# **Open Cases Submitted to Hitachi**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Received** | **Closed** | **Subject** |
| 70980 | Tue 3/12/2019 10:26 AM | Fri 4/12/2019 9:51 AM | Calendar Block Shift Code |
| 70999 | Tue 3/12/2019 12:10 PM | Fri 4/12/2019 9:51 AM | Time Count Function |
| 71000 | Tue 3/12/2019 12:11 PM | Fri 4/12/2019 9:51 AM | Counter block |
| 71014 | Tue 3/12/2019 1:53 PM | Fri 4/12/2019 9:51 AM | Calendar Block |
| 71277 | Sun 3/17/2019 12:56 PM | Fri 4/12/2019 9:51 AM | Debugging cijConnect |
| 71389 | Tue 3/19/2019 8:45 AM | Fri 4/12/2019 9:51 AM | IJP Operation Function |
| 71400 | Tue 3/19/2019 10:36 AM | Fri 4/12/2019 9:53 AM | Managing messages saved on the printer |
| 71435 | Tue 3/19/2019 3:21 PM | Fri 4/12/2019 9:51 AM | Print Data Management |
| 71476 | Wed 3/20/2019 10:12 AM | Fri 4/12/2019 9:51 AM | User pattern processing |
| 71488 | Wed 3/20/2019 11:35 AM | Fri 4/12/2019 9:51 AM | Environment Settings |
| 71501 | Wed 3/20/2019 1:31 PM | Fri 4/12/2019 9:51 AM | Unit Information |
| 71535 | Thu 3/21/2019 8:54 AM | Fri 4/12/2019 9:51 AM | Print Format Function |
| 71678 | Mon 3/25/2019 9:38 AM | Fri 4/12/2019 9:51 AM | Add/Delete/Insert columns |
| 71710 | Mon 3/25/2019 1:11 PM | Fri 4/12/2019 9:51 AM | Multi-line setup |
| 72039 | Fri 3/29/2019 12:48 PM |  | Auto Reflection issue |
| 72285 | Wed 4/3/2019 12:35 PM |  | Character Size |

# **Calendar Block Shift Code**

I am trying to use EtherNet/IP to initialize the Shift Settings.

The code looks like

**#region Item #4**

**// Add and select item #4**

**ServiceAttribute(ClassCode.Print\_format, (byte)ccPF.Add\_Column, 0);**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Item, 4);**

**// Point the calendar block  to Item #4**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Calendar\_Block, 4);**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Dot\_Matrix, "5x8");**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.InterCharacter\_Space, 1);**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Print\_Character\_String, "=>{{EE}}<=");**

**// Set < Shift Number="1" StartHour="00" StartMinute="00" EndHour="11" EndMinute="59" Text="AA" />**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Calendar\_Block, 1);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_Start\_Hour, 0);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_Start\_Minute, 0);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_String\_Value, "AA");**

**// Set < Shift Number="2" StartHour="12" StartMinute="00" EndHour="23" EndMinute="59" Text="BB" />**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Calendar\_Block, 2);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_Start\_Hour, 12);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_Start\_Minute, 0);**

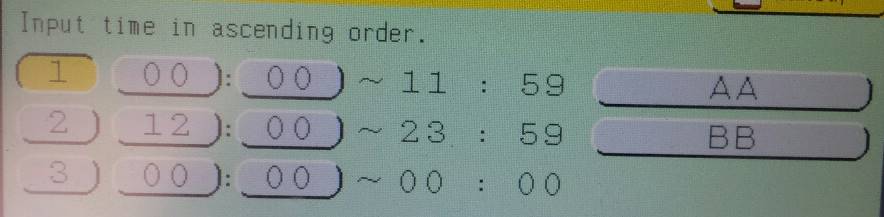
**SetAttribute(ClassCode.Calendar, (byte)ccCal.Shift\_String\_Value, "BB");**

**#endregion**

The traffic looks like

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item | 2 | 4 | 00 04 |
| 00 -- O.K. -- 32 7A 01 6F | Set | Index | Calendar\_Block | 1 | 4 | 04 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix | 1 | 5x8 | 03 |
| 00 -- O.K. -- 32 67 01 75 | Set | Print\_format | InterCharacter\_Space | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String | 11 | =>{{EE}}<= | 3D 3E 7B 7B 45 45 7D 7D 3C 3D 00 |
| 00 -- O.K. -- 32 7A 01 6F | Set | Index | Calendar\_Block | 1 | 1 | 01 |
| 00 -- O.K. -- 32 69 01 80 | Set | Calendar | Shift\_Start\_Hour | 1 | 0 | 00 |
| 00 -- O.K. -- 32 69 01 81 | Set | Calendar | Shift\_Start\_Minute | 1 | 0 | 00 |
| 00 -- O.K. -- 32 69 01 84 | Set | Calendar | Shift\_String\_Value | 3 | AA | 41 41 00 |
| 00 -- O.K. -- 32 7A 01 6F | Set | Index | Calendar\_Block | 1 | 2 | 02 |
| 00 -- O.K. -- 32 69 01 80 | Set | Calendar | Shift\_Start\_Hour | 1 | 12 | 0C |
| 00 -- O.K. -- 32 69 01 81 | Set | Calendar | Shift\_Start\_Minute | 1 | 0 | 00 |
| 00 -- O.K. -- 32 69 01 84 | Set | Calendar | Shift\_String\_Value | 3 | BB | 42 42 00 |

The view from the printer



No COM errors were generated in the process.

When I tried to read the shift codes back, I got

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Count OK** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 6F | True | Index | Calendar\_Block |  |  |  | 1 | 1 | 01 |
| 00 -- O.K. -- 33 69 01 80 | True | Calendar | Shift\_Start\_Hour | 1 | 255 | FF |  |  |  |
| 00 -- O.K. -- 33 69 01 81 | True | Calendar | Shift\_Start\_Minute | 1 | 255 | FF |  |  |  |
| 00 -- O.K. -- 33 69 01 82 | True | Calendar | Shift\_End\_Hour | 1 | 12 | 0C |  |  |  |
| 00 -- O.K. -- 33 69 01 83 | True | Calendar | Shift\_End\_Minute | 1 | 0 | 00 |  |  |  |
| 00 -- O.K. -- 33 69 01 84 | True | Calendar | Shift\_String\_Value | 1 | " | 00 |  |  |  |
| 00 -- O.K. -- 32 7A 01 6F | True | Index | Calendar\_Block |  |  |  | 1 | 2 | 02 |
| 00 -- O.K. -- 33 69 01 80 | True | Calendar | Shift\_Start\_Hour | 1 | 255 | FF |  |  |  |
| 00 -- O.K. -- 33 69 01 81 | True | Calendar | Shift\_Start\_Minute | 1 | 255 | FF |  |  |  |
| 00 -- O.K. -- 33 69 01 82 | True | Calendar | Shift\_End\_Hour | 1 | 12 | 0C |  |  |  |
| 00 -- O.K. -- 33 69 01 83 | True | Calendar | Shift\_End\_Minute | 1 | 0 | 00 |  |  |  |
| 00 -- O.K. -- 33 69 01 84 | True | Calendar | Shift\_String\_Value | 1 | " | 00 |  |  |  |

Not what I expected.

However, I have a question.

I did not send the Shift End Hour or Shift End Minute.  Is it safe never to send them?

Any help would be appreciated.

# **Time Count Function**

I am trying to set up the Time Count function using EtherNet/IP.  The code looks like

**#region Item #2**

**// Add and select item #2**

**ServiceAttribute(ClassCode.Print\_format, (byte)ccPF.Add\_Column, 0);**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Item, 2);**

**// Point to calendar block #2**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Calendar\_Block, 2);**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Dot\_Matrix, "5x8");**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.InterCharacter\_Space, 1);**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Print\_Character\_String, "=>{{FF}}<=");**

**// Set <TimeCount Start="AA" End="JJ" Reset="AA" ResetTime="6" RenewalPeriod="30 Minutes" />**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Time\_Count\_Start\_Value, "AA");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Time\_Count\_End\_Value, "KK");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Time\_Count\_Reset\_Value, "AA");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Reset\_Time\_Value, 6);**

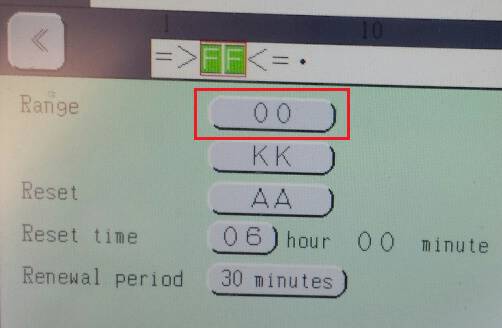
**SetAttribute(ClassCode.Calendar, (byte)ccCal.Update\_Interval\_Value, "30 Minutes");**

**#endregion**

The traffic looks like.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item | 2 | 2 | 00 02 |
| 00 -- O.K. -- 32 7A 01 6F | Set | Index | Calendar\_Block | 1 | 2 | 02 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix | 1 | 5x8 | 03 |
| 00 -- O.K. -- 32 67 01 75 | Set | Print\_format | InterCharacter\_Space | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String | 11 | =>{{FF}}<= | 3D 3E 7B 7B 46 46 7D 7D 3C 3D 00 |
| 00 -- O.K. -- 32 69 01 7B | Set | Calendar | Time\_Count\_Start\_Value | 3 | AA | 41 41 00 |
| 00 -- O.K. -- 32 69 01 7C | Set | Calendar | Time\_Count\_End\_Value | 3 | KK | 4B 4B 00 |
| 00 -- O.K. -- 32 69 01 7D | Set | Calendar | Time\_Count\_Reset\_Value | 3 | AA | 41 41 00 |
| 00 -- O.K. -- 32 69 01 7E | Set | Calendar | Reset\_Time\_Value | 1 | 6 | 06 |
| 00 -- O.K. -- 32 69 01 7F | Set | Calendar | Update\_Interval\_Value | 1 | 30 Minutes | 06 |

The view from the printer



Everything worked except the initial part of the range.  No COM Errors were generated.

Where I read the settings back, I get

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** |
| 00 -- O.K. -- 33 69 01 7B | Get | Calendar | Time\_Count\_Start\_Value | 3 | "000" | 30 30 30 |
| 00 -- O.K. -- 33 69 01 7C | Get | Calendar | Time\_Count\_End\_Value | 3 | "9KK" | 39 4B 4B |
| 00 -- O.K. -- 33 69 01 7D | Get | Calendar | Time\_Count\_Reset\_Value | 3 | "0AA" | 30 41 41 |
| 00 -- O.K. -- 33 69 01 7E | Get | Calendar | Reset\_Time\_Value | 1 | 6 | 06 |
| 00 -- O.K. -- 33 69 01 7F | Get | Calendar | Update\_Interval\_Value | 4 | 117440268 | 06 FF FF 0C |

The start value agrees with the printer screen but not what I sent.

Start/End/Reset values all have an extra character as a prefix.

Update Interval value returns four bytes.  The first byte (the 6) is correct.

Any help on this would be appreciated.

# **Counter block**

I am trying to build a counter and read it back using the EtherNet/IP Protocol.

Here is the code.  It shows the Class, Attribute, and Data that is being sent.  Human readable values like “Enable” or “Disable” are translated to the correct value before they are sent.

**// Set to first item**

**int item = 1;**

**// Select item #1**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Item, item);**

**// Set item number in count block**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Count\_Block, item);**

**// Set font, ICS, and Text is a 4 digit counter**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Dot\_Matrix, "5x8");**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.InterCharacter\_Space, 1);**

**SetAttribute(ClassCode.Print\_format, (byte)ccPF.Print\_Character\_String, "{{CCCC}}");**

**// Set <Counter InitialValue="0001" Range1="0000" Range2="9999" JumpFrom="6666" JumpTo ="7777"**

**//      Increment="1" Direction="Up" ZeroSuppression="Enable" UpdateIP="0" UpdateUnit="1"**

**//      Multiplier ="2" CountSkip="0" Reset="0001" ExternalSignal="Disable" ResetSignal="Signal 1" />**

**SetAttribute(ClassCode.Count, (byte)ccCount.Initial\_Value, "0001");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Count\_Range\_1, "0000");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Count\_Range\_2, "9999");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Jump\_From, "6666");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Jump\_To, "7777");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Increment\_Value, 1);**

**SetAttribute(ClassCode.Count, (byte)ccCount.Direction\_Value, "Up");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Zero\_Suppression, "Enable");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Update\_Unit\_Halfway, 0);**

**SetAttribute(ClassCode.Count, (byte)ccCount.Update\_Unit\_Unit, 1);**

**SetAttribute(ClassCode.Count, (byte)ccCount.Count\_Multiplier, "2");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Reset\_Value, "0001");**

**SetAttribute(ClassCode.Count, (byte)ccCount.Type\_Of\_Reset\_Signal, "Signal 1");**

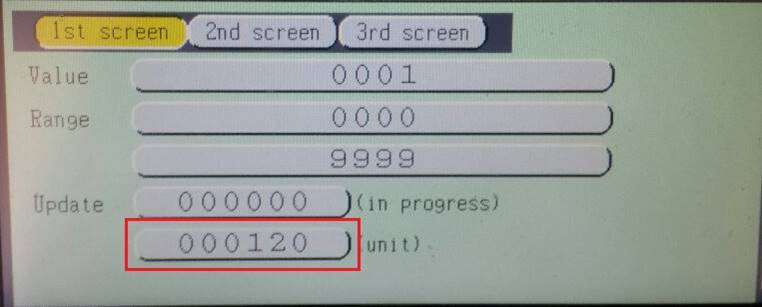
**SetAttribute(ClassCode.Count, (byte)ccCount.External\_Count, "Disable");**

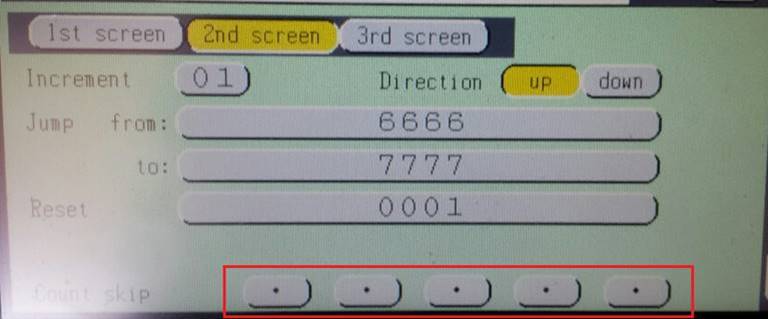
**SetAttribute(ClassCode.Count, (byte)ccCount.Count\_Skip, "0");**

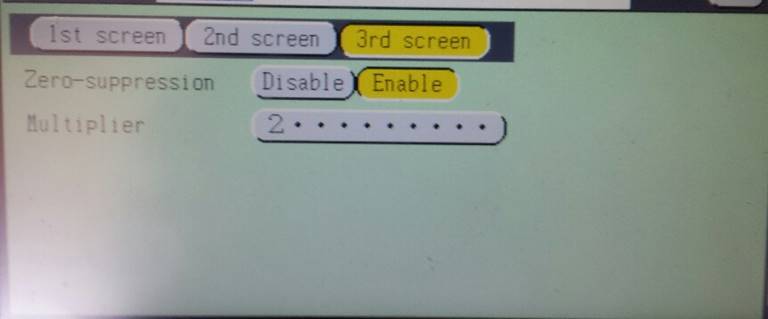
EtherNet/IP

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item | 2 | 1 | 00 01 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix | 1 | 5x8 | 03 |
| 00 -- O.K. -- 32 67 01 75 | Set | Print\_format | InterCharacter\_Space | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String | 4 | 1 | 20 31 20 00 |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item | 2 | 1 | 00 01 |
| 00 -- O.K. -- 32 7A 01 6E | Set | Index | Count\_Block | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix | 1 | 5x8 | 03 |
| 00 -- O.K. -- 32 67 01 75 | Set | Print\_format | InterCharacter\_Space | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String | 9 | {{CCCC}} | 7B 7B 43 43 43 43 7D 7D 00 |
| 00 -- O.K. -- 32 79 01 67 | Set | Count | Initial\_Value | 5 | 0001 | 30 30 30 31 00 |
| 00 -- O.K. -- 32 79 01 68 | Set | Count | Count\_Range\_1 | 5 | 0000 | 30 30 30 30 00 |
| 00 -- O.K. -- 32 79 01 69 | Set | Count | Count\_Range\_2 | 5 | 9999 | 39 39 39 39 00 |
| 00 -- O.K. -- 32 79 01 6E | Set | Count | Jump\_From | 5 | 6666 | 36 36 36 36 00 |
| 00 -- O.K. -- 32 79 01 6F | Set | Count | Jump\_To | 5 | 7777 | 37 37 37 37 00 |
| 00 -- O.K. -- 32 79 01 6C | Set | Count | Increment\_Value | 1 | 1 | 01 |
| 00 -- O.K. -- 32 79 01 6D | Set | Count | Direction\_Value | 1 | Up | 01 |
| 00 -- O.K. -- 32 79 01 73 | Set | Count | Zero\_Suppression | 1 | Enable | 01 |
| 00 -- O.K. -- 32 79 01 6A | Set | Count | Update\_Unit\_Halfway | 3 | 0 | 00 00 00 |
| 00 -- O.K. -- 32 79 01 6B | Set | Count | Update\_Unit\_Unit | 3 | 1 | 00 00 01 |
| 00 -- O.K. -- 32 79 01 74 | Set | Count | Count\_Multiplier | 2 | 2 | 32 00 |
| 00 -- O.K. -- 32 79 01 70 | Set | Count | Reset\_Value | 5 | 0001 | 30 30 30 31 00 |
| 00 -- O.K. -- 32 79 01 71 | Set | Count | Type\_Of\_Reset\_Signal | 1 | Signal 1 | 01 |
| 00 -- O.K. -- 32 79 01 72 | Set | Count | External\_Count | 1 | Disable | 00 |
| 00 -- O.K. -- 32 79 01 75 | Set | Count | Count\_Skip | 2 | 0 | 30 00 |

Here is the view from the printer







Update Unit always get set to 120 no matter what value is set.

I do not understand Count Skip so have no idea on what should be displayed.

Here is the data when it is read back:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** |
| 00 -- O.K. -- 33 75 01 6F | Get | IJP\_operation | Online\_Offline | 1 | 1 | 01 |
| 00 -- O.K. -- 33 7A 01 65 | Get | Index | Automatic\_reflection | 1 | 0 | 00 |
| 00 -- O.K. -- 33 7A 01 64 | Get | Index | Start\_Stop\_Management\_Flag | 1 | 1 | 01 |
| 00 -- O.K. -- 33 79 01 66 | Get | Count | Number\_Of\_Count\_Block | 1 | 1 | 01 |
| 00 -- O.K. -- 33 79 01 67 | Get | Count | Initial\_Value | 4 | "0001" | 30 30 30 31 |
| 00 -- O.K. -- 33 79 01 68 | Get | Count | Count\_Range\_1 | 4 | "0000" | 30 30 30 30 |
| 00 -- O.K. -- 33 79 01 69 | Get | Count | Count\_Range\_2 | 4 | "9999" | 39 39 39 39 |
| 00 -- O.K. -- 33 79 01 6A | Get | Count | Update\_Unit\_Halfway | 3 | 0 | 00 00 00 |
| 00 -- O.K. -- 33 79 01 6B | Get | Count | Update\_Unit\_Unit | 3 | 120 | 00 00 78 |
| 00 -- O.K. -- 33 79 01 6C | Get | Count | Increment\_Value | 1 | 1 | 01 |
| 00 -- O.K. -- 33 79 01 6D | Get | Count | Direction\_Value | 1 | 1 | 01 |
| 00 -- O.K. -- 33 79 01 6E | Get | Count | Jump\_From | 4 | "6666" | 36 36 36 36 |
| 00 -- O.K. -- 33 79 01 6F | Get | Count | Jump\_To | 4 | "7777" | 37 37 37 37 |
| 00 -- O.K. -- 33 79 01 70 | Get | Count | Reset\_Value | 4 | "0001" | 30 30 30 31 |
| 00 -- O.K. -- 33 79 01 71 | Get | Count | Type\_Of\_Reset\_Signal | 4 | 808464433 | 30 30 30 31 |
| 00 -- O.K. -- 33 79 01 72 | Get | Count | External\_Count |  | 0 |  |
| 00 -- O.K. -- 33 79 01 73 | Get | Count | Zero\_Suppression | 1 | 1 | 01 |
| 00 -- O.K. -- 33 79 01 74 | Get | Count | Count\_Multiplier | 1 | "2" | 32 |
| 00 -- O.K. -- 33 79 01 75 | Get | Count | Count\_Skip | 1 | " " | 20 |

Everything comes back as expected except:

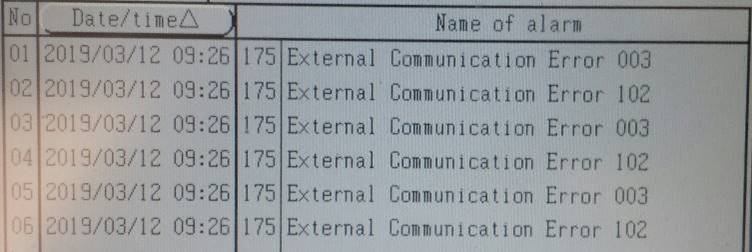
Update Unit Unit is 120

Type of reset signal is “000”

External count returns no data.

Count Skip is a Space “ “ (0x20)

The printer fault log looks like



The COM Errors are caused by

SetAttribute(ClassCode.Count, (byte)ccCount.Update\_Unit\_Halfway, 0);           // Causes COM Error

SetAttribute(ClassCode.Count, (byte)ccCount.Update\_Unit\_Unit, 1);              // Causes COM Error

SetAttribute(ClassCode.Count, (byte)ccCount.Type\_Of\_Reset\_Signal, "Signal 1"); // Causes COM Error

SetAttribute(ClassCode.Count, (byte)ccCount.External\_Count, "Disable");        // Causes COM Error

Any assistance would be appreciated.

# **Calendar Block**

I am trying to set up the Calendar Block using EtherNet/IP.  There are a lot of parts to it.

Here is the code for step 1.  Setting Month and Day-of-Week in substitution Rule #2

**// Set <Substitution Rule="01" StartYear="2010" Delimeter="/">**

**char delimeter = '/';**

**SetAttribute(ClassCode.Index, (byte)ccIDX.Substitution\_Rules\_Setting, Rule);**

**SetAttribute(ClassCode.Substitution\_rules, (byte)ccSR.Start\_Year, 2010);**

**// Set <Month Base="1">JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT/NOV/DEC</Month>**

**string[] months = "JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT/NOV/DEC".Split(delimeter);**

**for (int i = 0; i < months.Length; i++) {**

**SetAttribute(ClassCode.Substitution\_rules, (byte)ccSR.Month, i + 1, months[i]);**

**}**

**// Set <DayOfWeek Base="1">MON/TUE/WED/THU/FRI/SAT/SUN</DayOfWeek>**

**string[] day = "MON/TUE/WED/THU/FRI/SAT/SUN".Split(delimeter);**

**for (int i = 0; i < day.Length; i++) {**

**SetAttribute(ClassCode.Substitution\_rules, (byte)ccSR.Day\_Of\_Week, i + 1, day[i]);**

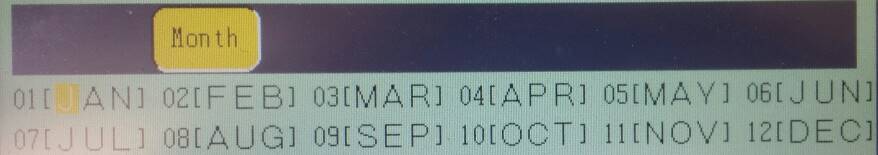
**}**

Here is the traffic sent to the printer

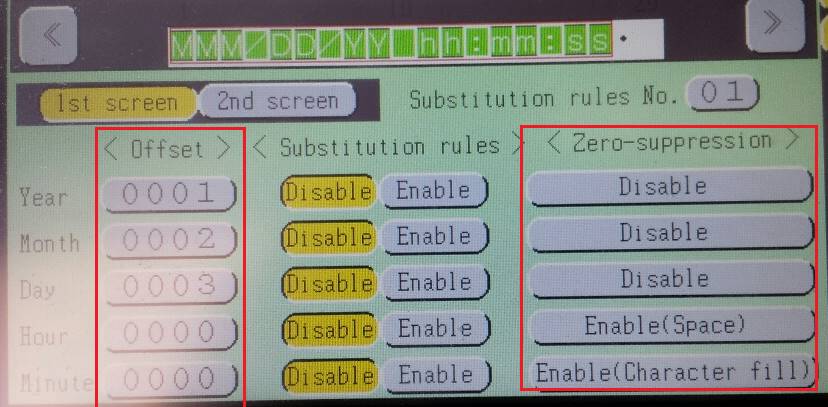
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Count OK** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 6C | True | Index | Substitution\_Rules\_Setting | 1 | 2 | 02 |
| 00 -- O.K. -- 32 6C 01 66 | True | Substitution\_rules | Start\_Year | 2 | 2010 | 07 DA |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | JAN | 01 4A 41 4E 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | FEB | 02 46 45 42 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | MAR | 03 4D 41 52 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | APR | 04 41 50 52 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | MAY | 05 4D 41 59 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | JUN | 06 4A 55 4E 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | JUL | 07 4A 55 4C 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | AUG | 08 41 55 47 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | SEP | 09 53 45 50 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | OCT | 0A 4F 43 54 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | NOV | 0B 4E 4F 56 00 |
| 00 -- O.K. -- 32 6C 01 68 | True | Substitution\_rules | Month | 5 | DEC | 0C 44 45 43 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | MON | 01 4D 4F 4E 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | TUE | 02 54 55 45 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | WED | 03 57 45 44 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | THU | 04 54 48 55 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | FRI | 05 46 52 49 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | SAT | 06 53 41 54 00 |
| 00 -- O.K. -- 32 6C 01 6D | True | Substitution\_rules | Day\_Of\_Week | 5 | SUN | 07 53 55 4E 00 |

Worked perfectly if I use substitution rule #1.  I can set Substitution rule #2 and they appear in the printer but cannot read them back.

Here is what the screen looked like



After the entire message was sent, Here is what the calendar block looked like.



Offsets Code:

**// Set <Offset Year="1" Month="2" Day="3" Hour="4" Minute="-5" />**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Offset\_Year, 1);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Offset\_Month, 2);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Offset\_Day, 3);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Offset\_Hour, 4);**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Offset\_Minute, -5);**

Substitution code

**// Set <EnableSubstitution SubstitutionRule="01" Year="False" Month="True"  Day="False"**

**//      Hour ="False" Minute="False" Week="False" DayOfWeek="False" />**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Substitute\_Year, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Substitute\_Month, "Enable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Substitute\_Day, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Substitute\_Hour, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Substitute\_Minute, "Disable");**

Zero Suppression code

**// Set <ZeroSuppress Year="Disable" Month="Disable" Day="Disable"**

**//      Hour ="Space Fill" Minute="Character Fill" />**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Zero\_Suppress\_Year, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Zero\_Suppress\_Month, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Zero\_Suppress\_Day, "Disable");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Zero\_Suppress\_Hour, "Space Fill");**

**SetAttribute(ClassCode.Calendar, (byte)ccCal.Zero\_Suppress\_Minute, "Character Fill");**

The traffic

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item | 2 | 1 | 00 01 |
| 00 -- O.K. -- 32 7A 01 6C | Set | Index | Substitution\_Rules\_Setting | 1 | 2 | 02 |
| 00 -- O.K. -- 32 7A 01 6F | Set | Index | Calendar\_Block | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix | 1 | 5x8 | 03 |
| 00 -- O.K. -- 32 67 01 75 | Set | Print\_format | InterCharacter\_Space | 1 | 1 | 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String | 33 | {{MMM}/{DD}/{YY} {hh}:{mm}:{ss}} | 7B 7B 4D 4D 4D 7D 2F 7B 44 44 7D 2F 7B 59 59 7D |
| 00 -- O.K. -- 32 69 01 74 | Set | Calendar | Substitute\_Year | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 75 | Set | Calendar | Substitute\_Month | 1 | Enable | 01 |
| 00 -- O.K. -- 32 69 01 76 | Set | Calendar | Substitute\_Day | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 77 | Set | Calendar | Substitute\_Hour | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 78 | Set | Calendar | Substitute\_Minute | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 68 | Set | Calendar | Offset\_Year | 1 | 1 | 01 |
| 00 -- O.K. -- 32 69 01 69 | Set | Calendar | Offset\_Month | 1 | 2 | 02 |
| 00 -- O.K. -- 32 69 01 6A | Set | Calendar | Offset\_Day | 2 | 3 | 00 03 |
| 00 -- O.K. -- 32 69 01 6B | Set | Calendar | Offset\_Hour | 2 | 4 | 00 04 |
| 00 -- O.K. -- 32 69 01 6C | Set | Calendar | Offset\_Minute | 2 | -5 | FF FB |
| 00 -- O.K. -- 32 69 01 6D | Set | Calendar | Zero\_Suppress\_Year | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 6E | Set | Calendar | Zero\_Suppress\_Month | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 6F | Set | Calendar | Zero\_Suppress\_Day | 1 | Disable | 00 |
| 00 -- O.K. -- 32 69 01 70 | Set | Calendar | Zero\_Suppress\_Hour | 1 | Space Fill | 01 |
| 00 -- O.K. -- 32 69 01 71 | Set | Calendar | Zero\_Suppress\_Minute | 1 | Character Fill | 02 |

Offset out for Hour (+4) and Minute (-5) were not set.  Zero suppression was all set properly.

All could be read back getting the values shown on the printer screen.

I am working to understand the Substitution rule settings.

* The substitution rule was not set in the calendar block
* Substitution rule settings that I sent were not used

Any input would be greatly appreciated.

Moving cijConnect from the Hitachi (Serial) protocol to the EtherNet/IP protocol will require a lot of testing.  Here are a couple of things that would be helpful:

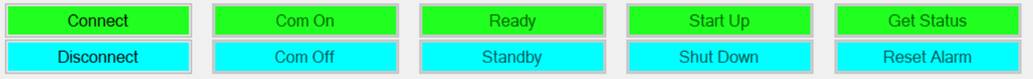
* Request status
  + The Serial Protocol returned two different status values
    - ACK == The request was accepted
    - NAK == The request was rejected
  + The EtherNet/IP Protocol just returns “Success” and returns data even if the request was rejected.  The data returned is not meaningful.
  + A helpful change would be for the EtherNet/IP Protocol to
    - return NAK if the request was rejected and return no data
* COM Errors
  + The Serial Protocol generated COM errors when a NAK was returned.  Without the SOP-04 Kit, there was no way to determine the COM Error.
  + The EtherNet/IP Protocol sets the “S/S Management Flag” to “1”
  + A helpful change would be for the EtherNet/IP Protocol to
    - Provides a method of reading the number of COM errors that current exist
    - Provide a method of reading the COM error information
    - Provide a method to clear the COM error list.

These two changes would make the move to EtherNet/IP Protocol faster, easier, and more reliable.

# **Debugging cijConnect**

How do I accomplish this?

cijConnect provides a number of controls for controlling the state of the printer using the Hitachi (Serial) Protocol:

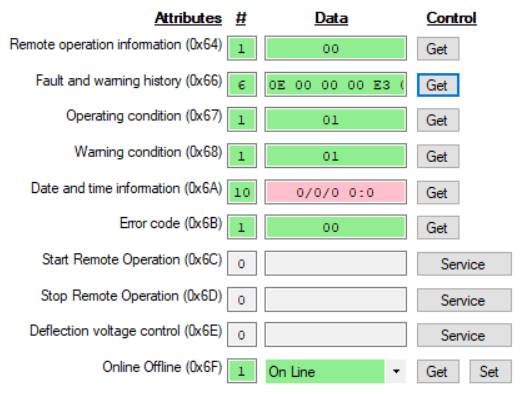


I would like to continue to provide them for the EtherNet/IP Protocol.

* Connect / Disconnect == These will probably not be provided in the new protocol.  There is no unsolicited input from the printer, so there is no need to keep the connection open.  The EtherNet/IP Protocol provides for a Session/Forward envelope around the requests for burst type transmission
* Com On / Com Off == is provided 0x75 0x6F
* Ready / Standby == There is a Service called “Deflection Voltage Control” but I do not know what it does.
* Start Up / Shut Down == There are two Service Call for “Start Remote Operation” and “Stop Remote Operation” that will work.  Curious as to why it is not a Get/Set like Online/Offline State?
* Get Status == There are a number of flags on the IJP Operation Function.  Would be nice to retrieve then with a single Get request that had the same components as the Serial Protocol.
* Reset Alarm == In the Serial Interface, there was a Fault Clear.  I do not see an equivalent Service request

# **IJP Operation Function**

For the IJP Operation Function in general



Here are the Attribute codes and retrieved values

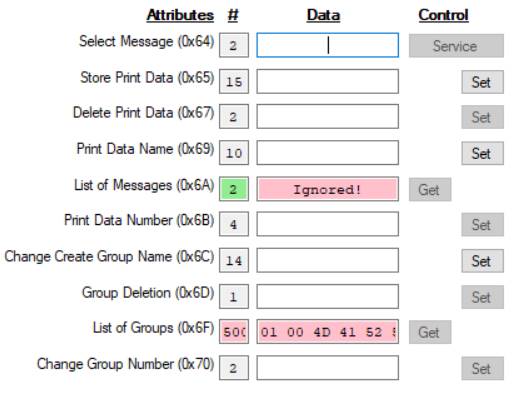
* 0x64 / 00 == Do not know what it means.  Reception not possible?
* 0x66 /  0E 00 00 00 E3 07 == Do not know what it means
  + 0E == 14 which is the number of Com errors currently in my printer
  + 00 00 00 == Do not know what they mean
  + E3 07 == 2019 in Little End format
* 0x67 / 01 == In standby?
* 0x68 / 01 == Ink Low warning? (I have no ink in my printer)
* 0x6A / 00 00 00 00 00 00 00 00 00 00 == Sometimes it shows a data and time
* 0x6B == Always shows 00.
* 0x6C & 0x6D == I understand
* 0x6E == What does this do?
* 0x6F == Works fine

Changing the Item Number in the Index Function seems to change some values but have not figured a pattern.

Would like to see a service call to clear all entries from the Alarm History.

# **Print Data Management**

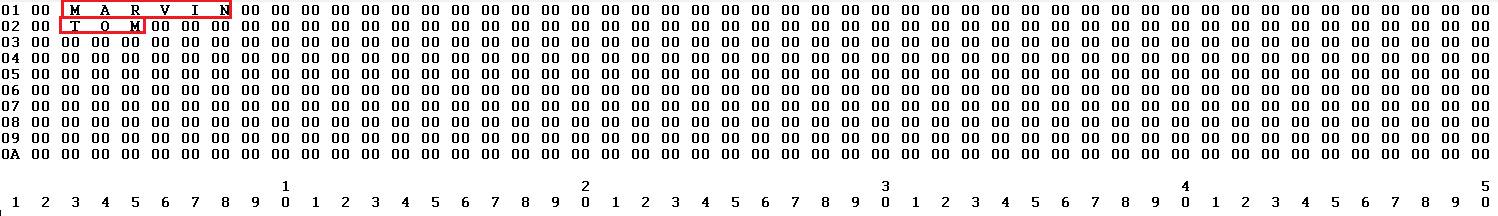
My observations working with Print Data Management Function



Attribute number and comment:

* 0x64 == Works as expected
* 0x67 – Works as expected
* 0x6A == Causes printer to hang.  Must turn EtherNet/IP off and back on to clear issue (A reset function would be nice)
* 0x6F == Need to specify 0 to 99, not 1 to 99.  See output below.
* The rest have not been tested yet.

For 0x6F, I created two groups (1 – MARVIN, 2 – TOM) and put messages in them.  Then I read them using 33 66 01 6F 00.



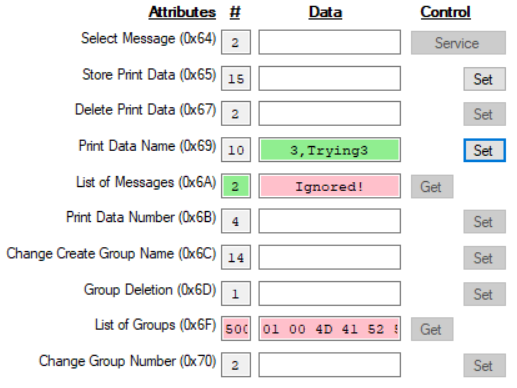
I got:

* 10 groups of 50 bytes (500 bytes in all)
* Each group contained:
  + Two byte group number in Little Endian format
  + The name in UTF8 format
  + Padding with nulls up to 48 bytes.

Marvin

# **Print Format Function**

Except for previously noted in Hitachi case 00071400, everything works as advertised except:



Attribute

* 0X65 == References =>Type + Group Number + Nickname + “00”<=.  What is type, how is it formatted for output, and what does this command accomplish?  I cannot figure it out.
* 0x69 == seems to act as follows:
  + Can change an existing name
  + If name is blank, the command is ignored but COM Error 006
  + If name duplicates another name in the list, the request is ignored but COM Error 006.

Would be nice to get a NAK if the request is not honored.

Does the 0x65 save the message or is there a command somewhere to cause the message to be saved?

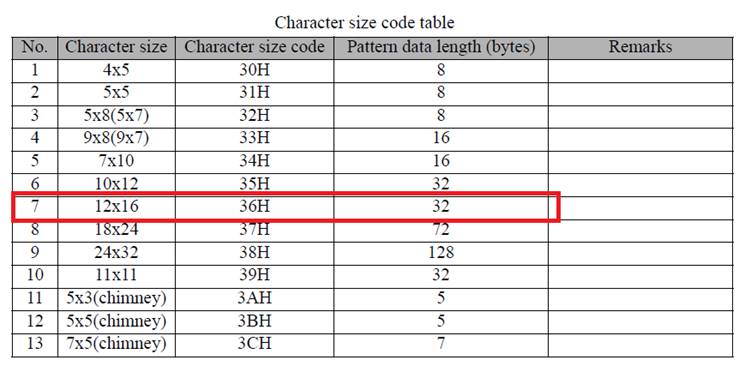
# **Print Format Function**

Some success.  I am trying to send and receive the following user trade mark pattern to the printer.

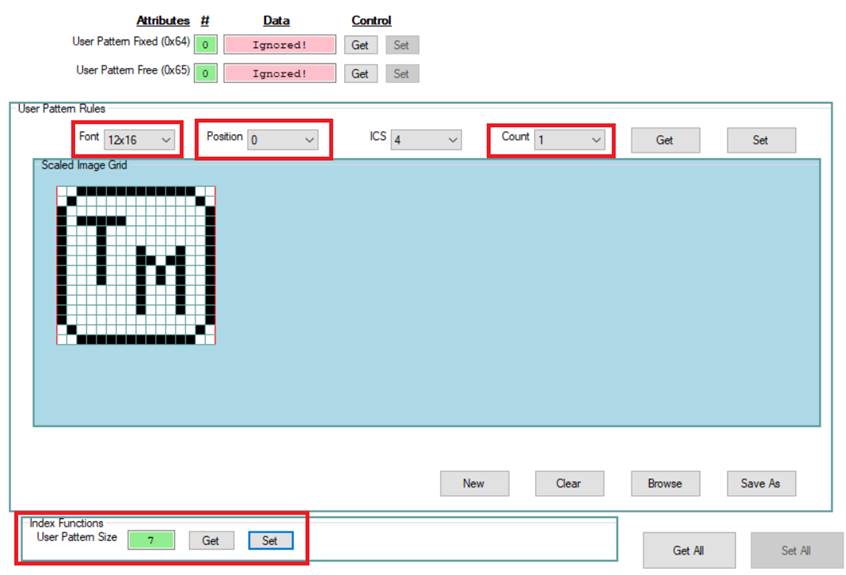
Specification

* Font = 12x16
* Position = 0
* Count = 1

From the RX Technical manual:



From the Hitachi Browser:



Attempt to send:

* Command = 32 6B 01 64
* Data = 07 00 FC 3F 02 40 01 90 01 90 C1 9F 01 90 01 90 01 80 F9 83 01 81 C1 80 01 81 F9 83 01 80 02 40 FC 3F
  + 07 = Font Size
  + 00 = Position
  + FC … 3F = 32 byte pattern

Result:

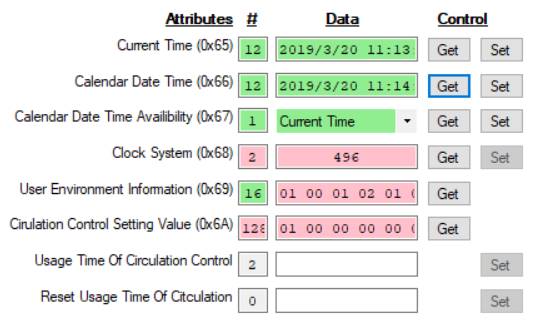
* The pattern reached the printer  
  
* Response = “Unable to read data from the transport connection: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.".

Attempt to read:

* Command = 33 6B 01 64
* Data = 07 00
  + 07 = Font Size 12x16
  + 00 = Position
* Response = “Unable to read data from the transport connection: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond."

# **Environment Settings**

Trying the Environment settings:

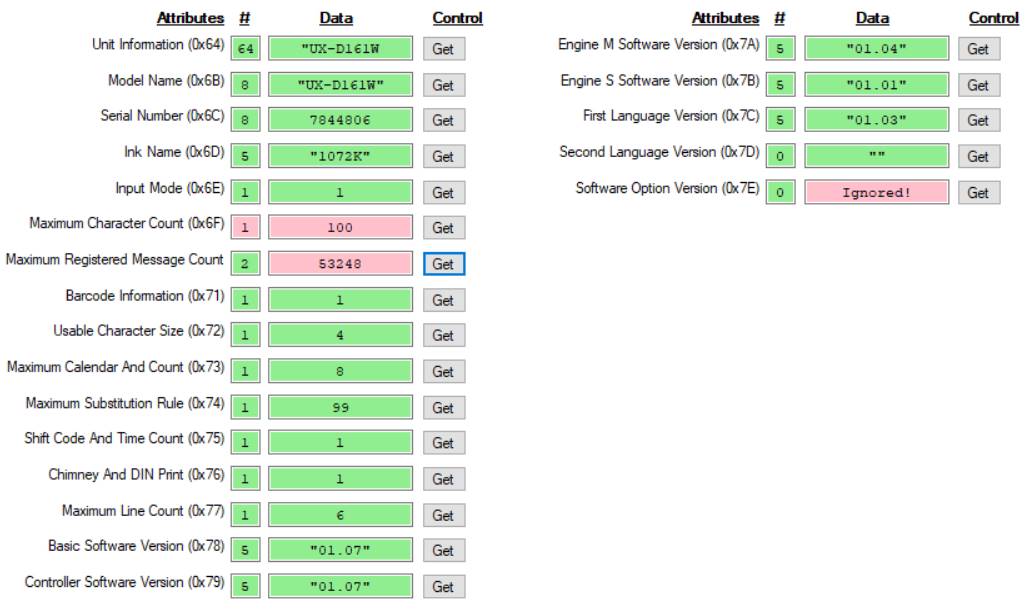


Attributes and comments

* 0x65 & 0x66 == Get
  + Command = 33 71 01 65
  + Response = E3 07 03 00 14 00 0B 00 10 00 31 00  
            YYYYY MMMMM DDDDD hhhhh mmmmm sssss
  + Two bytes each in Little Endian format
* 0x65 == Set (attempt to change month )
  + Command = 32 71 01 65
  + Data = E3 07 04 00 14 00 0B 00 10 00 31 00  
         YYYYY MMMMM DDDDD hhhhh mmmmm sssss
  + Response = Success The time was not changed in the printer
* 0x67 == Worked
* 0x68 == has an issue
  + Get == 33 71 01 68
  + Response = 01 F0 (an extra byte is returned.  The first byte is correct)
  + Set = 32 71 01 68
  + Data = 02
  + Response = Success (the clock system was changed)
* 0x69 & 0x6A == Need documentation
* 0x6B & 0x6C == Do not know how to test or when I would use them.

# **Unit Information**

The Unit information function has a few issues



Attribute and comments

* 0x64 == Returns 64 bytes
  + Get = 33 73 01 64
  + Response = 55 58 2D 44 31 36 31 57 00 00 00 00 C6 B3 77 00 31 30 37 32 4B 2B DD BE 69 B2 0D 2C 53 31 30 31 38 2B DD BE 69 B2 0D 00 00 00 00 00 02 00 E8 03 D0 07 01 00 04 00 08 00 63 00 01 00 01 00 06 00
    - 55 58 2D 44 31 36 31 57 ("UX-D161W")
    - 00 00 00 00 C6 B3 77 00 == 0x6D Response( 7844806 machine serial number in some unknown format)
    - 31 30 37 32 4B = 0x6D Response (“1072K”)
    - 2C 53 31 30 31 38 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 == Just hanging here.  Do not know where it goes.  Might be part of the ink type (“,S1018”)
    - 02 00 == 0x6E Response (This is what is actually in the machine but 0x6E reports “01”)
    - E8 03 == 0x6F Response (This is 1000 and should be the response for 0x6F.  But 0x6f reports a “64”)
    - D0 07 == 0x70 Response (This is 2000 and should be the response for 0x70.  But 0x70 reports “D0 00”)
    - 01 00 == 0x71 Response valid
    - 04 00 == 0x72 Response valid
    - 08 00 == 0x73 Response valid
    - 63 00 == 0x74 Response valid
    - 01 00 == 0x75 Response valid
    - 01 00 == 0x76 Response valid
    - 06 00 == 0x77 Response valid
* 0x6B thru 0x77 are described above
* 0x78 thru 0x7d == Work fine
* 0x7E == Causes a printer hang.  {“Unable to read data from the transport connection: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.")

# **Print Format Function**

There are two sections to print format

* Message Description (In red) (Should this be moved to print specification?)
* Item Description (In green)



Attributes and comments in red section:

* 0x64 == Attempting to read the message name hangs up the printer {“Unable to read data from the transport connection: A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.")
* 0x65 == Working properly
* 0x66 == Reports number of items when in free layout
* 0x67 & 0x6D == These two move in lock step with each other.  Both take on values 1 to 3. (Not 0 to 2).  Why are two needed?
* 0x69 thru 0x6F == Service requests.  Still trying all of them.  The serial interface had a “Delete all but one” command.  Would be nice here.  Otherwise, I have to keep deleting columns, etc. to clear the display
* 0x6C == Not sure what it does.  Need help here.

Attributes and comments in green section:

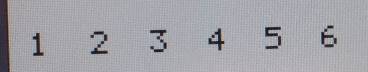
* Several of these do not always apply.  It would be nice to get a NAK back if they do not apply for the layout involved.
* 0x7A == X/Y coordinates.  Would be nice to be able to read then when Individual and Overall layout is being used.  Could then:
  + send a message to the printer
  + read the X/Y coordinates for each item
  + Adjust cijConnect’s display to reflect where the printer actually placed them
* 0X7b == Could not get a response even when I set the message to variable ICS manually in the printer.

# **Add/Delete/Insert columns**

Comments on column creation and deletion

**1 == Add Column (34 67 01 6B) == Adds a new column at the end of the current list of columns**

Before



Commands

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column | 0 |  |  |

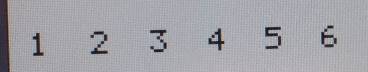
After

cid:image005.jpg@01D4E2EE.6491AF30

Works as expected.

**2 == Insert Column (34 67 01 69) == Uses Index Function (Column) to insert a column**

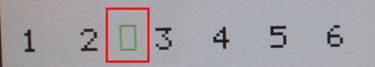
Before



Commands

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column | 2 | 3 | 00 03 |
| 00 -- O.K. -- 34 67 01 69 | Service | Print\_format | Insert\_Column | 0 |  |  |

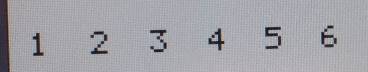
After



New item added at the third position.  Column index uses 1-origin positioning.

**3 == Delete Column (34 67 01 6A)  == Uses Index Function (Column) to delete a column**

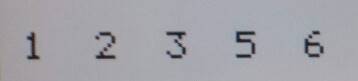
Before



Commands

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#Out** | **Data Out** | **Raw Out** |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column | 2 | 3 | 00 03 |
| 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column | 0 |  |  |

After



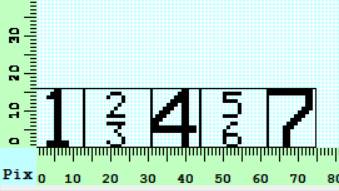
Column was deleted at the fourth position.  Column index uses 0-origin positioning.

# **Multi-line setup**

For cijConnect, messages with multiple lines were set up as follows

* The message was cleared using “Delete all items but one”.
* The message was built as a single row(line)
* Text Setup Rules / Line Count was used to stack the items as needed.

Consider the message (an actual message layout but I changed the text):



To build the message with EtherNet/IP

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Status/Path** | **Access** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** | **#Out** | **Data Out** | **Raw Out** |
| **Normal startup.  Be sure com is on, get other settings of interest** | | | | | | | | | |
| 00 -- O.K. -- 32 75 01 6F | Set | IJP\_operation | Online\_Offline |  |  |  | 1 | 1 | 01 |
| 00 -- O.K. -- 33 75 01 6F | Get | IJP\_operation | Online\_Offline | 1 | 1 | 01 |  |  |  |
| 00 -- O.K. -- 33 7A 01 65 | Get | Index | Automatic\_reflection | 1 | 0 | 00 |  |  |  |
| 00 -- O.K. -- 33 7A 01 64 | Get | Index | Start\_Stop\_Management\_Flag | 1 | 0 | 00 |  |  |  |
| 00 -- O.K. -- 33 7A 01 64 | Get | Index | Start\_Stop\_Management\_Flag | 1 | 0 | 00 |  |  |  |
| 00 -- O.K. -- 33 7A 01 65 | Get | Index | Automatic\_reflection | 1 | 0 | 00 |  |  |  |
| **Read the number of columns and then delete all but one** | | | | | | | | | |
| 00 -- O.K. -- 33 67 01 66 | Get | Print\_format | Number\_Of\_Columns | 1 | 5 | 05 |  |  |  |
| 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 1 | 01 |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 4 | 00 04 |
| 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 3 | 00 03 |
| 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 2 | 00 02 |
| 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 1 | 00 01 |
| 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 1 | 01 |
| **Now process all the commands** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 0 | 00 |
| 00 -- O.K. -- 32 7A 01 64 | Set | Index | Start\_Stop\_Management\_Flag |  |  |  | 1 | 2 | 02 |
| **Process only after all the information has been sent** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 1 | 01 |
| **Add four more columns** | | | | | | | | | |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  | 0 |  |  |
| 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  | 0 |  |  |
| **Stack up columns 2 and 4** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| **Add print string and font for item #1** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "1" | 31 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| **Add print string and font for item #2** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 2 " | 20 32 20 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| **Add print string and font for item #3** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 3 | 00 03 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 3 " | 20 33 20 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| **Add print string and font for item #4** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "4" | 34 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| **Add print string and font for item #5** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 5 | 00 05 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 5 " | 20 35 20 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| **Add print string and font for item #6** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 6 | 00 06 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 6 " | 20 36 20 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| **Add print string and font for item #7** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 7 | 00 07 |
| 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "7" | 37 00 |
| 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| **Now process all the commands** | | | | | | | | | |
| 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 0 | 00 |
| 00 -- O.K. -- 32 7A 01 64 | Set | Index | Start\_Stop\_Management\_Flag |  |  |  | 1 | 2 | 02 |

The steps needed to accomplish the task

* Be sure COM is on
* Delete any existing message
  + Delete all the columns in the current message except 1
  + Set the number of lines in the first column to 1.  To do this, the column number (in this case 1) has to be set as the Index/Item value (not the Index/Column) before setting the lines to 1
* Add four columns to the message
* Set number of lines to 2 for columns 2 and 4.  Again, Index/Item is set to the column number.
* The last thing is to step thru the items setting font and text.

It can be done.  The main issue is:

* To set the line number, the column number is set in Index/Item Count.
* To read the line number, the column number must be set in Index/Column

What would be nice to see

* A single “Delete all but one” to clean up the old message
* Have setting the number of lines stack existing items like the old Serial Protocol.
* Have Set and Get of number of lines use Index/Columns

One last issue.  For the Index function class (7A 66), the description says “Item Count”.  It is used for indexing through many lists.  It is used for Items but has nothing to do with count.  Should be renamed ad “Index”

.

# **Auto Reflection issue**

Using Auto Reflection is 4 times faster that without Auto Reflection.  However, there is a problem with Get access when reflection is on.

Saving away a Get to execute later is meaningless since there is no way for the result of a get to be pass back using the EtherNet/IP protocol.  For that reason, the Get request should ignore Auto Reflection flag and pass back the Get data immediately.  The returned data should reflect the state of the machine at the time Auto Reflection was set to 1.

A case where it would be useful:

* A complete message is being sent to the printer
* I request the number of columns in the old message
* That number is then used to delete the old message (down to a single item)

In order to do this, I now have to:

* turn off Auto Reflection
* Do a Get
* Turn on auto reflection and continue.

This causes the printer to partially build the message rather than build it all at once.

Below is the traffic logs with Auto Reflection On (0.9804 sec) and then Off (4.6054 sec).

A Note:  In the traffic file, the Elapsed time for a Get/Set/Service is relative to the last request.  The Elapsed time for a Forward Close is relative to the Forward Open.  Same for Session and Connection.

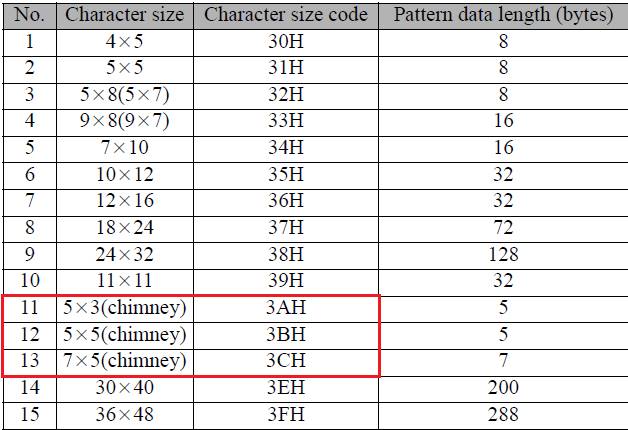
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Elapsed** | **Status/Path** | **Access** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** | **#Out** | **Data Out** | **Raw Out** |
| 4.8003 | Connection Open! |  |  |  |  |  |  |  |  |  |
| 4.8143 | Session Open! |  |  |  |  |  |  |  |  |  |
| 4.8273 | Forward Open! |  |  |  |  |  |  |  |  |  |
| 0.0290 | 00 -- O.K. -- 33 67 01 66 | Get | Print\_format | Number\_Of\_Columns | 1 | 5 | 05 |  |  |  |
| 0.0200 | 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 1 | 01 |
| 0.0240 | 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 1 | 00 01 |
| 0.0210 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.0300 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.0170 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.0250 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.0260 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 0.0160 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 1 | 01 |
| 0.0320 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.0210 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.0340 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.0210 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.0290 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 0.0190 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| 0.0240 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 0.0280 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| 0.0120 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 0.0240 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "1" | 31 00 |
| 0.0360 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 0.0150 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 0.0190 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 2 " | 20 32 20 00 |
| 0.0250 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0250 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 3 | 00 03 |
| 0.0180 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 3 " | 20 33 20 00 |
| 0.0280 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0230 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 0.0190 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "4" | 34 00 |
| 0.0190 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 0.0270 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 5 | 00 05 |
| 0.0200 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 5 " | 20 35 20 00 |
| 0.0260 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0320 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 6 | 00 06 |
| 0.0160 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 6 " | 20 36 20 00 |
| 0.0290 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0190 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 7 | 00 07 |
| 0.0140 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "7" | 37 00 |
| 0.0160 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 0.0120 | 00 -- O.K. -- 32 7A 01 65 | Set | Index | Automatic\_reflection |  |  |  | 1 | 0 | 00 |
| 0.0550 | 00 -- O.K. -- 32 7A 01 64 | Set | Index | Start\_Stop\_Management\_Flag |  |  |  | 1 | 2 | 02 |
| 0.9804 | Forward Close! |  |  |  |  |  |  |  |  |  |
| 0.9994 | Session Close! |  |  |  |  |  |  |  |  |  |
| 1.0154 | Connection Close! |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Elapsed** | **Status/Path** | **Access** | **Class** | **Attribute** | **#In** | **Data In** | **Raw In** | **#Out** | **Data Out** | **Raw Out** |
| 8.4068 | Connection Open! |  |  |  |  |  |  |  |  |  |
| 8.4228 | Session Open! |  |  |  |  |  |  |  |  |  |
| 8.4328 | Forward Open! |  |  |  |  |  |  |  |  |  |
| 0.0190 | 00 -- O.K. -- 33 67 01 66 | Get | Print\_format | Number\_Of\_Columns | 1 | 5 | 05 |  |  |  |
| 0.0140 | 00 -- O.K. -- 32 7A 01 67 | Set | Index | Column |  |  |  | 2 | 1 | 00 01 |
| 0.0780 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.1650 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.1560 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.1600 | 00 -- O.K. -- 34 67 01 6A | Service | Print\_format | Delete\_Column |  |  |  |  |  |  |
| 0.0410 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 0.1130 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 1 | 01 |
| 0.1010 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.1830 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.2380 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.1590 | 00 -- O.K. -- 34 67 01 6B | Service | Print\_format | Add\_Column |  |  |  |  |  |  |
| 0.0410 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 0.1200 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| 0.0520 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 0.1040 | 00 -- O.K. -- 32 67 01 72 | Set | Print\_format | Line\_Count |  |  |  | 1 | 2 | 02 |
| 0.0460 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 1 | 00 01 |
| 0.1200 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "1" | 31 00 |
| 0.1190 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 0.1180 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 2 | 00 02 |
| 0.1210 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 2 " | 20 32 20 00 |
| 0.2900 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0440 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 3 | 00 03 |
| 0.1040 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 3 " | 20 33 20 00 |
| 0.1880 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0430 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 4 | 00 04 |
| 0.0920 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "4" | 34 00 |
| 0.3250 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 0.0340 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 5 | 00 05 |
| 0.1810 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 5 " | 20 35 20 00 |
| 0.2190 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.0370 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 6 | 00 06 |
| 0.1430 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 4 | " 6 " | 20 36 20 00 |
| 0.2110 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 5x8 | 03 |
| 0.1040 | 00 -- O.K. -- 32 7A 01 66 | Set | Index | Item |  |  |  | 2 | 7 | 00 07 |
| 0.1140 | 00 -- O.K. -- 32 67 01 71 | Set | Print\_format | Print\_Character\_String |  |  |  | 2 | "7" | 37 00 |
| 0.1830 | 00 -- O.K. -- 32 67 01 74 | Set | Print\_format | Dot\_Matrix |  |  |  | 1 | 12x16 | 07 |
| 4.6054 | Forward Close! |  |  |  |  |  |  |  |  |  |
| 4.6203 | Session Close! |  |  |  |  |  |  |  |  |  |
| 4.6393 | Connection Close! |  |  |  |  |  |  |  |  |  |

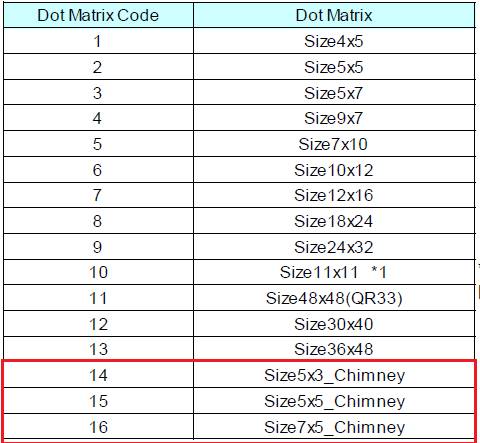
# **Character Size**

There are two different mappings for character size:

User Pattern Function (0x6B) Attributes 0x64 and 0x65 refer to Table 5.3.8-3 of the UX Technical Spec



Print Format Function (0x67) Attribute 0x74 refers to table 7.4 of the EtherNet/IP document



Issues with implementation:

* The UX Spec is 0-Origin == 0 thru 15 (missing 13 (3DH))
* The EIP Spec is 1-origin == 1 thru 16
* The location of the chimney fonts differs.

This causes issues with the implementation.  It would be nice for both tables to use the EIP Mapping.