i^3 Daylight Saving I/O Tutorial

Introduction

The purpose of this tutorial is to show how the Real Time Clock of the i^3 can be programmed to automatically adjust for daylight saving time. It is recommended that before beginning this tutorial that the both the RTC and Basic Operation tutorials are understood.

This tutorial will show how to make the i^3 cope with the automatic alteration of the RTC System Registers without entering an infinite loop, and a way of selecting a single moment in time despite the RTC being spread over 7 separate Registers.

Run the I/O configuration wizard and download it into your i^3 controller.

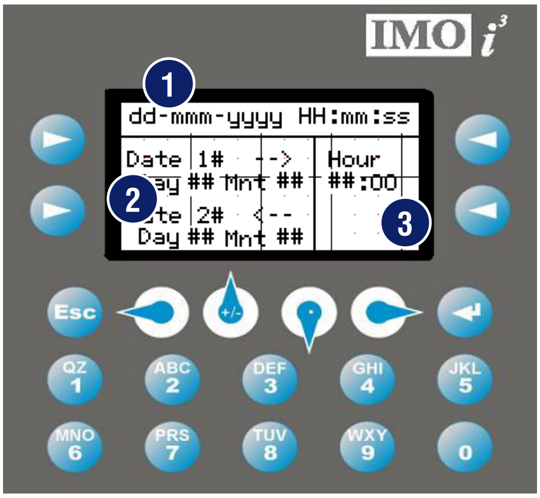
Setting the Control Screen

Before starting, three pieces of information are required. The 1st date is for when the RTC jumps forward, the 2nd date is when the clock jumps back, and the final piece of information is the hour of the day that the operator wishes the change to occur. Also, for reference, the current date and time is displayed at the top of the screen.

1. Two editable Time Data objects are required to alter the date and are useful for testing the function.

2. For each reference date, two editable Numerical Data objects must be assigned to available %R registers.

3. Another editable Numeric Data object is needed for the changeover hour, also assigned to a %R register.



For this function to work correctly, the i^3 needs to create a window between the two dates, monitor the actual time and date, and then create a one-time action once that window is entered or exited.

This cannot be done by using the date references as single moments in time; ie; creating an action at the exact moment when the current time equals the time specified by one of the reference dates. If the i^3 is powered down during that time when daylight-saving adjustment should occur; then the change will not occur. Although the RTC clock runs on without power to the i3, the logic that controls the daylight-saving function does not.

By creating a window, now matter when we enter, or exit the window, the change should still occur.

The Real Time Clock

In the table below, we see how the RTC is divided into multiple registers from %SR44 to %SR50.

SR Number	Description	Name	Limits
%SR0044	RTC Seconds	RTC_SEC	0-59
%SR0045	RTC Minutes	RTC_MIN	0-59
%SR0046	RTC Hours	RTC_HOUR	0-23
%SR0047	RTC Day of the Month	RTC_DATE	1-31
%SR0048	RTC Month	RTC_MON	1-12 (1 = January)
%SR0049	RTC Year	RTC_YEAR	1996-2095
%SR0050	RTC Day of the Week	RTC_DAY	1-7 (1 = Sunday)

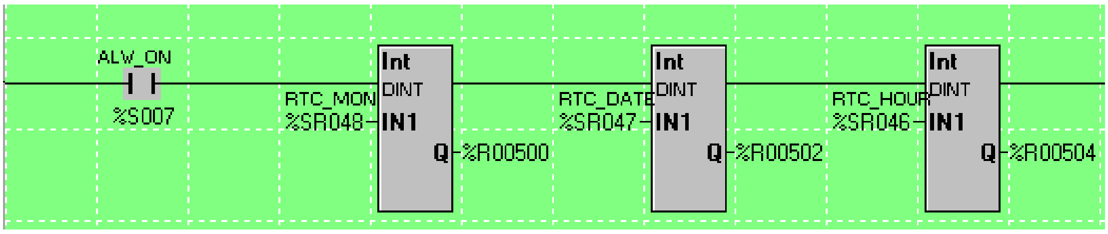
Creating a single reference window using comparator functions to look at multiple registers is almost impossible and would become cumbersome – seven overlapping windows would have to be created with a degree of control to make them function as one.

A much simpler solution is to condense the important parts of the RTC into one 32-bit Register. This lone register will contain all the information we need to construct the reference window.

i^3 Daylight Saving I/O Tutorial

Convert Singles to Doubles and Merge RTC

All of the RTC system registers are single integers. Single integers have a maximum value of 65536, but the reference requires a maximum of 123112 (December 31st 12:00), to get around this the single integers are converted into doubles, as shown below.



In reality, only the month, the day, and the hour are needed to make a functional frame of reference. As none of the three values ever exceed 100, multiplying each one by 100 shifts their value by two places along our reference DINT ready for the next to be added, i.e.

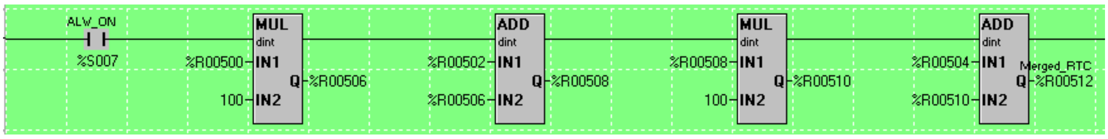
13:00hrs on 12th November (11) becomes:

11 x 100 = 1100

1100 + 12 = 1112

1112 x 100 = 111200

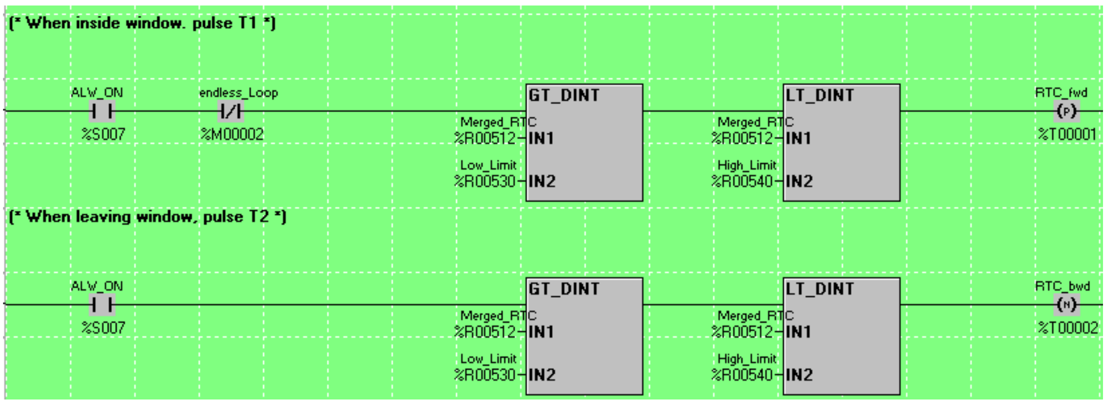
111200 + 13 = 111213



The same is also done for the two Date reference values #1 and #2 to make them comparable with the merged RTC, and create the 'End-stop' limits for our reference window.

Creating the Reference Window

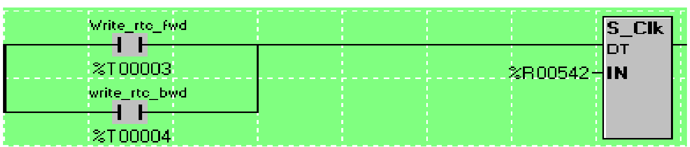
Using comparator functions to monitor when the RTC is greater than the Low limit and less than the High limit.



There is a positive transition coil (P) made when entering the window, and a negative transition coil (N) made when it is exited. This is to make sure that the function operates within a single scan of the program in either circumstance.

Correcting the Clock

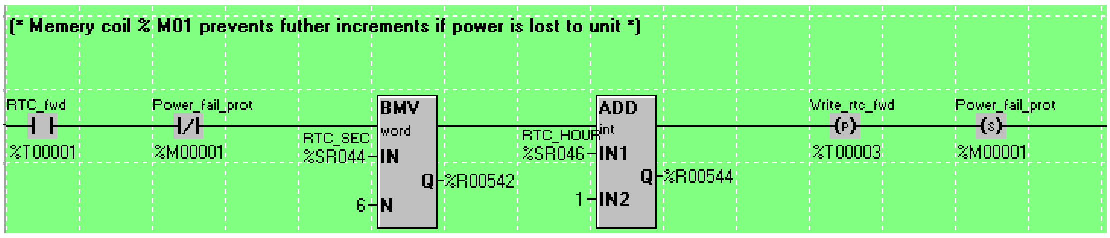
If the clock is being set forward or backward the function for writing to it is the same. A block of seven consecutive Single registers must be prepared before writing using the S_Clk function as shown here.



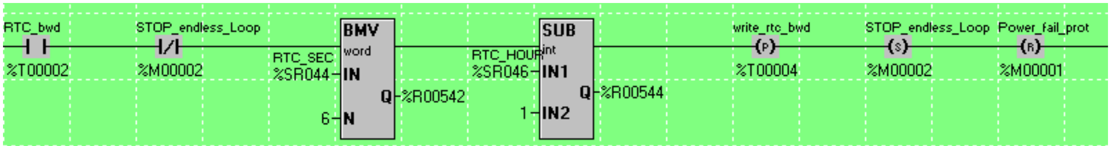
i^3 Daylight Saving I/O Tutorial

Knowledge

When preparing the block of registers for the clock to move forward, 1 must be added to the register containing the Hour reference. Once this has been done %T0003 is pulsed to write the values to the RTC System Registers, and Retentive Memory Bit %M0001 is set to prevent the clock from advancing by 1 hour each time the power is cycled on the i^3 while the time and date is within the Reference Window.



Likewise, for the clock to be moved back, 1 must be subtracted.

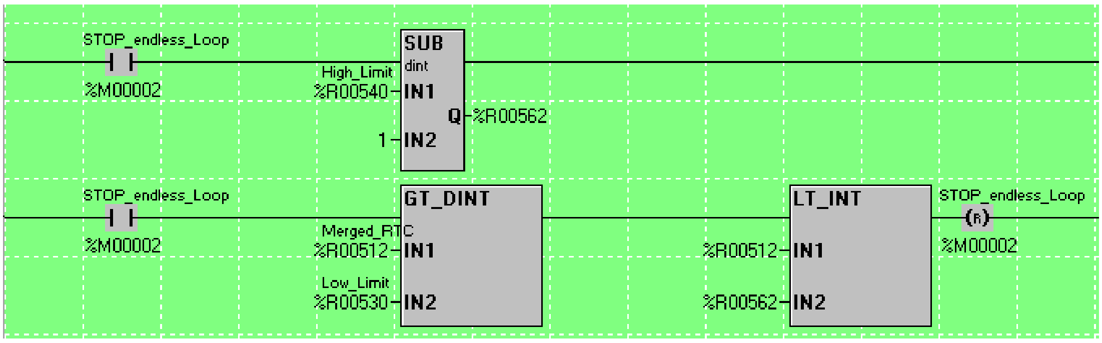


Above we can also see that Retentive Memory Bit %M0002 is set, and %M0001 is reset. Resetting %M0001 allows the clock to be advanced again when the time is right. Setting %M0002 begins a sequence that prevents an endless loop.

Stopping the Endless Loop

When the time passes completely through the Reference Window and transitions through the High Limit; the clock is set back by one hour. This action would push the RTC back inside the reference window, and hence, advance it by one hour, causing it to yet again be outside of the window. This endless loop has to be stopped, and in a way that allows the RTC to be changed only once and not leave the i^3 in a state of limbo for 2 hours until the High Limit is well and truly cleared.

To achieve this, a fake Window is created using the High Limit Minus 1, and %M0002 also disables rung 33 that controls the true Window.



Once the RTC has transitioned safely through the fake Window; the function can be reset ready for the next instance.

www.imopc.com

IMO Precision Controls Ltd
Unit 3, The Interchange, Frobisher Way
Hatfield, Hertfordshire AL10 9TG UK

Tel: +44 (0)1707 414 444
Fax: +44 (0)1707 414 445
Email: sales@imopc.com
Web: www.imopc.com