If we wish to run a piece of code at regular intervals we can use a **periodic interrupt**. The SysTick timer is a simple way to create periodic interrupts. A periodic interrupt is one that is requested on a fixed-time basis. This interfacing technique is required for data acquisition and control systems, because software servicing must be performed at accurate time intervals. This is analogous to setting your alarm to wake you up at 7:00AM every morning, or having a bus schedule that has a bus arrive at the bus stop every 30 minutes. Smoke detectors use periodic interrupts. Every 30 seconds the detector will wake up, flash an LED, and then see if there is any smoke. If there is smoke the alarm goes off, and if there is no smoke, then it goes back to sleep.

For a data acquisition system, it is important to establish an accurate sampling rate. The time in between ADC samples must be equal and known in order for the digital signal processing to function properly. Similarly for microcontroller-based control systems, it is important to maintain a periodic rate for reading data from the sensors and outputing commands to the actuators.

One application of periodic interrupts is called **intermittent polling** or **periodic polling**. Figure 12.5 shows busy wait side by side with periodic polling. In busy-wait synchronization, the main program polls the I/O devices continuously. With periodic polling, the I/O devices are polled on a regular basis (established by the periodic interrupt.) If no device needs service, then the interrupt simply returns.

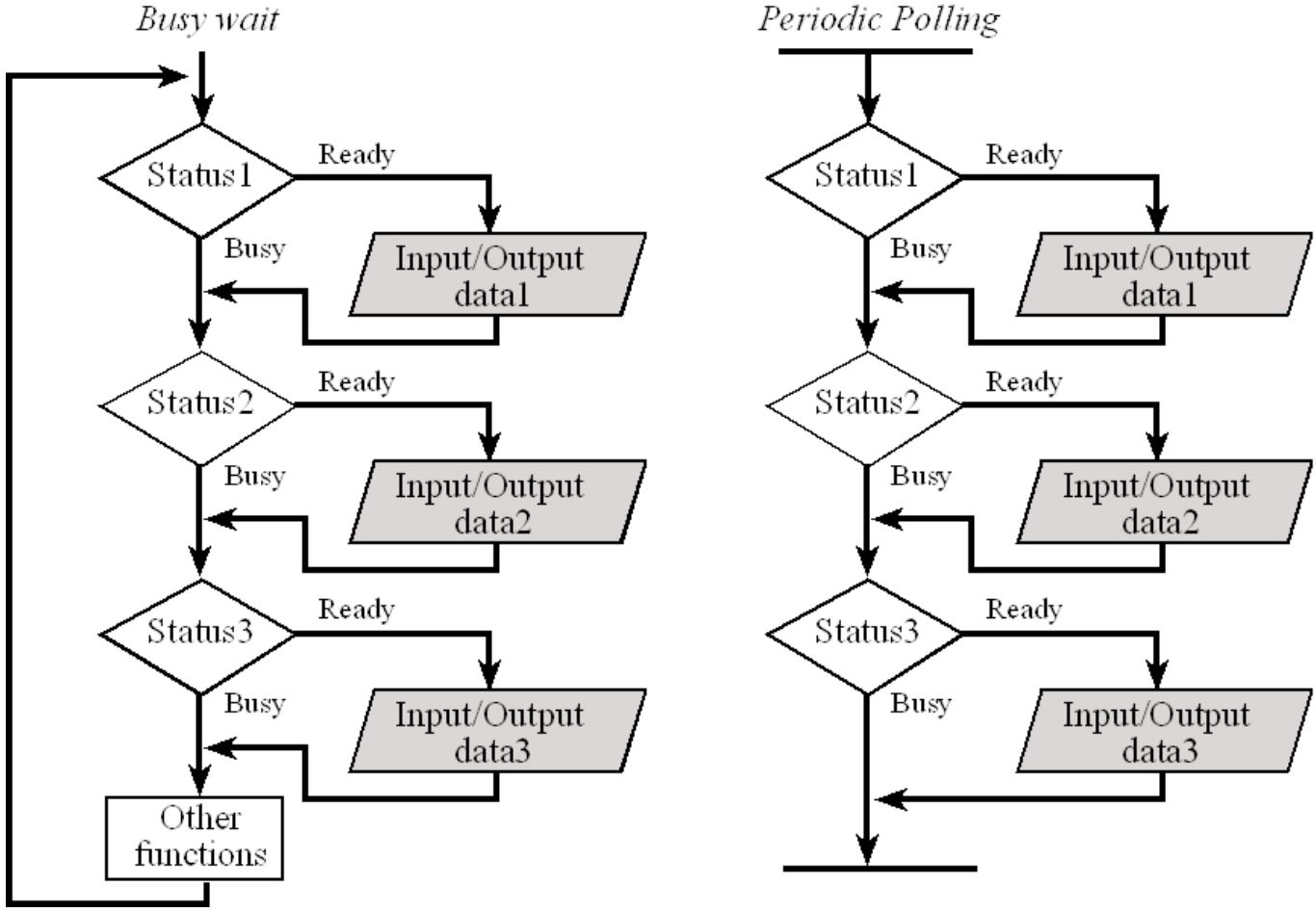


*Figure 12.5. On the left is busy-wait, and on the right is periodic polling.*

**Latency** is defined as the elapsed time between a request for service and the servicing of that request. If the polling period is Δt, then on average the interface latency will be ½Δt, and the worst case latency will be Δt. Periodic polling is appropriate for low bandwidth devices where real-time response is not necessary. This method frees the main program to perform other functions. We use periodic polling if the following two conditions apply:

1. The I/O hardware cannot generate interrupts directly
2. We wish to perform the I/O functions in the background

MOOCs are interactive and subjects include computer science, public health, and artificial intelligence.