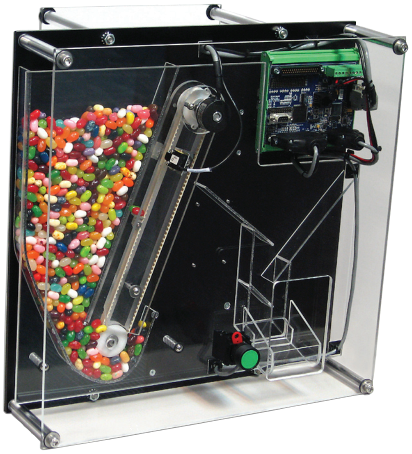
APPLICATION NOTE: Remote Controlled Jellybean Dispenser

INTRODUCTION

All computer controlled machines use electronic interfaces to bridge the gap between computer and machine. Although it is usually possible to locate those interfaces inside the computer, it is often problematic to do so. For example, complex machines can have numerous interfaces, which require thick bundles of wires that can be difficult to route out of the computer. To make matters worse, those wire bundles can be quite long if the computer and machinery are far apart, resulting in high manufacturing cost and lower reliability. Expensive, bulky termination boards are required because field wiring can't be directly connected to computer-resident interfaces. Also, computer-resident interfaces become obsolete over time and can't be easily replaced. Fortunately, there is a simple and elegant way to eliminate all of these problems: ***embed the interfaces in the machine and connect them to the computer with an Ethernet cable.***

This paper discusses the design of a simple jellybean dispensing machine that is remotely controlled via Ethernet. The machine demonstrates how commonly used machine components such as incremental encoders, motors, and switches can be monitored and controlled over a local area network.

MACHINE OVERVIEW

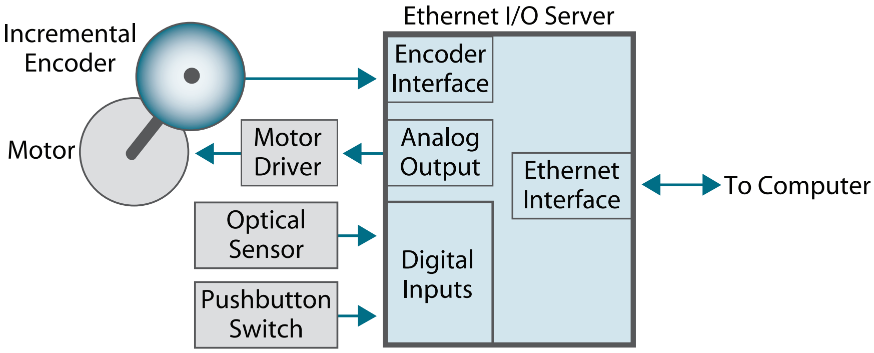


The Jellybean Machine (JBM) dispenses jellybeans by fetching them from a reservoir and dropping them into an exit chute. When the front panel pushbutton is pressed, a small "bucket" — which is mounted to a motorized timing belt — will try to scoop up jellybeans from the reservoir and position them for inspection by an optical sensor. If the bucket is empty, it will make a second pass through the reservoir and pause again for inspection. In either case, if the optical sensor "sees" jellybeans in the bucket, the bucket will deliver the jellybeans to the exit chute. The reservoir is assumed to be empty if the bucket comes up empty on a second pass.

SYSTEM ARCHITECTURE

The JBM has no internal control computer — it is monitored and controlled by a remote computer over a local area network. Remote control is facilitated by a **Sensoray model 2426 I/O server module**, which provides Ethernet access to a variety of built-in interfaces including analog I/O, digital I/O, serial COM, and incremental encoder.

Two of the I/O server's interfaces are used to implement closed-loop motor control: an analog output is used to control motor torque, and an encoder interface provides motor control feedback by tracking counts from an **Encoder Products Company model 15T incremental thru-bore encoder**. The remote computer uses these interfaces in conjunction with a PID algorithm to control bucket velocity and position.



INCREMENTAL ENCODER

The Encoder Products Company model 15T encoder is available with several signal output types (differential RS-422, TTL/CMOS, open collector), all of which are supported by Sensoray's encoder interface. Differential signaling is typically used to ensure reliable operation in cases where the encoder cable is long, but this doesn't apply to the JBM because its encoder interface is embedded in the machine, in close proximity to the encoder. Even so, the JBM design team opted to use differential RS-422 signaling to guarantee flawless performance.

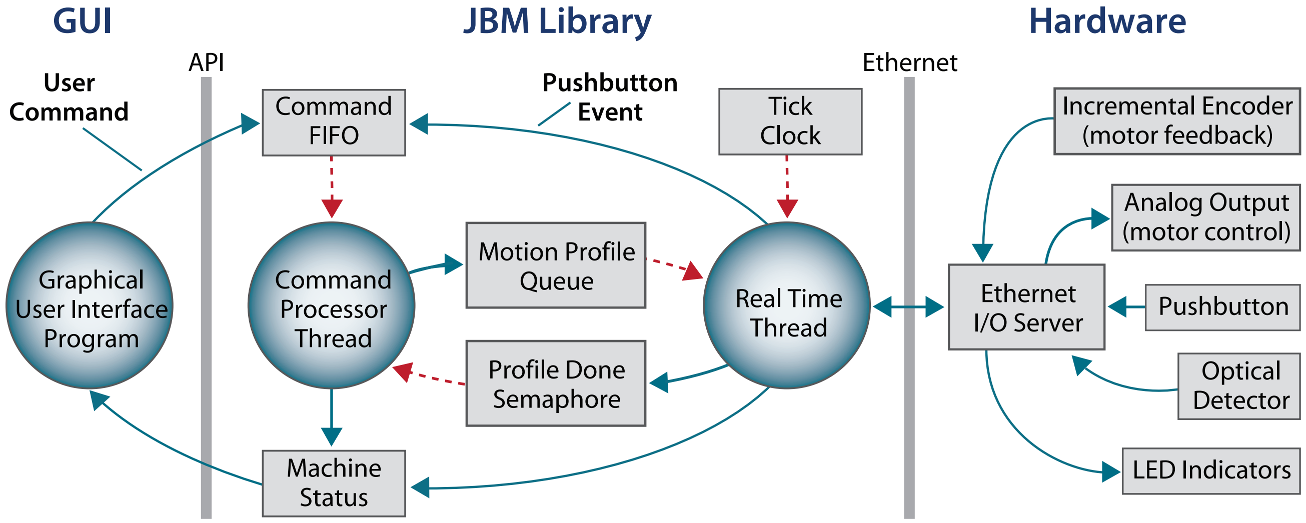
ENCODER INTERFACE

Ethernet and Windows are both non-deterministic, but it's often possible (and desirable) to work around this so that they can be used in control applications. The JBM is a case in point. When the I/O server sends encoder counts to the computer, it also sends a time stamp that indicates when the counts were sampled. By taking into account both encoder counts and time stamps, the PID algorithm is able to compensate for variable network and operating system latencies.

SOFTWARE

The software has two components: a machine control library and a graphical user interface (GUI) program. The library performs all real time machine control and monitoring, completely independent of the GUI. The library was coded in C and structured so that it could be easily ported to another operating system such as Linux or a real time operating system.

The GUI sends commands to the library to invoke machine operations, and it receives machine status information from the library. GUI development was quick because the GUI has no real time responsibilities or direct involvement with the inner workings of the machine. The GUI was written in VB.NET for rapid development, though it could just as easily have been written in another programming language due to its symbiotic relationship with the library.



As shown in the above diagram, the JBM library has two threads. The command processor thread is responsible for managing overall machine state. The command processor thread receives and executes commands from the GUI and pushbutton, and it computes and issues the motion profiles needed to satisfy those commands.

The real time thread programs the analog output at regular "tick" intervals, using encoder feedback and a PID algorithm to control belt speed and position in accordance with the motion profiles dictated by the command processor. It also controls LEDs and monitors the pushbutton and optical sensor. Not shown in the diagram is the I/O server's API library (free download from Sensoray), which allows the real time thread to communicate with the I/O server across Ethernet.

CAVEATS AND CREDITS

The JBM was designed by participants in Sensoray's Summer Engineering Internship Program. It is intended to serve as a practical demonstration of remote machine control over Ethernet. The JBM is not intended for production applications, nor is it an FDA-approved jellybean dispensing system. Please enjoy its highly palatable output.