Journal of Engineering Science and Technology Review 15 (6) (2022) 20 - 27

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Innovation

# A Technical Report on an Innovative Plan to Migrate from Traditional PCI Card based Control System to USB Port Interfaced Control System that has been Implemented Successfully in an Active Synchrotron Beam Line

**Avijit Das**

*Saha Institute of Nuclear Physics, Kolkata 700064, India*

Received 5 July 2022; Accepted 21 October 2022

***Abstract***

This technical document has been written based on a modification work done on a supervisory control system, where a novel idea has been implemented successfully to perform equivalent tasks of a PCI Data Acquisition (DAQ) Card with the help of a custom-made Microcontroller unit (McU) based circuit - but certainly, compromising the data transmission speed. An existing relay drive, sensor and gauge monitoring system, which was fully dependent on two traditional PCI cards for interfacing, was totally switched over to a modified supervisory system and plugged to the control PC through a single USB port. That not only saves money and time against procurement of new modern peripheral cards but also minimizes the shutdown period of a functioning experimental setup. The target of this project was twofold - firstly, the urgent up-gradation of the old computer and secondly, the permanent elimination of any future dependability on such an obsolete interface system. Here, a detailed explanation has been given about modification of the existing and running control network, which includes partial alternation of the previous graphical programming code, as well as the inclusion of a new 8-bit processor-based electronic circuit that was able to perform similar activities of the obsolete PCI card based DAQ system. This article has not been written about the innovation of any new theory or machinery but illustrates a non-conventional process to confront a harsh, unforeseen situation encountered in a running project. This concept might have important significance in the field of automation work - where a quick solution for an inoperative or discarded control hardware is urgently required, keeping all other existing, old, but active device operations unaffected. This article would be beneficial to the audience who are looking for an immediate and cost-effective way to replace old-fashioned peripheral hardware where a high throughput rate is not very essential.

*Keywords:* PCI card, Data Acquisition Card, Microcontroller, LabVIEW, USB Hub

## 1. Introduction

This report demonstrates an innovative idea of hardware migration from a back-dated interfacing technique, like PCI (Peripheral Component Interconnect) card-based system (almost discontinued nowadays) to a single USB port-driven control system. The modification work has been executed at the Grazing Incidence X-ray Scattering (GIXS) Beamline (BL-13) of Indus-II Synchrotron Radiation Sources in Raja Ramanna Centre for Advanced Technology (RRCAT) at Indore. It is being maintained by a group of members from Surface Physics and Material Science Division of Saha Institute of Nuclear Physics, Kolkata for the last 15 years [1]. The project has been funded by the Department of Atomic Energy, Government of India.

PCI was mostly a popular connection interface method for attaching computer peripheral cards between 1995 and 2005 and had been superseded by PCI Express (PCIe) in 2004. Modern computers mainly have PCIe and are not compatible to fit older PCI cards [2, 3]. In the year 2020, it was urgently required to upgrade the fifteen years old Windows XP OS-based computer when a software crash ruined the supervisory control network. But, as PCI slots are not commonly available in present-day commercial desktop computers, a quick decision had been taken to interface all running devices through a USB-to-multi-serial port converter and a custom-designed Microcontroller (McU) based circuit. It provided ample freedom to the users to upgrade the old desktop computer with a modern one, which obviously does not equip with any conventional PCI slot on its Motherboard. This multi-serial port USB hub is the central part of a star-like network topology with a low-cost, self-designed, aforementioned circuit board, attached to one of its branch lines, that was able to perform essential tasks of 24 numbers of digital I/O pins of the PCI-DAQ card. This new theme of monitored control scheme has been executed successfully keeping all the previous physical connections and their functionalities intact. But, undoubtedly, in terms of speed and reliability - the performance of such a circuit is incomparable with a commercially available DAQ card.

## 2. Previous Control and Monitoring scheme worked till the year 2019

Oxford-Danfysik, UK (presently known as FMB-Oxford) supplied all essential beamline instruments for beam guiding, tuning, and focusing with their own control electronics and application software. Three monochromatic fluorescent screens (commonly known as Beam Position Monitor or BPM) and a Safety Shutter (SS) placed within the experimental hutch, though supplied by Oxford-Danfysik, were left free to be handled by the users themselves [1]. Along with these instruments, there were

*E-mail address: avijit.das@saha.ac.in
ISSN: 1791-2377 
doi:10.25103/jestr.156.03

five pneumatic vacuum gate valves (manufacturer : VAT VAKUUMVENTILE AG, Switzerland) in the beamline and arrangement was made to operate these valves from a different Graphical User Interface (GUI) screen, that had been developed with LabVIEW 7.1 on Windows XP environment - all software codes and actuator driver hardware electronics were designed and developed in-house during the years 2008-14 [1]. There were three vacuum pumping stations comprised of a Turbo-molecular pump, a rotary vane backing pump, and a cold cathode gauge mounted on the beam pipeline, with its electronic controller – all supplied by Pfeiffer Vacuum GmbH, Germany [1]. The mirror system in the beamline, comprises of two mirrors, Vertically Collimating Mirror (VCM) and Vertically Focussing Mirror (VFM) supplied by Oxford-Danfysik, which has two ion gauges mounted on the chamber wall, were controlled by a single Varian Dual Gauge controller. All these gauge controllers were connected to the control PC via a PCI Multi-Serial Port Card (following RS232 serial communication method) to help the users to keep a constant watch on the vacuum level at different regions inside the long beam pipeline.

A Graphical User Interface application for actuator control and gauge monitoring system was developed around the year 2010 by LabVIEW software package version 7.1 (from National Instruments, USA) on Windows XP operating system [1]. The main interfacing hardware was consists of two PCI cards mounted on traditional PCI slots on a desktop PC motherboard - a National Instruments NI-6251 DAQmx card (High-Speed M-Series Multifunction Data Acquisition Board) and an eight-serial-port Specialix IO8+ PCI card (manufacturer: Perle Systems Inc, USA) as shown in Fig-1 of Ref-1.

## 3. Recent modification to migrate from dual PCI Cards to single USB port control option

At end of the year 2019, due to some software crash, the data acquisition card (NI-DAQ) stopped working. A few essential old versions of drivers were not readily available to run on Windows XP environment. In that alarming situation, a decision was adopted to replace the old computer (that was running for fifteen years on Windows XP OS) with a modern desktop computer. But the most important fact was that - dual PCI slots are not commonly available in present-day commercial desktop computers. It has almost become obsolete except for very few motherboards that still like to maintain backward compatibility. Few Industrial Computers with PCI slots may still be available but were comparatively more costly. Hence, a strategic decision had been taken (shown as a schematic diagram in Fig. 1) to interface all running devices (keeping their existing connections unaltered) through a USB-to-multi-serial port converter and another custom-made 8-bit Microcontroller (McU) based circuit - designed and assembled in-house (circuit shown in Fig. 2).

The McU board was very much compatible with the previous DAQ card activities – because the voltage transition of the port pins in both cases was 0v to 5v logic output. Previously, NI SCB-68 E-series quick connector shielded I/O termination box was being used to screw up wires to link the PCI card with the relay driver Darlington-pair ICs ULN2003 (shown in Fig. 7, Ref 1). The same relay driver circuit has been used without any alteration in this modified interfacing plan (developed in 2021) too.

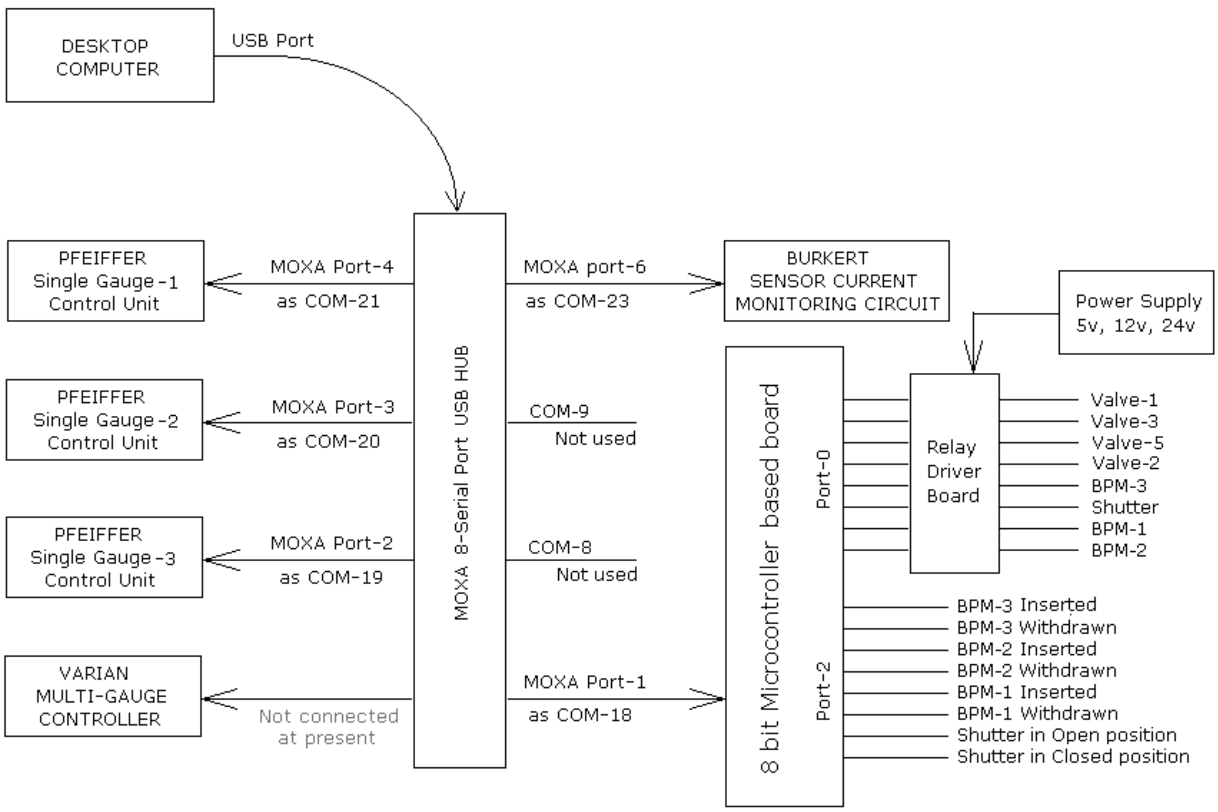


**Fig. 1.** New control and monitoring scheme with Multi-Serial port USB Hub is in use since November 2021.

MOXA 8-Serial port USB Hub, that gives a user the advantage of interfacing eight RS-232 serial port devices connected to a USB port of any computer. It has eight DB9 male type 9-pin ports at its rear side that provides easy plug and play of such an instrument. After the installation of a proper driver, when it was connected to a USB port, COM Ports 1 to 8 appeared in the *Device Manager* as COM-18 to 25.

Though the DAQ card was able to perform multiple functions very fast like, digital/analogue signal input and output, analogue/digital triggering, as counter/timer, waveform generator etc. - so far we had utilized its digital port pins only as input or output. Hence, it was convenient for us to replace the PCI card with a microcontroller-based digital circuit to perform the same tasks.

### 3.1 Microcontroller unit (McU) based electronic circuit introduced in 2021

The 8-bit microcontroller-based circuit (Fig. 2) was designed to replace the NI-6251 DAQmx PCI card while still keeping all the digital port pin's activity same. The microcontroller I used here was ATMEL AT89C52 which has four numbers of eight-bit ports. I used Port-0 for driving eight 12v relays (shown in Table 1) and Port-2 for reading the position status of the safety shutter and all three beam position monitors (shown in Table 2). The digital input/output line numbers were kept almost similar to the previously used DAQ card. As the vacuum chamber gate valves are not being operated still now from the computer GUI, a little change has been made in the design narrated in Table 1.

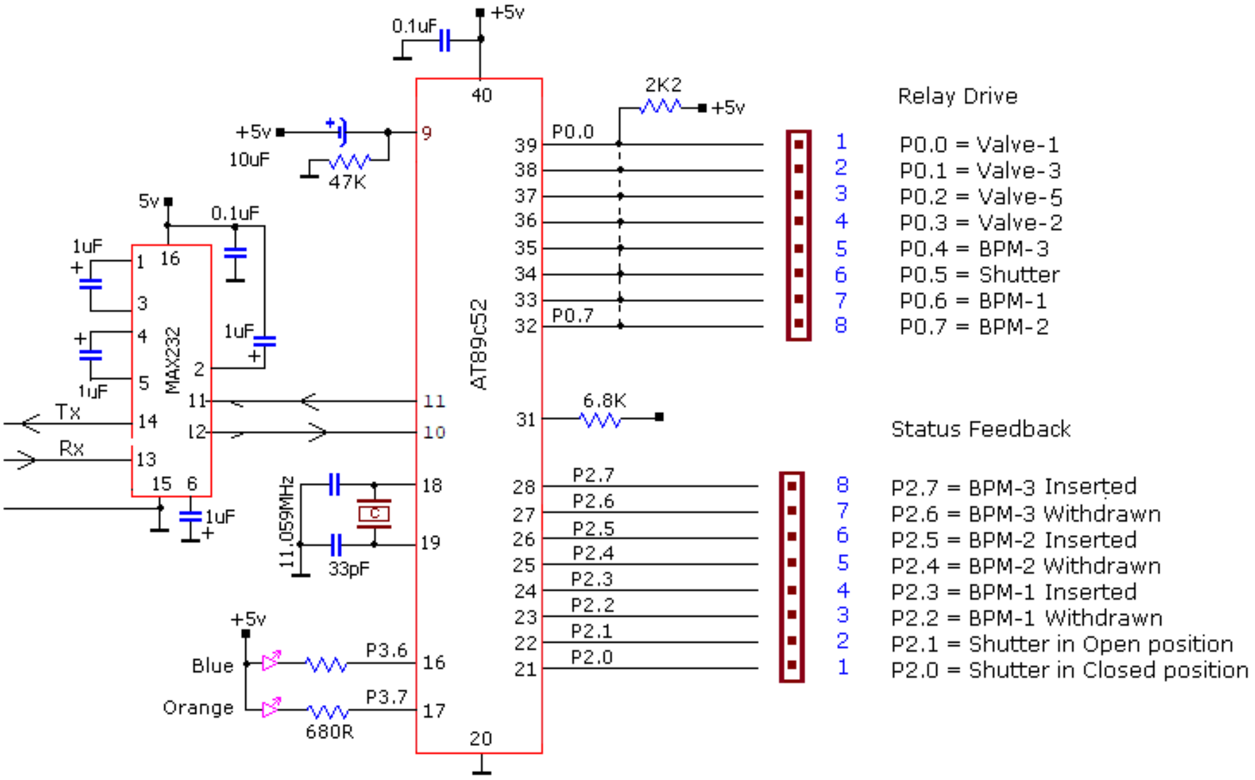


**Fig. 2.** McU-based Circuit to drive 12v relays through Port-0 and read positions of leaner actuators of beamline components.

**Table 1.** Connection details at a glance for relay drive from McU, with little alteration made on Port-0, Pin-4. Instead of GV-4, it operates BPM-3.

| Components to operate | Relay Number | DAQmx card Port pins *(in old scheme)* | AT89C52 McU Port Pins *(in new scheme)* |
|---|---|---|---|
| **Gate Valve 1** | Relay-1 | Port-0, Pin-0 | Port-0, Pin-0 |
| **Gate Valve 3** | Relay-2 | Port-0, Pin-1 | Port-0, Pin-1 |
| **Gate Valve 5** | Relay-3 | Port-0, Pin-2 | Port-0, Pin-2 |
| **Gate Valve 2** | Relay-4 | Port-0, Pin-3 | Port-0, Pin-3 |
| **Gate Valve 4** | Relay-5 | Port-0, Pin-4 | N/C |
| **Safety Shutter** | Relay-6 | Port-0, Pin-5 | Port-0, Pin-5 |
| **BPM-1** | Relay-7 | Port-0, Pin-6 | Port-0, Pin-6 |
| **BPM-2** | Relay-8 | Port-0, Pin-7 | Port-0, Pin-7 |
| **BPM-3** | Relay-9 | Port-1, Pin-0 | Port-0, Pin-4 |
| *No device attached* | | Port-1, Pin No. 1 – 7 | Port-1, Pin No. 0 –7 |

**Table 2.** Status feedback input lines of all three BPMs and Safety Shutter.

| **AT89C52 IC Port pin details** | **Device position Status** | **LED on Controller Front Panel** | ***Boolean* type LED Indicators on GUI screen** | |
|---|---|---|---|---|
| | | | ***True*** | ***False*** |
| Port-2, Pin No. 0 | Shutter Closed | Green | ON | OFF |
| Port-2, Pin No. 1 | Shutter Open | Red | ON | OFF |
| Port-2, Pin No. 2 | Screen-1 out of the beam path | Blue | ON | OFF |
| Port-2, Pin No. 3 | Screen-1 inside the beam path | Red | ON | OFF |
| Port-2, Pin No. 4 | Screen-2 out of the beam path | Blue | ON | OFF |
| Port-2, Pin No. 5 | Screen-2 inside the beam path | Red | ON | OFF |
| Port-2, Pin No. 6 | Screen-3 out of the beam path | Blue | ON | OFF |
| Port-2, Pin No. 7 | Screen-3 inside the beam path | Red | ON | OFF |

The control programme was written in embedded C language on M-IDE Studio for MCS-51 (Release 0.2.5.18) which is a free version to use. It uses SDCC compiler which is a free open-source ANSI-C compiler designed for 8-bit Microprocessors. The Microcontroller used here was AT89C52 (Manufactured by ATMEL Corp.) as the heart of the circuit (circuit shown in Fig. 2). It also had been programmed by the same USB port linked programmer supplied by Robokits India. It is an 8-bit controller with 80C51CPU architecture, on-chip flash memory and four interrupt priority levels. Among the four interrupts, I used only serial interrupt to awaken the processor to receive and analyze the incoming data bytes sent from the PC and react accordingly. Here also I had used a MAX232 IC to convert the voltage level of RS232 signals to 0 to 5v suitable for digital circuits.

Timer-2 had been used to generate a baud rate of 9600 by setting the values of the internal registers RCAP2H as 0xFF and RCAP2L as 0xDC with an 11.059 MHz crystal attached. Timer-2 control register T2CON was set to 0x34 to configure the operation mode of Timer-2 as a baud rate generator. Interrupt Enable Register IE was set to 0x90 where global disable/enable bit and serial port interrupt enable bit was set to 1.

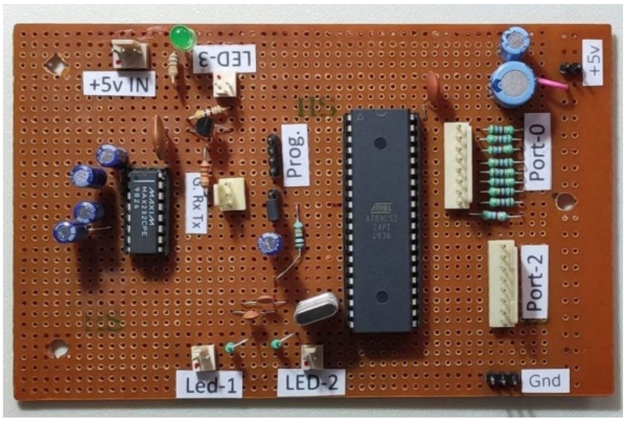


**Fig. 3.** Circuit with McU and other components assembled on a Vero board.

The string of eleven command bytes sent from the LabVIEW GUI was comprised of an address byte (though it was not really required in RS-232 communication, but still set it as 0x08), a Function code byte, and eight Port-0 pin values (either 0x31 or 0x00 to represent 1 or 0) appended with a carriage return (0x0D) at the end. A sub-programme was written in embedded C language, such that it would accept and analyze the incoming data string, only when the processor receives a carriage return (0x0D) at the end. Among the eight data bytes, the first one was used to set the Port-0 pin-7 and the last one was used for the Port-0 pin-0.

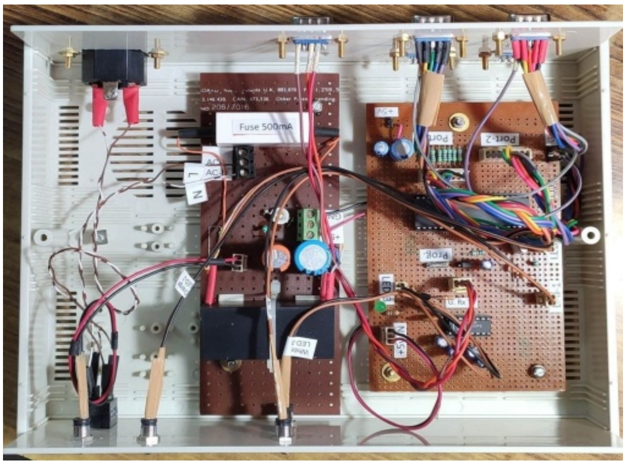

**Fig. 4.** Picture of the complete McU based Controller.

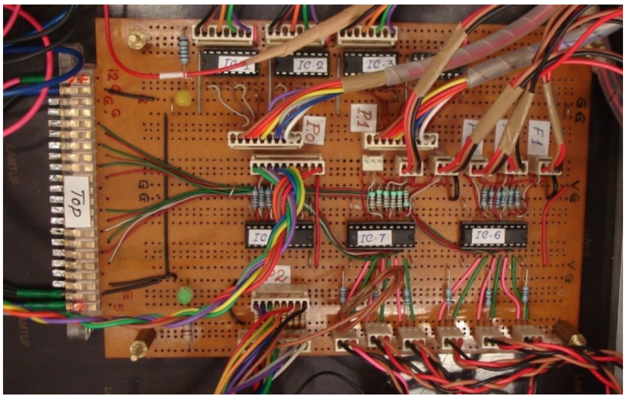

**Fig. 5.** Relay driver ULN-2003 IC board assembled in the year 2009

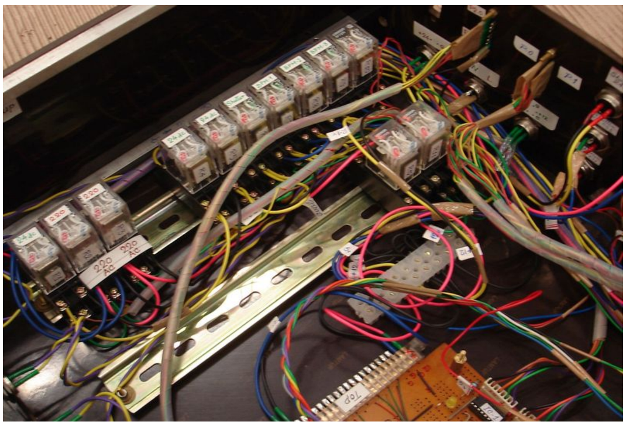

**Fig. 6.** Series of 12v OMRON Relays to supply 24v DC voltage to the pneumatic actuators. (still working since 2010)

The Function Code used was either a character 'D' to denote relay drive action or a character 'F' to read Port-2 value as component position status feedback. The same 12v relay driver board made in the year 2008 with ULN2003 IC chips had been used without making any changes.

### 3.2 LabVIEW programme code to drive Pneumatic Linear Actuators

LabVIEW programme runs in idle mode with a *Flat Sequence Structure* inside a *While Loop*. It is a cyclic operation of the following actions [1].

1. Read the status of positions of Shutter, BPM-1, BPM-2, and BPM-3
2. Read pressure from Pfeiffer Cold Cathode gauges.
3. Read an array of bytes for water flow sensor parameters from the McU-based circuit
4. Reading of vacuum level from Varian Multi-Gauge Controller was withheld as the Controller electronics went bad in the year 2017.

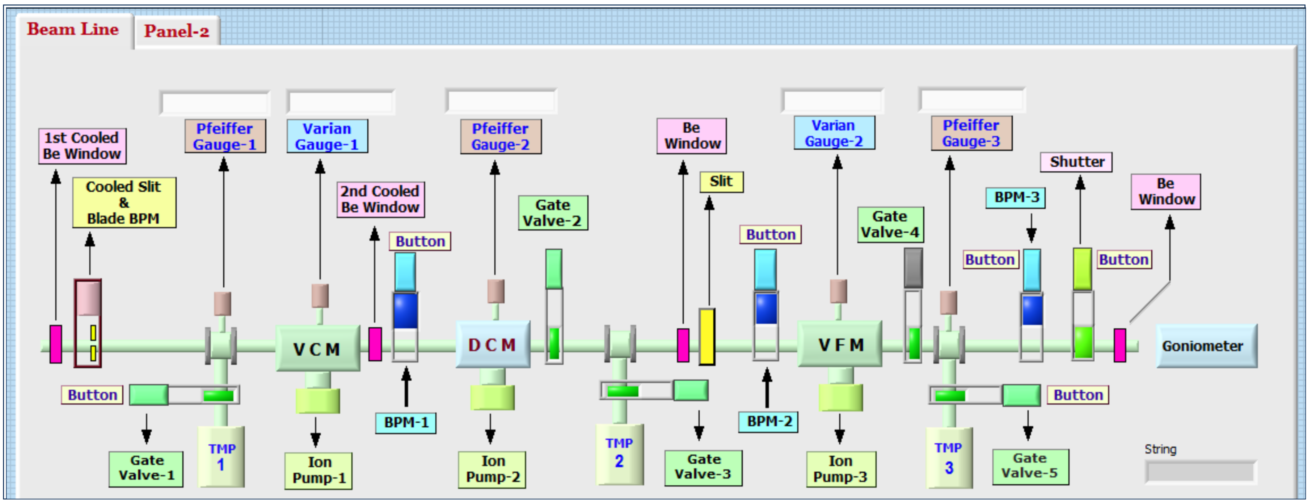


**Fig. 7.** Present view (screenshot) of the main user interface screen (modified in the year 2021), primarily developed in the year 2010 with LabVIEW 7.1. GV-4 operating button is deactivated, that has been indicated in Table 1.

LabVIEW graphical programme code shown in Fig. 8 illustrates how the eight *Front Panel Buttons* construct an array of characters in the *Block Diagram* to be sent to the - board via a new *Sub VI* created, called McURelayDriveBoard.vi. Each button contributes either a character '0' or '1' in a *Case Structure* - depending upon its state. These characters were concatenated one after another to form a character string (Fig. 8). All these buttons were placed inside an *Event Structure* to activate only during toggling-action of any one of the eight buttons. This block diagram was placed in another *While Loop* in parallel to the main *While Loop* and as a result, every button-press action is instant and it sends the updated character string to the McU board immediately.

But it should be noted, that the status change of the shutter and BPMs position reflects on the GUI screen with some delay - as because the software instruction to read the of Port-2 value from the McU board is being called from a *Flat Sequence Structure* loop. Thus, the loop cycle delay comes into account.

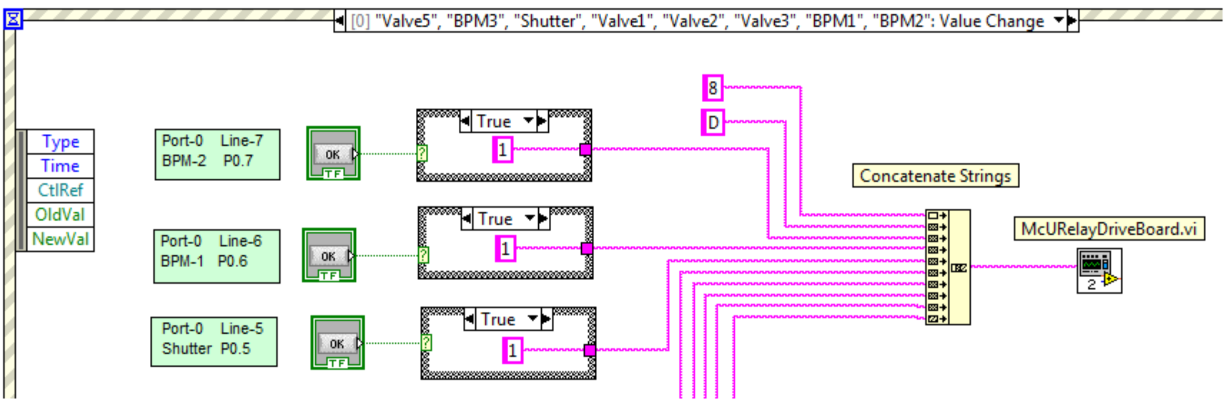


**Fig. 8.** Part of the graphical code to concatenate single characters to form a ten character string to send to the *Sub VI* McURelayDriveBoard.vi

### 3.3 Embedded C programme code to drive an Relay

The address that was set for the McU was decimal 8 (0x38) and the character D (0x44) denotes port pin drive action. A byte 0x0D (Carriage Return) was appended after this array of bytes to indicate the end of data string while interrogating with the McU board.

**Table 3**. Sequence of array of bytes sent from LabVIEW programme as a string of characters, with each mouse click on any of the active buttons on the GUI *Front Panel*.

| | Byte No. 1 | Byte No. 2 | Byte No. 3 | Byte No. 4 | Byte No. 5 | Byte No. 6 | Byte No. 7 | Byte No. 8 | Byte No. 9 | Byte No. 10 | Byte No 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Address | Function code | Eight numbers of Data Bytes | | | | | | | | CR |
| LV o/p Bytes Array | 0x38 | 0x44 | Data Byte 7 | Data Byte 6 | Data Byte 5 | Data Byte 4 | Data Byte 3 | Data Byte 2 | Data Byte 1 | Data Byte 0 | 0x0D |

Inside the Microcontroller AT89C52, initially all the Port-0 pins were forcefully set to zero by bitwise AND operation.
P0 = P0 & 0x00;

Then the programme used to read next bytes one after another and set or reset the Port-0 pins starting from pin-7 up to pin-0 by bitwise OR operation each time on the updated P0 port value.

```
if( in_bytes[1] == 0x44 )  // 'D'
{
   P0 = P0 & 0x00;

   if( in_bytes[2] == 0x31 )
      P0 = P0 | 0b10000000;      // 7th bit
   if( in_bytes[3] == 0x31 )
      P0 = P0 | 0b01000000;      // 6th bit
   if( in_bytes[4] == 0x31 )
      P0 = P0 | 0b00100000;      // 5th bit
   if( in_bytes[5] == 0x31 )
      P0 = P0 | 0b00010000;      // 4th bit
   if( in_bytes[6] == 0x31 )
      P0 = P0 | 0b00001000;      // 3rd bit
   if( in_bytes[7] == 0x31 )
      P0 = P0 | 0b00000100;      // 2nd bit
   if( in_bytes[8] == 0x31 )
      P0 = P0 | 0b00000010;      // 1st bit
   if( in_bytes[9] == 0x31 )
      P0 = P0 | 0b00000001;      // zero position bit
}
```

**Fig. 9.** Part of the C code that segregates and looks for a character '1' (or 0x31 in hexadecimal) from the array of bytes to determine which button on the LabVIEW GUI has been pressed. Accordin gly, it sets only one Port-0 pin (P0_7 to P0_0) on receiving a command string. Table 4 will illustrate the logic clearly

### 3.4 Embedded C programme code to monitor position status of the components

To read the position of Safety Shutter and BPM-1, 2 and 3, LabVIEW programme sends an enquiry command '8F' (denotes feedback enquiry from a device of address 8) to the McU board that reads Port-2 value at that moment and sends corresponding bytes to the PC via MOXA COM Port-1 (designated as COM-18 on Device manager).

Part of the C code is shown here,

```
else if( in_bytes[1] == 0x46 ) // 'F'
{
   port_2 = P2;
   sendChar_SerialPort( 0x38 );    // address = 8
   sendChar_SerialPort( 0x46 );    // 'F'

   if( P2_7 == 1)
      sendChar_SerialPort( 0x31 );    // send '1'
   else
      sendChar_SerialPort( 0x30 );    // send '0'

   if( P2_6 == 1)
      sendChar_SerialPort( 0x31 );    // send '1'
   else
      sendChar_SerialPort( 0x30 );    // send '0'

   if( P2_5 == 1)
      sendChar_SerialPort( 0x31 );    // send '1'
   else
      sendChar_SerialPort( 0x30 );    // send '0'

   if( P2_4 == 1)
      sendChar_SerialPort( 0x31 );    // send '1'
   else
      sendChar_SerialPort( 0x30 );    // send '0'
```

**Fig. 10.** Part of the C code that monitors individual Port-2 pins and construct an array of bytes - send it to the PC via IC MAX232 and COM-1 Port (designated as COM-18 in Windows *Device Manager*).

**Table 4.** Data Byte-7 to Data Byte-0 are designated as *unsigned char array* variables *in_bytes[2]* to *in_bytes[9]* in embedded C code. A Command String sets only one AT89C52 IC Port-0 pin at a time to actuate a single 12v Relay corresponding to the button used for Shutter, BPM-1, BPM-2, BPM-3, GV-1, GV-2, GV-3 or GV-5. (GV-4 was unused in the new scheme that was adopted in 2021).

| **Data Byte 7** | **Data Byte 6** | **Data Byte 5** | **Data Byte 4** | **Data Byte 3** | **Data Byte 2** | **Data Byte 1** | **Data Byte 0** | **Port-0 AT89C52** | **Relay No.** | **Device Activated** |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x80 | 8 | BPM 2 |
| 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x40 | 7 | BPM 1 |
| 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x20 | 6 | Safety Shutter |
| 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x10 | 5 | BPM 3 |
| 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x08 | 4 | Gate Valve 2 |
| 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x04 | 9 | Gate Valve 5 |
| 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x02 | 2 | Gate Valve 3 |
| 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x01 | 1 | Gate Valve 1 |

**Table 5.** This table shows the Array of Bytes generated by the McU and sends to the PC via MOXA COM-1 Port. These Bytes are manipulated in LabVIEW programme as a String of Characters.

| | Address | Function code | Data Byte 7 | Data Byte 6 | Data Byte 5 | Data Byte 4 | Data Byte 3 | Data Byte 2 | Data Byte 1 | Data Byte 0 | CR |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Data Bytes sent from the McU and received by LabVIEW GUI as a String of Characters | | | | | | | | | | |
| **Byte Count** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| **Read Port-2 Status** | | | | | | | | | | | |
| 0x80 | 0x38 | 0x46 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x0D |
| 0x40 | 0x38 | 0x46 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x0D |
| 0x20 | 0x38 | 0x46 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x0D |
| 0x10 | 0x38 | 0x46 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x30 | 0x0D |
| 0x08 | 0x38 | 0x46 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x30 | 0x0D |
| 0x04 | 0x38 | 0x46 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x30 | 0x0D |
| 0x02 | 0x38 | 0x46 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x30 | 0x0D |
| 0x01 | 0x38 | 0x46 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x30 | 0x31 | 0x0D |

### 3.5 LabVIEW programme code to read position status

Microcontroller AT89C52 sends a carriage return (0x0D) byte at end of this byte array to indicate 'end' of the data string. LabVIEW Programme analyzes this input string (part of the C code shown in Fig. 10) and set the Front Panel Boolean LED indicators ON or OFF to show status of the Oxford-Danfysik components graphically as illustrated in Table 2.

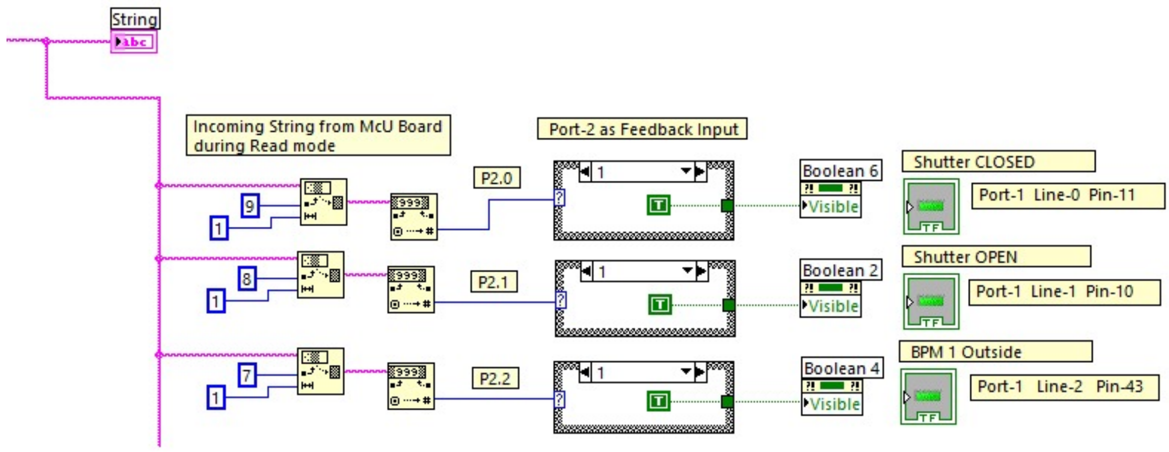


**Fig. 11.** Segregation of individual bits from the input character string with the help of *String Subset* VI and *Decimal String to Number* VI.

## 4. Conclusion

This article has been projected on technical modification work done with an innovative idea to replace a Data Acquisition Card (DAQ) with a very low-cost custom-designed 8-bit Microcontroller-based circuit. It is worth noting that, in reality, this McU-based circuit is not able to handle the same data transfer speed as a PCI DAQ card - as the circuit board is connected to the desktop computer through RS-232 serial data communication method. Hence, its response is comparatively slower. Moreover, NI-6251 DAQmx card has a few analogue input and output channels with 16-bit ADC, which my circuit board does not possess. This feature was not considered while the selection of the Microcontroller was made - as in our project, all operations were based on digital pulses only. People, who are intended to use analogue signals, may think of AVR processors. Alternatively, an Arduino board that uses AVR controllers, might also be used. The operation of the Microcontroller has been explained in detail with the construction of data bytes and screenshots of partial C code. PCI card-based automated test and measurement systems were very popular for scientific and technical applications till the last decade and this project model may become relevant to people who are looking for a quick and low-cost alternative to obsolete peripheral hardware for any of their running automation projects, where the response speed could be compromised. Programmers, who would like to use any other PC programming languages to build a GUI, may use the same logical methodology as I have followed in my LabVIEW graphical programming and embedded C code. During the switch-over process from dual PCI cards to a single USB port-driven system, it was a challenging task to keep the activities related to each active button on the GUI screen unaltered even after the hardware switching. It was also necessary to keep the time consumption at a minimum during this alternation process. This technical note demonstrates a simple and low-cost substitute for PCI Digital Data Acquisition hardware in the field of instrumentation, especially with the new generation of computers, where compatibility issues often create a challenging situation.

**Acknowledgment**

I am grateful to Prof. M. K. Sanyal for involving me with this beamline project. I am thankful to Prof. M. K.

Mukhopadhyay and Prof. S. Bhunia for their suggestions and for providing the opportunity to execute such a risky venture on a running synchrotron beamline. I would like to express my sincere thanks to other group members, like Dr. S. K. Mandal and Dr. R. Urkude for their cooperation during the execution of the plan.



## References

1. Das A., "A Technical Report on Design Process and Implementation of a LabVIEW based GUI with Associated Electronics for Operating and Monitoring Vacuum Components and Sensors of a Synchrotron Beam Line", *International Journal of Scientific and Engineering Research*, 13(6), 2022, pp.344-351.
2. "Peripheral Component Interconnect", Retrieved from: https://en.m.wikipedia.org/wiki/Peripheral_Component_Interconnect, 2022-07-10.
3. Mark Casey, "What Is Peripheral Component Interconnect (PCI)?". Retrieved from https://www.lifewire.com/what-is-pci-2640242, 2021-10-17
4. National Instruments Corporation, USA, "Product Catalogue: High-Speed M Series Multifunction DAQ – 16-Bit, up to 1.25 MS/s, up to 80 Analogue Inputs".
5. Maxim Integrated Products, Inc. USA, "*Technical Datasheet: +5V-Powered, Multichannel RS-232 Drivers/Receivers*", Retrieved from https://www.analog.com/media/en/technical-documentation/data-sheets/MAX220-MAX249.pdf, 2019/2022-10-19.
6. Atmel Corporation, USA, "*AT89C52: 8-bit Microcontroller with 8K Bytes Flash*", Retrieved from https://ww1.microchip.com/downloads/en/DeviceDoc/doc0313.pdf 2021-10-17
7. *"Raja Ramanna Centre for Advanced Technology"* Retrieved from https://en.wikipedia.org/wiki/Raja_Ramanna_Centre_for_Advanced_Technology, 2021-10-17
8. "Indus 2", Retrieved from https://en.wikipedia.org/wiki/Indus_2, 2021-10-17