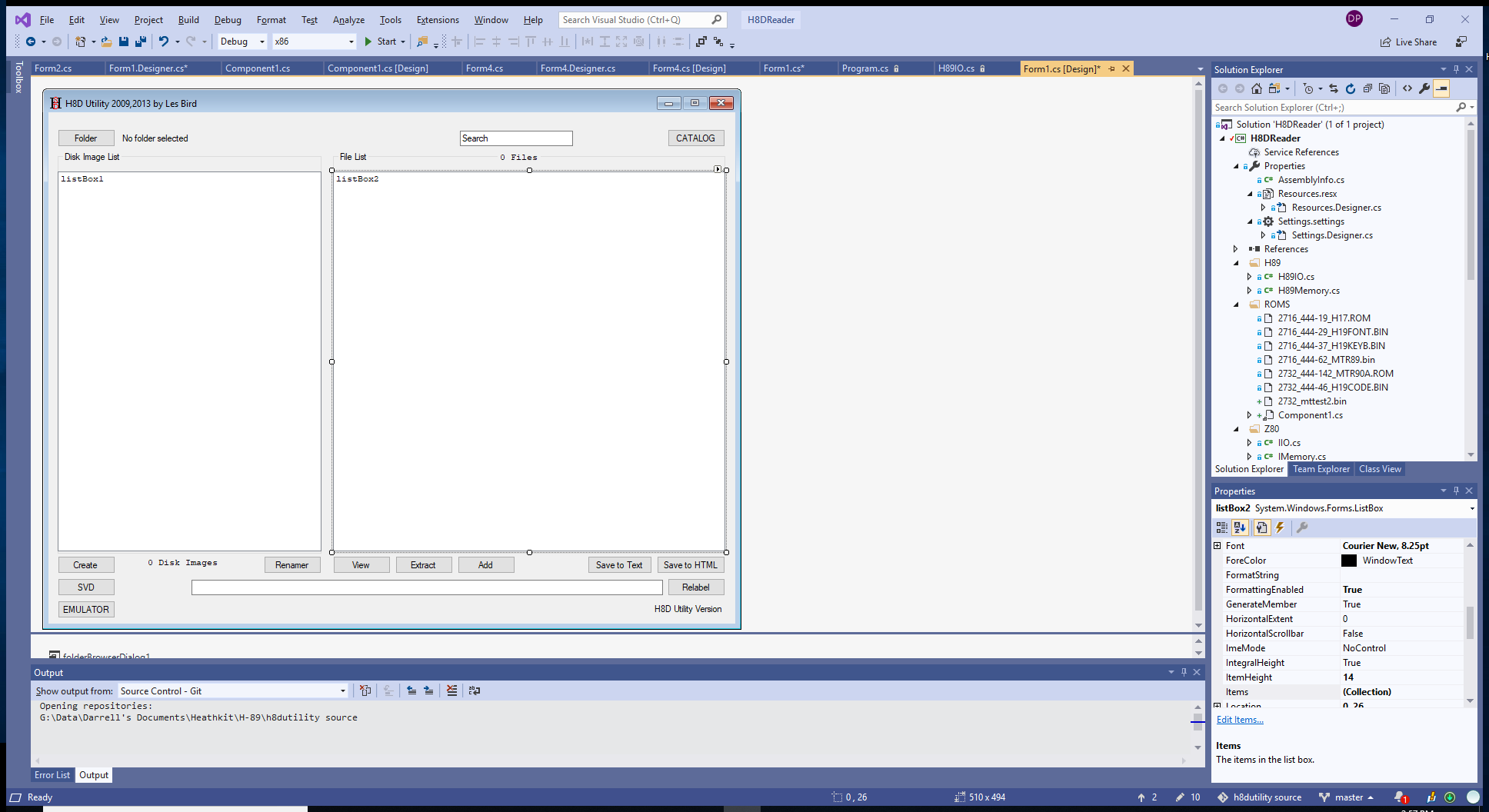
H8D Utility Changes

Darrell Pelan

* Version 1.52
  + Added CP/M file extract capability
  + Added initialization for Folder button to read directory from last saved working directory
  + Added try/catch to ReadCPMDir Entry for empty images
  + Added capability to select Boot ROM from directory listing
  + Changed CPM File size message to indicate bytes instead of KB
  + Added disk size calculation based on stored size to account for 400k .H8D disks
  + Changed from ASCIIencoding to UTF8Encoding to deal with ASCII chars with bit 7 set high being encoded as ‘?’ instead of the proper character
* Version 1.60
  + Added CP/M capability to Add button for H8D disks
* Version 1.70
  + Added .imd file read capability to convert to H37 format
* Updated CPMFile code

Questions

* In the H8D format, which directories start at 0x2200 or 0x2600?
* Is there a disk “marker” to indicate a 400k LLL format disk?



**Form1.cs**

* Catalog - button2 line 253
* ReadCPMImage() line 645
* ReadCPMDirEntry() line 715

(listBox1.Items[i].ToString().Contains(".H8D")

* Folder – button1
* Search – textBox1
* Extract = button7
  + line 1256, new code 1414
* Add Button6

Form4.cs ROM Boot Line 159

Form4.cs Function LoadROM line 1839

**H8D Format**

diskdef h8d  
   seclen 256  
   tracks 40  
   sectrk 10  
   blocksize 1024  
   maxdir 32  
   skew 4  
   boottrk 3  
   os 2.2

256 \* 10 \* 40 = 102,400

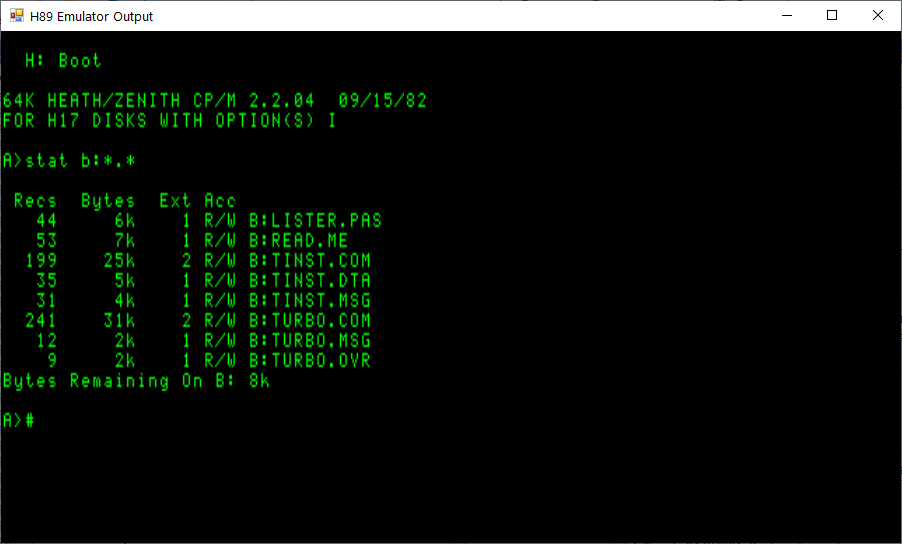
Interleave = 8

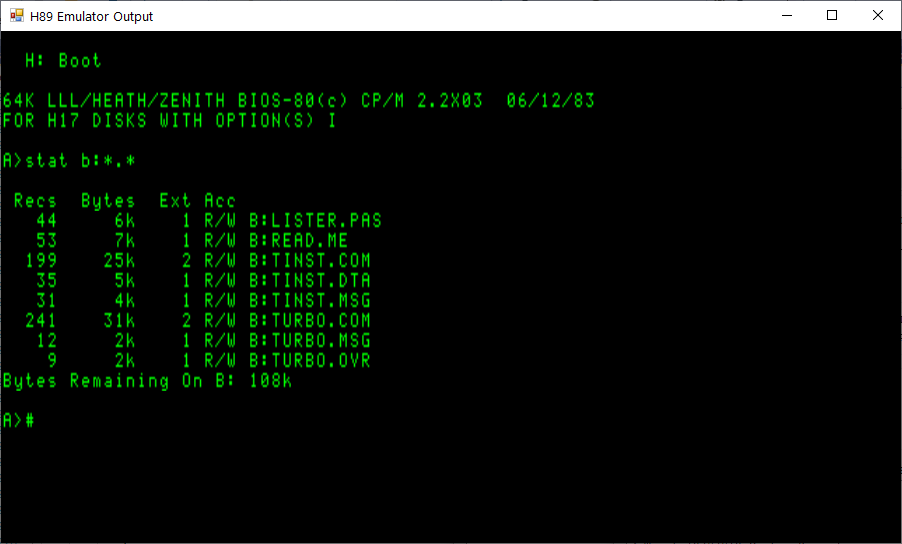
Directory start 1E00

Directory byte 15 = number of 128 records in this entry

* Record number is max 0x80 = 128.
* There are 16 allocation block numbers per directory entry
* Allocation block size is 1024 bytes. 16 \* 1024 /128 = 128 records per directory entry (0x80)
* 16k per directory entry
* Physical sectors are 256 bytes. Logical sectors are 128 bytes

ReadCPMImage counts on directory entries being sequential





Hi,

I've completed my fist set of updates to the H8DUtility and attached a zip file with the exe file. The changes are listed below. Please let me know if you try it out and if (when) you encounter a problem.. I still need to add comments to my code and clean up a few things, so I'll release the code a little bit later. As this was my first C# coding experience, there was a lot of learning on my part. "The Programmers CP/M Handbook" was invaluable.

• Added CP/M file extract capability

• Added initialization for Folder button to read directory from last saved working directory

• Added try/catch to ReadCPMDir Entry for empty images

• Added capability to select Boot ROM from directory listing. Boot ROMs must begin with "b\_". This was driven by a need to test new ROM code to save burn/install/test cycles. The failure of my ROM test code on the hardware was duplicated in the emulator.

• Changed CPM File size message to indicate bytes instead of KB

• Added disk size calculation based on stored size to account for 400k .H8D disks

• Changed from ASCIIencoding to UTF8Encoding to deal with ASCII chars with bit 7 set high being encoded as ‘?’ instead of the proper character

Future efforts include

Add support for .IMD files. I looked at the .HFE files, but they are stored in a MFM encoded format making manipulation much more difficult.

Add CP/M file capability to the Add button.

Darrell

<**style name="Base.V28.Theme.AppCompat" parent="Base.V26.Theme.AppCompat"**>  
 *<!-- We can use the platform styles on API 28+ -->* <**item name="dialogCornerRadius"**>?android:attr/dialogCornerRadius</**item**>  
</**style**>  
<**style name="Base.V28.Theme.AppCompat.Light" parent="Base.V26.Theme.AppCompat.Light"**>  
 *<!-- We can use the platform styles on API 28+ -->* <**item name="dialogCornerRadius"**>?android:attr/dialogCornerRadius</**item**>  
</**style**>

<**style name="Base.Theme.AppCompat" parent="Base.V28.Theme.AppCompat"**/>  
<**style name="Base.Theme.AppCompat.Light" parent="Base.V28.Theme.AppCompat.Light"**/>  
<**style name="Base.V28.Theme.AppCompat" parent="Base.V26.Theme.AppCompat"**>  
 *<!-- We can use the platform styles on API 28+ -->* <**item name="dialogCornerRadius"**>?android:attr/dialogCornerRadius</**item**>  
</**style**>  
<**style name="Base.V28.Theme.AppCompat.Light" parent="Base.V26.Theme.AppCompat.Light"**>  
 *<!-- We can use the platform styles on API 28+ -->* <**item name="dialogCornerRadius"**>?android:attr/dialogCornerRadius</**item**>  
</**style**>