecting them on the combo box, changing there value and press [Apply]. If the user hits the [Select] button on the “Panel Setpoints & configuration” view beside a setpoint, that setpoint will be selected automatically when the dialogue is loaded.

Screen shot of “Modify Setpoint” dialogue: -



Note: All views save the window size & position when closed onto the Windows registry, which are restored when loaded: -



Next, need to add Genset controls [Start], [Stop], [Estop], [Reset]: -

The control code has been added to the timer, 8 more states have been added: TX_START, RX_START, TX_STOP, RX_STOP, TX_ESTOP, RX_ESTOP, TX_RESET, RX_RESET.

If timeout occurs the query message will be repeated, this will happen a max of 3 times and state will move back to TX_CONTROLS. If successful state moves back to TX_CONTROLS. These controls have been tested on the laptop and the [Reset] control has also been tested on the panel it self, all work as expected. When generator simulator becomes available next week the other control will be tested fully. All views, Controls, [Disconnect], [Connect], etc. are available on the menu bar, toolbars and the context menu.

Screen shot of context menu show below (note my bitmap menu code from my database program has been reused): -

Screen shot of main toolbar shown below: -

Note hyperlink code from my Access 4000 database program has also been reused: -

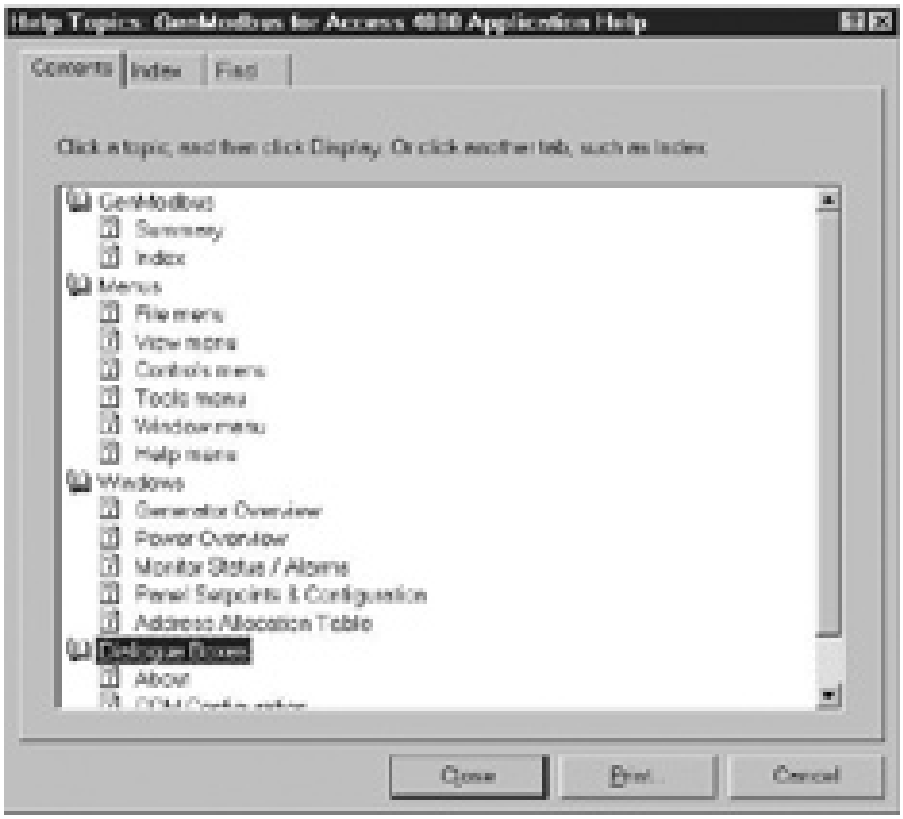
Week 28: Monday 05/02/2001 to Friday 09/02/2001

Continue working on GenModus monitoring program, improved code structure and reliability. Added an icon on the status bar which flashes green then yellow when Modbus communications are operating correctly the Icon is red when in disconnect mode. If communication fails the icon will not flash. Screen shot of the status bar below: -

Program was tested on Access 4000 control panel using hardware simulator to simulate the Genset. The program works fine, no problems found yet!

There is another change of plan; the program will now be placed on the web for customers to download hence a small help file had to be created. It was decided that the program will make a good test program, and will be available to the customers of Access 4000 free of charge.

Created a help file using “Help Workshop” and “Microsoft word”, this is the first time I’ve created a windows help file and it took some time to learn how to compile such a file. See Screen shot below: -



All Windows, Dialogues and menu items are directly linked to the help file; for example if you press F1 when the “COM Configuration” dialogue is active. The help topic “COM Configuration2 will appear.

Thursday & Friday was spent in the electronic lab. Helping to carrier out voltage transient tests on the new CI panel. It had previously fail some of these tests at Queen University Belfast; we were checking the modifications made worked.

The first test was to send these voltage transient signals through the main voltage input on the CI panel, each transient test lasted for 1 minute. Tests included: -

- +1kV transient on Ground line.
- -1kV transient on Ground line.
- +1kV transient on Live.
- -1kV transient on Live.
- +1kV transient on Neutral.
- -1kV transient on Neutral.
- +2kV transient on Ground line.
- -2kV transient on Ground line.
- +2kV transient on Live.
- -2kV transient on Live.
- +2kV transient on Neutral.
- -2kV transient on Neutral.

The CI panel continued working as normal at 1kV, at 2kV RS485 communication died, but recovered when voltage transient was stopped. No permit damage was caused to the CI panel during any of these tests.

Next, using a coupling clamp all external cables were subjected to these transient signals. Cables tested included RS485, AutoDialer faults and the telephone cables (both Autodialer and modem). In every case the coupling clamp was charged to 1kVolts, both Positive and Negative signal where tested. RS232 (GenAccess) to RS485 (Access 4000) communication slowed when voltage transient signals were active but recovered as soon as each test was complete.

Large voltages such as 1kVolt & 2kVolt signals where being used, hence it was vital to follow strict safety producers. CI panel did not pass all tests first time, modifications where made and tests repeated until it passed. The CI panel will be tested officially on Saturday at Queens University Belfast. Production cannot start until the CI panel passes all of these tests, if any fail it could hold back production for weeks.

Week 29: Monday 12/02/2001 to Friday 16/02/2001

Started work on a small dialog based windows program, propose of which is to load *.txt and *.S19 files to the Access 4000 control panel. There already exists a dos version called load.exe, but for it to work correctly you must leave Windows. The communication protocol used is extremely simply compared with Modbus, plus I had a copy of the source code form the MS dos load.exe.

Using the source code from the MS dos load.exe as a guild I started to write the communication protocol for sending Text and S19 files. The communication protocol for text and S19 files differ so it was important to detect, which was being sent.

The text file communication protocol was really simple; it consisted of sending the text file out in one large block then waiting for ACK once complete. The S19 (main code) file communication protocol was also simple; it consisted of sending the S19 file out in 80 byte blocks, waiting for ACK after every block. If timeout occurs or NAK the last block is repeated, max number of NAKs is 10 (then error message to many NAKs will be displayed) and max number of timeouts is 5.

The baud rate used is 9600bps, 1 stop bit, and no party. This format is fixed and cannot be change, reliability is more important than speed. It takes about 4 min 30 seconds to load 220Kb S19 file (Access 4000 – Ver 1.19K), about 5 second for 3195 byte language text file and about