

Hitachi UX Model 161 EtherNet/IP

A comprehensive example

Here is a complete example of

- Building a message,
- Sending it to the Hitachi UX 161 printer,
- Retrieving it from the Hitachi UX 161 printer,
- Comparing the Sent message to the Retrieved message.

This is all done using the EtherNet/IP protocol.

However, it uses 5 calendar objects as well as Shift Codes and Time Counts. These are features that require an extra key in the printer to activate them.

While I will try to keep this up to date, the code here and the code in the Hitachi Browser may differ from time to time.

Code Structure for a Test

The connection to the printer is not kept open. It is opened for a request and closed. If multiple requests are needed, the code need to be structured like this:

```
if (EIP.StartSession(true)) {  
    if (EIP.ForwardOpen()) {  
        try {  
  
            // Place your code here  
  
        } catch (EIPIOException e1) {  
            // In case of an EIP I/O error  
            string name = $"{EIP.GetAttributeName(e1.ClassCode, e1.Attribute)}";  
            string msg = $"EIP I/O Error on {e1.AccessCode}/{e1.ClassCode}/{name}";  
            MessageBox.Show(msg, "EIP I/O Error", MessageBoxButtons.OK);  
        } catch (Exception e2) {  
            // You are on your own here  
        }  
    }  
    EIP.ForwardClose();  
}  
EIP.EndSession();
```

After you have finished with the error processing, you must still execute the “ForwardClose” and “EndSession” to leave the printer in an idle state.

Within the code block using the above structure, you can call another method with the same structure. EIP will sort it out and only one connection will be established.

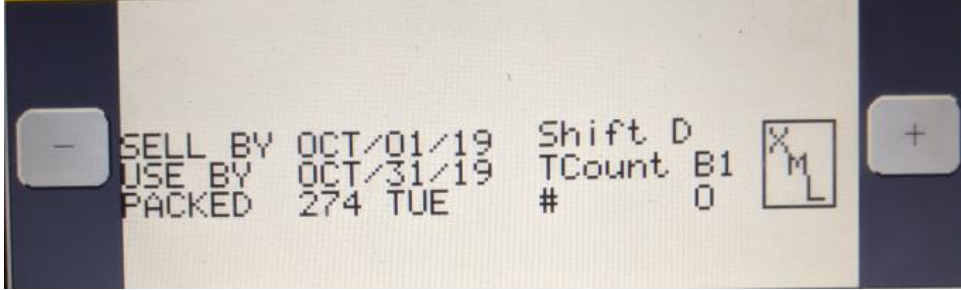
The parameter “true” on the “StartSession” tells EIP to report the “EIPIOException”. Otherwise the errors are ignored.

If only a single command is needed, that command can be issued without the above structure. EIP will open a connection, make the request, and close the connection.

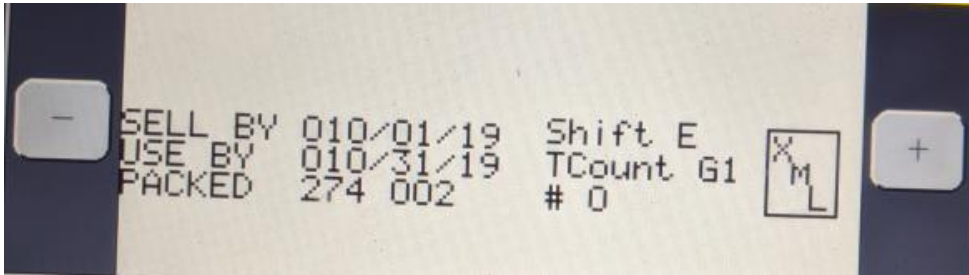
Building the message

The first step is to define a way of describing the message in an industry standard for that is independent of printer model. The method I chose is the Extensible Markup Language (XML). XML is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable.

The test message was built by cijConnect (plus a bit of manual manipulation at the printer) and looks like:



Attempting to build it with the same message with EtherNet/IP yielded



It has plain text, dates, counters, and a logo.

The overall layout of the XML document that will be used here is (<!-- ... --> is a comment):

```
<Label Version="1">
  <Printer Model="UX-D161W" Make="Hitachi">
    <!-- Data that applies to the whole message -->
    <TwinNozzle>
      <!-- Data that pertains to optional printer features -->
    </TwinNozzle />
    <Substitution Rule="01" StartYear="2019" Delimiter="/">
      <!-- Data that is stored in the substitution files in the printer -->
    </Substitution>
  </Printer>
  <Objects>
    <!-- Data unique to each item. Types are Text, Date, and Counter -->
  </Objects>
</Label>
```

All XML data is represented as

```
<Section Parameter="Value" ... >
```

Where “Section” defines the area of interest, Parameter defines the Attribute, and “Value” specifies the setting.

When the value is a number in the EtherNet/IP document, only a number can be specified. If the value has a discrete set of symbolic values, either the discrete value or the associated numerical value can be specified.

Printer Section

The printer section of the XML document generally reflects the information in the Print Specification Class (0x68) of the Hitachi EtherNet/IP manual. An example of the XML:

```
<Printer Model="UX-D161W" Make="Hitachi">
  <PrintHead Orientation="Inverted/Forward" />
  <ContinuousPrinting RepeatInterval="0" PrintsPerTrigger="0" />
  <TargetSensor Filter="Until End of Print" SetupValue="50" Timer="0" />
  <CharacterSize Height="0" Width="90" />
  <PrintStartDelay Reverse="96" Forward="96" />
  <EncoderSettings HighSpeedPrinting="HM" Divisor="1" ExternalEncoder="None" />
  <InkStream InkDropUse="2" ChargeRule="Standard" />
</Printer>
```

As stated above, a parameter like Orientation can be specified as “Inverted/Forward” or as “2”. A full list of the symbolic values is included as Appendix B.

Substitution Rules Sub-Section

The Substitution Rules Section represents the data found in the “REPPX” files in the printer. It generally reflects the information in the Substitution Class (0x6C) of the Hitachi EtherNet/IP manual. In total, it would look like:

```
<Substitution Rule="01" StartYear="2019" Delimiter="/">
  <Year Base="00">00/01/02/03/04/05/06/07/08/09</Year>
  <Year Base="10">10/11/12/13/14/15/16/17/18/19</Year>
  <Year Base="20">20/21/22/23</Year>
  <Month Base="1">JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT/NOV/DEC</Month>
  <Day Base="01">01/02/03/04/05/06/07/08/09/10</Day>
  <Day Base="11">11/02/13/14/15/16/17/18/19/20</Day>
  <Day Base="21">21/22/23/24/25/26/27/28/29/30/31</Day>
  <Hour Base="00">00/01/02/03/04/05/06/07/08/09</Hour>
  <Hour Base="10">10/11/12/13/14/15/16/17/18/19</Hour>
  <Hour Base="20">20/21/22/23</Hour>
  <Minute Base="00">00/01/02/03/04/05/06/07/08/09</Minute>
  <Minute Base="10">10/11/12/13/14/15/16/17/18/19</Minute>
  <Minute Base="20">20/21/22/23/24/25/26/27/28/29</Minute>
  <Minute Base="30">30/31/32/33/34/35/36/37/38/39</Minute>
  <Minute Base="40">40/41/42/43/44/45/46/47/48/49</Minute>
  <Minute Base="50">50/51/52/53/54/55/56/57/58/59</Minute>
  <Week Base="01">01/02/03/04/05/06/07/08/09/10</Week>
  <Week Base="11">11/12/13/14/15/16/17/18/19/20</Week>
  <Week Base="21">21/22/23/24/25/26/27/28/29/30</Week>
  <Week Base="31">31/32/33/34/35/36/37/38/39/40</Week>
  <Week Base="41">41/42/43/44/45/46/47/48/49/50/51/52/53</Week>
  <DayOfWeek Base="1">MON/TUE/WED/THU/FRI/SAT/SUN</DayOfWeek>
</Substitution>
```

Objects Section

The objects section has one object for each Item in the printer message. There are three different types of objects:

- Text Object == Does not rely on the printer for automatic updating from print to print.
- Date Object == Contains information that is kept in sync with the clock in the printer.
- Counter Object == Contains information that is kept in sync with the print counter hardware.

Dates and Counters cannot both appear in the same printer item. Logos (Fixed and Free) can appear in any object.

Object Type Text

The information in the Text Object defines the Font Face as it appears in printed form. It generally reflects the information in the Print Format Class (0x67) of the Hitachi EtherNet/IP manual. A simple example :

```
<Object Type="Text">
  <Font BarCode="None" IncreasedWidth="1" LineSpace="0" CharacterSpace="1">5x8 (5x7) </Font>
  <Location ItemNumber="1" Row="1" Column="1" />
  <Text>1</Text>
</Object>
```

Item Number, Row and Column attributes for the Location section are all 1-Origin.

Object Sub-Type Date

The Date Object is a subsection of the text Object. It defines the formatting of the printer clock settings used in the printed message. It generally reflects the information in the Calendar Class (0x69) of the Hitachi EtherNet/IP manual. Multiple Date Sections can appear in a Text Object

```
<Date Block="1">
  <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
  <ZeroSuppress Month="Disable" />
  <EnableSubstitution Month="Enable" />
</Date>
<Date Block="2">
  <Offset Year="0" Month="0" Day="23" Hour="0" Minute="0" />
  <ZeroSuppress Year="Disable" Day="Disable" />
  <EnableSubstitution Year="Disable" Day="Disable" />
</Date>
<Text>BEST BY {{MMM}} {{DD}}, {{YYYY}}</Text>
```

The Text Section was shown again to clarify the Date Sections.

Object Sub-Type Counter

The Counter Object is a subsection of the Text Object. It defines the formatting of the printer counter settings used in the printed message. It generally reflects the information in the Count Class (0x79) of the Hitachi EtherNet/IP manual. Multiple Counter Sections can appear in a Text Object

```
<Counter Block="1" Reset="9000" CountUp="Up" Increment="1" JumpTo="8000" JumpFrom="5999"
  UpdateUnit="1" UpdateIP="0" Range2="9999" Range1="0000" InitialValue="0000"
  Multiplier="" ZeroSuppression="Enable" />
<Text>THE CURRENT COUNT {{CCCC}}</Text>
```

The Text Section was shown again to clarify the Counter Sections.

The complete XML for this message is in Appendix A.

Messages to/from Printer

Load XML into the printer

To load an XML file into the printer:

```
using (OpenFileDialog dlg = new OpenFileDialog()) {
    dlg.Title = "Select XML formatted file!";
    dlg.Filter = "XML (*.xml)|*.xml|All (*.*)|*.*";
    DialogResult dlgResult = DialogResult.Retry;
    while (dlgResult == DialogResult.Retry) {
        dlgResult = dlg.ShowDialog();
        if (dlgResult == DialogResult.OK) {
            try {
                EIP.SendXmlToPrinter(dlg.FileName);
            } catch (Exception ex) {
                MessageBox.Show(parent, ex.Message, "Cannot load XML File!");
            }
        }
    }
}
```

To load an XML string into the printer:

```
try {
    string xmlString = "...";
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.PreserveWhitespace = true;
    xmlDoc.LoadXml(xmlString);
    EIP.SendXmlToPrinter(xmlDoc);
} catch (Exception ex) {
    MessageBox.Show(parent, ex.Message, "Cannot load XML File!");
}
```

Retrieve XML from the printer

To get the XML for a message currently loaded in the printer (might want a “try” block around the call)

```
string XMLText = EIP.ConvertLayoutToXML(); // Retrieve the printer settings
XmlDocument xmlDoc = new XmlDocument(); // Turning it back into an XML Document
xmlDoc.PreserveWhitespace = true;
xmlDoc.LoadXml(XMLText);
```

Verify the printer load

To verify that the printer load accomplished the desired result (might want a “try” block around the call):

```
XmlDocument xmlDoc = new XmlDocument();
xmlDoc.PreserveWhitespace = true;
xmlDoc.Load(FileName);
VerifyXmlVsPrinter(xmlDoc);
```

Verify sample results

Valid	XML Name	Class	Attribute	Item	Block	Sub Rule	Data Out	Data In
TRUE	Orientation	Print_specification	Character_Orientation	N/A	N/A	N/A	Inverted/Forward	Inverted/Forward
TRUE	RepeatInterval	Print_specification	Repeat_Interval	N/A	N/A	N/A	0	0
TRUE	PrintsPerTrigger	Print_specification	Repeat_Count	N/A	N/A	N/A	0	0
TRUE	Filter	Print_specification	Target_Sensor_Filter	N/A	N/A	N/A	Until End of Print	Until End of Print
TRUE	SetupValue	Print_specification	Target_Sensor_Filter_Value	N/A	N/A	N/A	50	50
TRUE	Timer	Print_specification	Target_Sensor_Timer	N/A	N/A	N/A	0	0
TRUE	Width	Print_specification	Character_Width	N/A	N/A	N/A	90	90
TRUE	Height	Print_specification	Character_Height	N/A	N/A	N/A	0	0
FALSE	Forward	Print_specification	Print_Start_Delay_Forward	N/A	N/A	N/A	96	83
FALSE	Reverse	Print_specification	Print_Start_Delay_Reverse	N/A	N/A	N/A	96	83
TRUE	HighSpeedPrinting	Print_specification	High_Speed_Print	N/A	N/A	N/A	HM	HM
TRUE	Divisor	Print_specification	Pulse_Rate_Division_Factor	N/A	N/A	N/A	1	1
TRUE	ExternalEncoder	Print_specification	Product_Speed_Matching	N/A	N/A	N/A	None	None
TRUE	InkDropUse	Print_specification	Ink_Drop_Use	N/A	N/A	N/A	2	2
TRUE	ChargeRule	Print_specification	Ink_Drop_Charge_Rule	N/A	N/A	N/A	Standard	Standard
TRUE	Font	Print_format	Dot_Matrix	1	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	1	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	1	N/A	N/A	1	1
TRUE	Text	Print_format	Print_Character_String	1	N/A	N/A	SELL BY {{MMM}}/{DD}}/{YY}}	SELL BY {{MMM}}/{DD}}/{YY}}
TRUE	Year	Calendar	Offset_Year	1	1	N/A	0	0
TRUE	Month	Calendar	Offset_Month	1	1	N/A	0	0
TRUE	Day	Calendar	Offset_Day	1	1	N/A	0	0
TRUE	Hour	Calendar	Offset_Hour	1	1	N/A	0	0
TRUE	Minute	Calendar	Offset_Minute	1	1	N/A	0	0
TRUE	Year	Calendar	Zero_Suppress_Year	1	1	N/A	Disable	Disable
TRUE	Month	Calendar	Zero_Suppress_Month	1	1	N/A	Disable	Disable
TRUE	Day	Calendar	Zero_Suppress_Day	1	1	N/A	Disable	Disable
TRUE	Year	Calendar	Substitute_Year	1	1	N/A	Disable	Disable
FALSE	Month	Calendar	Substitute_Month	1	1	N/A	Enable	Disable
TRUE	Day	Calendar	Substitute_Day	1	1	N/A	Disable	Disable
TRUE	Font	Print_format	Dot_Matrix	2	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	2	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	2	N/A	N/A	1	1

TRUE	Text	Print_format	Print_Character_String	2	N/A	N/A	USE BY {{MMM}}/{{DD}}/{{YY}}	USE BY {{MMM}}/{{DD}}/{{YY}}
TRUE	Year	Calendar	Offset_Year	2	2	N/A	0	0
TRUE	Month	Calendar	Offset_Month	2	2	N/A	0	0
TRUE	Day	Calendar	Offset_Day	2	2	N/A	30	30
TRUE	Hour	Calendar	Offset_Hour	2	2	N/A	0	0
TRUE	Minute	Calendar	Offset_Minute	2	2	N/A	0	0
TRUE	Year	Calendar	Zero_Suppress_Year	2	2	N/A	Disable	Disable
TRUE	Month	Calendar	Zero_Suppress_Month	2	2	N/A	Disable	Disable
TRUE	Day	Calendar	Zero_Suppress_Day	2	2	N/A	Space Fill	Space Fill
TRUE	Year	Calendar	Substitute_Year	2	2	N/A	Disable	Disable
FALSE	Month	Calendar	Substitute_Month	2	2	N/A	Enable	Disable
TRUE	Day	Calendar	Substitute_Day	2	2	N/A	Disable	Disable
TRUE	Font	Print_format	Dot_Matrix	3	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	3	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	3	N/A	N/A	1	1
TRUE	Text	Print_format	Print_Character_String	3	N/A	N/A	PACKED {{TTT}} {777}}	PACKED {{TTT}} {777}}
TRUE	Year	Calendar	Offset_Year	3	3	N/A	0	0
TRUE	Month	Calendar	Offset_Month	3	3	N/A	0	0
TRUE	Day	Calendar	Offset_Day	3	3	N/A	0	0
TRUE	Hour	Calendar	Offset_Hour	3	3	N/A	0	0
TRUE	Minute	Calendar	Offset_Minute	3	3	N/A	0	0
TRUE	DayOfWeek	Calendar	Zero_Suppress_Day_Of_Week	3	3	N/A	Disable	Disable
FALSE	DayOfWeek	Calendar	Substitute_Day_Of_Week	3	3	N/A	Enable	Disable
TRUE	Font	Print_format	Dot_Matrix	4	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	4	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	4	N/A	N/A	1	1
FALSE	Text	Print_format	Print_Character_String	4	N/A	N/A	Shift {{E}}	Shift {E}}
TRUE	Year	Calendar	Offset_Year	4	4	N/A	0	0
TRUE	Month	Calendar	Offset_Month	4	4	N/A	0	0
TRUE	Day	Calendar	Offset_Day	4	4	N/A	0	0
TRUE	Hour	Calendar	Offset_Hour	4	4	N/A	0	0
TRUE	Minute	Calendar	Offset_Minute	4	4	N/A	0	0
TRUE	Font	Print_format	Dot_Matrix	5	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	5	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	5	N/A	N/A	1	1
TRUE	Text	Print_format	Print_Character_String	5	N/A	N/A	TCount {{FF}}	TCount {{FF}}
TRUE	Year	Calendar	Offset_Year	5	5	N/A	0	0

TRUE	Month	Calendar	Offset_Month	5	5	N/A	0	0
TRUE	Day	Calendar	Offset_Day	5	5	N/A	0	0
TRUE	Hour	Calendar	Offset_Hour	5	5	N/A	0	0
TRUE	Minute	Calendar	Offset_Minute	5	5	N/A	0	0
FALSE	Start	Calendar	Time_Count_Start_Value	5	5	N/A	A1	00
TRUE	End	Calendar	Time_Count_End_Value	5	5	N/A	X2	X2
TRUE	ResetTime	Calendar	Reset_Time_Value	5	5	N/A	6	6
TRUE	Font	Print_format	Dot_Matrix	6	N/A	N/A	5x8(5x7)	5x8(5x7)
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	6	N/A	N/A	1	1
TRUE	IncreasedWidth	Print_format	Character_Bold	6	N/A	N/A	1	1
TRUE	Text	Print_format	Print_Character_String	6	N/A	N/A	# {{{CCCCC}}}	# {{{CCCCC}}}
TRUE	Reset	Count	Reset_Value	6	1	N/A	000001	000001
TRUE	CountUp	Count	Direction_Value	6	1	N/A	Down	Down
TRUE	Increment	Count	Increment_Value	6	1	N/A	2	2
TRUE	JumpTo	Count	Jump_To	6	1	N/A	000300	000300
TRUE	JumpFrom	Count	Jump_From	6	1	N/A	000199	000199
TRUE	UpdateUnit	Count	Update_Unit_Unit	6	1	N/A	1	1
TRUE	UpdateIP	Count	Update_Unit_Halfway	6	1	N/A	0	0
TRUE	Range2	Count	Count_Range_2	6	1	N/A	999999	999999
TRUE	Range1	Count	Count_Range_1	6	1	N/A	000000	000000
FALSE	InitialValue	Count	Initial_Value	6	1	N/A	000001	000000
TRUE	Multiplier	Count	Count_Multiplier	6	1	N/A		
TRUE	ZeroSuppression	Count	Zero_Suppression	6	1	N/A	Enable	Enable
TRUE	Font	Print_format	Dot_Matrix	7	N/A	N/A	18x24	18x24
TRUE	InterCharacterSpace	Print_format	InterCharacter_Space	7	N/A	N/A	2	2
TRUE	IncreasedWidth	Print_format	Character_Bold	7	N/A	N/A	1	1
TRUE	Text	Print_format	Print_Character_String	7	N/A	N/A	{X/000}	{X/000}

Hard Coded Example

Here is what the code might look like

```
// Create a message with three rows, two columns, and a Logo (that contains one of everything)
private bool Comprehensive() {
    bool success = true;
    string[] itemText = new string[] {
        "SELL BY {{MMM}}/{{DD}}/{{YY}} ", "USE BY  {{MMM}}/{{DD}}/{{YY}} ", "PACKED  {{TTT}} {{777}} ",
        "Shift {{E}}", "TCount {{FF}} ", "# {{CCCCC}} ", "{X/0}"
    };
    int firstBlock = 1;
    if (EIP.StartSession()) {
        if (EIP.ForwardOpen()) {
            try {
                // Clean up the display
                {
                    EIP.GetAttribute(ccPF.Number_Of_Columns, out int cols);
                    if (cols > 1) {
                        EIP.SetAttribute(ccIDX.Column, 1); // Actually column 2
                        while (--cols > 0) {
                            EIP.ServiceAttribute(ccPF.Delete_Column);
                        }
                    }
                    EIP.SetAttribute(ccIDX.Item, 1);
                    EIP.SetAttribute(ccPF.Line_Count, 1);
                    // Avoid issues with add columns
                    EIP.SetAttribute(ccPF.Dot_Matrix, "5x8");
                    EIP.SetAttribute(ccPF.Barcode_Type, "None");
                    EIP.SetAttribute(ccPF.Print_Character_String, "1");
                }

                // Set up the rows and columns
                {
                    // First column is already there, just create the second and third columns
                    EIP.ServiceAttribute(ccPF.Add_Column);
                    EIP.ServiceAttribute(ccPF.Add_Column);
                    // Allocate the items in each column (Should this be Column and not Item?)
                    EIP.SetAttribute(ccIDX.Item, 1);
                    EIP.SetAttribute(ccPF.Line_Count, 3);
                    EIP.SetAttribute(ccIDX.Item, 2);
                    EIP.SetAttribute(ccPF.Line_Count, 3);
                    EIP.SetAttribute(ccIDX.Item, 3);
                    EIP.SetAttribute(ccPF.Line_Count, 1);
                    // Set the Interline Spacing
                    EIP.SetAttribute(ccIDX.Column, 1);
                    EIP.SetAttribute(ccPF.Line_Spacing, 1);
                    EIP.SetAttribute(ccIDX.Column, 2);
                    EIP.SetAttribute(ccPF.Line_Spacing, 2);
                }

                // Format the items
                {
                    // Set the format constant for all six items
                    for (int i = 1; i <= 6; i++) {
                        EIP.SetAttribute(ccIDX.Item, i);
                        EIP.SetAttribute(ccPF.Dot_Matrix, "5x8");
                        EIP.SetAttribute(ccPF.InterCharacter_Space, 1);
                        EIP.SetAttribute(ccPF.Print_Character_String, itemText[i - 1]);
                    }

                    // Set a logo into the seventh item
                    EIP.SetAttribute(ccIDX.Item, 7);
                    EIP.SetAttribute(ccPF.Dot_Matrix, "18x24");
                    EIP.SetAttribute(ccPF.InterCharacter_Space, 2);
                }
            }
        }
    }
}
```

```

    EIP.SetAttribute(ccPF.Print_Character_String, itemText[6]);
}

// Set up the clock for item 1
{
    EIP.SetAttribute(ccIDX.Item, 1);
    EIP.GetAttribute(ccCal.First_Calendar_Block, out firstBlock);
    EIP.SetAttribute(ccIDX.Calendar_Block, firstBlock);
    EIP.SetAttribute(ccCal.Substitute_Month, "Enable");
}

// Set up the clock for item 2
{
    EIP.SetAttribute(ccIDX.Item, 2);
    EIP.GetAttribute(ccCal.First_Calendar_Block, out firstBlock);
    EIP.SetAttribute(ccIDX.Calendar_Block, firstBlock);
    EIP.SetAttribute(ccCal.Substitute_Month, "Enable");
    EIP.SetAttribute(ccCal.Offset_Day, 30);
    EIP.SetAttribute(ccPF.Calendar_Offset, "From Yesterday");
}

// Set up the clock for item 3
{
    EIP.SetAttribute(ccIDX.Item, 3);
    EIP.GetAttribute(ccCal.First_Calendar_Block, out firstBlock);
    EIP.SetAttribute(ccIDX.Calendar_Block, firstBlock);
    EIP.SetAttribute(ccCal.Substitute_Day_Of_Week, "Enable");
}

// Set up the clock for item 4
{
    EIP.SetAttribute(ccIDX.Item, 4);
    EIP.GetAttribute(ccCal.First_Calendar_Block, out firstBlock);
    EIP.SetAttribute(ccIDX.Calendar_Block, firstBlock);

    // Set < Shift Number="1" StartHour="00" StartMinute="00" Text="D" />
    EIP.SetAttribute(ccIDX.Calendar_Block, 1);
    EIP.SetAttribute(ccCal.Shift_Start_Hour, 0);
    EIP.SetAttribute(ccCal.Shift_Start_Minute, 0);
    EIP.SetAttribute(ccCal.Shift_String_Value, "D");

    // Set < Shift Number="2" StartHour="8" StartMinute="00" Text="E" />
    EIP.SetAttribute(ccIDX.Calendar_Block, 2);
    EIP.SetAttribute(ccCal.Shift_Start_Hour, 8);
    EIP.SetAttribute(ccCal.Shift_Start_Minute, 0);
    EIP.SetAttribute(ccCal.Shift_String_Value, "E");

    // Set < Shift Number="2" StartHour="16" StartMinute="00" Text="F" />
    EIP.SetAttribute(ccIDX.Calendar_Block, 3);
    EIP.SetAttribute(ccCal.Shift_Start_Hour, 16);
    EIP.SetAttribute(ccCal.Shift_Start_Minute, 0);
    EIP.SetAttribute(ccCal.Shift_String_Value, "F");
}

// Set up the clock for item 5
{
    EIP.SetAttribute(ccIDX.Item, 5);
    EIP.GetAttribute(ccCal.First_Calendar_Block, out firstBlock);
    EIP.SetAttribute(ccIDX.Calendar_Block, firstBlock);

    // Set <TimeCount Start="A1" End="X2" Reset="A1" ResetTime="6" RenewalPeriod="30 Minutes" />
    EIP.SetAttribute(ccCal.Update_Interval_Value, "30 Minutes");
    EIP.SetAttribute(ccCal.Time_Count_Start_Value, "A1");
    EIP.SetAttribute(ccCal.Time_Count_End_Value, "X2");
    EIP.SetAttribute(ccCal.Reset_Time_Value, 6);
    EIP.SetAttribute(ccCal.Time_Count_Reset_Value, "A1");
}

// Set up the counter for item 6
{
    EIP.SetAttribute(ccIDX.Item, 6);

```

```
EIP.GetAttribute(ccCount.First_Count_Block, out firstBlock);
EIP.SetAttribute(ccIDX.Count_Block, firstBlock);
```

```
EIP.SetAttribute(ccCount.Initial_Value, "000001");
EIP.SetAttribute(ccCount.Count_Range_1, "000000");
EIP.SetAttribute(ccCount.Count_Range_2, "999999");
EIP.SetAttribute(ccCount.Jump_From, "000199");
EIP.SetAttribute(ccCount.Jump_To, "000300");
EIP.SetAttribute(ccCount.Increment_Value, 2);
EIP.SetAttribute(ccCount.Direction_Value, "Down");
EIP.SetAttribute(ccCount.Zero_Suppression, "Disable");
EIP.SetAttribute(ccCount.Count_Multiplier, "2");
EIP.SetAttribute(ccCount.Reset_Value, "000001");
EIP.SetAttribute(ccCount.Count_Skip, "0");
```

```
EIP.SetAttribute(ccCount.Update_Unit_Halfway, 0);
EIP.SetAttribute(ccCount.Update_Unit_Unit, 1);
EIP.SetAttribute(ccCount.Type_Of_Reset_Signal, "Signal 1");
EIP.SetAttribute(ccCount.External_Count, "Disable");
```

```
}
```

```
} catch {
    success = false;
```

```
}
```

```
}
```

```
EIP.ForwardClose();
```

```
}
```

```
EIP.EndSession();
```

```
return success;
```

```
}
```

XML Retrieved from the printer

```
<Label Version="1">
  <Printer Make="Hitachi" Model="UX-D161W">
    <PrintHead Orientation="Inverted/Forward" />
    <ContinuousPrinting RepeatInterval="0" PrintsPerTrigger="0" />
    <TargetSensor Filter="Until End of Print" SetupValue="50" Timer="0" />
    <CharacterSize Height="0" Width="90" />
    <PrintStartDelay Reverse="96" Forward="96" />
    <EncoderSettings HighSpeedPrinting="HM" Divisor="1" ExternalEncoder="None" />
    <InkStream InkDropUse="2" ChargeRule="Standard" />
    <TwinNozzle />
    <Substitution Delimiter="/" StartYear="2019" Rule="1">
      <Month Base="1">JAN/FEB/MAR/APR/MAY/JUN/JUL/AUG/SEP/OCT</Month>
      <Month Base="11">NOV/DEC</Month>
      <DayOfWeek Base="1">MON/TUE/WED/THU/FRI/SAT/SUN</DayOfWeek>
    </Substitution>
  </Printer>
  <Objects>
    <Object Type="Date">
      <Font BarCode="None" IncreasedWidth="1" InterLineSpace="1" InterCharacterSpace="1">5x8 (5x7)</Font>
      <Location ItemNumber="1" Row="3" Column="1" />
      <Date Block="1">
        <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
        <ZeroSuppress Year="Disable" Month="Disable" Day="Disable" />
        <EnableSubstitution Year="Disable" Month="Enable" Day="Disable" />
      </Date>
      <Text>SELL BY {{MMM}}/{DD}}/{YY}} </Text>
    </Object>
    <Object Type="Date">
      <Font BarCode="None" IncreasedWidth="1" InterLineSpace="1" InterCharacterSpace="1">5x8 (5x7)</Font>
      <Location ItemNumber="2" Row="2" Column="1" />
      <Date Block="1">
        <Offset Year="0" Month="0" Day="30" Hour="0" Minute="0" />
        <ZeroSuppress Year="Disable" Month="Disable" Day="Disable" />
        <EnableSubstitution Year="Disable" Month="Enable" Day="Disable" />
      </Date>
      <Text>USE BY {{MMM}}/{DD}}/{YY}} </Text>
    </Object>
    <Object Type="Date">
      <Font BarCode="None" IncreasedWidth="1" InterLineSpace="1" InterCharacterSpace="1">5x8 (5x7)</Font>
      <Location ItemNumber="3" Row="1" Column="1" />
      <Date Block="1">
        <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
        <ZeroSuppress DayOfWeek="Disable" />
        <EnableSubstitution DayOfWeek="Disable" />
      </Date>
    </Object>
  </Objects>
</Label>
```

```

    <Text>PACKED  {{TTT} {777}} </Text>
</Object>
<Object Type="Date">
    <Font BarCode="None" IncreasedWidth="1" InterLineSpace="2" InterCharacterSpace="1">5x8 (5x7)</Font>
    <Location ItemNumber="4" Row="3" Column="2" />
    <Date Block="1">
        <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
        <ShiftCode Shift="1" StartHour="0" StartMinute="0" EndHour="7" EndMinute="59" ShiftCode="D" />
        <ShiftCode Shift="2" StartHour="8" StartMinute="0" EndHour="15" EndMinute="59" ShiftCode="E" />
        <ShiftCode Shift="3" StartHour="16" StartMinute="0" EndHour="23" EndMinute="59" ShiftCode="F" />
    </Date>
    <Text>Shift {E}</Text>
</Object>
<Object Type="Date">
    <Font BarCode="None" IncreasedWidth="1" InterLineSpace="2" InterCharacterSpace="1">5x8 (5x7)</Font>
    <Location ItemNumber="5" Row="2" Column="2" />
    <Date Block="1">
        <Offset Year="0" Month="0" Day="0" Hour="0" Minute="0" />
        <TimeCount Interval="-1200116811" Start="A1" End="X2" ResetTime="6" ResetValue="A1" />
    </Date>
    <Text>TCount {{FF}} </Text>
</Object>
<Object Type="Counter">
    <Font BarCode="None" IncreasedWidth="1" InterLineSpace="2" InterCharacterSpace="1">5x8 (5x7)</Font>
    <Location ItemNumber="6" Row="1" Column="2" />
    <Counter Block="1" Reset="          " CountUp="Up" Increment="1" JumpTo="          " JumpFrom="          " UpdateUnit="1"
UpdateIP="0" Range2="999999" Range1="000000" InitialValue="000000" Multiplier="" ZeroSuppression="Enable" />
    <Text># {{CCCCC}} </Text>
</Object>
<Object Type="Text">
    <Font BarCode="None" IncreasedWidth="1" InterLineSpace="0" InterCharacterSpace="2">18x24</Font>
    <Location ItemNumber="7" Row="1" Column="3" />
    <Text>{X/000}</Text>
</Object>
</Objects>
</Label>

```

Traffic for Hard Coded Routine

Status/Path	Access	Class	Attribute	#In	Data In	Raw In	#Out	Data Out	Raw Out
33 67 01 66	Get	Print format	Number Of Columns	1	1	01			
32 7A 01 66	Set	Index	Item				2	1	00 01
32 67 01 72	Set	Print format	Line Count				1	1	01
32 67 01 74	Set	Print format	Dot Matrix				1	5x8(5x7)	03
32 67 01 77	Set	Print format	Barcode Type				1	None	00
32 67 01 71	Set	Print format	Print Character String				2	"1"	31 00
33 67 01 66	Get	Print format	Number Of Columns	1	1	01			
32 7A 01 66	Set	Index	Item				2	1	00 01
32 67 01 72	Set	Print format	Line Count				1	1	01
32 67 01 74	Set	Print format	Dot Matrix				1	5x8(5x7)	03
32 67 01 77	Set	Print format	Barcode Type				1	None	00
32 67 01 71	Set	Print format	Print Character String				2	"1"	31 00
32 67 01 6D	Set	Print format	Format Setup				1	Individual	01
32 68 01 68	Set	Print specification	Character Orientation				1	Normal/Forward	00
32 68 01 75	Set	Print specification	Target Sensor Filter				1	Time Setup	00
32 68 01 76	Set	Print specification	Target Sensor Filter Value				2	50	00 32
32 68 01 74	Set	Print specification	Target Sensor Timer				2	0	00 00
32 68 01 64	Set	Print specification	Character Height				1	99	63
32 68 01 67	Set	Print specification	Character Width				2	10	00 0A
32 68 01 69	Set	Print specification	Print Start Delay Forward				2	55	00 37
32 68 01 6A	Set	Print specification	Print Start Delay Reverse				2	45	00 2D
32 68 01 65	Set	Print specification	Ink Drop Use				1	2	02
32 68 01 77	Set	Print specification	Ink Drop Charge Rule				1	Mixed	01
32 68 01 6B	Set	Print specification	Product Speed Matching				1	Auto	02
34 67 01 6B	Service	Print format	Add Column				1		00
34 67 01 6B	Service	Print format	Add Column				1		00
32 7A 01 66	Set	Index	Item				2	1	00 01
32 67 01 72	Set	Print format	Line Count				1	3	03
32 7A 01 66	Set	Index	Item				2	2	00 02
32 67 01 72	Set	Print format	Line Count				1	3	03
32 7A 01 66	Set	Index	Item				2	3	00 03
32 67 01 72	Set	Print format	Line Count				1	1	01
32 7A 01 67	Set	Index	Column				2	1	00 01
32 67 01 73	Set	Print format	Line Spacing				1	1	01
32 7A 01 67	Set	Index	Column				2	2	00 02
32 67 01 73	Set	Print format	Line Spacing				1	2	02
32 7A 01 66	Set	Index	Item				2	1	00 01
32 67 01 74	Set	Print format	Dot Matrix				1	5x8(5x7)	03
32 67 01 75	Set	Print format	InterCharacter Space				1	1	01
32 67 01 71	Set	Print format	Print Character String				28	"SELL BY {{MMM}}/{DD}}/{YY}} "	53 45 4C 4C ...
32 7A 01 66	Set	Index	Item				2	2	00 02
32 67 01 74	Set	Print format	Dot Matrix				1	5x8(5x7)	03

32 67 01 75	Set	Print format	InterCharacter Space	1	1	01
32 67 01 71	Set	Print format	Print Character String	28	"USE BY {{MMM}}/{DD}}/{YY}} "	55 53 45 20 ...
32 7A 01 66	Set	Index	Item	2	3	00 03
32 67 01 74	Set	Print format	Dot Matrix	1	5x8(5x7)	03
32 67 01 75	Set	Print format	InterCharacter Space	1	1	01
32 67 01 71	Set	Print format	Print Character String	23	"PACKED {{TTT}}{777}} "	50 41 43 4B ...
32 7A 01 66	Set	Index	Item	2	4	00 04
32 67 01 74	Set	Print format	Dot Matrix	1	5x8(5x7)	03
32 67 01 75	Set	Print format	InterCharacter Space	1	1	01
32 67 01 71	Set	Print format	Print Character String	12	"Shift {{E}}"	53 68 69 66 ...
32 7A 01 66	Set	Index	Item	2	5	00 05
32 67 01 74	Set	Print format	Dot Matrix	1	5x8(5x7)	03
32 67 01 75	Set	Print format	InterCharacter Space	1	1	01
32 67 01 71	Set	Print format	Print Character String	15	"TCount {{FF}} "	54 43 6F 75 ...
32 7A 01 66	Set	Index	Item	2	6	00 06
32 67 01 74	Set	Print format	Dot Matrix	1	5x8(5x7)	03
32 67 01 75	Set	Print format	InterCharacter Space	1	1	01
32 67 01 71	Set	Print format	Print Character String	14	"# {{CCCCC}} "	23 20 7B 7B ...
32 7A 01 66	Set	Index	Item	2	7	00 07
32 67 01 74	Set	Print format	Dot Matrix	1	18x24	08
32 67 01 75	Set	Print format	InterCharacter Space	1	2	02
32 67 01 71	Set	Print format	Print Character String	6	"{X/0}"	7B 58 2F 30 ...
32 7A 01 66	Set	Index	Item	2	1	00 01
33 69 01 66	Get	Calendar	First Calendar Block	1	1	01
32 7A 01 6F	Set	Index	Calendar Block	1	1	01
32 69 01 75	Set	Calendar	Substitute Month	1	Enable	01
32 7A 01 66	Set	Index	Item	2	2	00 02
33 69 01 66	Get	Calendar	First Calendar Block	1	2	02
32 7A 01 6F	Set	Index	Calendar Block	1	2	02
32 69 01 75	Set	Calendar	Substitute Month	1	Enable	01
32 69 01 6A	Set	Calendar	Offset Day	2	30	00 1E
32 67 01 8D	Set	Print format	Calendar Offset	1	From Yesterday	00
32 7A 01 66	Set	Index	Item	2	3	00 03
33 69 01 66	Get	Calendar	First Calendar Block	1	3	03
32 7A 01 6F	Set	Index	Calendar Block	1	3	03
32 69 01 7A	Set	Calendar	Substitute Day Of Week	1	Enable	01
32 7A 01 66	Set	Index	Item	2	4	00 04
33 69 01 66	Get	Calendar	First Calendar Block	1	4	04
32 7A 01 6F	Set	Index	Calendar Block	1	4	04
32 7A 01 6F	Set	Index	Calendar Block	1	1	01
32 69 01 80	Set	Calendar	Shift Start Hour	1	0	00
32 69 01 81	Set	Calendar	Shift Start Minute	1	0	00
32 69 01 84	Set	Calendar	Shift String Value	1	"D"	44
32 7A 01 6F	Set	Index	Calendar Block	1	2	02

32 69 01 80	Set	Calendar	Shift Start Hour	1	8	08
32 69 01 81	Set	Calendar	Shift Start Minute	1	0	00
32 69 01 84	Set	Calendar	Shift String Value	1	"E"	45
32 7A 01 6F	Set	Index	Calendar Block	1	3	03
32 69 01 80	Set	Calendar	Shift Start Hour	1	16	10
32 69 01 81	Set	Calendar	Shift Start Minute	1	0	00
32 69 01 84	Set	Calendar	Shift String Value	1	"F"	46
32 7A 01 66	Set	Index	Item	2	5	00 05
33 69 01 66	Get	Calendar	First Calendar Block	1	5	05
32 7A 01 6F	Set	Index	Calendar Block	1	5	05
32 69 01 7F	Set	Calendar	Update Interval Value	1	30 Minutes	06
32 69 01 7B	Set	Calendar	Time Count Start Value	3	"A1"	41 31 00
32 69 01 7C	Set	Calendar	Time Count End Value	3	"X2"	58 32 00
32 69 01 7E	Set	Calendar	Reset Time Value	1	6	06
32 69 01 7D	Set	Calendar	Time Count Reset Value	3	"A1"	41 31 00
32 7A 01 66	Set	Index	Item	2	6	00 06
33 79 01 65	Get	Count	First Count Block	1	1	01
32 7A 01 6E	Set	Index	Count Block	1	1	01
32 79 01 67	Set	Count	Initial Value	7	"000001"	30 30 30 30 30 31 00
32 79 01 68	Set	Count	Count Range 1	7	"000000"	30 30 30 30 30 30 00
32 79 01 69	Set	Count	Count Range 2	7	"999999"	39 39 39 39 39 39 00
32 79 01 6E	Set	Count	Jump From	7	"000199"	30 30 30 31 39 39 00
32 79 01 6F	Set	Count	Jump To	7	"000300"	30 30 30 33 30 30 00
32 79 01 6C	Set	Count	Increment Value	1	2	02
32 79 01 6D	Set	Count	Direction Value	1	Down	02
32 79 01 73	Set	Count	Zero Suppression	1	Disable	00
32 79 01 74	Set	Count	Count Multiplier	2	"2"	32 00
32 79 01 70	Set	Count	Reset Value	7	"000001"	30 30 30 30 30 31 00
32 79 01 75	Set	Count	Count Skip	2	"0"	30 00
32 79 01 6A	Set	Count	Update Unit Halfway	3	0	00 00 00
32 79 01 6B	Set	Count	Update Unit Unit	3	1	00 00 01
32 79 01 71	Set	Count	Type Of Reset Signal	1	Signal 1	01
32 79 01 72	Set	Count	External Count	1	Disable	00