A Forth Driven, Networked system for Applied Automation

Donald C. Long Jr.
Florida Drum Co.
3300 N. Hutchinson St.
Pine Bluff Ar. 71602

ABSTRACT

Finding flexible and cost effective ways of interfacing microcomputers for data acquisition and control has been a stumbling block for many applications.

The availability of low cost Single Board Computers and intelligent input/output systems provides exciting new capabilities for the automation of systems and entire facilities.

This paper describes an applications command language known as the Master Control Program. Supported hardware presently includes the Optomux family of intelligent interface boards. The advantages, system performance, and limitations of this form of networked distributed control are reviewed.

Introduction :

Widespread practical applications of microcomputers seem to lag the technology by several years. The majority of computer installations are excluded from many auxiliary applications due to limited I/O. Inputs are typically limited to keyboards and outputs to displays and printers.

The interface with environment and real world systems is crucial for many applications. The link between computer logic and the real world is easily handled with optically isolated input/output modules. But distributing the computer logic is rarely a straightfoward task.

Intelligent, off-the-shelf interface systems are now available. Such systems are approaching refinement to the point where time is saved in developing the interfaces, allowing one to spend more energy on the application.

My previous experiences with I/O boards and kludge-it-to-the-buss schemes were not always rewarding. Many applications do not lend themselves for computer interfacing. Industrial grade programmable controllers are fine for simple relay type logic but presently do not have the smarts for adaptive systems. After blowing up a computer with line voltage, I figured there must be a better way!

System Description:

About two years ago, Opto 22, a company well known for high quality solid state relays and I/O boards, introduced a family of Networked based intelligent controllers know as Optomux™ 1 (Fig 1). Any computer with a serial port can be the host network controller with the help of a RS-232 to RS-422 converter. RS-422 serial communications standard provides excellent noise immune and long distance remote links via 4 wire (two twisted pairs) balanced line operation. A single host serves multiple tasks located thousands of meters apart. Up to 256 boards may be addressed per line for a total of 4096 channels.

Either input or output modules can be plugged into each of the 16 positions on the board. Each position is configurable for a certain function such as a timer, counter, latched input, etc. Modules are available for a wide range of voltage levels. The PC host can be remotely located in a protected environment thereby lowering cost and enhancing system reliability. Additional points can be added at any time at very reasonable cost.

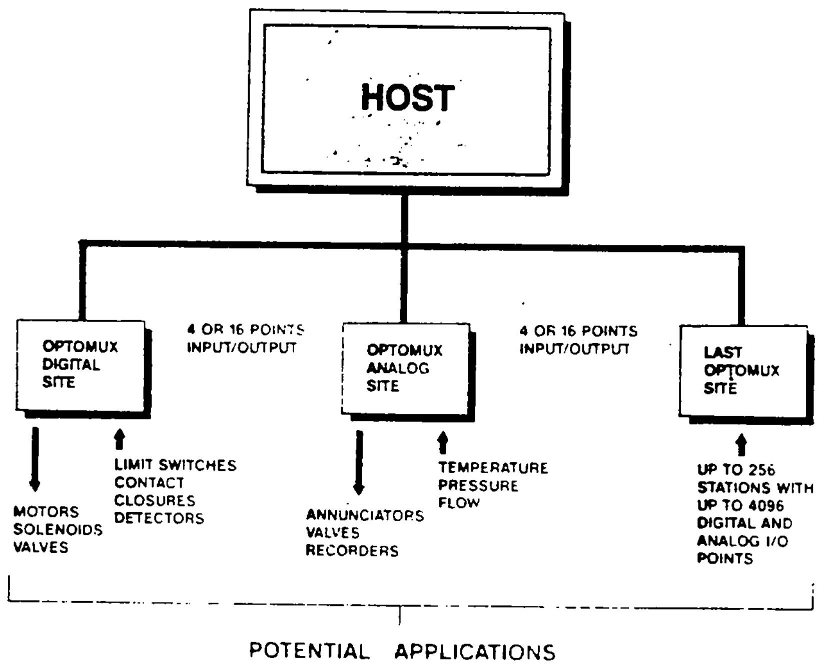

Fig 1

1 OPTOMUX is a trademark of OPTO 22

Software Environment:

The Master Control Program was developed with LMI's Forth 83 PC/Forth 3.1. Network communications is performed by a resident MS DOS assembly language device driver. The customized driver functions in any PC or AT Compatible environment. Upon execution of an I/O command, six data arrays pointers are pushed onto the stack and the driver is called via an interrupt. The command kernal could easily be ported to any version of Forth. Many of the advanced propriety system features of PC/Forth such as multi-tasking and window support has proved invaluable in the first installation.

The command language is comprised of over 100 commands providing control structures tailored for automation. These commands fall into several categories: Optomux Digital, Optomux Analog, Optomux Network Control, System Control, and Application Tools.

The Digital Commands perform:

- Input/Output
- Pulse Generation
- Event Latching
- Event Timing
- Event Counting

The Analog Commands perform:

- Input/Output
- Input Averaging
- Threshold detection
- Temperature Conversion to Degrees
- Thermocouple Linerazition

Network Control Commands perform:

- System security
- Setting of communications parameters
- Network diagonistics
- Error handling

Application Tools include:

- Autoincrementing counters
- Scheduler
- Logical control blocks
- Mouse menus

Extensive HELP commands guide the user in building applications.

Donald C. Long, Jr.

System Performance and Security:

Many complex tasks may be performed concurrently. Many of the critical tasks such as high-speed counting, timing, latching, pulse generation, etc are performed by a single-chip micro on each OPTOMUX board.

How fast is fast ??

The maximum communication rate of the Network is 19200 baud. Except for very long runs or very noisy environments, 19200 baud is the recommended rate for maximum throughput. Typical throughput is over 100 I/O commands per second on a 6 Mhz AT, versus 80 commands per second on a "vanilla" PC. Many commands process all 16 I/O channels per board. While not the speed of a direct connection, it is fast enough for most applications. Up to several hundred I/Os can be used while still keeping scan time under 100 ms for any particular point. Using a separate processor for the network communications would improve throughput of the system considerably.

Repeatability of time-critical functions are excellent since such functions are performed by a dedicated CPU. Programmed timers have 10-ms resolution and Counters count up to 100 Hz with 25% duty cycle. Inputs must be valid for two milliseconds to assure latching.

For integrity checksums verifies data in both directions before any command is processed. A four-pass protocol is supported that echos and verifies each character, but I've never found this mode necessary. Watchdog commands program outputs to deactivate if the network goes silent. System security has been adequate even in critical applications.

Donald C. Long, Jr.

A Simple Applications Example:

The Wye-Delta Starting arrangement is commonly used for starting three-phase motors for demand reduction. This simply involves swapping contacters after a certain time period. A similar arrangement can be used for two speed conveyer start up so that items are not thrown off.

```
3 COMMPORT  192 BAUDRATE   \ Select and Initialize comm port 3
A0  PUCLEAR                \ Power Up Clear Optomux at Address 0
A0 1 & 2 & 3 & CONFIGOUT   \ Modules 1,2,3 Configured for Output
A0 4 &          CONFIGIN   \ Module  4      Configured for Input

A0 2 &  500 SET DELAYON    \ Delay 5 sec when instructed ON
A0 3 &  500 SET TIMEON     \ Stay ON for 5 sec when instructed ON
A0 4 &      LATCH-ON>OFF   \ Latch any input for ON

: PUMP  A0  1 & 2 & 3 &  ; \ Define Address and Modules for Pump
```

Typing from console or executing WITHIN a defination

```
PUMP ON                  \ Turn Pump on for 5 seconds Y connection
                         \ Automatically transfers to Delta
                         \ connection in 5 sec

PUMP OFF                 \ Kill the pump
```

To control the level in a tank:

```
: TANK_HIGH?  ( -- f   TRUE IF TANK LEVEL IS HIGH)
             A0 4  READLATCH  1=           ( check latch)
             DUP IF A0 4 & CLEARLATCH THEN ( reset if true) ;

: CONTROL_TANK_LEVEL ( Place word in task for periodic execution)

      TANK_HIGH?
                IF    PUMP ON
                ELSE  PUMP OFF
                THEN                  ;
```

The word "A0" selects Address 0 and clears the data arrays. The word "&" adds the Modules to an array which selects the channels for the following command.

Donald C. Long, Jr.

Summary:

In the first application, a single host PC is performing the work of many large programmable controllers at a fraction of the cost without the inherent limitations. A background task may control entire machines or processes. If single-board computers were utilized instead of full-blown Personal Computers the cost could be reduced further.

Examples and techniques found in the Forth journals were instrumental in developing a working knowledge of Forth.

Over a year of effort went into developing the system to it's present form. The most difficult phase of the project the development of an interrupt callible resident device driver.

The primary goal was to develop a system not only replacing existing control schemes, but a system providing ample resources, programmibility, and expandability for applications of ever increasing complexity. The high level commands allow even non programmers to turnkey applications quickly. Programmers familiar with Forth can adapt the concept to their system or extend the language to handle many applications.

Presently, the Primary limitation of this system is the communications rate on the serial link. A higher rate link (>500 Kbaud) or possibly even fiber optics would allow such a system to be used for even the most demanding data acquisition jobs. For now one must wait until the network chips sets become available or invest in the higher speed factory automation protocalls such as MAP (Manufacturing Automation Protocol).

One of the most exciting aspect of this system is the overall cost effectiveness. The language is continuously evolving and will be used in several future applications including automating entire manufacturing facilities. A general purpose version of the language is available for those who have applications and wish to be spared time and development costs of designing an interface.

Donald C. Long, Jr.