Pak. j. sci. ind. res., vol.38, no.5-6, May-June 1995

PERSONAL COMPUTER BASED RAILWAY MONITORING AND SIGNALING SYSTEM

B. S. Chowdhry, . A. R. Memon, M.D. Makhdoom, and D. R. Gautam

Department of Electronic Engineering, Mehran University of Engineering and Technology, Jamshoro-Pakistan

(Received March 17, 1993 ; revised March 25, 1995)

To make railways a safer and more reliable mode of transport, it is quite necessary to introduce modernisation into existing railway signaling and control systems, because railway accidents cause great concern throughout the country. Major and minor railway accidents result from a seemingly intractable combination of human error, faulty signals, negligence and so on. When the frequency of such occurrence reaches alarming proportions, there comes a great need to improve signaling and communication systems. This clearly needs some rethinking. We propose and describe an implementation of a personal computer controlled railway monitoring and signaling system to avoid such types of disasters.

Key words. Personal computer, Railway signalling, Programmable interrupt controller.

Introduction

In the past, the conventional railway systems have been operated by railway personnel and by available communication means amongst them. Railway personnel encompasses the station master, signalling operator, and the driver. Each of them coordinates their activities for safe train control; nonetheless, the conventional systems, having been subjected to human judgment, are not fault free and can cause human life if and when the train encounters an accident*.

It is patent that the extensive uses of computers results in saving manpower and the creation of an error free environment with speedy performance [1-4]. Keeping this in mind, we have designed a system which utilizes personal computers for railway monitoring and signaling system. The block diagram of the complete system is shown in Fig. 1. Piezoelectric sensors [5] detect the position and movement of train on the railway track. After amplifying the sensor output, the signal is fed to the computer through a programmable interrupt controller. As the personal computer (PC) receives the signal as an interrupt, it executes the interrupt service routine corresponding to the interrupt type. There are two interrupt routines to be serviced, called arrival and departure routines, respectively. The arrival service routine checks the following sequence:

Which train is coming ?
Whether it is stopping at station or not.
If stopping then at which platform number.
That platform is free or not.
Give appropriate signal to the train.
Return from the routine.

On the otherhand, the departure service routine executes in the following ways:

Which train is leaving?
From which platform it leaves.
Give clearance signal.
Return from the routine.

Depending upon the interrupt type, PC gives the appropriate signal at the output latch. As the voltage of digital signal is not sufficient to drive the signal arm or electric bulbs, it has to be amplified to the desired voltage. This amplified signal is then used to light the electric bulb and run motors which ultimately drive the signal arm as desired.

For the prototype purpose, push switches have been simulated to give input signal instead of piezoelectric sensors and amplifiers as shown in Fig. 1. Whereas LED's are used for output detection as a replacement for electric bulbs and motors.

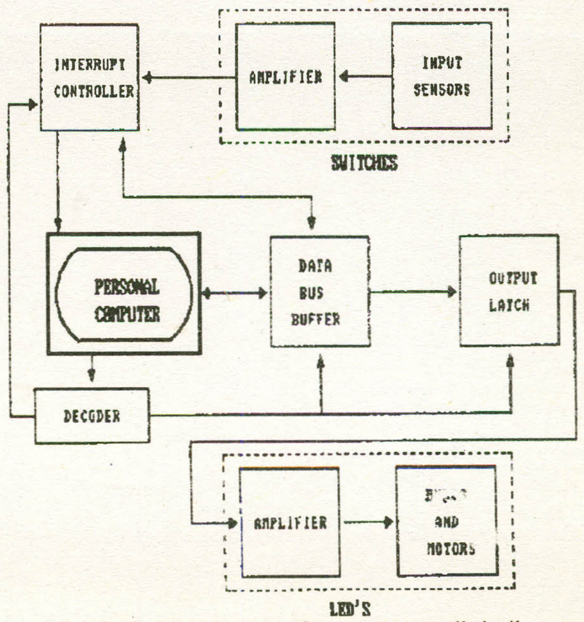

Fig. 1. Block diagram of a personal computer controlled railway monitoring and signalling system.

* As was the case in the head on collision of two shuttle trains between Depot Hills and Gilani Statios on Karachi Circular Railway Network..

Input and output signal conditioning. In computer controlled railway monitoring and signaling systems, sensors are needed to detect the arrival and departure of trains and they must be conditioned to T T L signal level before feeding to a interrupt controller. On the output side, the data on the port must be converted to voltage required by motors and electric bulbs. The technique and devices needed for sensing and conditioning of the data are presented in this section.

Input sensors. A piezoelectric sensor has been chosen for sensing the presence of a train because the piezoelectric transducer is simple and rugged, suitable for many applications where sufficient mechanical force is available. When piezoelectric material is subjected to mechanical stress, it generates surface charge due to deformation of the crystal lattice. This charge can be made to leak away through an external resistance, thus producing a voltage proportional to the imposed stress [6].

It is assumed that there are two main lines for the entrance or exit from a particular railway station. The sensor can be placed on the inner side of the top flange. When the train moves over the rail having this sensor, because of the speed and weight of the train, it will produce the signal of some millivolts. The amount of voltage generated is proportional to the piezoelectric used. So it can be properly designed to give the required voltage signal of some millivolts value.

For the station having two main lines, four sensors can be arranged on the track as shown in Fig. 2 (a). The location of sensor on the track is shown in Fig. 2(b). From the fig. 2(a), it is clear that, if input sensor 1 detects the arrival of up train on the station then input sensor 2 detects the departure of up train from the station. Similarly input sensor 4 and input sensor 3 detect the arrival and departure of the down trains respectively. When the train arrives on any of the main line or leaves from the main line, the sensor on that line detects its presence. As already explained, the output from the sensors will be in millivolts, therefore it has to be amplified to the digital voltage

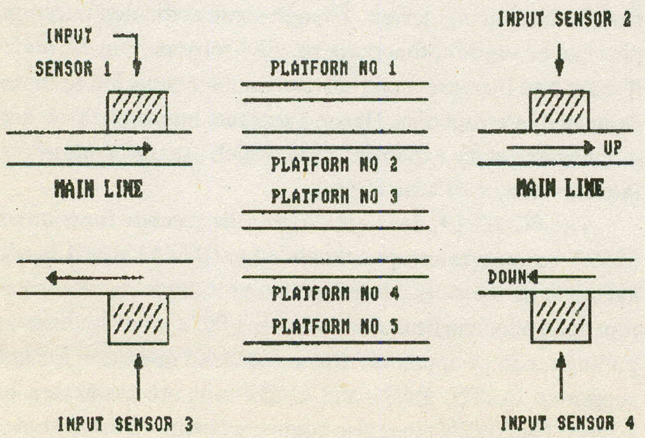

Fig. 2a. Placement of sensors on main line of railway track.

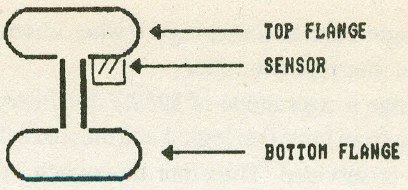

Fig. 2b. Location of sensor on rail.

level of +5 volts to represent logic 1. All the outputs from the amplifier are connected to the interrupt inputs of the programmable interrupt controller 8259A (Fig. 3a).

In the simulated model, switches have been used to indicate arrival or departure of the train, instead of piezoelectric sensors. The sensors are simulated by using signal pole double throw (SPDT) switches. The connection of switches are shown in Fig. 3a. When the switch is activated, the first bounce sets latch, thus eliminating bounce on the activation. Similarly, the latch is reset on the first bounce of the deactivation and thus, is clearly reset. Since the switches have both normally open and normally close contacts and it breaks the normally closed contacts before the normally open contact, debounce effect is easily eliminated. The timing diagram is shown in Fig. 3b. When the switch is at NC position, the outputs SWO to SW3 are at logic 1 whereas if the switch moves to NO position, the outputs of switch SWO to SW3 goes to logic 0. Hence, the outputs of SWO to SW3 are

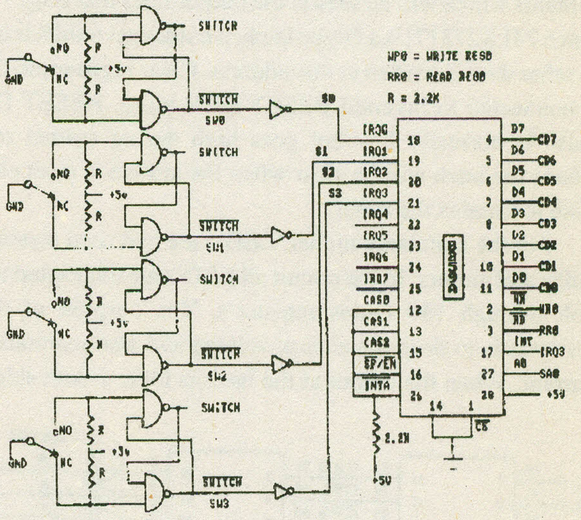

Fig. 3a. Connection of switches and interrupt controller.

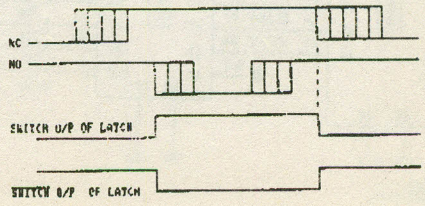

Fig. 3b. Timing diagram of latched switch output.

normally at logic 1 and becomes logic 0 when interrupt signal is given to the interrupt controller.

On the edge trigger mode of 8259A, the interrupt input should change from logic 0 to logic 1 and must be at logic 1 till the interrupt is serviced. Therefore the switch outputs are inverted by using inverters which are normally 'O' and becomes logic 1 when the switch is closed.

Data transceiver and output Latch. Hardware connection of the data transceiver is shown in Fig.4. Chip select line from the decoder is connected to enable pin E and IOR signal from slot is directly connected to the DR pin. As seen from the decoding circuit when chip select line is low, that is, the address 0300H-0307H is selected and IOR is low, microprocessor is doing read operation. Hence data on the card data bus (B side) is transferred to A side connected to slots data bus. On the contrary, when Chip Select is low and IOR is high, microprocessor is doing write operation on the card. In this way, the transceiver transfers data from slots that is A side to data bus of card, that is side B. When chip select is A high, it isolates the card data bus from the slot bus signals.

The Dtype latch is needed to grab data when output latch is selected for write operation. From the decoding circuitry, it can be seen that WRITE REGO pin will be low when microprocessor does write operation at I/O address 0304H or 0305H. This signal enable the latch and grab the data bus contents which will be seen at the output sides that is Q1 - Q8. Since 74LS273 [7] is a Dtype latch, it holds data until it is clear or other data is written at this address. Clear register pin CLR is connected to inverted RESET DRV pulse. RESET DRV pulse is normally low but goes high during system reset. Hence the latch will be clear when the system is reset otherwise it remains the same.

For the simulated model, LED's are and used instead of bulbs and motors. In the circuit all LED's are connected to +5 volt through 180 Ohms register's. The purpose of these resistors is to drop the excess voltage and also maintain the current. When the output at the latch is logic 1 both sides of

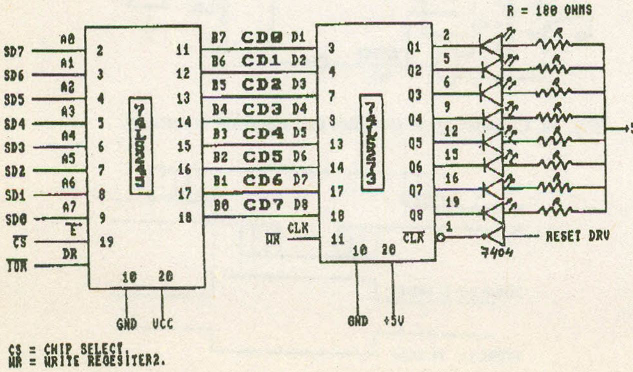

Fig. 4. Data Transceiver and output latch.

LED's are at same potential and hence it will not glow. On the otherhand when the output at the latch is logic O that is almost O volt, there will be potential difference between two sides of LED which causes current to flow through it and therefore LED glows accordingly. It means logic zero must be written at the port to the LED. During reset operation, RESET DRV signal clears all the Dtype flipflops causing all LED's to glow. This can be used to check the conditions of LED's. After checking, during initialization Hex FF can be written at port address 0304H to off all the LED's. Now the port becomes ready to receive the proper signals and glow the LED's accordingly.

Amplifiers and motors. Though there are no amplifiers and motors on the simulated model, these can be added to the output port for actual implementation of the system. It is clear that, the output voltage on the port will not be more than + 5volt and therefore if electric bulbs for signal indication and motors to drive signal arms are required then this voltage level needs to be amplified.

Interfacing to the PC. There were two choices for sending input signal from the sensors or switches to the PC, either by input port or interrupt method. If input port method was employed to read the data directly to the processor, then the microprocessor has to check that input port on certain interval of milliseconds or so and will be busy most of the time performing only this task. Furthermore, if more than one sensor gives input at the same time then there can be problem, because it may not be able to choose the priority. Both of these hurdles can be overcome by using programmable interrupt controller 8259A. The Programmable Interrupt Controller 8259A [8] can accept up to eight different interrupts (IRO-IR7) and will enable one at a time according to their priority. Hence, this solves two problems. Firstly, more than one sensor input can be acquired and secondly, it can detect which sensor has given signal first. There are two 8259A available on the motherboard of the PC-AT, but none of the interrupt lines are free for interfacing design. Though some dedicated interrupt pins can be used for this purpose, all 4 sensors (may be more if main line increases) outputs can not be connected to these dedicated interrupt pins. Hence a separate interrupt controller is connected at the expansion card, which uses only one of the interrupt source of base 8259A.

The PC-AT [9] does not support the cascade function of 8259A because interrupt acknowledge (INTA) signal is not available on the slots. However, the prototype board's interrupt controller can be connected to the PC's interrupt lines in polling mode of operation. Since cascaded operation are not supported CASO, CAS1 and CAS2 pins are connected to +5volt. The SP/EN pin is also connected to the +5 volt indicating that device is master one. As the arrival and departure of the train timing are not fixed, the controller should always be enable and hence its CS is connected to ground. Interrupt acknowledge INTA pin is connected to +5volt because PC-AT I/O slot signal does not provide INTA signal. Data line DO-D7 are connected to the prototype board's data bus and interrupt line IRQO-IRQ3 are connected to the output SO-S3 which are inverted switch outputs. IRQ3 interrupt assigned for serial communication 2 is used for prototype board so INT output pin of expansion controller is connected to IRQ3 of slots signal. SAO pin of slot signal is connected to the AO pin of the 8259A to have two different addresses for internal register. The RD and WR pins are connected to decoder output. With these hardware connection, the interrupt controller is ready for initialization and it accepts interrupt after initialization is completed. The detail of initialization of expansion 8259A can be seen from the data sheet or reference books [10].

Address decoding. Input/output devices can be used in two modes, namely memory mapped I/O mode and I/O mapped I/O mode. In memory mapped I/O, the I/O device is considered as a memory and all the instructions used by memory devices can be used for I/O devices. On the contrary in I/O mapped I/O technique, fixed I/O address is available and special instructions like IN and OUT must be used for data transfer. All devices on the simulated I/O card are used in I/O mapped I/O mode. The IMB PC-AT supports both 8 bit and 16 bit peripheral devices that map into special I/O addresses. The system board is designed so that I/O devices are accessed using special commands within their own separate I/O memory space. I/O memory resides within addresses OOOOH to FFFFH (64K bytes) although little of this addressable space is actually used. From this 64K byte of I/O space, the address from 0300H-031FH is specially assigned for a prototype card and this space is used in our application.

Since all the devices data bus and address bus are connected to the common data and address bus respectively, one and only one device should receive or send the data/addresses on the bus at the same time. If two devices output data at the same time, they create data error problem and to solve this, address decoding technique is used. Address decoder selects one device at a time by activating its chip enable signal and it disables all other devices. Various programmable and nonprogrammable devices can be used for address decoding purpose. Dual 2 to 4 line decoder is used in this application to implement address decoding. Before designing any address decoding circuitry, the first thing to do is to get the required address of the devices. In this application only two devices need addresses and they are interrupt controller (2 locations) and output latch (1 location). Though, they need only three locations, AO is already used for the selection of either of two addresses of the interrupt controller and therefore can not be used in address decoder. It is necessary to use A1 and A2 to select either of two devices from four possible devices. This will also make provision that two more devices can be added in future if expansion is required.

The address lines A3 to A9 and AEN pin are used to generate chip select line. They are connected to different gates as shown in fig.5 in such a way that chip select line goes low when 0300H to 0307H I/O address is selected. Dual 2 to 4 decoder (74LS139) uses this chip select line together with SA1 and SA2 to generate the required read/write and chip select signals for interrupt controller, output latch and transceiver. The address of interrupt controller is 0300H and 0301H, whereas output latch occupies 0304H.

Software development. The software of this PC based railway monitoring and signaling system is comprised of three portions. They are 'initialization', 'interrupt routines' and 'application programe'. The role of initialization routine is to initialize the system base interrupt controller (8259A). As the system works on interrupt basis, interrupt routines actually do the function of monitoring and signaling. The application program is written to facilitate the user or operator for entering, displaying, updating, appending and removing the train data.

Initialization of the expansion interrupt controller. As already mentioned, the cascaded function of the interrupt controller on the system board is not supported. Hence, to expand the interrupts, the expansion interrupt controller must be operated in poll mode. Since the expansion interrupt

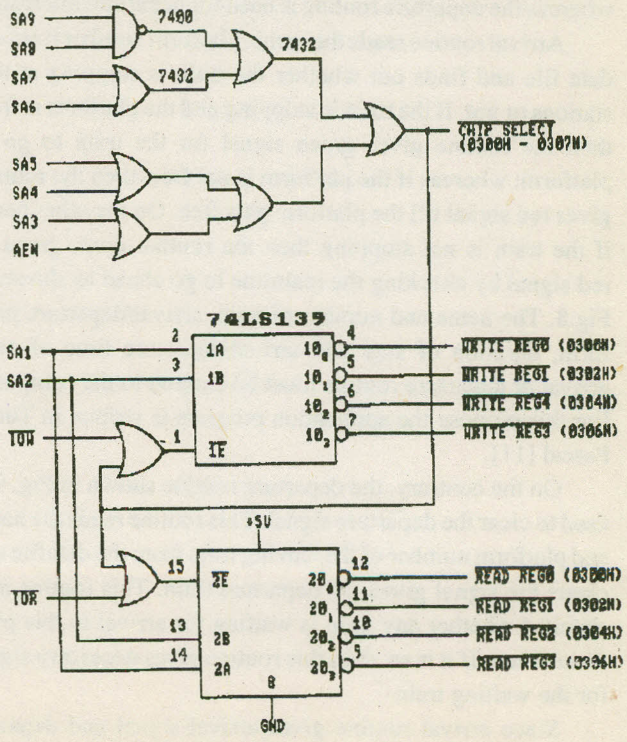

Fig. 5. Address decoder circuitry.

controller is not required to operate in the same mode as that of the controller on the baseboard, it gives the interface designer greater freedom to select the modes of operation that are best suited for the application. However there are some modes that must be selected of the scheme to operate properly. During initialization, the expansion interrupt controller has to be selected to operated in master mode and 8085 mode. The expansion interrupt controller is initialized to operate in a fully nested mode with edge triggered interrupt request inputs and a normal end of interrupt mode as shown in Fig. 6. The interrupts that are generated as inputs to the expansion interrupt controller causes interrupt requests on the base system boards interrupt controller. Thus, a special interrupt service routine is required to handle the expansion interrupts. Fig. 7 shows flowcharts of initialization and service routine for handling and directing interrupts accordingly.

Interrupt service routine. The general flowchart of the expansion interrupt service routine is shown in Fig. 7(b). Although the service routine used will be of a different type, it follows all the house keeping given in the flowchart of (Fig. 7(b). Presently, there are four interrupts on the PC based railway monitoring and signaling system and they can be easily expanded to eight by adding a small amount of hardware and software. Among four interrupts, the first two are arrival interrupts and the other two are departure interrupts. Therefore, for arrival interrupts, the arrival routine is used; whereas the departure routine is used for departure interrupts.

Arrival routine reads the name, platform number from the data file and finds out whether the train is stopping at this stations or not. If the train is stopping and the platform is free, then the routine gives green signal for the train to go to platform; whereas if the platform is not free, then the routine gives red signal till the platform gets free. On the other hand, if the train is not stopping then the routine gives green or red signal by checking the mainline to go ahead as shown in Fig.8. The name and number of train, arrival/departure platform, duration of stay and arrival/departure time of each arrival or departure routine must be entered to the computer. For this purpose the application program is written in Turbo Pascal [11].

On the contrary, the departure routine shown in Fig. 9 is used to clear the departure signal. This routine reads the name and platform number of the leaving train from the datafile and clears the signal given for departure train. This routine also checked whether any train is waiting for arrival to this platform or not. If it is so, then this routine gives necessary signal for the waiting train.

Since arrival routine gives arrival signal and depature routine clears the departure signal, another routine is needed to clear the arrival signal after a certain interval of time and

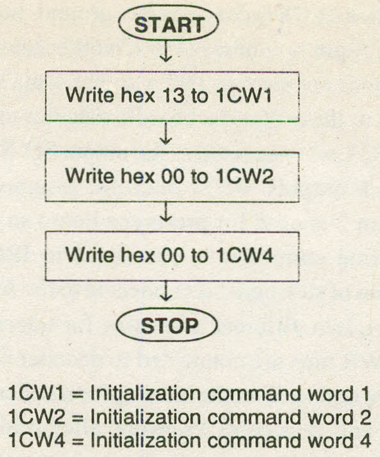

Fig. 6. Flow chart for initialization command of expansion interrupt controller (8259A).

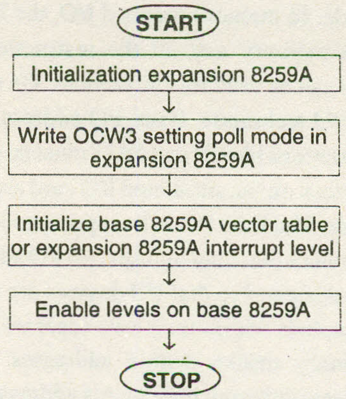

Fig. 7a. Initialization routine for the expansion interrupt controller.

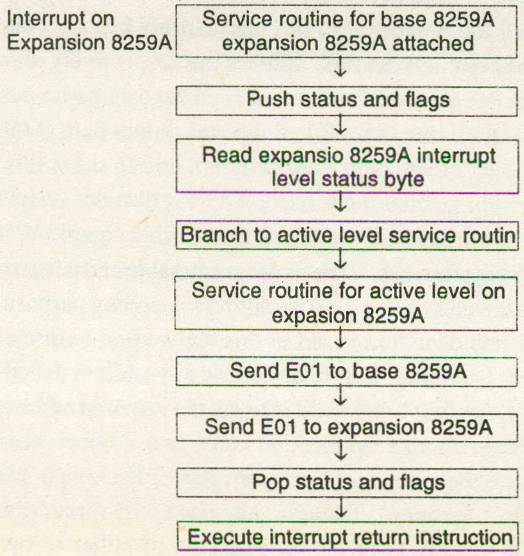

Fig. 7b. Interrupt service routine for the expansion interrupt controller.

also to give departure signal to the train when it is supposed to leave. User timer interrupt is used for this purpose which is available on the BIOS interrupt routine.

As these routines are more related with hardware control, therefore these are implemented in 80286 assembly language [12].

INT OBH Service Routine. The complete flowchart of the interrupt routine for interrupt INT OBH is shown in Fig. 10. The routine uses four data files to read the appropriate data depending upon the interrupt types. These data files are set from the application routine. The data variable PFNUMBER, DSTAY, INTTRYPE, and STATUS are used to hold the platform number, duration of stay, interrupt type and status of signals respectively. Platform number and duration of stay are read from the data file. INTTYPE will be 0,1, 2 or 3 depending upon the interrupt such as arrival up, arrival down, depature up or departure down respectively. A single byte is used to hold the status to the signals. The eight bits of this byte are assigned for different purposes as shown below. Since two mainlines are assumed there must be two signals for two arrival trains and there must be two signals for two depature trains.

7	6	5	4	3	2	1	0 bit
ARRUP green signal	ARRUP red signal	ARRDW green signal	ARRDW red signal	DEPUP green signal	DEPUP red signal	DEPDW green signal	DEPDW red signal

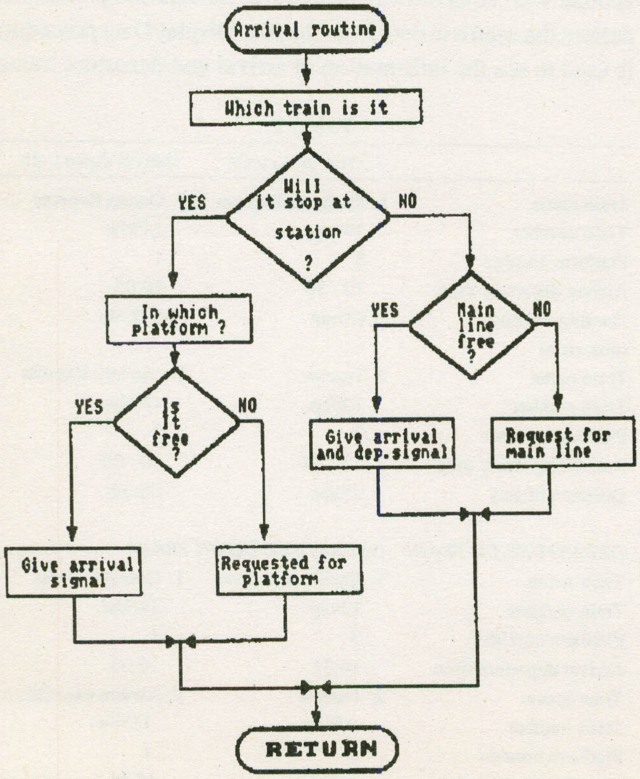

Fig. 8. Arrival rotine flowchart.

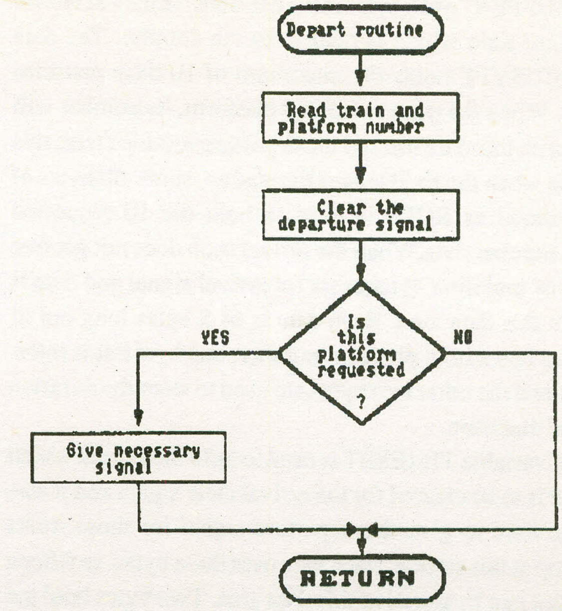

Fig. 9. Departure routine flowchart.

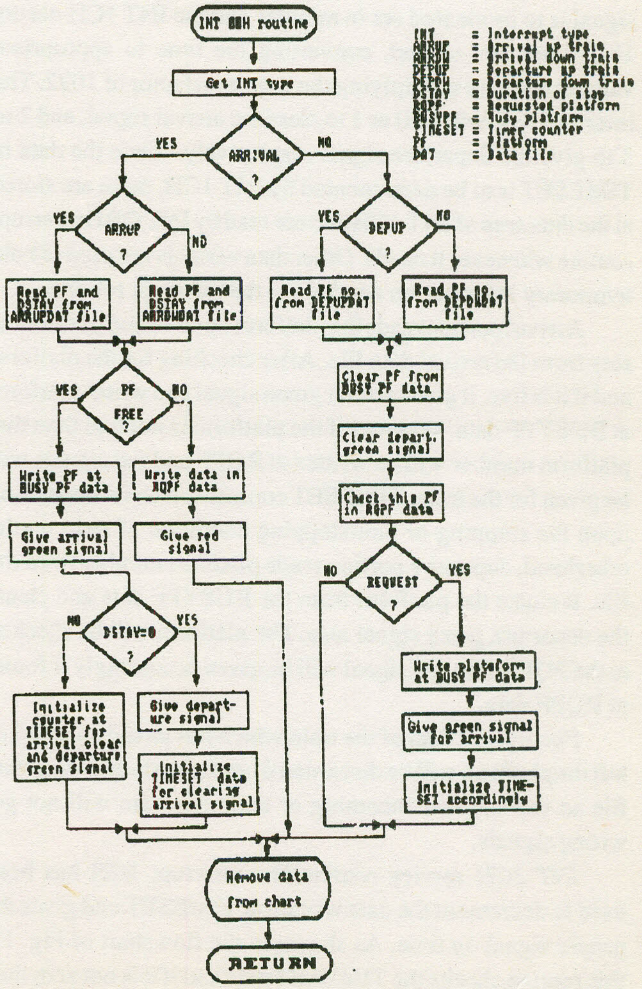

Fig. 10. Interrupt service routine flowchart for interrupt INTOBH.

A "BUFFER" of 48 bytes is used to hold the data of arrival or departure train which is read from the datafile. The data variable BUSYPE holds the maximum of 10 busy platform numbers. When the train enters the platform, its number will be written on this data file and the departure routine clears this data value when the train leaves the station. some 50 bytes of data separated as RQPF is used to hold the 10 requested platform number data. When the arrival train does not get free platform or mainline, it requests for arrival signal and data is written in this data area. Each data is of 5 bytes long out of which two bytes hold platform number, one byte holds interrupt type and the other two bytes are used to store the duration of stay of that train.

Data variable TIMESET is used to hold the time at which the signal is to be cleared for the arrival clear signal and it also holds the time to give the departure signal for those trains which stop at this station. Each data uses three bytes, so fifteen data values can be stored in this data area. Two bytes hold the converted values of time and the third byte holds the type of the interrupt. Since the duration of stay and the time at which signal is to be cleared are in minutes and the INT 1CH occurs 18.2 times per second, converting the time to appropriate value is done by multiplying the time by a factor of 1092. The interrupt type will be 0 or 1 to clear the arrival signal, and 2 or 3 to give the departure signal respectively. Since the data in TIMESET is to be decremented by INT 1CH, these are stored at the data area of INT 1CH and are read by INT OBH interrupt routine whenever it needs. Other data variable are used to hold temporary information needed for the interrupt routine.

Arrival portion reads the platform number and duration of stay from the arrival data file. After checking for the platform and if it is free, it gives arrival green signal and writes platform at BUSYPF data. Whereas if the platform is not free then that platform number will be written at RQPF and red signals will be given for the train. TIMESET counter can be set depending upon the stopping or non-stopping condition of train on the otherhand, departure portion reads platform number from the file. It clears the platform from the BUSYPF data and clears the departure green signal also. The platform will be checked at the PQPF data and signal will be given accordingly if found at RQPF data.

Finally, the data of the train which has already arrived or left the platform will be discarded from the corresponding data file so that another incoming or outgoing train will not get wrong signals.

INT 1CH service routine. The interrupt 1CH has been used to decrement the data written at TIMESET and gives the proper signal on time. As shown in the flowchart of Fig. 11, this routine checks the TIMESET data and if it is not zero then decrement procedure begins. As soon as it detects TIMESET equal to zero, appropriate signal will be given by this routine depending on the interrupt number. This routine is executed at the speed of 18.2 times per seconds.

Application programs. These are the programs which are specially developed for entering and manipulating the data and are called application programs or utility programs. To make the system more versatile and user friendly, these type of application programs are essential. In this railway monitoring and signaling system, the following procedures are written in Turbo Pascal for entering and manipulating the data.

* Read data
* Display data
* Append data
* Update data
* Delete data

To make the system easier all the procedures are menu driven and they can be called by the main program accordingly.

Working of simulated system. To describe the complete working of the system let us assume the following dummy data as the sequence of arrival and departure of trains Table 1.

The arrival and departure train data can be stored in their respective files using the application program. For example, to enter data the first time, Read Data procedure can be used. Once data has been entered, changes can be done from UpdateData routine; whereas adding data to existing data file is done with AppendData procedure. DeleteData procedures deletes the used trin data from the file. DisplayData procedure is used to see the information of arrival and departure trains.

Table 1.

	Arrival of up train	Arrival down train
Train name	1. Shalimar Express	1. Quetta Express
Train number	139up	145dw
Platform number	3	1
Arrival departure time	10 : 30	10:05
Duration of stay (non-stop)	03min	OOmin
Train name	2. Tezrow	2. Awami Express
Train number	150up	163dw
Platform number	3	1
ArrivalDeparture time	12 : 30	10 : 50
Duration of stay	05min	05min
DEPARTURE UP TRAIN	*DEPARTURE DOWN TRAIN*	
Train name	1. Shalimar express	1. Quetta express
Train number	139up	145dw
Platform number	3	1
Arrival departure time	10:33	10:05
Train name	2. Tezrow	2. Awami express
Train number	150up	163dw
Platform number	3	1
Arrival departure time	12:35	10:55

Once the data entering and manipulating process is complete, the personal computer becomes ready to receive interrupt and responds accordingly.

Now, suppose if arrival upward interrupt is given by first switch, the computer knows that it is ARRUP train interrupt and hence opens ARRUPDAT file and reads Shalimar express train data. Since all the platforms are free initially, PC gives green signal to go to the platform number 3 and initializes two counters, one to clear arrival upward green signal (say after 1 min) and other of duration 3 minutes to give departure green signal. After 1 minute the arrival up green signal will be cleared and departure green signal will be given after 3 min. Now, if the interrupt comes again from ARRUP switch then PC first checks whether Shalimar Express has left platform number 3 or not because Tezrow is also supposed to go at platform number 3. If Shalimar Express had already left, that is, given DEPUP interrupt from switch three, then PC gives arrival green signal and initializes two counters as explained for the first train. On the contrary if the platform is not free then platform 3 will be written in the requested platform list and a red signal gives for Tezrow. As Shalimar leaves platform number 3, departure routine first clears the departure up green signal and then checks for requested platform. Since platform number 3 is on the requested list platform, departure routine gives green signal for Tezrow and removes platform number from requested platform list

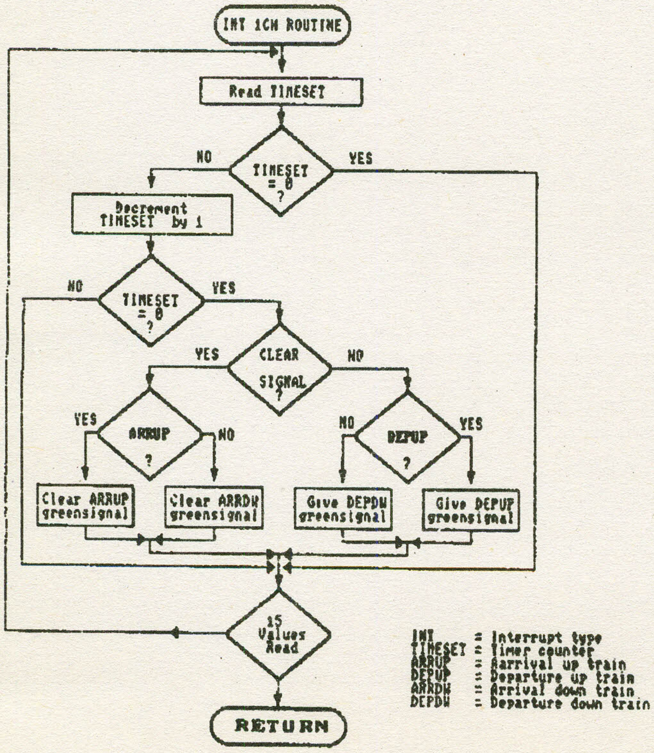

Fig. 11. Interrupt service routine flowchart for interrupt INT1CH.

In addition, if any nonstopping train comes on platform number 1 then the nonstopping train gets both arrival green and departure green signal at once. To clear arrival green signal, a counter is used; whereas departure green signal will be cleared through interrupt when the train leaves the station. The serial.of train names must be ordered according to the arrival or departure timing sequence and the data must be changed using application programs if the sequence is to be modified. The arrival and departure down routines also work exactly in the same manner. In this way arrival and departure routines give proper signal to the incoming and outgoing trains.

Conclusions and Suggestions. In this paper, initially the need for monitoring and signaling in respect of railway system was emphasized. A solution based on a personal computer utilizing programmable interrupt controller (PIC) has been proposed and a laboratory prototype as such designed and demonstrated. The working of the system can be summarized as follows:

When a train arrives, sensor on the track detects its arrival and gives signal to the PC through programmable interrupt controller. After receiving the signal, PC checks; whether the train is going to stop or not and gives necessary signal if it is not stopping. If it is stopping, first it checks the platform and gives necessary signal if platform is vacant. When train leaves the station, sensor on outgoing track detects its departure and hence clears the platform and mainline for the incoming trains.However, in the prototype model sensors are simulated by push switches to indicate input signal; whereas LED's are used for output detection as a replacement of electric bulbs. A complete I/O card for the PC interface is designed for getting signals from the switches, processing the data and giving signals accordingly. Main interrupt routines are written in assembly language and application programs to manipulate data are written in Turbo Pascal.Although, only two main lines have been assumed, this work can be expanded for any number of main lines by simply adding adequate circuitry. As interrupts are used to detect the presence of trains, therefore, whenever a train comes or leaves, it automatically executes the control routines. This means the PC can be used for any other purpose when there could not be any train routine is not executing. To make the system error free, reliable, and versatile, all the possible alternatives were studied and the best possible system developed. However, the limitations cannot be ruled out in any new idea.It is hoped that by the implication of this modern system, in conjunction with the existing facilities, for monitoring and signaling of the train by the railway department, railway disasters may largely be avoided and railway travel would become safer and more reliable for the passengers.

Acknowledgments. One of us (D. R. Gautam) is thankful to Ministry of Education, Government of Nepal and Ministry of Finance, Economic Affairs Division, Government of Pakistan for giving assistanceship under Pakistan Technical Assistant Scheme. Thanks are also due to M.K. Kella for drawing nice flowcharts and figures. The authors are grateful to Mehran University of Engineering and Technology for providing facilities to do the research work in the department of Electronic Engineering. The listing of software and complete hardware circuit can be obtained from the authors.

References

1. R.F. Herrman, A Microprocessor Controlled Railway Signalling Interlock in Industrial Applications of Microprocessor, IECI Proceedings, 2-7 March (1977).
2. K. Mori *et al,* Autonomous Decentralized Loop Network and Its Applications to Train traffic Control System, IECON'84, Tokyo, October (1984).
3. Mian Ghiasuddin, Modernization of Pakistan Railway, The Pakistan Engineer, May (1993).
4. K. Takashi, Concept of Friendly Autonomous Decentralized System for Next Generation Train Traffic Control, Proceeding of ISADS 93 (International Symposium on Autonomous Decentralized System), Kawasaki, Japan, March-30 , April - 1 (1993).
5. Kurt S. Lion, *Instrumentation in Scientific Research Electrical Input Transducers* (Mc Graw-Hill, 1959).
6. G.C. Barney, *Intelligent Instrumentation:* Microprocessor Application in Measurement and Control (Prentice Hall International, 1988).
7. TTL Data, Texas Instruments.
8. *Microprocessors and Peripherals Handbook,* (Intel Corporation, 1989), Vol.1.
9. Lewis C. Eggebrecht, *Interfacing to the IBM Personal Computer* (Mc Millan Computer Publishing U.S.A., 1992).
10. Douglas V. Hall, *Microprocessors and Interfacing Programming and Hardware* (Mc Graw-Hill International Edition, 1992).
11. *Trubo Pascal Rereference Manual* (Borland International Inc. U.S.A., 1991).
12. Krip R. Irvine, *Assembly Language for the IBM PC* (Maxwell Macmillan International Edition, 1990).