# Text for initial video description

The screen on the left is the Terminal display application. The picture on the right is an original Heathkit H-89 showing the screen fonts. The Load button opens a file dialog box to get the original font file. The number in the text box to the right of the Show button is the offset to a character in the font file. The asterisk in large text box represents one pixel from the font file.

The H19 button starts a new window with a bitmap image. The initialization function writes ABCDEF in the bitmap. The next letters were typed from the keyboard. Each time a character is written to the bitmap, the bitmap is saved to variable representing the bitmap. The bitmap is fixed in size and does not scale with the window frame.

The next program is the serial interface to a CP/M system. The application on the left is my application, currently using a textbox. I will integrate the bitmap solution into this application. The application on the right is Absolute Telnet used to connect to the CP/M system console via Telnet. My application is connected at 19,200 baud through a serial port. I would like my bitmap solution to scale like Absolute Telnet (expand and shrink fonts with window size)

# Design Notes 1/17/2023

Alrighty...now we're talking😎...some questions/thoughts:

0) Why is there no Debug configuration?

1) You'll have to offer more details, but C# programs are generally organized by namespace/class/members (functions and data) w/large classes (sometimes) broken across files using 'partial classes'.

2) I'm intrigued...Are you drawing each char pixel-by-pixel? As for when to update - I suspect you want a termh19\_Paint() handler. Paint is the Windows event where you Should do all your redrawing.

3) Obviously related to (2) but check out the termh19.SizeMode property. One of its five settings may help. Docking/Anchoring may also help. [c# - resizing pictureBox proportionally to Form resizing - Stack Overflow](<https://stackoverflow.com/questions/23982033/resizing-picturebox-proportionally-to-form-resizing>)

4) [ToAscii function (winuser.h) - Win32 apps | Microsoft Learn](<https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-toascii>)

5) Threading and Windows is notoriously tricky due to an ugly history that requires update to Windows Controls to be done on the 'main' thread - the one where they're created. Consider the Background Worker Component from the Toolbox instead (or at least to start/experiment). [How to: Download a File in the Background - Windows Forms .NET Framework | Microsoft Learn](<https://learn.microsoft.com/en-us/dotnet/desktop/winforms/controls/how-to-download-a-file-in-the-background?view=netframeworkdesktop-4.8>)

Let me know if this helps and what's next. If you'd like to meet, I have time during the day tomorrow if you're available.