Fieldbus 103

Loop scheduling

- Overview
- Basic scheduling
- Multiple loops on the same segment

Overview

What makes each control action happen when it should?

Good process control is time-dependent. If control actions don't happen when they should, the resulting process variability can increase energy and feedstock use, reduce yields, and reduce product quality.

FOUNDATION fieldbus solves this problem by executing control on a deterministic, real-time schedule. The technology is designed to accommodate the full range of control situations you're likely to face.

This course summarizes how FOUNDATION fieldbus does this.

Hint: As you go through the topics in this course, watch for answers to these questions:

- *What determines when a function block or communication action begins?*
- *How do you avoid overlapping communications when you have more than one loop on the same segment?*
- *How can you accommodate both fast and slow loops?*

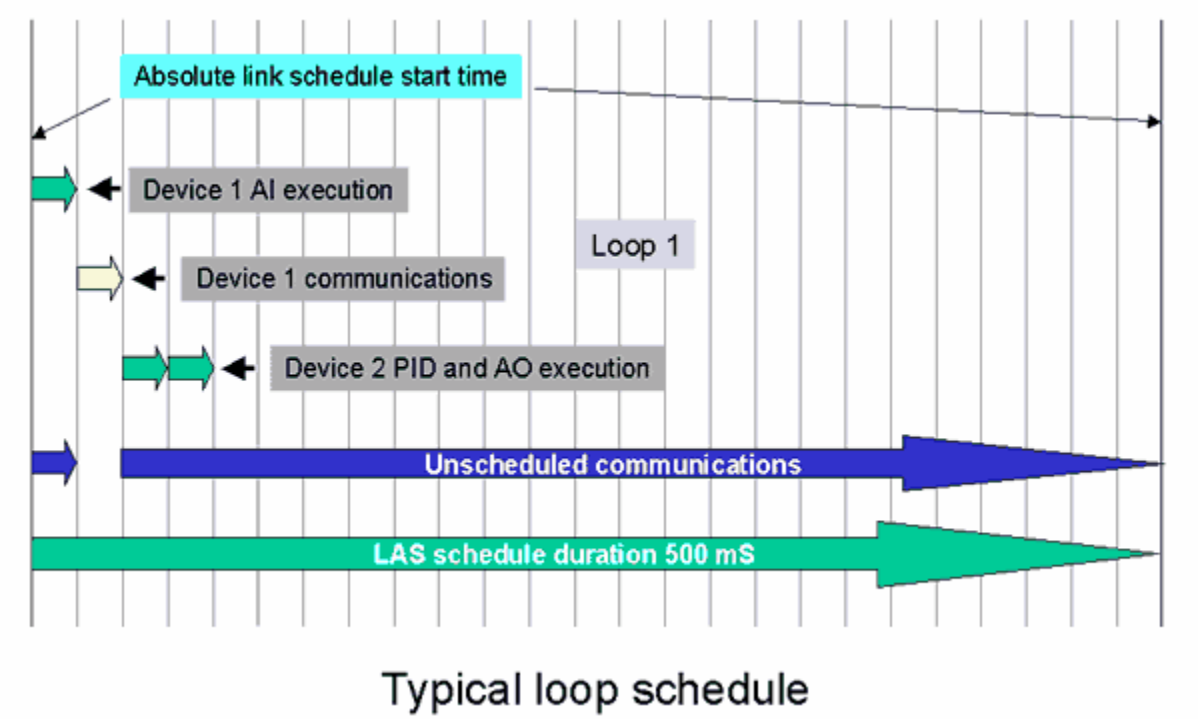
Basic scheduling

In FOUNDATION fieldbus, control-related communications and function blocks execute at precisely defined intervals, and in the proper scheduled order for correct process control.

Methods for ensuring correct timing of fieldbus communications — including the **application clock** and **link active scheduler** — are covered in the previous course, and the course that follows this one covers **function blocks** in detail. This section explains how the two are synchronized to provide the best possible control performance.

The overall schedule is called a **macrocycle**. The macrocycles for all devices on a segment are precisely scheduled and all use the same absolute start time. Function blocks and communications execute at specified offsets from this absolute time.

This diagram shows the schedule for a typical loop where the PID function is in the valve controller (Device 2). Each activity occurs at a defined offset from the absolute start time.



Typical loop schedule

This cycle repeats on an exact, ongoing schedule. Unscheduled (acyclic) messages can be communicated any time scheduled (cyclic) messages are not being sent.

Practical pointer

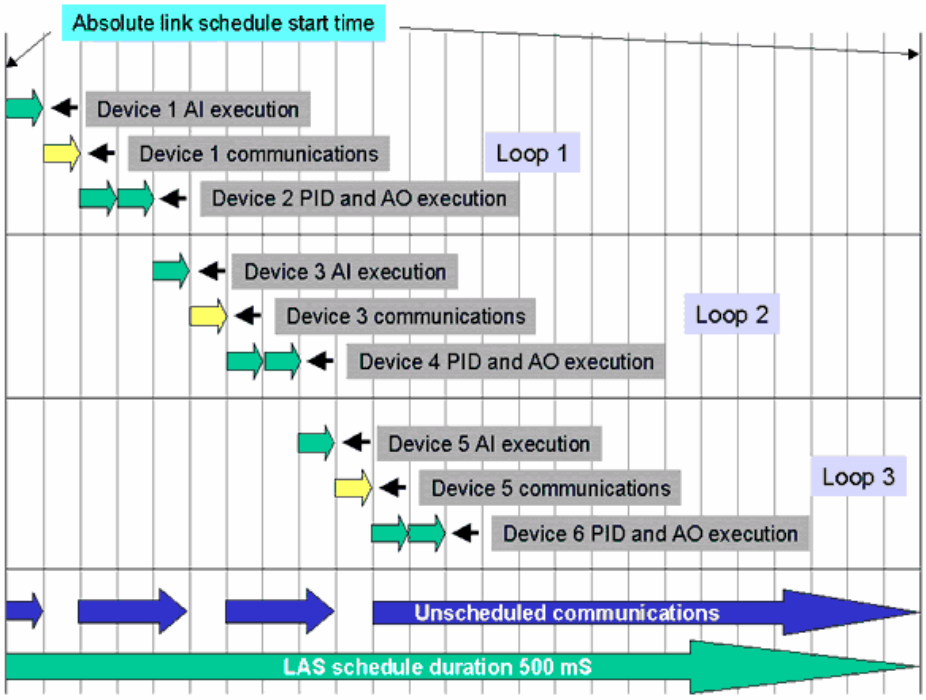
Use some caution in scheduling loops. Functions will execute in the order you specify, even if that order is incorrect. Scheduling the AO first, the PID next, and the AI last will add a large and needless delay to your overall loop processing.

The PlantWeb advantage

PlantWeb architecture makes loop scheduling a breeze. Scheduling is done automatically within the DeltaV and Ovation configuration tools. You simply drag and drop blocks on a graphical configuration sheet. As the blocks are dropped in place, they are automatically numbered in execution order and macrocycle is automatically calculated. If you build them in the order they are to execute, no further scheduling is needed.

If you decide to change the order of execution, the DeltaV and Ovation configuration tools let you change the order of execution by simply clicking on the blocks in the order you want them to execute.

Multiple loops on the same segment



Multiple loop schedule

As this example shows, you can have several function blocks executing at the same time on the same segment, provided that they're in different devices and have different start times. The example has three loops, with PID in the valve controller.
However, you can't have more than one device communicating on the bus at the same time. The example schedule prevents communication overlap by staggering the function block communication start times so one block doesn't start until the previous one has finished.

For the sake of simplicity, the diagram shows blocks executing in sequence, with no processing overlap. In reality, multiple blocks can execute at the same time as long as they're in different devices, and the data can be communicated as soon as the processing is complete. Multiple devices cannot communicate at the same time.

Practical pointer

Don't assume all devices have equal performance. Different devices may require different times to execute a function block. For example, one device may execute a PID block in 30 ms; another may take 75. Check with your vendor to see if they offer an automated scheduling tool that takes different device execution times into consideration.