Industrial Automation

Tech Note 29 Working with Time in Crimson®

Abstract:

This document describes working with time in Crimson; including, but not limited to, the settings and the system variables associated with Time Manager.

Products:

G3 Series HMI / G3 Kadet HMI / Graphite® HMI / Graphite® Controllers / Modular Controller / Data Station Plus / ProducTVity Station

Use Case: Real-time Clock Manipulation

The real-time clock accuracy specification of the G3 HMIs, Graphite HMIs and Controllers, the Data Station Plus, Modular Controller, and ProducTVity Station is that there will be less than one minute per month drift. The 4" Kadet does not have a real-time clock chip, so it loses its time when it is powered down. This document covers the options available to keep the clock as accurate as required by the application.

Required Software:

Crimson 2.0 or Crimson 3.0

Required Firmware:

Crimson 2.0 Build 39 or later

Setting the Clock from User Interface

1. Add a Time and Date primitive to a display page.
2. Open the properties of the Time and Date primitive.
3. Set the Operation to *Data Entry.*
4. Click *OK.*

Setting the Clock with a PC

Crimson

From the menu bar at the top of the Crimson window:

1. Click *Link-Options.*
2. Select connection method.
3. Click *OK.*
4. Physically connect to the unit.
5. Click *Link-Send Time*.

Command Line

1. Open a command prompt window.
2. Navigate to: C:\Program Files\Red Lion Controls\Crimson 3.0\Utils.
3. Run the send executable: send using the –time and appropriate link switches.
   EX: send -time -usb

Synchronizing the Clock Using Time Manager

Settings

Control

Enable Time Manager**:** Enable this setting to use advanced time management. Once time management is enabled, the unit's time zone will get set via a PC (using *Link-Send Time*) or by using the TimeZone or TimeZoneMins and UseDST system variables.

Time Server

Enable SNTP: Enable this setting to expose the unit's time via SNTP.

NOTE: This server only works with Red Lion Clients; third party clients are not supported.

Time Client

Enable SNTP: Enable this setting to sync the unit's clock with another SNTP time source, such as a time server on the network.

Linked DST: Enable this setting to sync the Daylight Savings setting with the time server.

NOTE: This only works with Red Lion servers.

SNTP Mode: Select how the SNTP client should obtain its configuration.

SNTP Server: Select how the SNTP IP information will be entered, and then enter the IP Address, name, or expression.

NOTE: In order to use one of the Named options, DNS must be configured on the TCP interface that will be used to connect to the server.

Enable GPS: Enable this setting to sync the unit's clock with a GPS receiver connected to the unit.

NOTE: Requires a GPS receiver connected via serial communications using the Garmin NMEA-0183 driver.

Frequency: Enter the frequency at which the unit's clock is synchronized.

Time Stamps

Time Source: Select the source of time stamps for data logs, event logs, and the alarm viewer.

Time Zone: Select if the time stamps should use local time or UTC.
NOTE: Only available when Time Source is Real Time Clock.

Sync Clock: Select if the unit's clock should synchronize with the Expression.
NOTE: Only available in Crimson 3.0, see below for programming that is available for both Crimson 2.0 and 3.0.

Expression: Expression to use for the time stamp, value must be in seconds elapsed since 1-1-1997.
NOTE: The Time and Date functions can be used to convert to the correct format, example below in step 3 of the Configuration Example.

Time Manager Related System Variables

TimeZone: Offset from UTC in hours (-12 to +12). Eastern Standard Time (EST) is -5.

TimeZoneMins: Offset from UTC in minutes (-720 to +720). Indian Standard Time (IST) is +330.

UseDST: Set during Daylight Savings Time, clear during standard time.

Configuring Unit for Local Time

As mentioned above, there are two methods that can be used to configure the unit to display the correct local time once the Time Manager is enabled.

1. Use a connected PC's settings. Click *Link-Send Time* to send the PC's clock, time zone, and day light savings settings to the unit.
2. Use the system variables described in the previous section. These variables could be added to the display to allow users to modify these settings, or used in actions or programs.

Setting the Clock with Values from an External Device

Programmatically setting the clock, much like the Expression option in the Time Manager, requires the number of seconds that have elapsed since January 1, 1997.

Functions

Crimson provides the Time and Data functions to convert the current Hours, Minutes, Seconds, Month, Date, and Year to the desired format.

Time(h, m, s): Returns a value representing the indicated time as a number of seconds since midnight.
Date(M, D, Y): Returns a value representing the indicated date as a number of seconds since the datum point of January 1, 1997.

Example Configuration

This example will provide instructions to synchronize the Red Lion unit's clock with an external device at the top of every hour (when the minutes change to 0).

1. Create 6 Numeric tags
   a. Hours
   b. Minutes
   c. Seconds
   d. Month
   e. Date
   f. Year

2. Map the tags to the appropriate registers in the external device
3. Create a new Program, named PLCTimeSync with the following code

```
// declare local variables
int emit; // emit is time spelled backwards
int etad; // etad is date spelled backwards

emit = Time(Hours, Minutes, Seconds);
etad = Date(Year, Month, Date);

SetNow(emit + etad);
```

NOTE: Checking for good communications to the external device or non-zero values in the Month, Date, or Year tags is advisable prior to setting the unit's time, to avoid setting it incorrectly.

4. Add a Data Match trigger to the Minutes tag with the following settings
   a. Value: *0*
   b. Delay: *0*ms
   c. Action: *PLCTimeSync()*

Accessing Components of Unit's Clock

There is a family of functions available in Crimson that gives programmers access to the components of the unit's real-time clock value. As mentioned earlier in this document, the unit's clock is stored as the number of seconds that have elapsed since January 1, 1997; these functions use a value with the same datum point as their argument.

FUNCTION	DESCRIPTION
GetDate	Returns the day-of-month portion of *time*.
GetDay	Returns the day-of-week portion of *time*.
GetDays	Returns the number of days in *time*.
GetHour	Returns the hours portion of *time*.
GetMin	Returns the minutes portion of *time*.
GetMonth	Returns the month portion of *time*.
GetSec	Returns the seconds portion of *time*.
GetWeek	Returns the week-of-year portion of *time*.
GetWeeks	Returns the number of weeks in *time*.
GetWeekYear	Returns the week year when using week numbers.
GetYear	Returns the year portion of *time*.

Usage

Most of the time these functions are used, they are used in relation to the current time, so the GetNow() function is the most common argument to use with these functions. Another common, albeit advanced, use of these functions is to find the name of the previous day's log file.

Trigger at Midnight

1. Create a new Flag tag named Midnight
2. Change the Source to: *GetHour(GetNow()) == 5*
3. Add an Active On trigger
4. Add the desired Action

Find Yesterday’s Date

1. Create a new Numeric tag named Yesterday
2. Change Yesterday’s Source to *GetNow() – 86400*
   NOTE: 86,400 is the number of seconds in one day
3. Create a new program that returns a string with the following code

```
// declare local variables
cstring YY, MM, DD, hh;

// get year from yesterday, and convert to 2 digit text
YY = IntToText(GetYear(Yesterday), 10, 2);
// get month from yesterday, and convert to 2 digit text
MM = IntToText(GetMonth(Yesterday), 10, 2);
// get date from yesterday, and convert to 2 digit text
DD = IntToText(GetDate(Yesterday), 10, 2);

return MM + "-" + DD + "-" + YY;
```

Send the Red Lion’s Time to an External Device

1. Create a new Numeric tag named Hours
2. Change the Source of Hours to *GetHour(GetNow())*
3. Repeat steps **1** and **2** for all of the components of time to be sent to the external device
4. Create a Gateway block under the device where the time information will be sent and use the new time tags as the source data