

## A simple optical fibre-linked remote control system for multiple devices

M A SUMESH<sup>1</sup>, BOBY JOSEPH<sup>2</sup> and D P MAHAPATRA<sup>2</sup>

<sup>1</sup>Indian Institute of Science, Bangalore 560 012, India

<sup>2</sup>Institute of Physics, Bhubaneswar 751 005, India

e-mail: dpm@iopb.res.in

MS received 11 June 2003

**Abstract.** We report on the development of a simple control system which can handle multiple devices through an optical fibre data link. The devices are controlled using a set of DACs through serial data communication via a serial port of a PC. Serial data from the PC get converted to parallel mode using a homemade “serial in parallel out” (SIPO) device at the remote end. The functioning of this to control multiple devices is presented and discussed.

**Keywords.** Optical fibre data link; remote control system; multiple devices.

### 1. Introduction

In most accelerator laboratories, usually at least one set of power supplies has to be controlled remotely. This requirement is either due to restrictions imposed because of radiation environments or due to high voltage/high magnetic fields etc. A fibre-optically linked communication system is the best choice for such an application. A standard solution is to use the Group3 control system (Website 2), which is a fibre-optically linked distributed control system commercialised by Group3 Technology, New Zealand. In the present case we have developed a much simpler remote communication system which uses the serial port of a PC together with a serial-to-parallel converter coupled by an optical fibre link. In this system there is no PC add-on card and one can do away with the intelligent processor which serves the various devices to be controlled at the remote end. The software as required for this is simple and is very easy to implement.

This system has been developed as a part of a project to control a negative ion source, SNICS (source of negative ions by cesium sputtering, NEC, USA) which forms the major component of a low energy negative ion implanter (see Middleton 1983, *Instruction Manual* 1985). This source sits at a high voltage of  $-75$  kV thereby producing singly charged negative ions which get accelerated to the ground potential with a final energy of 75 keV. It can produce ion beams for all elements which form stable negative ions. It uses accelerated cesium ions striking a cold cathode to produce negative ion beams of the cathode material. Cesium vapour comes from the cesium oven into an enclosed area between the cooled cathode and the heated ionizing surface.  $\text{Cs}^+$  ions are accelerated to the negatively biased cathode, which then sputter

negative ions away. A thin layer of cesium condensed on the cathode surface enhances the negative ion production. In general, negative ion beam current is a function of the cathode composition, the cathode potential, the cesium ion flux and cathode temperature. There are six power supplies: one for Cs oven heater, one for cathode voltage, one for extraction voltage, one for ionizer current supply, one for focusing the emergent ion beam and one for applying the source bias. Out of these, the first five are the ones which float at a high negative potential (bias) and need to be controlled remotely with complete electrical isolation. For optimum output (of the beam current) we have to control various parameters such as cathode voltage, cesium oven temperature, ionizer current, extractor voltage, focus voltage etc.

The power supplies used for the SNICS have both local and remote programming facilities.

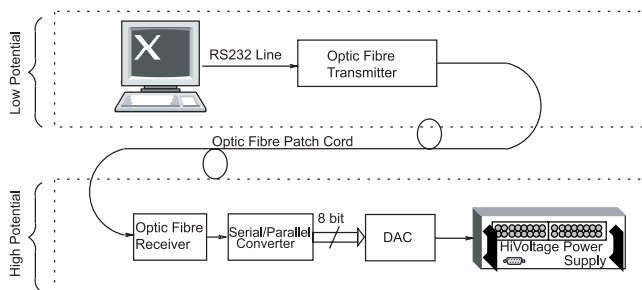
The potentiometers mounted on the front panel of the power supply cabinet facilitate local programming, while remote programming is made possible through an analog port on the rear side. The remote control of the power supplies requires an analog voltage of 0–10 V. The outputs of the power supplies are directly proportional to the input voltage at the analog port. In the SNICS controller we are making use of this facility for remote control. The conventional method for communication with this analog port is through an electrical cable. But this is not possible in the case of SNICS control since all power supplies (excepting one, i.e. bias) actually sit in a high voltage enclosure. The present system has primarily been developed because of this.

In the following section, we present a description of the scheme where we have shown how to control both single and multiple power supplies. Implementation of the scheme for SNICS is described after that. The electronics involved and its implementation is presented in the next section which is followed by a section on typical results and conclusions.

## 2. Scheme

In this section we explain the basic scheme to control a single power supply by sending serial data from a PC. Figure 1 shows the block diagram of the scheme. The programming voltage required for the remote control of the power supply can be derived from a DAC, with a reference voltage of 10 V. The serial data from the COM-port is used to drive an optical fibre transmitter. An optical fibre carries the serial data from the transmitter to the receiver at the remote end. This serial data need to be now converted into parallel mode for controlling the device. This can be done using a “Serial in parallel out”(SIPO) device followed by a DAC.

To achieve control over more than one power supply at high resolution, one needs to modify the primary system described above, by adding three 8-bit latches (8-bit address latch and 16-bit data latch), at the output of the SIPO (refer figure 2.) Also, through software, one



**Figure 1.** Scheme to control a single high voltage power supply.

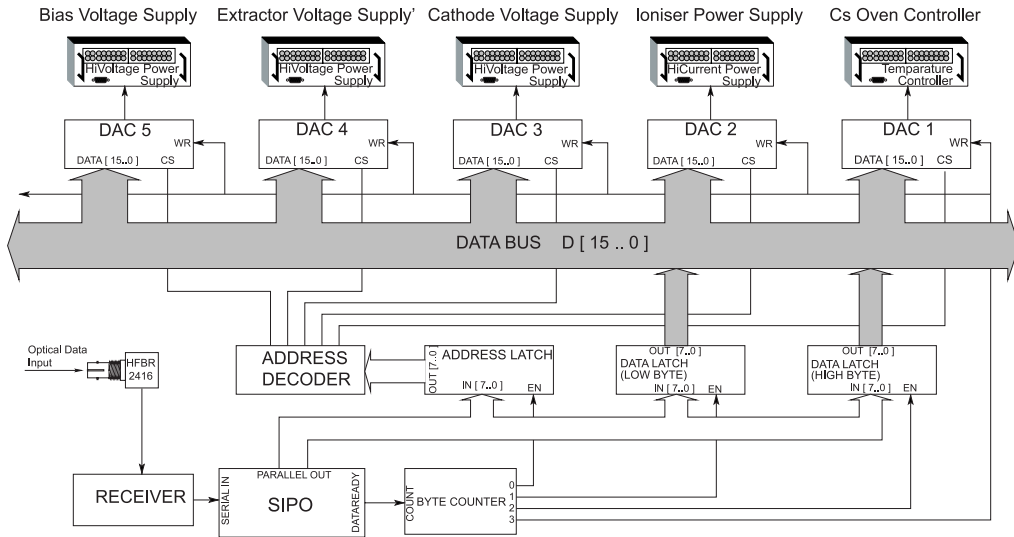


Figure 2. Scheme to control more than one power supply.

needs to modify the data packet, combining 16-bit data with an 8-bit address. The modified data packet is shown in figure 3, and contains an 8-bit address followed by a 16-bit data and control byte. In this scheme we can use a DAC with a maximum resolution of 16-bits.

After the controller is switched ON, the first data byte, is recognized by the controller as the address byte. Only that particular device which has the same address is selected to receive the 16-bit data after the address. On the reception of the control byte this particular device reads the data bytes latched at the data latch.

### 3. An example: SNICS control

In our requirement, each of the power supplies in the SNICS is assigned an 8-bit address. The analog ports of these power supplies are driven by five 16-bit DACs. We used AD669 which is a monolithic CMOS DAC from Analog Devices (Website 1), with built-in data latch and reference voltage (10 V). The 8-bit address latched in the address latch of the controller is used to select the corresponding DAC. The 16-bit data are written to the DAC during the presence of the control byte. The control byte can be any 8-bit data.

Power supplies used in SNICS, which need remote control, are cesium oven temperature control, cathode, extractor and focus power supplies and ionizer supply. Here, as an example, we describe the controlling of one of them (say, for example, the extractor power supply). Let the address assigned to this be  $82_H$ . This means, the decode output for  $82_H$  is used to enable

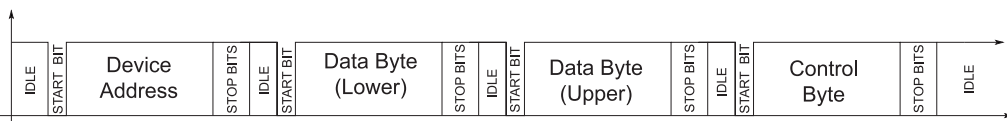


Figure 3. Timing diagram of the data packet.

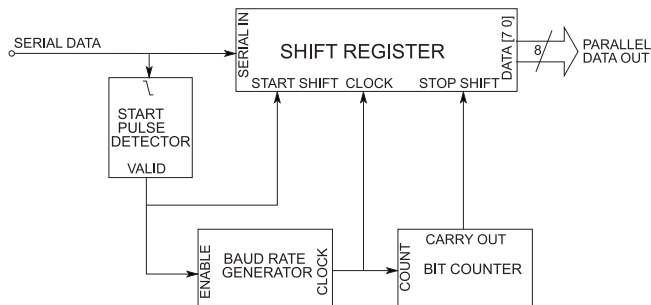
the DAC corresponding to the extractor power supply. Now the lower byte of the data is send through the serial channel followed by the upper byte. The lower and upper bytes are then written to the internal data latch of the DAC on reception of the control byte. Thus the output of the power supply can be controlled with a resolution of  $1/(64k - 1)$ . Similarly we can control all other power supplies. Interestingly, in our case the ionizer power supply requires a programming voltage between 0–5 V. The only difference here is the analog port voltage range. In this case we used an external 5 V reference voltage.

#### 4. Details of electronics

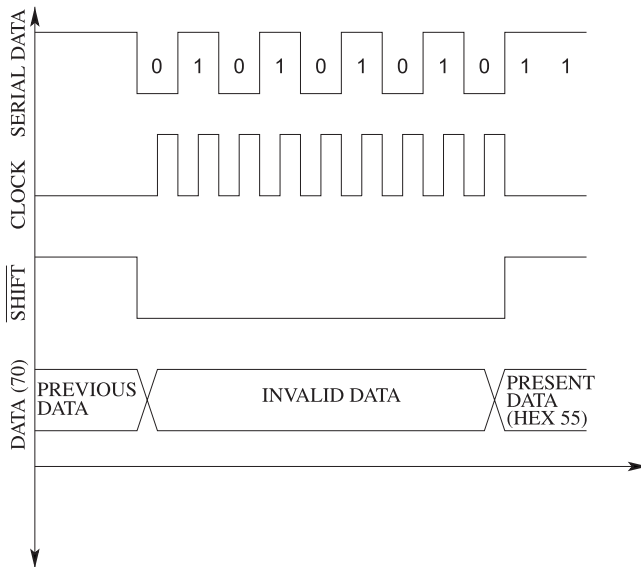
The serial port sends a user given data (8-bit) as a frame of one byte sandwiched between one *START* and two *STOP* bits at a given (programmable) baud-rate. We have used a baud-rate of 2400. This data must conform to the RS-232 protocol. The negative RS-232 compliant data output is not compatible with the optical fibre transmitter which requires TTL logic levels. The required level shifting is done using a commonly available IC 1489. The TTL serial data are then used to drive an optical fibre transmitter ( $\lambda = 632 \text{ nm}$ ) using a driving circuit. For the transmitter, we have used an ST Module, HFBR 1414. A standard  $125 \mu\text{m}$  multi-mode optical fibre (commonly used in computer networking) is used for carrying the optical signals from the transmitter to the receiver. A photodiode receiver HFBR 2416 is used to convert the optical signal to corresponding electrical signal. This signal is suitably amplified using op-amp TLO82 and shaped using a Schmitt trigger wired around IC 555 (Floyd 2002) before going to the SIPO.

A block diagram of SIPO is shown in figure 4. The heart of the SIPO is an 8-bit shift register which converts the serial data into parallel mode and makes it available at the output. A start pulse detector enables the shift register and a baud-rate generator. The baud-rate generator synchronises the shifting operation of the register with serial data input. A bit counter supervises shift register by counting the required number of shifts.

The timing diagram of the SIPO is given in figure 5. In the figure, the serial data stream is shown at the top. This is user specified and forms the input data stream from the RS-232 COM port. As an example, in the figure we have taken  $55_H$  as the value with some fixed time difference between the bits. When there is no data transfer, the serial line is held at logic HIGH (i.e. MARK state). A HIGH to LOW transition of this line implies start of data transfer. Now shift register starts shifting until it shifts nine times. On completion of the 9th shift the *START* bit will get shifted out and discarded. Now the data present in the shift register is in the parallel form of the serial data stream (after the elimination of the *START* and *STOP* bits).



**Figure 4.** Block diagram describing the serial to parallel converter (SIPO).



**Figure 5.** Timing diagram of serial to parallel data converter.

Data flow through the optical fibre at a baud-rate of 2400 as fixed in the present case. One of the important aspect is to synchronize the operation of the SIPO with incoming data stream. This is achieved by generating an internal clock running at 16 times the baud-rate. Details of the circuit diagram are given in figure ?? . An IC 555 wired as an astable multi-vibrator provides the required internal clock at the given rate of 38.4 kHz. This clock frequency is divided down to the baud-rate by a “divide by 16 counter” 74LS193 (referred to as baud-rate generator). The clock generator starts at the falling edge of the serial line. The MSB line, QD is derived as the CLOCK signal, that goes HIGH exactly at middle of the serial data bit. This way we synchronize the CLOCK with the input data stream with a maximum possible error of 6.25% (assuming the local oscillator is running exactly at 16 times the baud-rate).

A HIGH to LOW transition of the serial data sets the D flip-flop U4A, whose D input is tied to HIGH. The active low output  $\overline{Q}$  provides the  $\overline{SHIFT}$ , which enables the baud-rate generator 74LS193 and the bit counter CD4017. The D flip-flop U4A remains SET until nine pluses from the baud-rate generator are out. The decode output corresponding to  $9_H$  is used for clearing U4A. Since the CLR inputs of 74LS74 (Heiserman 1993) are asynchronous, we have to synchronize this signal with the falling edge of the CLOCK. This is required because an asynchronous CLEAR using decode  $9_H$  output will cause a short term CLK pulse (pin number 6 of IC 74LS164) whose width may not be sufficient to be recognized by the shift register and in turn may cause uncertainty on the ninth shift. Finally, we extract 8 data bits out of the serial data stream discarding the START and STOP bits. The SHIFT signal can be used to count the number of bytes which is needed as described below.

A byte counter (CD4017) discriminates between the address byte, the lower data byte, the upper data byte and the control byte. When the controller is idle (with no data transmission through fibre), output of byte counter is  $0_H$ . As the first byte is received, the counter counts up by 1, and the corresponding decoded output saves this data into the address latch. Similarly, the next two bytes (lower and upper bytes of the data) are saved in the 16-bit data latch. The next byte is recognized as the control byte the value of which is insignificant. The  $\overline{SHIFT}$  signal from the SIPO, corresponding to the control byte is used as  $\overline{WR}$  for the DACs. After this the byte counter resets and the controller gets ready to receive a new data packet. The output

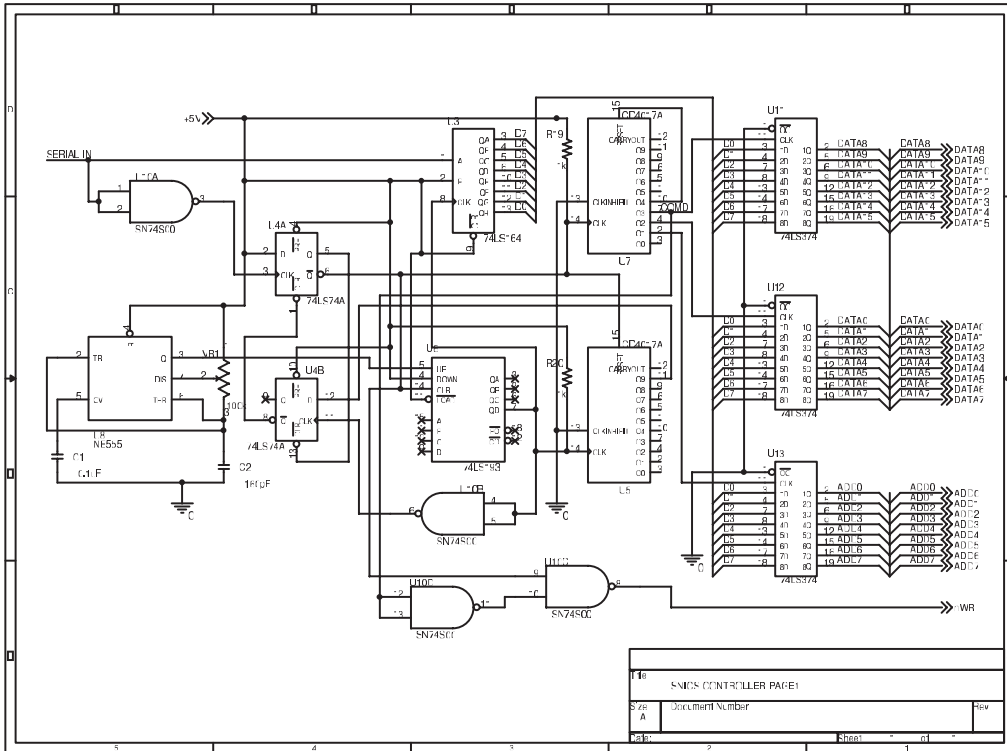


Figure 6. Circuit diagram 1.

of address and data latch form common address and data buses respectively. Use of tristate buffer is not necessary here since all devices connected to these buses are input devices.

While controlling multiple devices, as in the present case, it is necessary to assign a specific address to each of them. At the same time it is convenient to have these addresses as adjacent numbers. For controlling five power supplies we have used addresses 80<sub>H</sub> through 84<sub>H</sub>. The first nibble is selected using the DIP switch (S1) and the next nibble is decoded using a 3-line-to-8-line decoder, 74LS138 (figure 7).

Details of electronics regarding the digital to analog conversion is given in figure ???. The 16-bit DAC AD669 is used to generate a 0–10 V output as required for the analog input port of the power supply to be controlled. The data lines DB0 through DB15 are connected to the common data bus. The chip select  $\overline{CS}$ , is driven by the address decoder and the  $\overline{WR}$  signal activates  $\overline{L1}$ . When both  $\overline{CS}$  and  $\overline{L1}$  signals are LOW, the data bus content is placed into DAC and corresponding voltage appears at the output. Three 10 $\mu$ F capacitors decouple the power lines. As has been mentioned earlier AD 669 has its own voltage reference (10 V) but an additional LM 336-5 or similar chip can be used for meeting the requirement of 0–5 V program whenever needed.

### 5. Sample results and conclusions

In this section we present some typical test results. As a test case we have taken a set of repetitive data of value 55<sub>H</sub>. (This value is selected in conformation with the figure 5.) The

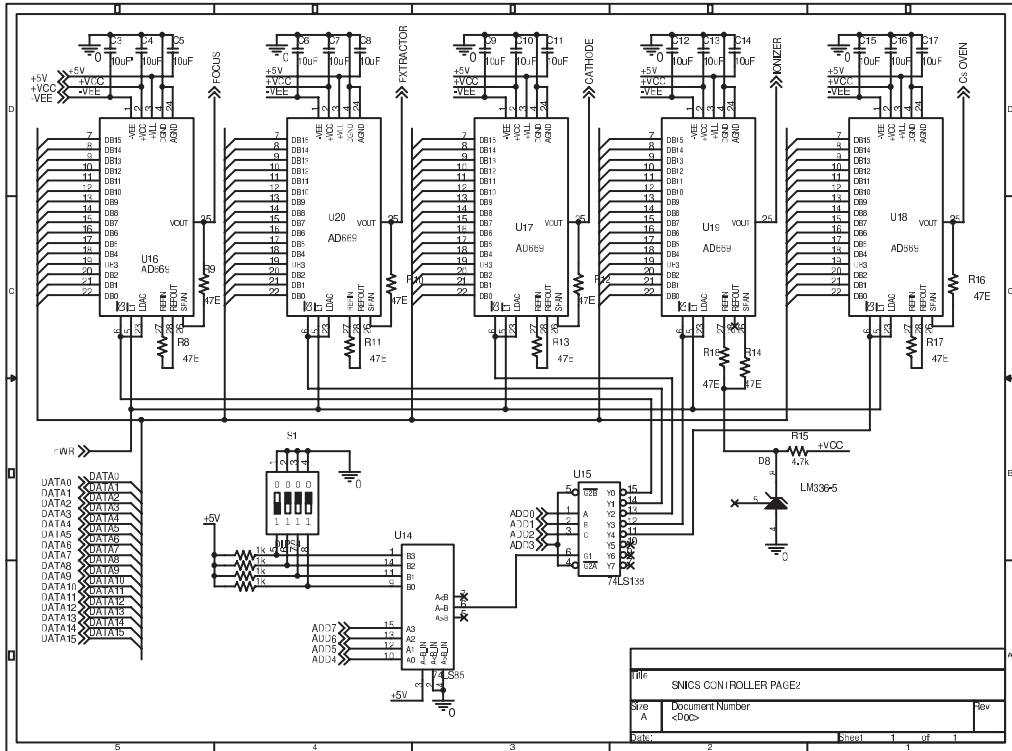


Figure 7. Circuit diagram 2.

interval between two data bytes (or characters) is taken to be 10 ms. This data stream together with corresponding data both at transmission and receiving ends are shown in figures 8 and 9. These are screen-shots of an oscilloscope used for testing the system. Figure 8 shows output

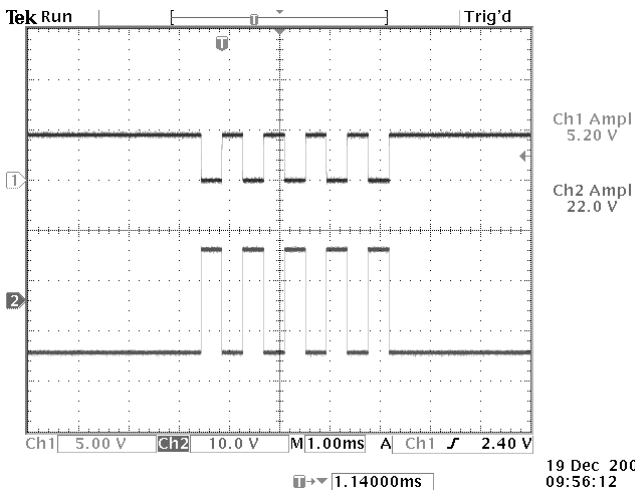
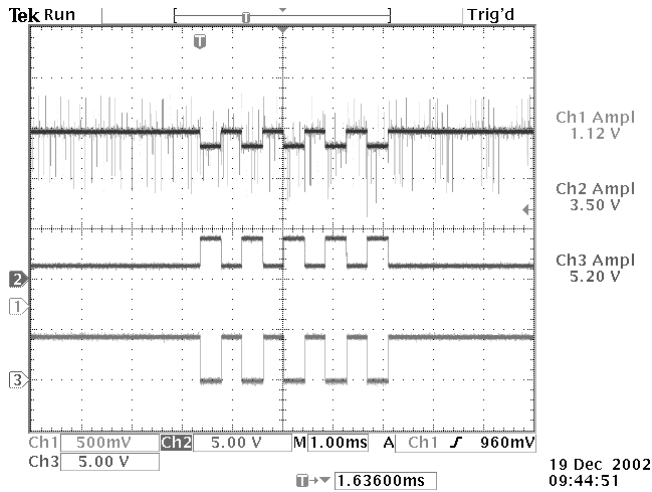


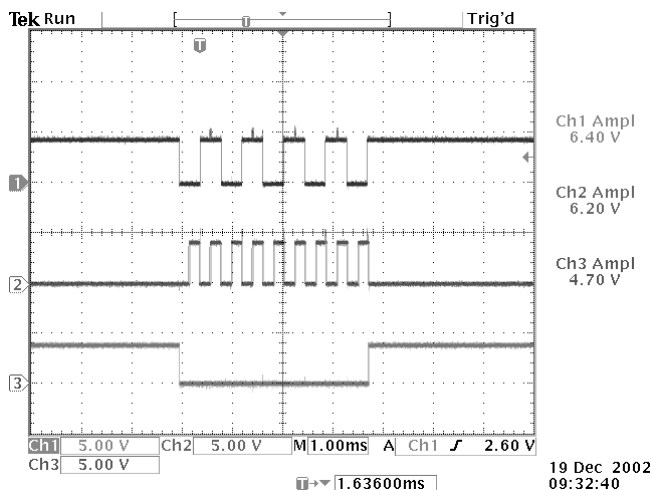
Figure 8. Oscilloscope screenshot showing the serial data in the transmitter side. Channel 1 is the TTL logic level and channel 2 gives the RS-232 signals.



**Figure 9.** Oscilloscope screenshot in the receiver side.

of the COM1-port (RS-232) and the TTL signal obtained after the level shifting. Figure 9 shows the signals at the receiver (remote) side. The top one is the output of the photodiode receiver. One can see a lot of noise superimposed on the signal. The second trace is the signal after the amplifier where the noise is greatly reduced since the amplifier also acts as a low-pass filter. This is fed to the Schmitt trigger. The bottom trace corresponds to the serial data recovered. Figure 10 shows the SIPO stage wave forms. One can compare these wave forms with those given in figure 5. This figure in terms of three traces shows the serial data, the *CLOCK* and the *SHIFT* signals.

Presently the controller is being tested for the control of five power supplies. However, more devices can be controlled independently using the same procedure. SNICS requires a sequence of events for its startup and shutdown. One can implement software algorithms for smooth startup and shutdown using the present system keeping the hardware intact. Since data transmission is digital, we can control valves and switches which need enable/disable



**Figure 10.** Oscilloscope screenshot at the SIPO stage.

pulses. In the current system data transmission is simplex (uni-directional). For monitoring parameters such as pressure, temperature and status signals from various power supplies, valves and switches, one needs duplex data transmission. The system presented here can be upgraded to such a bi-directional one, the scope of which extends to a fully automated feed back control system.

To conclude, we state that an optical fibre-linked control system has been developed for controlling negative ion source (SNICS) parameters in the Ion Beam Laboratory, Institute of Physics, Bhubaneswar. This controller is very simple in design and does not require any additional add-on card. The data communication takes place through the serial port of a PC (COM1 or COM2) using a simple SIPO technique the circuit of which is also developed and presented.

One of the authors (MAS) wishes to thank Dr R K Choudhury, Institute of Physics for encouragement and financial support during his stay at the institute.

## References

- Campbell J 1993 *C Programmer's guide to serial communication* 2nd edn (Indianapolis, IN: Sams)  
Floyd T L 2002 *Digital fundamentals* (New Delhi: UBS)  
Govindarajalu B 1992 *IBM PC and clones* (New Delhi: Tata McGraw-Hill)  
Heiserman D L 1993 *Logic IC master reference* (San Diego, CA: Academic Press)  
Horowitz P, Hill W 1997 *Art of electronics* 2nd edn. (London: Cambridge Univ. Press)  
*Instruction manual* 1985 SNICS-II ION SOURCE, National Electrostatic Corp. Wisconsin, pp 1–22  
Middleton R 1983 *Nucl. Instrum. Methods* 233: 193

## Websites

- [1] <http://www.analog.com/>  
[2] <http://www.group3.com/>