

CANE TESTING SERVICE LOOKS AT LABORATORY COMPUTERISATION

By S. KING and E. P. EAST
Sugar Industry Central Board

and A. D. EVANS
South African Sugar Association

Abstract

Data relating to the sampling and analysis of consignments of sugar cane are captured directly from weighbridges and laboratory instruments by means of a mini-computer. Labour requirements are reduced and errors inherent in the manual capture of data are eliminated. Interface hardware was designed and developed to overcome problems caused by the factory environment and to reduce costs.

Introduction

The Sugar Industry Central Board is responsible for sampling and analysing sugar cane delivered to the sugar mills by farmers, for the purpose of cane payment. Each season, over a period of about 40 weeks, more than 20 million tons of cane are delivered to 18 sugar mills spread geographically from Port Shepstone on the South Coast of Natal to Malelane in the Eastern Transvaal. This sugar cane comprises over 1 million consignments. Each consignment has data relating to it which encodes details such as growers (or farm) identity, where the cane was transhipped, the type of transport used, the variety of the cane, nett mass, analytical results, transport contractor information, time delay, etc.

At present, this data is transcribed onto source documents which are transported by road to Durban where the data is captured by key-punch operators and processed on a main frame computer (IBM 370-138). The printed output is transported back to the mills by road again. In all, of the order of 4 million characters of information are processed in this manner each week. It is labour intensive, time consuming and prone to human error.

A project was conceived whereby all this data could be captured on site, using a mini-computer. Thus, an instrument reading would be captured by pressing a button instead of writing it down. This would eliminate transcription errors and the labour involved in key-punching the data in Durban. The computer could be used for on-site calculations presently carried out by clerical staff, with further labour saving. Further, the computer could carry out immediate validity and range checks on incoming data, significantly reducing errors that would normally have to be checked for and amended manually.

The Basic System

In broad terms, the system consists of a visual display unit with keyboard (VDU) in the weighbridge, where a large proportion of the descriptive data is keyed in. Gross and tare masses for each consignment are captured directly from the automated weighbridges. Facilities also exist for the capture of data concerning products other than sugar cane. A printer is used to print identification numbers on self-adhesive labels and a second printer is used to print receipts for cane delivered.

A further VDU is situated in a cabin overlooking the cane offloading site for the input of sample reference data. In the

laboratory, one VDU is alongside the chemical balance and a further VDU is in the constant temperature room alongside the saccharimeter and refractometer.

Whenever data is captured, it is edited for correct formatting and checked against preset ranges and tolerances. Error messages are returned via the VDU's to the operators for action. At the end of the day, listings of all deliveries and advice notes to growers are printed out on a matrix printer in the laboratory. All data is spooled onto diskettes for further processing in Durban.

Hardware

Mini-Computer. The computer chosen was the IBM Series 1. This decision was based on the following considerations :

- (i) Price/performance in terms of the instruction repertoire, cost of storage capacity and the number of levels of interrupt.
- (ii) Ease of on-line linking to IBM 370-138 at Durban DP department.
- (iii) A floppy disc drive available as an interim measure to (ii) above.
- (iv) Capable of modular expansion to very large capacity.

For every advantage, there has to be a disadvantage and with IBM, the user is not able to carry out his own maintenance and repairs and vendor servicing is expensive. This is considered to be a disadvantage, although perhaps not all users would think so. Certainly, it will be borne in mind when the project reaches the stage of purchasing further mini-computers.

Hardware Purchased. The hardware purchased on the IBM Series 1 was :

1. 96 K bytes of mos memory.
2. A 14 megabyte fixed head disc.
3. A diskette (floppy disc) drive.
4. Two IBM VDU's.
5. Two Beehive VDU's which are RS 232C compatible with a teletype compatible interface on the IBM side.
6. A matrix printer — 80 CPS.
7. A timer.
8. A dido card which is *digital in / digital out* and has dual ports.
9. An expansion chassis to handle all the interfaces.

Software

Requirements.

In an on-line application of this type, it is essential that the software be able to support multiple terminals, digital and analogue I/O to instruments, scientific and commercial arithmetic functions. Multi-programming and multi-tasking with inter-program and intertask communications and with

task control and synchronisation would be of great advantage in optimising the system. A long term requirement is ease of host computer communication.

After reviewing the available offerings from the vendor it was decided to use the Event Drive Executive (EDX) operating system (which includes its own high level application language). It supports all the above requirements and has the further advantage of being an event driven operating system with a reasonably small size (at present 28K).

A major disadvantage of EDX is the primitive file accessing support (it only allows sequential and/or relative record number random file accesses). The records are of fixed length (256 byte) and with the EDX operating system only EDX programming is allowed. It was felt that these disadvantages were outweighed by the advantages.

Programs.

The on-line system comprises 6 main programs which reside in storage at all times. To reduce the memory requirements of the system extensive use is made of overlay programs, the loading of which is controlled by the main programs.

The six main programs are :

Interface Control Program.

Queues and controls all requests from any application programs for data from any peripherals and sends commands to and receives and translates data coming from the hardware multiplexor.

Weighbridge Control Program.

Controls all operations related to the acquisition of data from cane and non-cane vehicles passing over the weighbridges. This program is operated by the weighbridge clerk from the weighbridge VDU.

Cane Yard Control Program.

Controls all cane yard operations, identifies cane feeding into the mill and retains an inventory of the amount and ownership of "in stock cane", providing the link between the cane sampling, cane analysis and cane delivery details. This program is operated from the sample supervisors' VDU.

Sample Preparation Control Program

Controls the procedure related to the preparation of cane and product samples for analysis. The program is operated from the laboratory VDU.

Analytical Determination Control Program.

Controls the procedures relating to the analysis of cane and other factory samples and is operated from the instrument room VDU by the analytical attendant.

Cane Samples Calculation Control Program.

Sequences and controls the calculations relating to cane samples and removes completed cane samples from the "in process" file.

Other programs are loaded only when required and include end-of-day and end-of-week procedures, file maintenance programs and enquiry programs.

Interface Hardware

As with many mini-computers the IBM has a 16 bit digital input/digital output card. Problems are experienced, in that peripherals cannot be directly coupled to these cards due to the following considerations :

1. Instrumentation generally has in excess of 16 bits of information which is above the dido limit on each channel.
2. The cost of dido cards is relatively high and expensive expansion chassis have to be bought if one exceeds the number of interfaces allowed by the first chassis. If one card per instrument is considered, this problem of expense is quickly realised.
3. As the dido causes an interrupt at the computer, timing and acknowledgement by the peripherals has to be considered, and obviously, laboratory and automation instrumentation and equipment does not have the ability to control these parameters.
4. The throughput of information from the instrumentation to the computer is relatively slow as most are manually operated.
5. The cost of multiple conductor cabling is very high and is not satisfactory in the mill environment due to noise affecting TTL voltage sense systems.
6. Not all instrumentation gives BCD and/or digital signals.
7. Character printers, parallel interfaced, have to be controlled at considerable distances from the computer (600 metres) with their own timing problems.
8. Expansion and flexibility must be available to enable virtually any instrument or equipment to be controlled and/or added to at any time.

Controller Specification.

With the above constraints in mind, it was decided to design and build a controller onto one section of the dido (which has two identical sections). This interface must have the ability to :

1. Control up to 64 separate peripherals or automative equipment inputs.
2. Information from peripherals must be able to be stored until the computer has obtained the information.
3. The controller must have a system of priority scheduling to enable highspeed or important information to get through first.
4. The controller must be able to have command over the input of interrupts.
5. The interface must be modular, that is, as more instruments are added extra integrated circuits are added to include the new peripherals. This is to avoid the expense of building a controller for 64 peripherals when you are in fact using less than 10.

Final Configuration.

It was decided, after experiencing the effects of noise from the large thyristor controlled and 3-phase motors on TTL type sensing inputs to standardise on a differential current loop system with high common mode rejection, between the peripherals and the computer. The closest peripheral, it must be remembered, is 40 metres from the computer and a number of them are located approximately half a kilometre from the computer.

As such, a 32 bit serial transmitter is attached directly onto the instrument or peripheral. This standard transmitter connects via shielded twisted pair wire to a receiver in the dido controller. The information is checked for validity on a 5 bit input code and, if valid, a strobe is issued to a storage buffer.

Transmission can take place either continuously or on event with a switch selected option on the PC board. Another bit is used to determine if the computer should be inter-

rupted or not. Note that the interrupt can only take place if a particular peripheral interrupt is enabled.

It was found after building these interfaces that speeds of 150 000 baud were easily obtained with virtually no ringing at all and rejection 15 volts AC common mode.

Interface Software

With the EDX operating system, only one program can use the dido card at any one time. To improve the system efficiency, an interface program was written to handle all data input/output from other programs. The main problems to be overcome were :

1. The return of data from the instrumentation is not necessarily in the same sequence as the data requests.
2. The various instruments return data in different formats (i.e. Binary Coded Decimal, 10 bit binary, etc.), which had to be decoded into one standard format (32 bit binary).
3. A large amount of printlines are generated by the application programs which have to be sent byte by byte to the remote character printers.
4. The response time to obtain data from the instruments needed to be as short as possible without interference from the printlines.

These points were resolved by designing a program with three asynchronous tasks;

Command Task.

Queues requests from application programs for instrument data, places printlines into a print buffer and sends data requests to the multiplexor.

Print Task.

A low priority task that sends printlines byte by byte to the multiplexor whenever the command task is not engaged in transmitting data requests.

Return Task.

Queues data returns from the multiplexor, translates the data into standard format, matches the data return with a waiting application program request and returns the data to the application program.

Present State of Project

The hardware was installed on 21st December, 1978 in time to ensure a hectic festive season. Some time was required for checking and testing hardware and software. On 10th January, 1979 a start was made in training operators for 24-hour operation. After this time, the system ran more or less continuously with stops of a few hours during the day-time to iron out minor software and hardware problems. From 18th January, the system ran fulltime 24 hours per day until the end of the crushing season (unfortunately only 4 days).

Basically it can be said that the first test run of the system was generally successful in that the system handles the workload well with few logical errors, the staff are able to use the system with very little training; and the test run highlighted problem areas that need examination and action before the start of the new season.

Acknowledgement

The authors wish to thank the Sugar Industry Central Board for permission to publish the paper.