# Kicker Sequence Module (KSM) Manual

Thomas Lindner (DAQ group)

Version 1: Aug 12, 2016 : Initial release

Version 2: Mar 30, 2017 : added defaults to register map

*Introduction*

The following is a manual describing how to operate the Kicker Sequence Module (hereafter KSM). This KSM is in charge of creating the sequence of kicks that will drive the UCN kicker power supply. The KSM will be controlled by an EPICS control system provided by the controls group; this manual is principally meant for the controls group.

The KSM was built in the context of the overall UCN kicker control system, which is described in TRIUMF design note 16-11.[[1]](#footnote-1) Please refer to that document for an overall picture of how the UCN kicker control is meant to work. This manual includes some figures and text from design note 16-11; but some material may differ slightly from what is in the design note, because of changes while actually building the module.

*KSM Hardware and firmware*

The KSM is a VME card. The KSM hardware is actually a re-use of an existing TRIUMF VME card called the Programmable Pulse Generator (PPG), which is used by the DAQ group for creating configurable sequences of outputs.[[2]](#footnote-2) The KSM hardware is identical to the standard PPG hardware, except for having a couple extra LEMO connectors soldered on.

The KSM has 4 lemo inputs and 12 lemo outputs; the current KSM firmware uses 2 inputs and 7 outputs. The lemo outputs can be set to output either TTL or NIM logic levels; each output can be individually toggled between TTL and NIM by a dip switch.

The kicker power supply actually requires an optical logic signal to control the magnet; so we plan to add an additional external box which does a TTL (or NIM) to optical conversion. Daryl Bishop has agreed to provide this converter.

The KSM has different firmware from the standard PPG. The new firmware is necessary for the particular KSM functionality; the firmware was written by Chris Pearson of the DAQ group.

*KSM Functionality*

As noted, this module is in charge of creating the sequence of kicks that will actually drive the kicker power supply. Figure 1 shows the logic inputs/outputs from this module, as well as the VME control interface are shown below. In this section we will describe the logic by which the KSM generates these outputs. We will also note the VME registers which control certain KSM behavior.

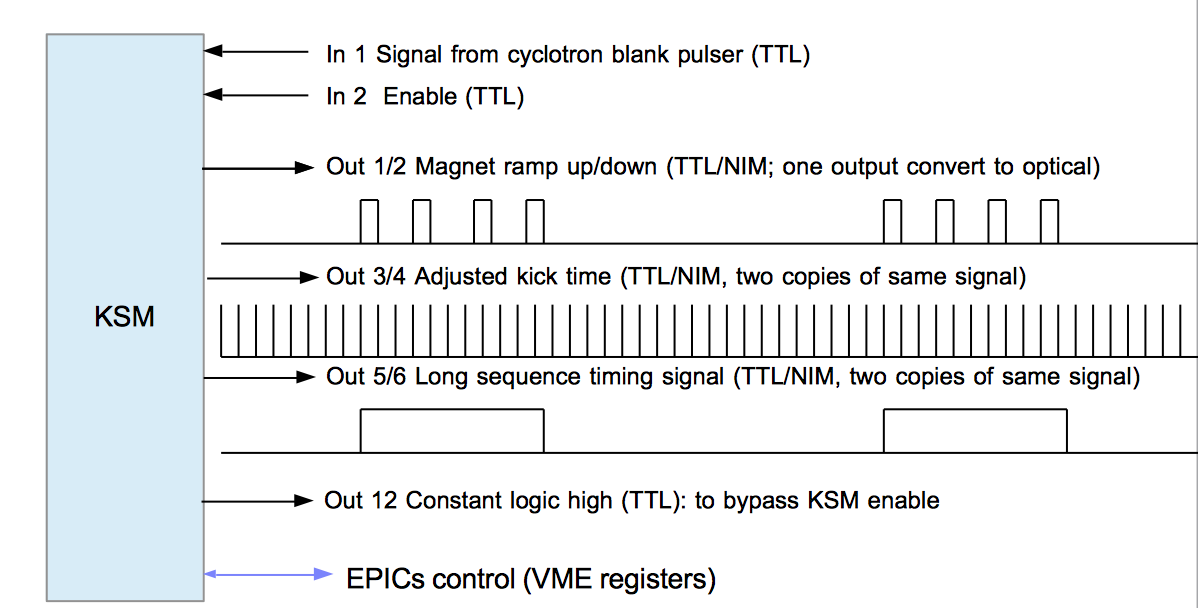


Figure : Logical inputs and outputs of KSM.

The KSM would start by defining the time at which the magnet kick should start. This time would be defined by adding a fixed offset to the time from the cyclotron blanking pulser signal (input 1). The operator would need the ability to change the time offset of the kick signal (VME register PULSER\_OFFSET). A copy of this **adjusted kick time** would be available on outputs 3 and 4.

The KSM would then define the actual short term and longer term sequence for the kicker magnet; i.e. define the short term cycle of kicking of a fraction of the beam buckets and the longer 1 min. beam-on / 3 min. beam-off cycle for normal operation. The actual magnet ramp up/down signals would be generated on outputs 1 and 2. Outputs 5 and 6 would indicate the longer beam-on/beam-off periods.

As mentioned earlier, there would need to be a large number of different modes and configurability in defining the KSM sequence. For a starting point we would want the ability to operate the KSM and kicker with the following different modes:

1. Operate the kicker magnet to permanently on so all beam gets directed to BL1U (VME register KICKER\_DC\_MODE). This would mean setting the magnet output (output 1) to permanent high.
2. Operate the kicker magnet where we kick a fraction (VME register KICK\_FRACTION) of beam buckets to beamline 1U. It would repeat kicking a fraction of buckets for a configurable ON period (VME register BEAM\_ON\_PERIOD) then shut off magnet for a configurable OFF period (VME register BEAM\_OFF\_PERIOD). It would repeat these ON / OFF periods for a configurable number of **long cycles** (VME register NUM\_LCYCLES).
3. Operate the kicker magnet in a manner identically to 2, except that the ON/OFF cycles would continue forever (VME register BEAM\_CONTINUOUS\_CYCLE).

The **sequences** in modes 2 and 3 would be started by writing to a VME register (START\_SEQUENCE). We show a diagram showing a mode 2 or 3 sequence in Figure 5.

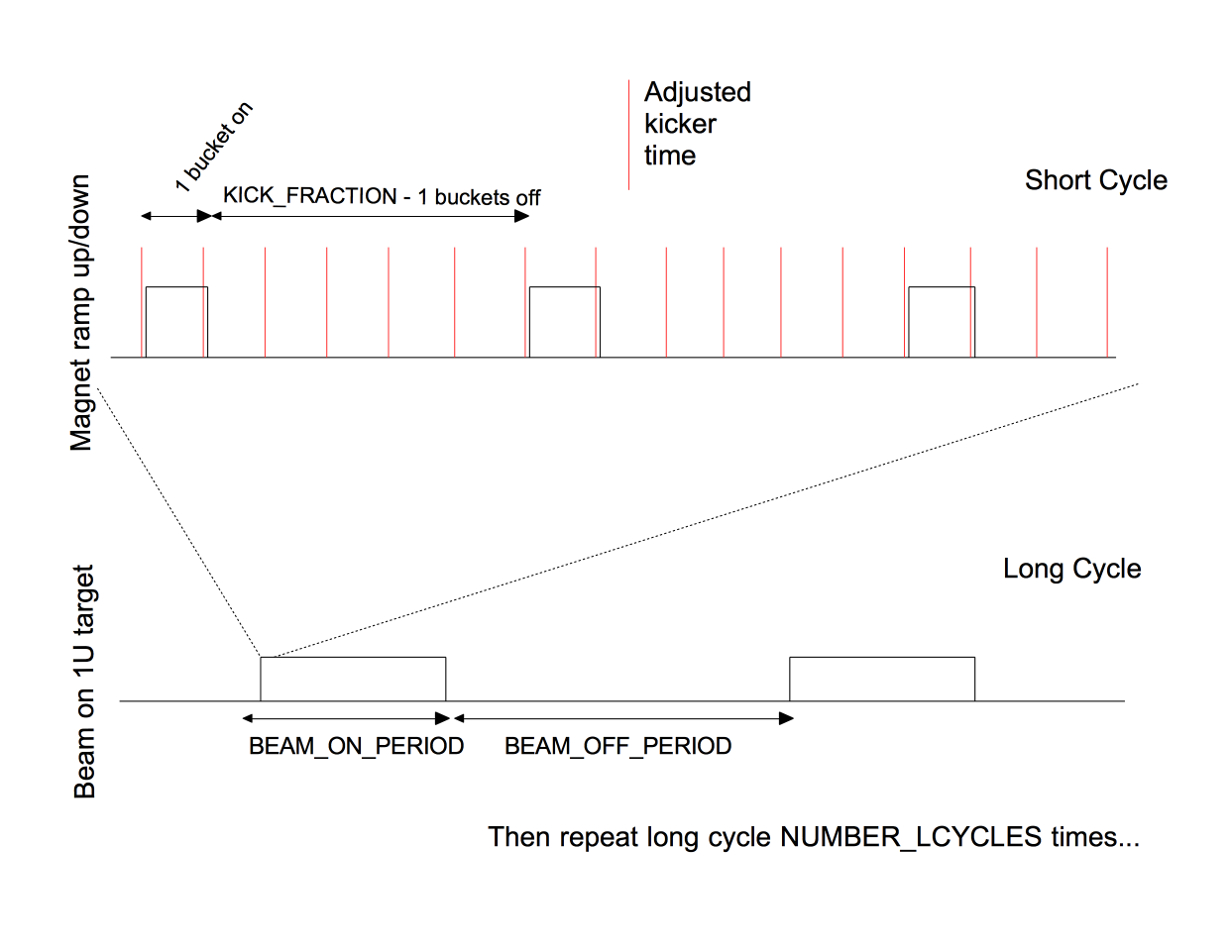


Figure : Diagram of short and long kicker cycles. In this example we are kicking 1 out of each 6 beam buckets.

*VME registers*

Below we list in more details the VME registers. Note that in addition to the set of configuration registers listed above, there is also three status registers, as well as a register for resetting the KSM.

Table 1: List of VME Parameters

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| VME Parameter Name | VME Address | | Parameter Function | Default value | Type ; unit |
| KSM\_ENABLE | 4 | If set to 1, then the KSM can start kicker sequences or set kicker in DC mode  If set to 0, then the KSM will not operate magnet (ie, output 1 will not go high). | | 0 | r/w ; n/a |
| KICKER\_DC\_MODE | 8 | Setting this parameter to 1 will turn on the kicker magnet permanently, ie set output 1 high. | | 0 | r/w ; n/a |
| PULSER\_OFFSET | 12 | Fixed offset to add to cyclotron blanking pulser before subsequent operations | | 0 | r/w; Clock units (10ns) |
| KICK\_FRACTION | 16 | Defines fraction of beam buckets kicked to beamline 1U. Specifically, means we will kick one bucket to 1U and then let KICK\_FRACTION-1 buckets go to beamline 1A; so 1/KICK\_FRACTION buckets will go to 1U. | | 3 | r/w ; n/a |
| BEAM\_ON\_PERIOD | 20 | Defines length of beam ON period (with fraction of buckets going to UCN) | | 1 | r/w; number of cyclotrons blanks. |
| BEAM\_OFF\_PERIOD | 24 | Defines length of beam OFF period (with no beam going to UCN) | | 1 | r/w; number of cyclotron blanks |
| NUM\_LCYCLES | 28 | The number of ON/OFF cycles to perform before stopping | | 2 | r/w; cycles |
| BEAM\_CONTINUOUS\_ CYCLE | 32 | Writing 1 to this register would set the KSM to use a continuous (never ending) sequence. Writing 0 would mean a cycle ends after NUM\_LCYCLES. | | 0 | r/w; n/a |
| START\_SEQUENCE | 36 | Writing 1 to this register would start the magnet kicks.  Writing 0 to this register would stop any ongoing sequence.  Register has no effect in DC mode. | | 0 | r/w ; n/a |
| KSM\_STATUS | 40 | Bit 0 : is magnet on Bit 1 : is in sequence Bit 2 : is in beam-on period Bit 3 : 1 is the KSM is disabled by input 2. | | n/a | r/o; bit-field |
| CYCLE\_COUNT | 44 | Number of cycles executed since sequence started. | | n/a | r/o |
| KICK\_COUNT | 48 | Number of magnet kicks in the current cycle. | | n/a | r/o |
| RESET | 60 | Writing to this register will turn off the magnet (if it is on) and reset all VME registers to their default values. | | n/a | Write-only |
|  |  |  | |  |  |

*KSM Machine Protection Functions*

The KSM will also have an enable input (input 2). If the enable input goes low then the KSM should immediately ramp down kicker magnet (if it is on) and stay down. Longer term this will allow the KSM to be inhibited by the Kicker Timing Module measurements of the 1VM4 capacitive pick-off. For the short term we will bypass this enable functionality by wiring KSM output 12 (always high) to KSM input 2.

Additionally, the KSM has logic where the magnet will always get ramped down after 2 ms (unless in DC mode). This will protect against a situation where the KSM ramps the magnet up on one blanking notch, but the signal from the blanking notch then goes away (for whatever reason).

*Related Documents*

* VME-PPG documentation: <http://www.triumf.info/wiki/DAQwiki/index.php/VME-PPG32>
* Kicker Control and Diagnostics Technical Note (TRI-DN-16-11)  
  <https://documents.triumf.ca/docushare/dsweb/Get/Document-132994/TRI-DN-16-11.pdf>

1. Design Note 16-11 can be found here:  
   <https://documents.triumf.ca/docushare/dsweb/Get/Document-132994/TRI-DN-16-11.pdf> [↑](#footnote-ref-1)
2. See Related Documents at end for link to further PPG documentation [↑](#footnote-ref-2)