# THE PROGRAM LINE SEQUENCER. A NEW APPROACH.

Julian Lewis and Paul Heymans
CERN, 1211 Geneva 23, Switzerland.


## Abstract

This paper describes how the CERN PS and PSB are sequenced for Pulse to Pulse Modulation (PPM), and then isolates the concepts involved. Based on these concepts a new general purpose sequencer, the Telegram-slave-unit, is described. It is then shown how this device has been used to sequence the LEP-preinjector.


## Introduction [1]

The machines of the CERN-wide accelerator complex work in cycles, each cycle being a compromise between the beams required and the constraints of the machine. The cycles of various machines therefore evolve independently except for a small fraction of the total time, when they interact during short intervals around the moments of beam transfer. For the latter, the magnetic fields, radio frequencies and beam characteristics must meet stringent conditions on both sides and this subtle rendez-vous requires the exchange of a number of signals.

Between transfers, the cycle of each accelerator evolves without input from the other ones, relying on its internal timing system which is peculiar to the specific needs of that machine and to the technology and thinking of the era in which it originated.

Several machines of the CERN-wide accelerator complex run interleaved cycles with different kind, quality, source and destination of the beams. They do so in periodic sequences called super-cycles. For this, numerous parameters must be refreshed from cycle to cycle, called Pulse to Pulse Modulation (PPM). Super-cycles must be orchestrated in such a way that the appropriate cycles meet at the correct instant for rendezvous. At the PS complex, the orchestra conductor comprises two sequencers: the Linac-Beam-Sequencer (LBS) which gives the beat and the Program-Line-Sequencer (PLS) which indicates to each machine what cycle is to be played next. Facilitating this, all cycle lengths are made multiples of one basic period of time.

In addition to its functions of cycle to cycle coordination and synchronisation, the PLS is provided with powerful and user-friendly editors and archive manipulation. These facilities have proved crucial for rapid changes in the programme of the PS complex. The electron/positron operations introduce additional programmes hence changes, which must be handled efficiently on the CERN-wide scale.

## The PS and PSB Program-line-sequencer

At the PS complex, orchestration between the cooperating accelerators is done by a subsystem of the PS control system, the so called Program-Line-Sequencer (PLS). The latter coordinates the accelerators of the PS complex by broadcasting messages (so called PLS telegrams) indicating the type of the present cycle and the next. The messages only contain WHAT should happen (eg cycle type); all detailed information on HOW it should happen is contained in tables in the process interface of each accelerator. At each cycle, the PLS telegram indicates which of those data are to be presented to the process equipment. The tables contain parameters such as magnet currents but also settings of preset counters governing the detailed timing of the cycle. Parameters in the tables can be adjusted from the consoles or loaded from archives. All cycles are usually multiples of one basic-period and synchronisation is insured by the SSC (start super-cycle event), a so called WHEN data from which all the cycles are triggered in sequence. (Fig 1).

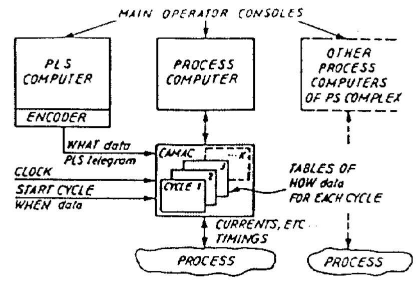


Fig 1. The old PLS system.

The PLS computer is a Nord-10 mini computer and a set of specific hardware connected together using serial Camac. The software consists of a set of interface routines to standardise access over the control system network, and local real time application programs which perform repetitive processes, manage the data tables, and perform input/output for the hardware. In addition a video display driver task maintains a detailed display of the control information being sent to the rest of the complex.

Information entered by the operators at the consoles is placed in data tables which drive the real time task constructing the telegrams. These telegrams are then distributed at fixed times during the machine cycles by so called PLS encoders, which are dedicated modules in a Camac crate driven by the PLS computer. The telegrams contain all the information required to describe the beam characteristics in a machine, for the present and next machine cycles. The telegram is a 10KHz bit serial message, each of its 256 bits having a specific function and being called a "programme-line", for historical reasons.

The meaning of the lines of a telegram depends on the machine whose state it describes, but for convenience, these meanings correspond when possible. So called "Groups" are defined as a contiguous set of bits in the telegram designated to describe one machine parameter such as beam destination or, beam intensity. Setting lines (bits) in a group corresponds to selecting a destination or intensity etc.

Two types of hardware modules are used to interpret the telegram at the receiving end, PLS-Receivers and PLS-Decoders. A PLS-Receiver stores the entire telegram for later access by application programs. A PLS-Decoder reacts to one preset line number, and its binary output state depends on whether or not the selected telegram bit was present. This output can be used to gate various hardware directly.

## Deriving timing pulses from the telegrams

The PS and PSB telegrams are sent out each machine cycle and contain all required information needed for the rest of the control system to coordinate and sequence the activities of the machines during each cycle of their super-cycles. Given that the operator preset timing values are already locally available in the memories of the process interface, all timings could then be derived from the telegrams, using the clock trains, preset counters, and PLS-Decoders.

The General-purpose-preset-counter (GPPC) contains all the necessary components. The pulse train can be selected under software control, and an option exists which allows the output pulse to be produced only when the inbuilt PLS-Decoder detects or not a specified PLS-line.

A further example of deriving timing and sequencing pulses from the telegrams and clock trains is the new Linac-Beam-Sequencer (LBS). The LBS can be viewed as a multi channel, general purpose, telegram controlled pulse generator, ie a black box into which the clock trains and telegrams enter, and from which timing pulses exit. It is completely table driven, constructed from standard hardware modules and uses a single software module. Timing pulses output by the LBS can be reprogrammed using interactive programs at the operator consoles. This device occupies a Camac crate containing a micro processor and a set of GPPCs. It outputs key pulses, related in time to the instant of beam transfers. In the absence of clock trains, the LBS can simulate its inputs, eg during machine development periods.

## Pulse to pulse modulation PPM [2]

The aquisition and control variables, for say a power-supply, may change from cycle to cycle. Microprocessor based Auxilary Crate Controllers (ACCs) in Camac crates contain data-tables relevant to each cycle of the super-cycle. Each table contains the control values of the operations programmed in that cycle as well as locations for the acquired values of the same. The control values of each table are transferred to the working registers in the interfacing Camac modules at specific instances, upon receipt of interrupts by key timing pulses in the cycles. The table is for each cycle chosen according to the telegram received. The latter is decoded in the PLS-Receiver which is interrogated by the ACC before the transfer. Acquisitions are made and placed in the table, following other interrupts, relevant to the process in question. Control values, labelled with one of the PLS conditions, are entered asynchronously through standard software interfaces called Equipment-Modules (EMs) in the relevant front-end-computer. Acquired values are likewise requested for display on consoles through the EMs.

Thus an applications program can access the values during particular cycles without affecting the values of the other cycles. The operators are thus able to work with the same machine but on different cycles, independently of each other, from different consoles.

## Problems with the existing Program-line-sequencer

After some eight years of successful use of the PLS, a number of limitations have come to light because of the need to adapt the control system to the ever expanding requirements of the PS, and because of obsolete hardware and software.

Several of the limitations may be lifted by changing the format of the telegrams, however each applications program that uses a telegram has built into it the knowledge of what the individual telegram lines represent. Hence it is impossible to change the format of the telegram without changing every applications program that uses it. This means that the telegrams cannot adapt to the changing requirements of the machines.

## Master slave relationships between accelerators

The PS can be considered as a junction which switches beams between the Proton and Ion Linacs, the PSB, LIL and EPA, SPS, LEAR, AA, and other machines and experiments. Within a super-cycle there could in principle be electrons, positrons, protons, anti-protons, Oxygen-ions etc, present during the individual cycles. If for example the SPS required positrons to be injected during a cycle ,then this implies what the PS, PSB, LIL and EPA must do, so that the beam will be injected correctly. In fact these slave machines need to be informed many cycles in advance in order for the beam transfer to take place. There is thus a master slave relationship between the SPS and PS ,and then one between the PS and PSB ,and so to the first machine to produce the beam. One may say that the LIL and EPA are slaves of the PS, and also that if there were an LPI-telegram, then it would be a slave of the PS-telegram.

We could imagine another black box, the Telegram-slave-unit (TSU), which has a master telegram as input and a slave telegram as output. If such a device could be made general purpose, then any number of machines could be coordinated by connecting TSUs together in the form of a tree representing the master slave relationship between the machines involved. We might even imagine a CERN wide sequencing tree such that all the individual super-cycles are created automatically from a single CERN telegram. Alternatively any TSU could be placed under local control and in that case all branches below that point in the tree would break off from the main CERN tree to form a little tree of their own. The first TSU is now producing the LPI-telegram from the PS-telegram and is a first step towards the CERN wide sequencer. (Fig 2.).

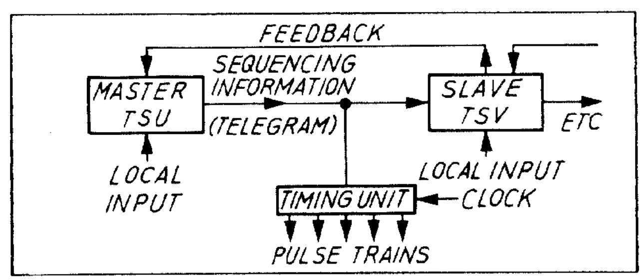


Fig 2. TSU Sequencer tree.

## Some concepts behind telegram decoding

At present the PS-telegrams consist of a 256 line (bit) message. Sets of these lines are combined together into groups which may describe some particular machine parameter such as intensity or beam destination. Thus the receiving end has additional information which is required in order to know what these lines mean, (eg how they are grouped). If the number of possible different telegrams had to increase, a free format may be considered. What is then required is a utility available to all telegram users to decode the incoming telegram. Hence the concepts of a telegram Group-Descriptions-Table or GDT, and a Telegram-to-Group decode procedure. Any applications program using these features will always correctly decode the incoming telegram if the relevant GDT is up to date. Hence each processor running equipment-modules has been provided with a set of GDTs and the decode procedures to interpret the telegrams, and all applications programs and real time tasks should use them. (Fig 3.).

The telegram reception logic buffers all telegrams in the whole super-cycle. This permits applications programs to access any cycle in the super-cycle, hence saving PLS-lines in the telegram by removing the need for groups to describe the next cycle. It further allows PPM to continue with the last sent super-cycle in the event of a PLS or TSU failure, and at machine development times one may locally simulate any super-cycle rather than follow the one operationally set. The GDT and telegram buffers, along with status information such as the current cycle number and the super-cycle length etc, are collected together into a single data structure called a Telegram-block or TBK.

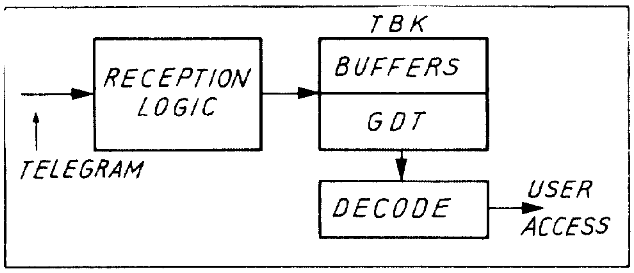


Fig 3. Telegram access logic.

Why make a Telegram-slave-unit

Alltogether the PLS system has proved to be a very flexible way to sequence the accelerators, so it was decided to use a similar approach to sequence the LEP-PreInjector. The existing PLS-system, and the corresponding console-computer interactive programs, would be very difficult to modify and cost a lot of beam down time. It would be much more sensible to develop the new system in parallel with the old one, using new and more powerful technology. (Fig 4.).

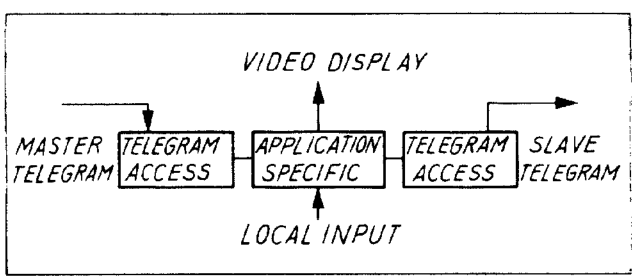


Fig 4. The TSU components.

The TSU consists of three parts, one which reads and decodes the incomming master telegram, one which executes the instructions which describe the relationship between the master and slave telegrams, and the final part, which sends out the slave telegram. The parts which receive and send out telegrams consist mainly of the telegram access logic and the master and slave TBKs. The central part which contains the instructions on how the slave is to be built is application specific, and hence depends on the particular TSU. In order to facilitate the programming of a telegram transformation in a TSU, a special language (FIDO) has been developed, and an interpreter for this language forms the heart of each TSU. Thus TSUs are very flexible as the telegram transformation program can be modified on-line using a language developed just for the job. The TSU provides the telegram-implementer with a custom built interpreter working with a predefined structured data space and operations which work with this space.

Discussion

The instructions or program which the TSU must perform in order to produce the slave telegram could be complex, and may involve looking at other master and slave cycles than the current ones. Thus depending on what operations must be performed, there will be a delay between the time when the information becomes available and the time when the newly calculated slave telegram is available. This delay can be much longer than a basic period, and in principle could be even longer than the super-cycle depending on what information is in the slave and how complex it is to calculate it. We use the fact that the super-cycle is a repetitive sequence in order to overcome some of these difficulties, and it is for this reason that the TSU buffers both all master telegrams and all slave telegrams. Hence the global implications of TSU real time response on a hierarchical control system must be considdered. We sometimes do need a real time response in handling sudden status changes which can not be precalculated for one reason or another. We might require fast reaction times for external conditions, or changes in the master super-cycle. If we are to guarantee that this response is within a few hundred milli seconds, then no great amount of calculation can be involved. In other words such responses might be logically much more primitive than a stable state (precalculable) mode of operation.

We have outlined two black boxes so far, represented by the TSU and the LBS. The TSU takes orders from above and produces more detailed orders for the level below, while the LBS takes orders and produces pulse trains to control the hardware. Hence by investing in the generality of these black boxes we can have a very flexible modular timing and sequencing system, and reduce the overall manpower-effort. To start with the master telegram format is that of the current PS, and the first slave format is the new LPI-telegram. If all the master telegram users are TSUs, then the PLS need only send telegrams when the super-cycle changes, the TSUs will do the local sequencing based on the last transmissions. All the problems outlined in the previous chapters can now be solved using this scheme. These principles are now further elaborated with the aim of a CERN wide sequencer and timing scheme.

References

[1] G. Baribaud, P. Heymans, On the sequencing of the CERN machines and on a unified particle transfer protocol. CERN/LEP/Note 564, 1986.

[2] P.P. Heymans and B. Kuiper for the PS Controls Group, in Lecture Notes in Physics 215, Computing in Accelerator Design and Operation Proceedings, Berlin 1983, Current control of interacting accelerators with particle beams of varying format and kind.