# Code outline: timeSyncDetectors\_source\_mod51.py

There are 7 main functions. They are, in order of being called,

**read\_polaris\_data**

**read\_LaBr\_data**

**apply\_coordinate\_transformations**

**find\_POLARIS\_sync\_pulses**

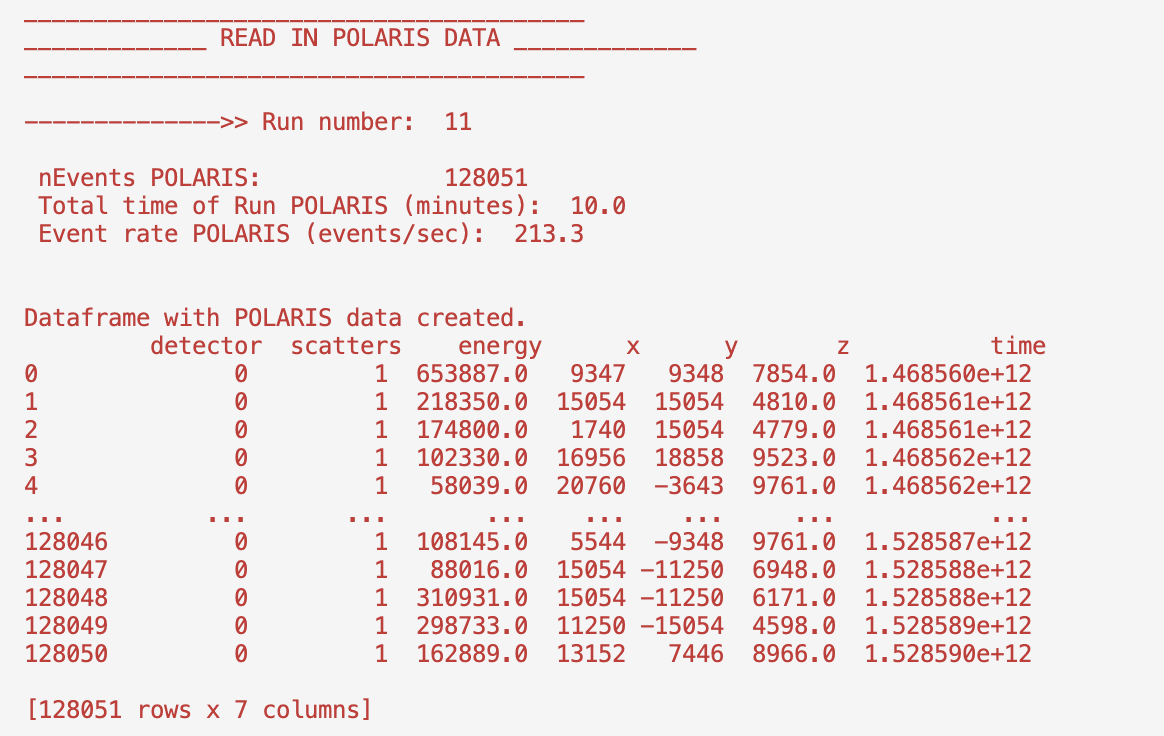
**find\_LaBr3\_sync\_pulses**

**time\_walk\_correction**

**merge\_two\_detector\_dataframes**

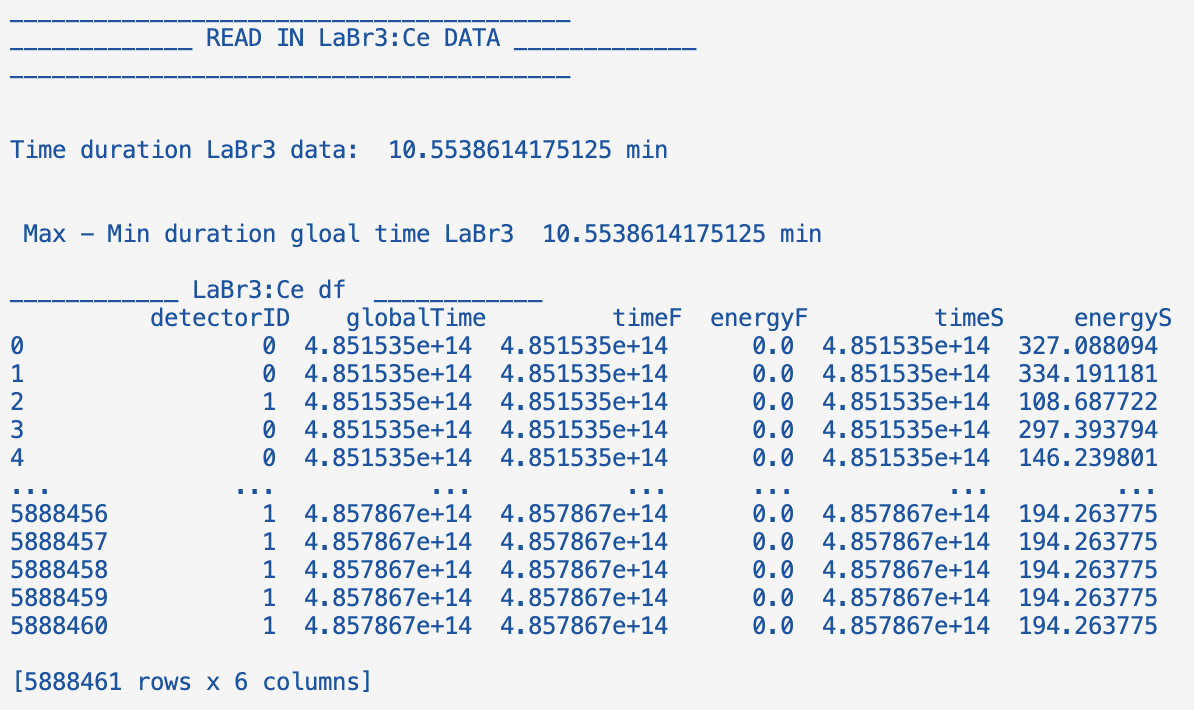
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. **read\_polaris\_data**

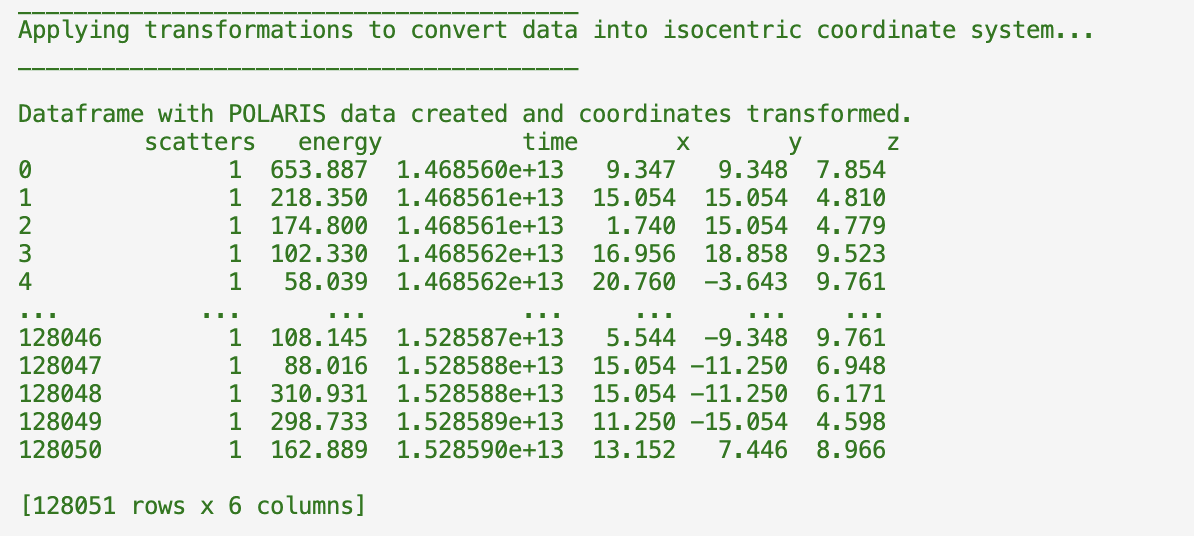
* read in the Polaris mod51.txt file: ’index’, ‘detector’, 'scatters', ‘energy’, 'x', 'y', 'z', 'time'
* filter the Polaris data to include only single scatter events and sync pulse events. The single scatter events are recorded as scatters “1” by the DAQ and the sync pulses are identified by a scatter value of “122”
* perform a check to see if the data is indeed in 10s of nanosecondsas is stated in the Polaris documentation, and for which we later correct.****

1. **read\_LaBr\_data**

* The LaBr3:Ce detector data is read in from a root file. This data ha sbeen sorted by two different sorting codes before being used in this post-processing. The first sorting code applies a time gate between events and reads the data in from the raw list mode data, calibrating the energy of the slow and the fast detector channels, and outputting a root file containing a TTree that houses the event data . The second sorting code applies a transformation to the energy-time matrix of the slow energy vs the time difference between the slow and the fast detector event time values. The time walk is reduced < 200 keV and eliminated beyond that. There is a 6-7 ns timing resolution between the fast and the slow detector times. This allows us to correspond the slow channel energy and the fast channel time from an event, allowing for +4% energy resolution and timing resolution ±300 ps.
* The Lar3:Ce data contained in the TTree is: ‘detectorID’,’timeF’,’energyF’,’timeS’,’energyS’.
* The detectorID is 0 for detector 1 (labelled 8), 1 for detector 2 (labelled 3), and 5 for the Polaris synchronization pulses.
* The time difference between the first and the last fast time is calculated to ensure that there is no lost data and to check that the time is indeed measured in nanoseconds . The LaBr3:Ce detector always has a longer time duration because of the nature of how the experiment is conducted. The LaBr3:Ce detector data acquisition tape is started, then the “connect camera” button in the Polaris GUI is enabled which begins to send the sync pulses into the Lar detector DAQ in 2 s intervals. When rates are seen in the MIDAS program in channel 15, the Polaris detector data acquisition is started and it is a timed run with a preset time duration. The Polaris data acquisition is completed and the LaBr data acquisition continues to acquire until stopped. Synchronisation pulses are sent continuously by the Polaris DAQ, even after the data acquisition has been stopped and only until the Polaris camera is “disconnected”. For these experiments, the Polaris camera is always disconnected before the Lar tape is stopped. The LaBr3 dataset should contain the same number of synchronization pulses as the Polaris dataset, however, the event data does not store the initial “connect camera” data and so there are more sync pulses in LaBr3 dataset than the Polaris dataset ( for a 10 min timed acquisition, the Polaris data will contain exactly 300 sync pulses).

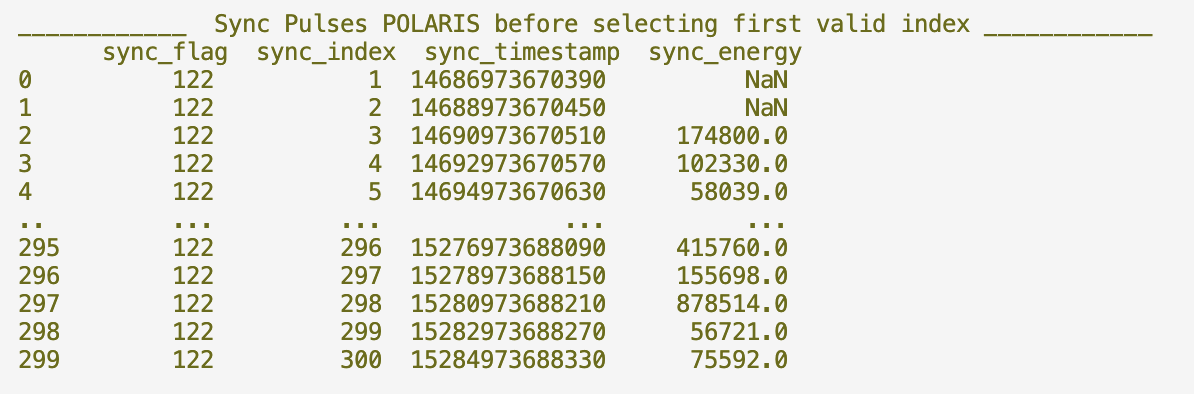
****

1. **apply\_coordinate\_transformations**

* The ‘x’,’y’,’z’ coordinate values of the Polaris detector are transformed by rotating them using mathematical formulas to transform the coordinates by a predetermined reference system.
* The energy, time and coordinate values are converted to the relevant units for the study. Energy is converted from eV to keV, time from tens of nanoseconds to nanoseconds and the position coordinates from meters to millimeters.
* This is the data that will be used for further Polaris detector event processing.

1. **find\_POLARIS\_sync\_pulses**

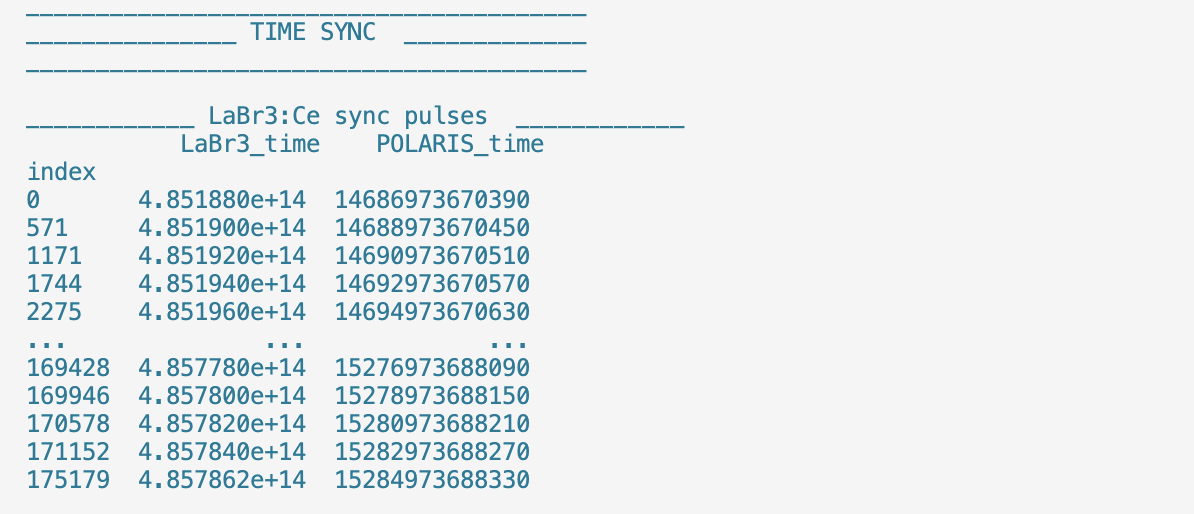
* The Polaris synchronisation pulses are configured to send to from the Polaris detector onboard DAQ module to the PIXIE-16 500MHz XIA card channel 15 every 2 ns (2000000060 ns). These are configured to be sent before the data acquisition of the Polaris detector has begun by pressing a button in the Polaris python GUI that says “connect camera”. This was configured in the Polaris detector capability to ensure accurate timing when connecting Polaris detectors to one another in coincidence measurements. The Polaris sync pulses have a flag sent to the “scatters” column in the list mode data read-out whereby all “scatters” equal to 122 correspond to Polaris sync pulses and should be 2000000060 ns apart in time.
* A second set of data is created in this function. The synchronisation pulses are pulled out of the Polaris event data.
* When the “connect camera” button is pushed in the Polaris UI, the sync pulses are sent out but their corresponding “energy” signal amplitude is null. When the data acquisition is set to begin, the sync pulses have an associated non-null energy magnitude.
* The first valid Polaris sync pulse corresponds to the first sync pulse that was sent after the data acquisition had been started. The integer number of sync pulses is saved for later use and the time duration from the first to the last valid Polaris synchronisation is calculated to ensure agreement with the known and run time of the experiment. The time difference between consecutive sync pulses is also calculated to check that it is in fact 2000000060 ns.

****

1. **find\_LaBr3\_sync\_pulses**

* The Polaris detector sends the 2 ns interval sync positive square-wave pulse to channel 15 of LaBr3 XIA DAQ card. The PIXIE system records the pulse height ‘energy’ and the time of this measurement using the CFD fast timing capability.
* Due to the square wave having a long duration in time, the number of non-zero fast energy- fast time events in channel 15 far exceeds the number predicted due to the experiment measurement time. The fast energy data was retained up to this point for the sole purpose of finding the Polaris time stamps. This is corrected for by doing the following: Calculating the time difference for every sync pulse relative to the first and then taking the integer of this time difference divided by 2000000060. We then obtain 0,0,0,1,1,1,2,2,2… etc. We keep only the first time-energy pair associated with each unique integer time\_difference/2000000060 value. We then check that the number of sync pulses we obtain \* 2 seconds is in agreement with our experiment run time. A further check is performed in taking the time difference between the new LaBr3 DAQ sync pulses matrix to see if it corresponds to 2000000060 ns.
* The integer difference between the number of sync pulses found in the LaBr3 event data and the Polaris event data is calculated. The first sync pulse in the LaBr3 DAQ is assumed to be located at the index + 1 of the integer number of excess sync pulses found between the two DAQs (which were associated with the “connect camera” operation, before the Polaris DAQ was started).
* A new column is added to this LaBr3:Ce sync pulse data called “sync\_flag’ and it is assigned the value 122.

1. **time\_sync**

* The Polaris detector sync pulse identifier ‘detectorID’=5 is dropped from the LaBr3 event data and the time difference between consecutive LaBr3 detector events is calculated and stored.
* The LaBr3 sync pulses with the 122 sync\_flag are inserted into the data at their relevant time positions.
* Each LaBr3 sync pulse time from the PIXIE DAQ is swopped with the corresponding sync pulse time from the Polaris DAQ 
* Starting from each sync pulse, the time difference is added cumulatively to the new polaris time for each entry between pulses. In this way, the LaBr3 data now has new time stamps associated with the Polaris DAQ clock time but retaining the original fast time behaviour of the LaBr3 detectors.
* ‘x’,’y’,’z’ columns are added to the data are added and filled with null values, and the slow time information is dropped from the events. The LaBr3 event data is now in Polaris clock time and has the columns : DetectorID (0 or 1), energy (in keV), time (in ns), x (null), y(null), and z (null). The energy data here was previously the energyS (slow channel energy) and the time data was the timeF (fast channel time).
* The polaris sync signals are exactly 2000000060 ns apart, whereas due to the DAQ deadtime or cable length etc, the labr3 received sync pulses are not (as seen in syncTimeDiff below). And so the Polaris sync pulse time values have been corrected for this shift by calculating 2000000060 – syncTimeDiff and then subtracting this corresponding value for the corresponding sync pulse signal.
* A screenshot of a computer

  Description automatically generated
* The time difference is again calculated from the start to the end of the data to ensure that no data has been corrupted or lost after these matrix operations have been performed. The sync pulses ( the events where sync\_flag == 122) are dropped.

1. **merge\_two\_detector\_dataframes**

* We assign the polaris detector events with a detectorID variable and let it equal to the integer 5. The polaris events are now ‘detectorID’ (equal to 5, ‘energy’ (in keV), time (in nanoseconds), and ‘x’,’y’,’z’ (in millimeters).
* The polaris and labr3 detector events are now on the same clock time and have the same data structure. They are merged and sorted according to time.