

USGS Outreach – Kiosk Project

By Todd Siskin

Background

The Kiosk project was initially started to develop a method of retrieving data from a field instrument located on the Cayuga Lake in Ithaca, NY, and graph the data in real-time within a PowerPoint presentation running on an enclosed computer at a restaurant located on the lake. This prototype will then be used to set up other kiosks throughout the state.

Development

The project was divided into several tasks. The first task was to develop a Visual Basic Application script within Excel to automate the updating of charts from a data source. Initially, this source was a text file. The script was developed to read values and place them in the spreadsheet, while updating the chart's parameters to keep the chart current for the date. Only the X-axis parameters need be changed, normally once a day, at midnight. The data would collect, with each new value adding a point on the chart. Once the script was developed, the next task was to change the data source from a text file to a serial port.

VBA would not access the serial port within Windows, so another method had to be found. The MSComm library was looked at, but did not integrate with VBA. (Visual Basic Professional was required, but considered beyond the scope of the project.) A third-party program called WinWedge was used to read the serial port and pass the data to Excel using Dynamic Data Exchange. The script was modified and streamlined to accept data from WinWedge and process it into the Excel spreadsheet. Additional sensors were added to the logging instrument and the script was updated for multiple charts at this time.

The next task was to integrate the Excel charts into the PowerPoint presentation. Initially, it was thought that Object Linking and Embedding would be used to link the charts into the presentation. This, in combination with the DDE link used to collect the data caused multiple problems with the project. DDE and OLE conflicts caused the original data collection module to execute repeatedly each time the presentation ran. Another conflict caused the OLE links to fail to update automatically while the presentation was running.

To solve the first problem, the data collection module had to be rewritten to allow control from WinWedge rather than from Excel. This became a moot point because of the second problem where the OLE charts wouldn't update automatically. It was determined to be a conflict with using DDE and OLE concurrently (for lack of a better explanation). The VBA script was modified to export the charts to JPEG files after the values were processed. The presentation was modified to reference the JPEG files rather than the OLE charts. This introduced a small problem because PowerPoint would cache the images, even when linked from external files.

It was hoped that the linked files would be reread each time they were referenced. This not being the case, several methods were tried, including using PowerPoint Viewer's list facility, publishing the presentation as a web page, and using a batch file to start the presentation repeatedly. This final method was the most promising, except that the original attempt did not make use of the Batch mode utility "Start", and the system crashed because it attempted to initiate an infinite number of multiple occurrences of the presentation rather than running one presentation at a time. Once the Start command was discovered, the script performed as desired.

An additional task, to allow external files to be transferred from the Unix server to the remote PC for inclusion into the presentation, was added for one of the proposed kiosks. It was thought that FTP might be a possible method, but Kermit was chosen since the remote communications will most likely be over long distance phone lines.

Design

As this project is to run unattended at Kiosk sites throughout the state, the program needs to be fairly self-sufficient. The system used for these projects should be a simple system without a lot of extra embellishments. The display monitor should be large enough to convey the desired information. As LCD flat-panel displays become more cost effective, they should be considered for these projects. A basic Intel Pentium system running Windows 9x, Microsoft Excel, Microsoft PowerPoint Viewer, MS-Kermit and TalTech WinWedge should be sufficient. Most or all of the Desktop icons should be hidden. The Desktop background should be set to the USGS Visual Identity logo.

The Wrapper

The system password prompt should be disabled so that the computer will start without user intervention. A link to the main script is placed in the C:\Windows\Start Menu\Programs\Startup folder. This script starts WinWedge with the appropriate configuration file. Excel is then started minimized, opening the data collection spreadsheet. Macro Security should be set to Low for this system. The PowerPoint script is then executed. This script starts the PowerPoint Viewer and displays the presentation once. When complete, the script runs itself to start the presentation all over again. Between each presentation run, the desktop will appear briefly.

Data Collection

WinWedge is configured to read the data stream being sent over the instrument's secondary data (dump) port and pass this data stream to the DDE channel for Excel to retrieve it. This allows the instrument to continue to provide the data through the dial-in port as well as to the kiosk. No translation is performed on the data stream. It is up to the Excel VBA module to parse to appropriate data. Since WinWedge does not perform any parsing, no additional changes to the configuration file should be necessary. As a DDE server, WinWedge signals the Excel data collection module whenever a new value is read.

The Spreadsheet

The Excel spreadsheet contains Visual Basic for Applications (VBA) code that activates when WinWedge receives data from the instrument. Upon first execution, the module determines where the first piece of data should be stored based on the difference between the current data and time, and the initial date stamp in cell A2 of the spreadsheet. If the cell is blank, the current date stamp is stored in that cell and is used in the computation. (All other cells in the column are formulae adding 15 minutes to the previous cell.) The X-axis parameters are updated to the current date range. Use of static variables allows information to be saved between module executions. The data stream is read and parsed into individual values and stored in the spreadsheet at the location determined during initialization. The cell pointer is incremented and if the date changes, the X-axis parameters are incremented as well. The charts are exported as JPEG files and the module ends.

The spreadsheet consists of several columns of dates and data as well as charts associated with each data set. For each chart, create a date column and parameter column(s). For the first data set, the date column should consist of a real value in the first row and the formula " $=A2+(1/96)$ " copied into each subsequent row. For each additional chart, enter the formula " $=A2$ " in the appropriate cell (ex. D2, F2) and copy the first formula into each subsequent row. (Copying the formula will allow Excel to substitute the appropriate row number for you.)

The VBA module will need to be modified for any additional parameters added to the spreadsheet. Up to seven parameters can be received from WinWedge. Two parameters can be parsed by taking the first 10 bytes and last 10 bytes as the two values. Additional parameters will require more advanced parsing.

Each chart references a separate data set and can be configured independently of each other to display the appropriate information.

The Presentation

The PowerPoint presentation has no specific criteria for development. To include a chart on any specific page, select the Insert menu, then Picture, then From File. Browse to the chart you want to insert and change "Insert" to "Link to File". Resize and position the chart as desired. If the chart is to be in the same location on more than one page, select the Edit menu, then Copy. Move to the desired page(s) and select the Edit menu, then Paste on each desired page. (Ctrl/V is the Paste shortcut.)

In order to allow the charts to be updated, the presentation is run from a recursive batch file. The charts are loaded from the directory each time the presentation starts. With Excel exporting the charts to a disk file, the presentation "updates" itself each time it starts.

Remote File Transfer

MS-Kermit is being used to connect to the office server and transfer files to be incorporated into the presentation. The system scheduler will execute the batch file at the designated time, which will invoke MS-Kermit-Medium, a feature-limited version of the Kermit protocol. The batch file simply starts Kermit with a customized initialization script and the Kermit command (Take) to perform the command script.

The command script contains the Kermit statements to dial the office terminal server, connect to the Unix server, log in as a cooperator (NYCDEP1), initiate the remote file transfer using C-Kermit, logout from the server and disconnect the modem. No user interaction is required and the file is overwritten each time the script is run. As with the Excel charts, the presentation will be read the new file once it has been transferred.