Programmable Logic Controllers

Interrupts

Interrupt Basics

- In terms of a PLC
  - What is an interrupt?
  - When can the controller operation be interrupted?
  - Priority of User Interrupts
  - Interrupt Latency
  - Interrupt Instructions
What is an Interrupt?

• An interrupt is an event that causes the controller to suspend the task it is currently performing, perform a different task, and then return to the suspended task at the point where it suspended.

• The Micrologix PLCs support the following User Interrupts:
  – User Fault Routine
  – Event Interrupts (4)
  – High-Speed Counter Interrupts (1)
  – Selectable Timed Interrupt

Interrupt Operation

• An interrupt must be configured and enabled to execute. When any one of the interrupts is configured (and enabled) and subsequently occurs, the user program:
  1. suspends its execution
  2. performs a defined task based upon which interrupt occurred
  3. returns to the suspended operation.

Interrupt Operation Example

Program File 2 is the main control program.
Program File 10 is the interrupt routine.
• An Interrupt Event occurs at rung 123.
• Program File 10 is executed.
• Program File 2 execution resumes immediately after rung 123.
Interrupt Operation (continued)

- Specifically, if the controller program is executing normally and an interrupt event occurs:
  1. the controller stops its normal execution
  2. determines which interrupt occurred
  3. goes immediately to rung 0 of the subroutine specified for that User Interrupt
  4. begins executing the User Interrupt subroutine (or set of subroutines if the specified subroutine calls a subsequent subroutine)
  5. completes the subroutine(s)
  6. resumes normal execution from the point where the controller program was interrupted

When Can the Controller Operation be Interrupted?

- The Micrologix controllers only allow interrupts to be serviced during certain periods of a program scan. They are:
  - At the start of a ladder rung
  - Anytime during End of Scan
  - Between data words in an expansion I/O scan
- The interrupt is only serviced by the controller at these opportunities.
- If the interrupt is disabled, the pending bit is set at the next occurrence of one of the three occasions listed above.
Priority of User Interrupts

• When multiple interrupts occur, the interrupts are serviced based upon their individual priority.

• When an interrupt occurs and another interrupt(s) has already occurred but has not been serviced, the new interrupt is scheduled for execution based on its priority relative to the other pending interrupts.
  – At the next point in time when an interrupt can be serviced, all the interrupts are executed in the sequence of highest priority to lowest priority.

Priority of User Interrupts

• If an interrupt occurs while a lower priority interrupt is being serviced (executed), the currently executing interrupt routine is suspended, and the higher priority interrupt is serviced.
  – Then the lower priority interrupt is allowed to complete before returning to normal processing.

• If an interrupt occurs while a higher priority interrupt is being serviced (executed), and the pending bit has been set for the lower priority interrupt, the currently executing interrupt routine continues to completion.
  – Then the lower priority interrupt runs before returning to normal processing.
**Interrupt Priorities**

The priorities from highest to lowest are:

- User Fault Routine
- Event Interrupt 0
- Event Interrupt 1
- High-Speed Counter Interrupt 0
- Event Interrupt 2
- Event Interrupt 3
- High-Speed Counter Interrupt 1 (MicroLogix 1500 only)
- Selectable Timed Interrupt

**Interrupt Latency**

- Interrupt Latency is defined as the worst case amount of time elapsed from when an interrupt occurs to when the interrupt subroutine starts to execute.

<table>
<thead>
<tr>
<th>Program Scan Activity</th>
<th>When an Interrupt Can Occur</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Scan</td>
<td>Between word updates</td>
</tr>
<tr>
<td>Ladder Scan</td>
<td>Start of Rung</td>
</tr>
<tr>
<td>Output Scan</td>
<td>Between word updates</td>
</tr>
<tr>
<td>Communications Service</td>
<td>Anytime</td>
</tr>
<tr>
<td>Housekeeping</td>
<td>Anytime</td>
</tr>
</tbody>
</table>
Determining Interrupt Latency

1. First determine the execution time for the longest executing rung in your control program (maximum rung time).
   - See MicroLogix 1100 Memory Usage and Instruction Execution Time handout for more information.
2. Multiply the maximum rung time by the Communications Multiplier corresponding to your configuration in the MicroLogix 1100 Scan Time Worksheet handout

Evaluate your results as follows:

<table>
<thead>
<tr>
<th>Controller</th>
<th>If the time calculated in step 2 is:</th>
<th>Then the Interrupt Latency is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroLogix 1200</td>
<td>less than 133 μs</td>
<td>411 μs</td>
</tr>
<tr>
<td></td>
<td>greater than 133 μs</td>
<td>the value calculated in step 2 plus 278 μs</td>
</tr>
<tr>
<td>MicroLogix 1500</td>
<td>less than 100 μs</td>
<td>360 μs</td>
</tr>
<tr>
<td></td>
<td>greater than 100 μs</td>
<td>the value calculated in step 2 plus 200 μs</td>
</tr>
</tbody>
</table>

INT – Interrupt Subroutine

Instruction Type: input

<table>
<thead>
<tr>
<th>Controller</th>
<th>When Rung Is:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True</td>
</tr>
<tr>
<td>MicroLogix 1200</td>
<td>1.0 μs</td>
</tr>
<tr>
<td>MicroLogix 1500</td>
<td>1.0 μs</td>
</tr>
</tbody>
</table>

- The INT instruction is used as a label to identify a user interrupt service routine (ISR).
- This instruction is placed as the first instruction on a rung and is always evaluated as true.
- Use of the INT instruction is optional.
UID - User Interrupt Disable

Instruction Type: output

<table>
<thead>
<tr>
<th>Controller</th>
<th>When Rung Is:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True</td>
</tr>
<tr>
<td>MicroLogix 1200</td>
<td>0.8 µs</td>
</tr>
<tr>
<td>MicroLogix 1500</td>
<td>0.0 µs</td>
</tr>
</tbody>
</table>

- The UID instruction is used to disable selected user interrupts.

Types of Interrupts Disabled by the UID Instruction

<table>
<thead>
<tr>
<th>Interrupt</th>
<th>Element</th>
<th>Decimal Value</th>
<th>Corresponding Bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>EII - Event Input Int.</td>
<td>Event 0</td>
<td>64</td>
<td>bit 6</td>
</tr>
<tr>
<td>EII - Event Input Int.</td>
<td>Event 1</td>
<td>32</td>
<td>bit 5</td>
</tr>
<tr>
<td>HSC - High-Speed Counter</td>
<td>HSC0</td>
<td>16</td>
<td>bit 4</td>
</tr>
<tr>
<td>EII - Event Input Int.</td>
<td>Event 2</td>
<td>8</td>
<td>bit 3</td>
</tr>
<tr>
<td>EII - Event Input Int.</td>
<td>Event 3</td>
<td>4</td>
<td>bit 2</td>
</tr>
<tr>
<td>HSC - High-Speed Counter</td>
<td>HSC1</td>
<td>2</td>
<td>bit 1</td>
</tr>
<tr>
<td>STI - Selectable Timed Int.</td>
<td>STI</td>
<td>1</td>
<td>bit 0</td>
</tr>
</tbody>
</table>

Note: Bits 7 to 15 must be set to zero.

(1) The MicroLogix 1200 has one HSC Interrupt, HSC0. The MicroLogix 1500 has two, HSC0 and HSC1.
Types of Interrupts Disabled by the UID Instruction

- To disable interrupt(s):
  1. Select which interrupts you want to disable.
  2. Find the Decimal Value for the interrupt(s) you selected.
  3. Add the Decimal Values if you selected more than one type of interrupt.
  4. Enter the sum into the UID instruction.
- For example, to disable EII Event 1 and EII Event 3:
  - EII Event 1 = 32, EII Event 3 = 4
  - 32 + 4 = 36 (enter this value)

UIE - User Interrupt Enable

- The UIE instruction is used to enable selected user interrupts.
- The parameter for the UIE instruction is computed exactly as the UID instruction.
UIF - User Interrupt Flush

- The UIF instruction is used to flush (remove pending interrupts from the system) selected user interrupts.
- The parameter for the UIF instruction is computed exactly as the UIE and UID instructions.

### UIF - User Interrupt Flush

<table>
<thead>
<tr>
<th>Controller</th>
<th>When Rung Is:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True</td>
</tr>
<tr>
<td>MicroLogix 1200</td>
<td>12.5 μs</td>
</tr>
<tr>
<td>MicroLogix 1500 1764-LSP</td>
<td>10.0 μs</td>
</tr>
<tr>
<td>MicroLogix 1500 1764-LRP</td>
<td>10.6 μs</td>
</tr>
</tbody>
</table>