

The New Tevatron Beam Position Monitor Front-End Software

Luciano Piccoli, Margaret Votava and Dehong Zhang

Abstract—The Tevatron is a proton anti-proton accelerator collider operating at the Fermi National Accelerator Laboratory. The machine is currently delivering beams for the CDF and D0 experiments, which expect increasing luminosity until the conclusion of Run II planned for 2009. The Laboratory defined a plan for achieving higher luminosity, and one of the tasks is the upgrade of the accelerator’s beam position monitor (BPM). The Tevatron was built during the early eighties and some of its control systems, including the BPMs, are still the original ones. This paper describes the front-end software of the Tevatron BPM upgrade, from the requirements to the implementation, and the underlying hardware setup. The design of the front-end software focuses on its modularity and reusability, allowing it to be applied to other Fermilab machines.

Index Terms—Accelerator measurement systems, Beam position monitor, Particle beam measurements

I. INTRODUCTION

THE original Tevatron BPM front-end system [1] has been used since the construction of the accelerator in the early 1980’s. The front-end system relied on Z80 microprocessors with very limited amount of memory (up to 32 kB). The software was written in assembler language and had to carefully manage scarce resources.

Despite the hardware limitations, the original system provided satisfactory beam position data, when improved accuracy and reliability became necessary for current and future Tevatron operations.

Requirements for the system upgrade include an improvement in position measurements by an order of magnitude [2]. The upgraded system is also expected to extend the BPM system capabilities by simultaneously providing position information for both proton (p) and anti-proton (\bar{p}) beams. The original system monitored either proton or anti-proton positions, therefore requiring only the p or \bar{p} signals at a time. Furthermore the system was unable to extract \bar{p} information, which becomes necessary as the \bar{p} intensity is expected to increase in the near future.

Particles circulate in the Tevatron in bunches formed by RF cavities at a frequency of 53.14 MHz, which varies slightly as the Tevatron is ramped in energy from 150 GeV to 980 GeV. The RF frequency in conjunction with the accelerator radius

of 1 km produces a total of 1113 equally spaced bunches. The usage of bunches depend on the machine configuration.

The Tevatron operations require position information for the first bunch of the p and \bar{p} beams. Measurements are based on single or multiple first bunch revolutions, which occurs at a frequency of 47.74 kHz, or an equivalent 20.94 μ s per turn.

The following section describes the Tevatron BPM upgraded hardware components. Section III lists the front-end software requirements, including the modes of operation for the system. Next the design of a generic front-end system is introduced, followed by the description of the software configuration to meet the Tevatron requirements. Finally, the performance of the complete system in the production environment is shown and its use on other Fermilab machines discussed.

II. SYSTEM OVERVIEW

The proton (p) and anti-proton (\bar{p}) beam signals are captured by BPM pickups mounted within the superconducting quadrupoles around the accelerator ring. These devices are part of the original system and remained the same for the upgrade. Each pickup is a pair of 50 Ω striplines 18 cm long, with a circular aperture of 7 cm.

Both striplines (plate A and plate B) have one connector at each end, which are used to harvest the beam signals. The pickups are arranged either horizontally or vertically (as in Fig. 1). The total number of pickups around the accelerator ring is 240, accessible at 27 service buildings.

Four cables bring the pickup signals up to the service building (also referred to as a *house*), into the upgraded front-end VME crate. Each crate contains the following components: one Motorola™ MVME2400 processor; one timing board; and up to six pairs of analog filter boards and digitizer boards. The components are described below.

1) *Processor Board*: The processor board chosen for the BPM upgrade is the Motorola™ MVME2400-0361 with 512 MB of memory. It runs the VxWorks® Real Time operating system (version 5.5) and is the host for the front-end software described in section IV.

2) *Timing Board*: Custom made board that generates timing signals that control and initiate acquisition of the BPM signals. Timing signals are based on inputs from the accelerator controls clock system, which include the Tevatron RF clock (RFCLK), originated from the low level RF used to accelerate the beam; the Tevatron beam sync clock (TVBS), synchronous to the beams; and the Tevatron event clock (TCLK) asynchronous to the beams. The board provides synchronization signals for up to eight digitizer boards. Signals to the digitizer

Manuscript received June 19, 2005; revised September 17, 2005. This work was supported by Fermi National Accelerator Laboratory, operated by Universities Research Association Inc. under contract No. DE-AC02-76CH03000 with the United States Department of Energy.

Authors are with the Fermi National Accelerator Laboratory, Batavia, IL 60510 USA (phone: (630) 840-6593; fax: (630) 840-6345; e-mail: piccoli@fnal.gov).

boards are propagated through individual cable connections between the boards.

3) *Filter Board*: Custom made board that receives the beam signals directly from the BPM pickups. The filter board conditions the pickup signals to be used as inputs of the digitizer boards. Each filter board contains eight channels, which can handle two BPM pickups (there are four channels per BPM). The filter board, combined with the timing board, is also capable of generating diagnostic signals. These signals can be directed to the digitizers or can be sent to the BPM pickups.

4) *Digitizer Board*: Commercial boards from Echotek™ corporation (model ECDR-GC814-FV-A) [3], containing high speed digitizers for the analog inputs (from the filter board). The board also contains a digital down converter, which produces in-phase (I) and in-quadrature (Q) outputs. Each sample is represented by one I and Q value pair. One digitizer has eight input channels, which handle signals from two BPM pickups.

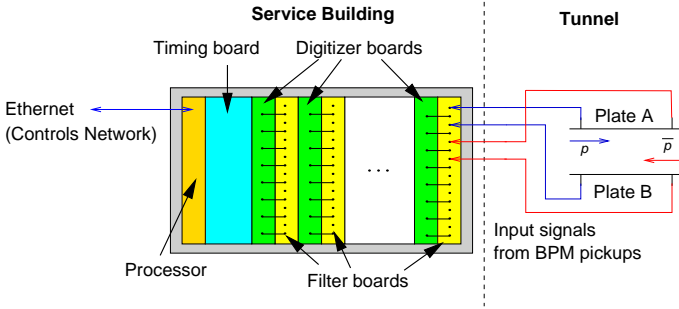


Fig. 1. Distribution of the BPM front-end components within the VME crate. Input signals are connected to the BPM pickups located in the accelerator tunnel (one out of up to twelve pickups is shown).

The modules are organized in the front-end crate as depicted in Fig. 1. The first slot contains the processor board, followed by the timing module. The number of filter and digitizer boards depends on the number of BPMs handled by the front-end. It varies from four to twelve BPMs, which can be handled by up to six filter board and digitizer board pairs. Data are exchanged between the front-end software and the online applications through an Ethernet connection.

The configuration and setup of the hardware described above is the responsibility of the front-end software. The different modes of operation and the type of data to be extracted from the system are described in the next section.

III. FRONT-END REQUIREMENTS

The front-end software has to constantly monitor beam position information acquired through the BPM pickups. The beam structure varies according to the mode of operation of the machine. The system has to configure itself based on mode changes and user requests.

The modes of operation defined for the front-end software are: closed orbit, turn-by-turn, injection, safe injection and diagnostics. For all modes, the front-end has to compute the beam position and intensity based on the raw I and Q values provided by the digitizer boards.

Based on the raw values and beam condition, the beam intensity is calculated based on the sum between signals from plates A and B , defined as $i = |A| + |B|$. The A and B signals are derived from the raw values ($|A| = \sqrt{I_A^2 + Q_A^2}$) for both plates and particle types.

The position is given by $p = s(|A| - |B|) / (|A| + |B|)$, where s is the scale factor. In addition, the front-end software should allow calibration factors, such as electrical and mechanical offsets, to be applied to the intensity and position calculations.

Due to the current 6:1 proton-to-anti-proton bunch intensity ratio and the inherent directivity of the pickup striplines, the \bar{p} intensity and position calculation requires corrections derived from the p measurements [4]. The front-end must apply the proper corrections for every \bar{p} measurements.

A. Closed Orbit

This is the default mode of operation for the front-end system. The closed orbit position for both p and \bar{p} beams is computed at a rate of 500 Hz. The measurement is taken by the digitizer boards running in a narrow band configuration during approximately 1.38 ms.

The system must keep several buffers associated with closed orbit measurements. They are: closed orbit, fast abort, slow abort, profile and display. Independent of the machine state, the closed orbit buffer is constantly updated. Those values must always be available.

The fast and slow abort buffers are frozen upon beam loss. Those buffers contain information that allow users to understand the beam motion previous to a beam loss event. The fast abort contains copies of closed orbit buffer elements and the slow abort is updated at 1Hz with an average of the last 100 elements from the fast abort buffer.

The last two closed orbit buffers, the profile and display, are filled with the latest closed orbit measurement at the reception of certain events. The events are broadcast through predefined TCLK signals, which are fired at certain beam conditions. For example, the profile measurements are taken during beam injection at distinct energy levels.

B. Turn-by-Turn

The turn-by-turn mode is enabled via user request and, differently from closed orbit, the position and intensity of the first bunch for every revolution must be measured. The system is configured to record information for 8192 consecutive turns. Additionally, the measurement must be tightly synchronized by the timing system across all houses.

The data collection for every turn is triggered by timing board upon the reception of the TVBS turn marker. A hardware signal is sent from the timing board to the digitizer boards, which take 32 samples at a rate of 74 MHz. The board is configured to select one of samples to represent the turn.

The data read out from the digitizer must be kept in a separate buffer, which can be read by the user at any time. After the conclusion of the turn-by-turn measurement the system must immediately return to the default closed orbit mode.

C. Injection

The injection is the turn-by-turn mode triggered by the proton injections in the machine. There can be several injections while the machine is tuned up. As in the previous mode, the system is required to gather information for the first 8192 turns.

Data from the injection mode must be kept in the injection buffer and must be available throughout the store, until the next injection. The system also must perform an averaging of the first turns, providing a closed orbit measurement for the injection. This calculation must be carried out by the processor instead of the digitizers because the hardware cannot be reconfigured quickly enough to provide injection closed orbit information. After the injection cycle is complete the system switches to the closed orbit mode.

D. Safe Injection

The safe injection mode is used when the front-end system cannot get reliable turn timing. This mode of operation is not expected to be used during normal conditions, but should be available as an option in case of timing failures.

At the start of the injection cycle the system is set up to sample the beam continuously at the digitizing rate of 74 MHz. Sampling is started by the reception of the TVBS injection event. The default number of samples for this mode is set to 8192, which in conjunction with digitizer decimation rates generates data for approximately 75 turns.

The data read out from the digitizers must be processed by the front-end, which searches for the intensity peaks. The samples with highest peaks are the candidates for the actual beam position. From the selected samples the system computes the initial closed orbit and returns to the default mode of operation.

E. Diagnostics

In diagnostics mode the system must provide several configuration options to the user. It must provide in depth raw data visualization and the capability of easily changing the configuration. This mode of operation is required for verifying cable connections, signal quality, find problematic channels and allow other procedures to insure the correct behavior of the system. The diagnostic signal generated by the timing and filter boards can be enabled on this mode.

IV. SOFTWARE DESIGN

The upgraded front-end software was designed to meet the Tevatron requirements keeping in mind its potential use in other BPM systems at Fermilab. As result, the functionality and data structures that can be shared between machines were isolated in a generic library. Object oriented design techniques using Unified Modeling Language (UML™) notation were used throughout the design phase.

The general principle of operation for the BPM front-end system follows a simple data acquisition (DAQ) sequence: arm, trigger and data read out. Every step is started by the

occurrence of an event in the system, for example the injection mode is armed when the machine starts an injection sequence.

The front-end software design is derived from that simple DAQ sequence. It can be defined as an event driven environment, where every operation must be preceded by the occurrence of some key event. Additional functionality is added to the model in order to define a complete generic environment. An overview of the general front-end software is shown on Fig. 2. Notice that events are received by tasks through individual event queues.

The active components of the system are represented by three types of tasks: control task, alarm task and data acquisition task. The control task provides general coordination and is responsible for starting up the system by creating data buffers and initializing the alarm and data acquisition tasks.

Any entity in the system is capable of generating alarm events, which are internally routed to the alarm task. The alarm task has internal thresholds and can be configured to dispatch alarms to external systems via the controls network.

The actual data acquisition job is carried out by the DAQ tasks. The basic DAQ task remains waiting for an event in its own input queue. Upon the reception of an event, the task performs the data read out. The DAQ task reads data from a *data source*, processes it if necessary, and stores it in a *data buffer*.

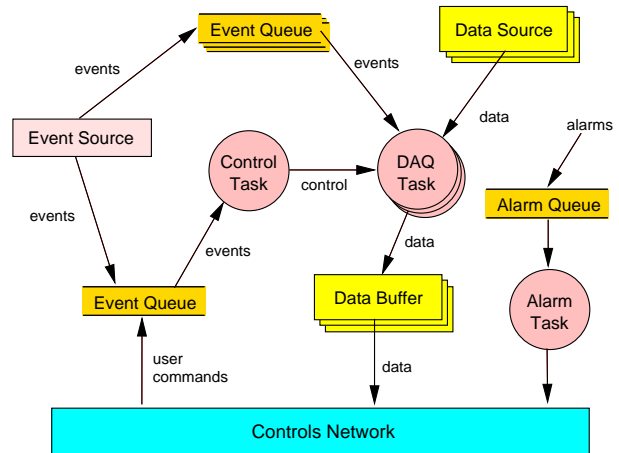


Fig. 2. Generic design for the BPM data acquisition environment.

A data source is an entity capable of generating data, and it implements methods to get single or multiple data elements. The data buffer is the source counterpart, allowing single or multiple data to be saved. Additionally, a data buffer can also perform as source, allowing information to be moved between buffers.

There are two types of data buffers: circular and FIFO. The circular buffer continuously accept new data and when it is full the oldest element is replaced. The FIFO buffer accepts data until it becomes full, and new data are rejected until the buffer is explicitly cleared. The size of the buffers is defined when the system is started up. There are no buffer size limitations other than the physical memory size.

All tasks in the system have an associated event queue. Events are wrappers for any information that needs to be

exchanged between tasks or other components of the system. They are implemented as a simple data structure containing an event type. The event types defined for the generic system include:

a) *Interrupt event*: generated upon the occurrence of an interrupt. The event generated contains information used by tasks to identify the source of the interrupt (e.g. interrupt vector).

b) *Mode change event*: generated when the system is requested to switch to a new mode of operation.

c) *Alarm event*: created internally to signal an abnormal condition, e.g. TVBS signal is not present.

d) *Time event*: used to trigger periodic tasks, such as the verification of TVBS signal presence.

e) *State change event*: produced when a *state device* has its state changed through an Ethernet broadcast. State devices are special ACNET devices (see sub-section IV-B) used for distributing changes in the operational state of certain devices, such as the Tevatron machine state.

f) *TCLK event*: generated upon the decoding of a predefined TCLK event.

These basic building blocks (tasks, queues and buffers, as shown in Fig. 2) can be arranged in any sequence to compose the Tevatron BPM front-end system. However, the front-end is not complete without adding components that are specific to the Tevatron system, such as the extension of the data acquisition tasks to perform data read out according to the upgrade requirements.

A. Tailoring System for the Tevatron

This sub-section describes how the basic building blocks defined in the generic library are used to configure the Tevatron front-end system according to the requirements listed in section III.

The modes of operation for the front-end system are reduced to two modes: closed orbit and turn-by-turn. The remaining modes are either extensions of those or require a slightly different configuration.

The closed orbit is the default mode of operation. It involves several data acquisition tasks, which are shown in Fig. 3. In closed orbit mode, only the closed orbit task has access to the digitizer boards. The timing, filter and digitizer boards are configured by the control task when entering this mode of operation. The data acquisition cycle is started by the timing board, which transmits a signal to the digitizers and generates an interrupt to the processor every two milliseconds. This periodicity is dictated by the upgrade requirements, nonetheless the system is capable of running slightly above its specifications.

Closed orbit measurements, which use the narrow band configuration, are started in the digitizer boards after receiving a trigger signal from the timing board. Meanwhile, the timing board interrupts the processor. As a result, a read out measurement event is generated and sent to the closed orbit task. The reception of this event starts the read out sequence by the closed orbit task, which consists of waiting for the completion of the measurement by the digitizer boards. The completion of

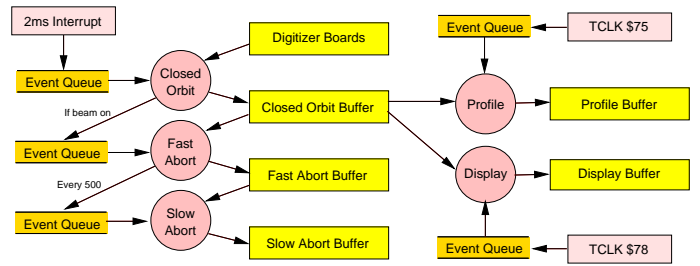


Fig. 3. Tevatron closed orbit data acquisition tasks

the measurement is signaled by an interrupt from the digitizers. Data are read, processed and stored in the closed orbit buffer and a new closed orbit measurement is started. The system keeps generating closed orbit measurements even if there is no beam in the machine. Position and intensity information are always available while the system is in this mode.

While there is beam in the machine, according to specific state devices, the closed orbit task generates a read out event to the fast abort task, which transfers the most recent value from the closed orbit buffer to the fast abort buffer. Similarly, the fast abort task generates a read out event to the slow abort task on every 500th event. The slow abort task processes the event by averaging the latest hundred elements from the fast abort buffer and saving the result as a new entry in the slow abort buffer. All three buffers, closed orbit, fast abort and slow abort are circular buffers with space for 8192 data points. The fast abort buffer contains a history of the past two seconds while the slow abort covers a longer period of time.

In addition to the abort tasks, the closed orbit mode also defines the profile and display data acquisition tasks. Both receive input events from the Tevatron clock (TCLK). The TCLK signal is decoded by the timing board, which interrupts the processor. If a profile (clock \$75) or display (clock \$78) TCLK is received the corresponding event is produced and sent to the respective task. When processing the event, the tasks get the latest value from the closed orbit buffer and save it in the profile or display buffers.

Differently from the closed orbit mode, the turn-by-turn mode produces position and intensity information for single turns. The configuration procedure is similar to the previous mode; all timing, filter and digitizer boards are configured when changing the mode.

In turn-by-turn, the timing board sends multiple triggers to the digitizers and a start interrupt to the processor. Every trigger corresponds to a turn, and the digitizer produces one sample per turn. The start interrupt causes the turn-by-turn task to wait for the digitization of the 8192 turns. Upon completion, the data from the digitizers are read by the processor using DMA operations. The data are saved in the turn-by-turn buffer, which can later be accessed by online applications.

Regular turn-by-turn measurements can be taken at any time while there is circulating beam in the machine. However, there are special cases for turn-by-turn measurements, as shown in Fig. 4. These are the injection mode and the safe mode.

Both injection and safe modes are used when beam is first injected in the machine. In injection mode, the 8192 turns

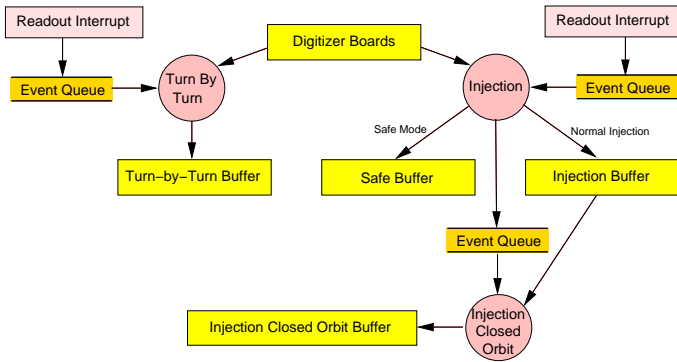


Fig. 4. Tevatron turn-by-turn data acquisition tasks

are taken after the timing system receives a TVBS clock $\$7C$ indicating that the injection kickers have been triggered and beam is about to be injected. The acquisition procedure is the same as in the regular turn-by-turn, except that data are saved in a special injection buffer and the injection task sends an event to the injection closed orbit task. The latter computes the average position and intensity from the injection, producing an initial closed orbit measurement, also available in a separate buffer.

The safe mode can be used as alternative to the injection turn-by-turn when the TVBS turn marker is not present or is not reliable. In this mode, the timing board is configured to send a single trigger to the digitizer, which is configured to collect 8192 consecutive samples (at the rate of 74 MHz and using filter decimation parameters that result on a coverage of 75 turns). The resulting data are stored in the safe buffer and later processed by a simple beam search algorithm used to find the intensity peaks. The selected samples are saved in the injection buffer and used to compute the injection closed orbit.

In addition to the modes described above, the system can be configured to produce diagnostics data. When configured to run in diagnostic mode, the front-end saves all samples taken by the digitizer boards in a special diagnostics buffer. The diagnostic data are read by a diagnostics user application that is capable of displaying several attributes of the data: magnitude, phase, position and intensity. Additionally through the user application it is possible to change triggers, acquisition controls and delays.

B. Controls Network

The 27 BPM front-end crates are individual entities and do not communicate with each other. The read out and analysis of the data acquired by the front-ends is handled by higher level software (referred to as online), which is able to address each crate individually.

The communication between online applications, such as the diagnostics application, and the front-end uses the Accelerator Controls Network — ACNET [5]. It defines a communication protocol over Ethernet, allowing control and monitoring data to be exchanged between front-end computers and online applications. Access to the data is possible through ACNET device readings and settings.

Every front-end defines up to 130 ACNET devices. System properties, such as channel delays, calibration and correction constants can be changed by setting devices. Status and beam position and intensity are read out using ACNET reading devices. The front-ends define a range of beam position readable devices, providing either the raw data directly from the digitizer or calibrated and corrected measurements.

There are no restrictions regarding access to the front-end ACNET devices. However, the constant read of buffers can impact the performance of the system. Buffers are critical regions and are protected by locks, shared by the ACNET devices and the data acquisition tasks. Therefore online applications should be careful when requesting large amounts of data from the front-ends.

Latest closed orbit data is used by online applications for machine monitoring. Data requests are limited to 15 Hz by the ACNET protocols. Data request for monitoring usually are positions or intensities. Access to fast and slow abort data is requested when a store is lost for diagnostics purposes. In such event the system stops feeding the abort buffers and online applications are able to request all buffered data without interfering the data acquisition.

On the other hand, requests for turn-by-turn data are submitted only a few times during a store, including requests for the injection turn-by-turn data. Processing of these have little impact on the system operation due to the very low probability of simultaneous data access from the online application requests and data acquisition tasks (a turn-by-turn measurement request is not likely to happen while turn-by-turn data from previous measurement is being read out). However these requests may produce large amounts of data, which may impact the network usage.

V. IMPLEMENTATION

The front-end system is implemented in the C++ language. The generic environment is contained in the so-called GBPM library whereas the Tevatron implementation is defined in the so-called TBPM library. Additional software is available in separate libraries, such as the digitizer driver code.

The front-end system provides several command line tools that allow experts to verify the current software status. It is possible to monitor the amount of data collected, the current configuration, and the operation history.

The system uses the TRACE tool [6] for helping debugging and profiling the system. The tool was developed at Fermilab and is available for the Linux and VxWorks[®] operating systems. The front-end libraries include TRACE directives throughout the code, which can be configured at runtime to produce a stack trace for given code sections.

VI. SYSTEM PERFORMANCE

Profiling measurements using TRACE show that the front-end system can sustain periodic closed orbit measurement at 500 Hz for a fully loaded crate (six filter-digitizer pairs). The closed orbit measurement cycle takes on average 1.8 ms to complete, leaving 10% idle time for additional processing used by the slow abort, profile and display data acquisition.

The closed orbit cycle is dominated by the averaging on the digitizing boards, which takes 1.38 ms. During the averaging the processor is available for lower priority processing, such as user data requests. After a measurement is completed, the hardware takes 235 μ s to be prepared for the next cycle. From the remaining 185 μ s, 110 μ s are required for transferring the data through VME; 50 μ s are used for calculating the intensity and position from the raw values and applying calibration and correction factors; and it takes 25 μ s for the additional tasks of saving the data and reorganizing internal buffers.

The closed orbit processing cost is minimal when compared with the cost of certain mode transitions. For example between the transition from turn-by-turn to closed orbit mode requires the digitizer boards to be reconfigured. The total setup time for a full crate is approximately 61 ms.

The total time to complete a turn-by-turn cycle and return to the default closed orbit mode takes 325 ms, which includes the 61 ms for loading the turn-by-turn digitizers setup at mode change time. The setup reloading time defines the minimum delay required by the front-end to arm for a turn-by-turn measurement. For the critical injection mode, the system receives a prepare for beam TCLK 2.7 seconds in advance, showing that the reload time is not critical for the operation.

After the turn-by-turn measurement is triggered it takes 170 ms for digitizers to sample information for the following 8192 turns. On completion, the data are read out by the processor through the VME backplane using DMA. The operation takes approximately 5.3 ms per board, resulting in a total time of 32 ms for the full system.

Immediately after the data are transferred the system switches back to closed orbit mode, which requires the remaining 61 ms for restoring the digitizer closed orbit configuration. The time required for configuring the timing board is negligible.

The intensity and position calculation along with the buffer management time for a turn-by-turn measurement is taken from the processor's idle closed orbit cycles, i.e. while the digitizers average beam data and during the 10% spare cycles from every measurement. The low priority turn-by-turn data processing is feasible because user applications do not need the turn-by-turn data immediately after the measurement is taken.

The front-end software takes advantage of the real time aspects of the operating system by controlling the priorities of the data acquisition tasks according to the mode of operation. Turn-by-turn data acquisition tasks have the priority increased when the system receives a turn-by-turn request and lowered as soon as the system gets back to the default closed orbit mode.

The Tevatron front-end software defines thirteen processing tasks, of which nine are dedicated to data acquisition, one is the control task, one is the alarm task, and the additional two are used for logging purposes. The total memory used by the system does not exceed 25% of the 512 MB available in the processor board, most of it is reserved by the driver for reading out the digitizer boards.

The upgraded BPM system was commissioned between February and May of 2005. The hardware was installed during

short Tevatron shutdowns and between stores. During the commissioning a couple of software and hardware major bugs were discovered and fixed without delaying the final delivery. The transition from the old system was smooth without impacting operations. The upgraded BPM front-end is producing reliable data for Tevatron operations since May 23, 2005.

VII. CONCLUSION

The paper has described the Tevatron upgraded front-end software. The proposed design and implementation met the upgrade requirements, delivering a flexible and reliable software. New modes of operation, data acquisition tasks and data buffers can easily be added to the system.

The installation of the upgraded electronics has been completed in May of 2005. The BPM system is reliably measuring proton position data and used in routine Tevatron operations. The \bar{p} position measurement implementation should be completed soon. In addition to the routine operations, the upgraded BPM system has collected data for accelerator physics analysis of the Tevatron.

In the near future, the Tevatron front-end software is expected to be used for the Main Injector BPM upgrade. It is possible to reuse the same structure defined in the GBPM package and have the system configured to meet different requirements. The Main Injector is structured differently than the Tevatron and requires additional modes of operation, which implies more data acquisition tasks and data buffers. Additionally, the generic environment can also be used for projects beyond the BPM scope. It is suitable and can be tailored to other types of data acquisition systems, such as slow controls.

REFERENCES

- [1] R. E. Shafer, R. E. Gerig, A. E. Baumbaugh, and C. R. Wegner, "The Tevatron beam position and beam loss monitoring systems," in *XII International Conference on High Energy Accelerators*, Batavia, IL, August 1983.
- [2] S. Wolbers, B. Banerjee, B. Barker, S. Bledsoe, T. Boes, M. Bowden, G. Cancelo, B. Forster, G. Duerling, B. Haynes, B. Hendricks, T. Kasza, R. Kutschke, R. Mahlum, M. Martens, M. Mengel, M. Olson, V. Pavlicek, T. Pham, L. Piccoli, J. Steimel, K. Treptow, M. Votava, B. West, R. Webber, and D. Zhang, "Tevatron beam position monitor upgrade," in *Particle Accelerator Conference (PAC05)*, Knoxville, TN, May 2005.
- [3] Echotek Corporation, "Echotek ECDR-814x user manual," [Online]. Available: http://www.echotek.com/a2d/ecdr-814_AD/ECDR-814Rev3p0062001.pdf.
- [4] R. K. Kutschke, J. Steimel, R. Webber, and S. Wolbers, "Simultaneous position measurements of protons and anti-protons in the Tevatron," in *Particle Accelerator Conference (PAC05)*, Knoxville, TN, May 2005.
- [5] J. Patrick, "Evolution of the Fermilab control system," in *IX International Conference on Accelerator and Large Experimental Physics Control System*, Gyeongju, Korea, October 2003.
- [6] R. Rechenmacher and D. Holmgren, "Fermi linux TRACE," [Online]. Available: <http://fermitools.fnal.gov/abstracts/trace/abstract.html>.