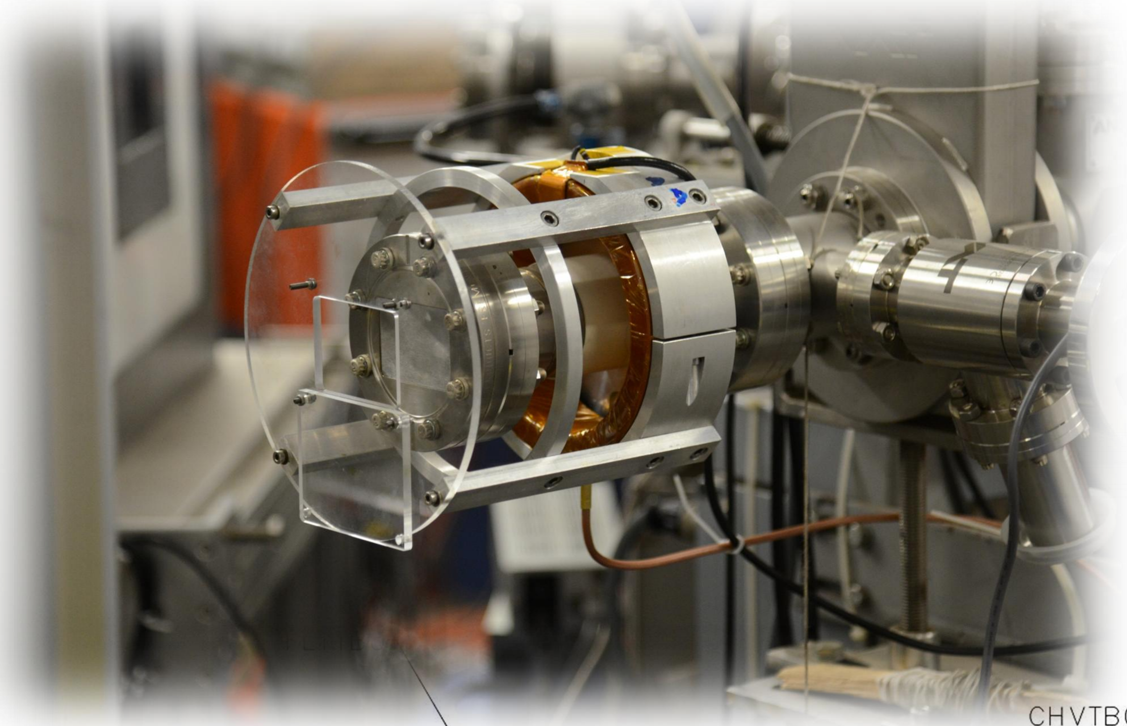
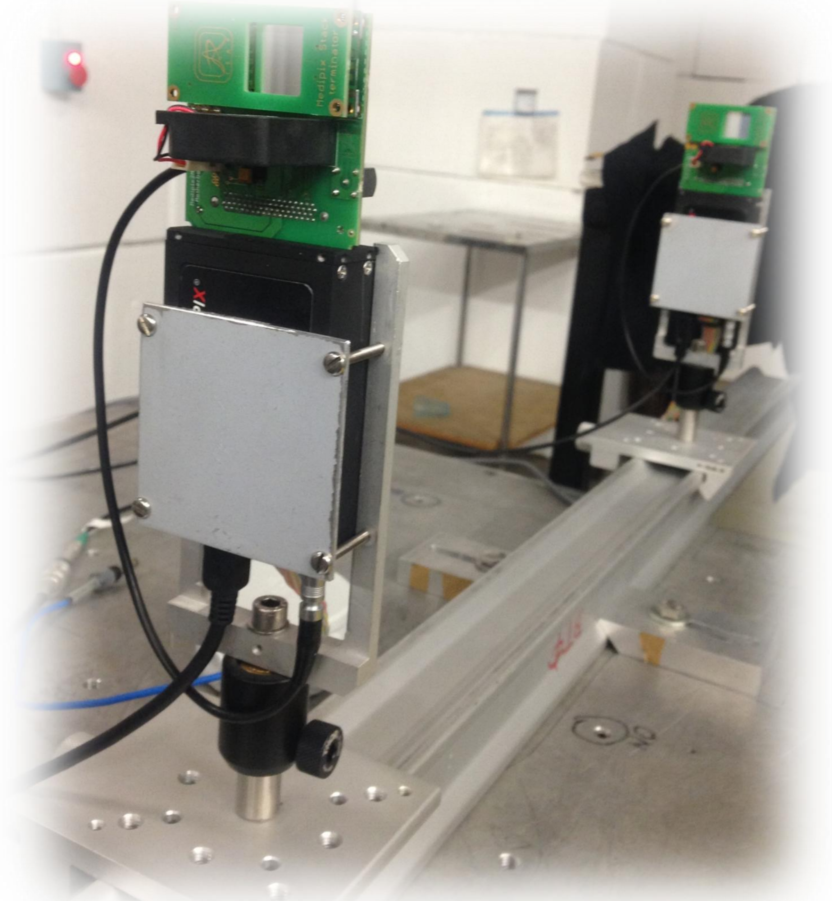
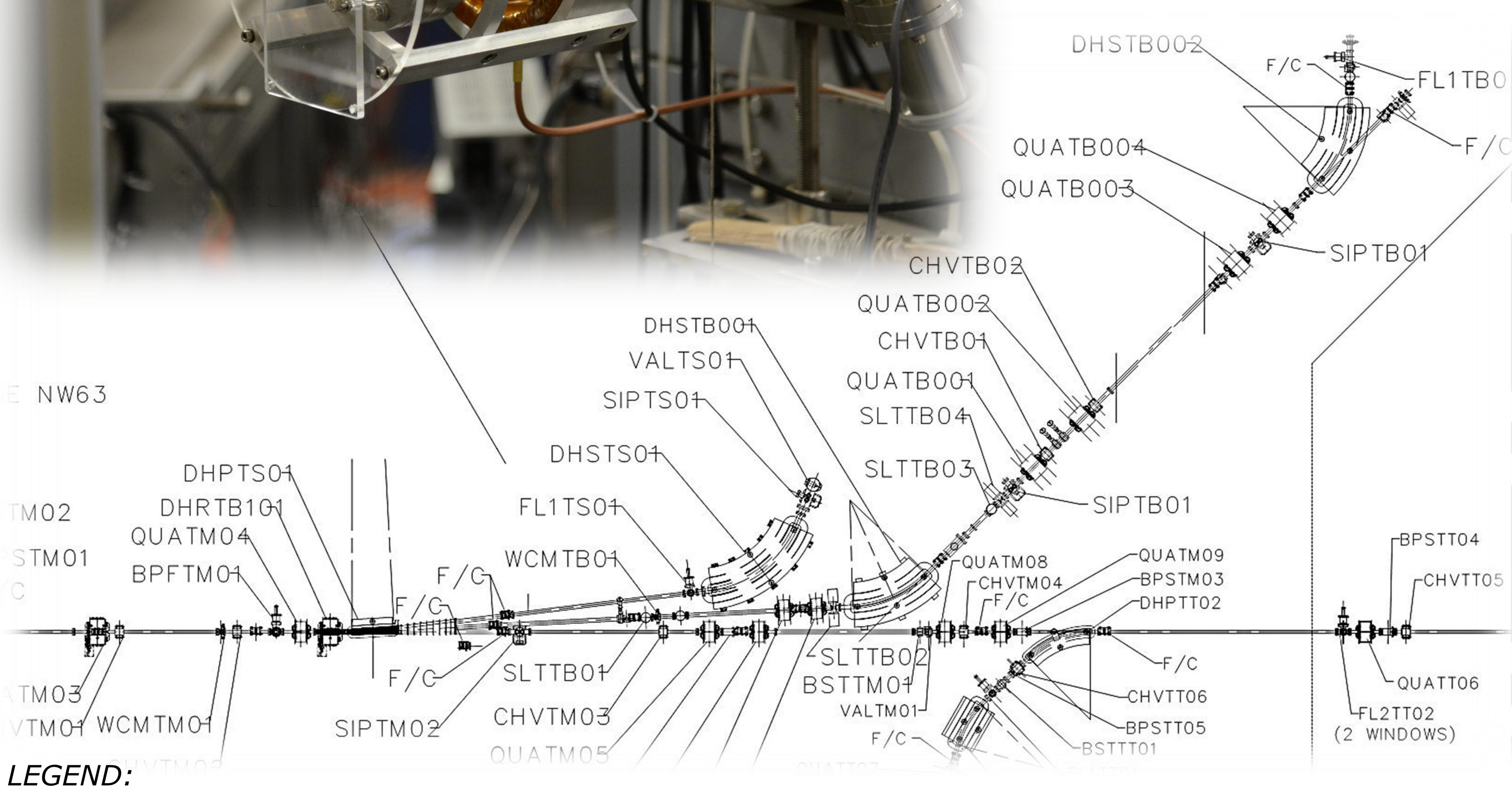
A HARDWARE AND SOFTWARE OVERVIEW ON THE NEW BTF TRANSVERSE PROFILE MONITOR

B.Buonomo, C. Di Giulio, L.G. Foggetta†, INFN – Laboratori Nazionali di Frascati, Frascati, (Italy)
P.Valente, INFN – Sezione di Roma, Rome (Italy)



The BTF (Beam Test Facility) is part of the DAΦNE accelerator complex: it is composed of a transfer line driven by a pulsed magnet allowing the diversion of electrons or positrons

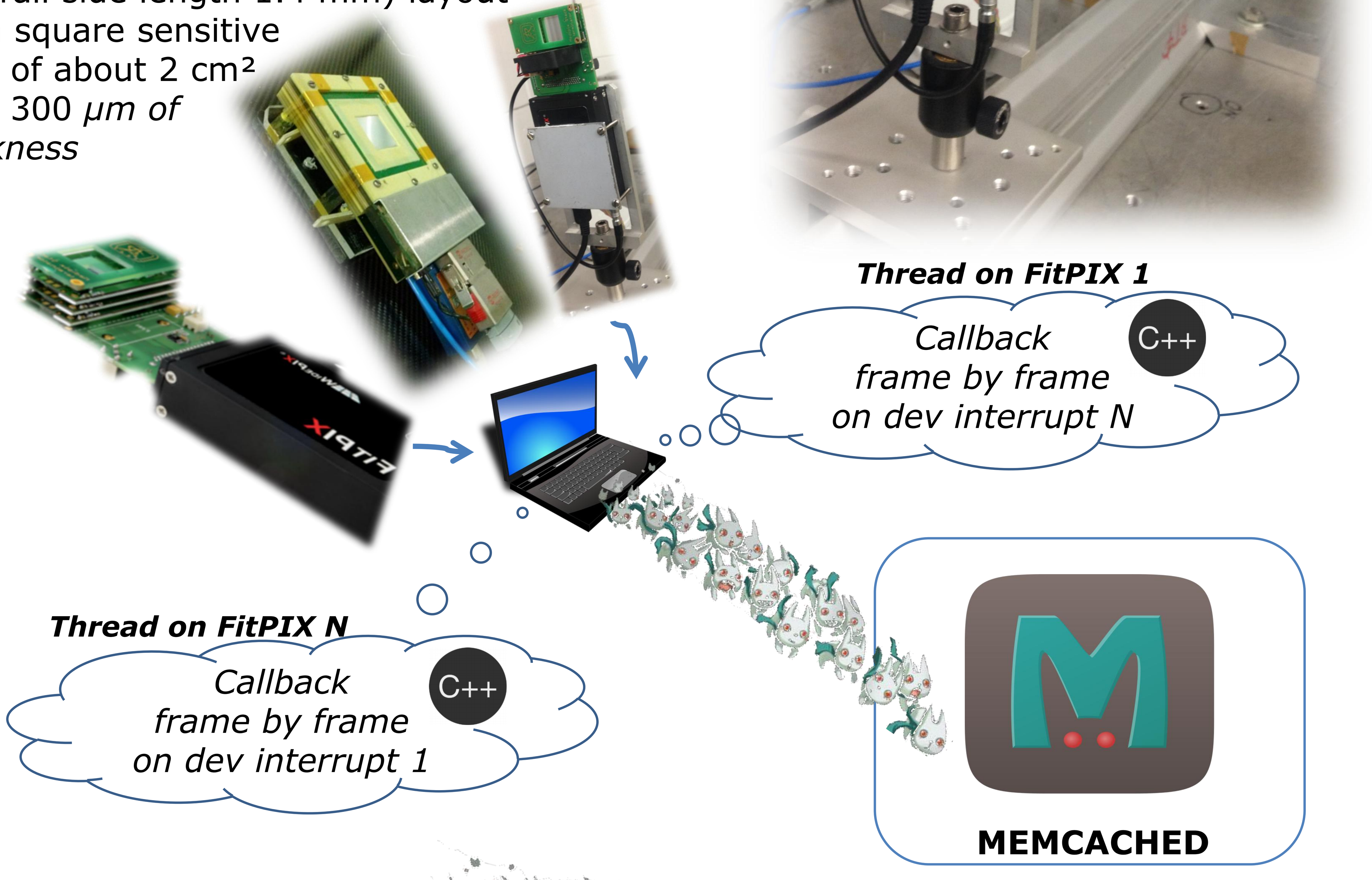




FITPIX Kit Detectors: sensors have a square pixel of 55 μm side in a 256×256 pixels (overall side length 1.4 mm) layout for a square sensitive area of about 2 cm² and 300 μm of *thickness*

LEGEND:
- *DHPTB101 = 3° pulsed dipole*
- *DHSTB001 = 42° DC dipole magnet,*
- *QUATB*** = 3x quadrupole doublets,*
- *SLTB*** = 4x scraper collimator,*
- *~20 m long drift line.*
- *DHSTB002 = second (45°) dipole magnet*
- *(two alternative windows)*

In order to have a completely arbitrary access to live data at the maximum frame-rate within the BTF duty cycle, we have implemented software architecture with a typical producer-consumer layout, implementing data caching on MEMCACHED server, to allow more than one different consumers at the same time.



This has been possible exploiting all the available programming solutions: starting with the python scripting capability of the provided software kit, down to low-level programming, which required some interaction with ADVACAM for a full exploitation of the Linux libraries. We have a fully working version of the software for the single **FitPIX**, the **multidetector-single FitPIX**, and more FitPIX devices in daisy chain. As a by-product, the same software architecture allowed us to have a similar implementation for a **GEMPIX** tracker, a four sensors detector with a software configuration similar to the four stacked layers in a single FitPIX.

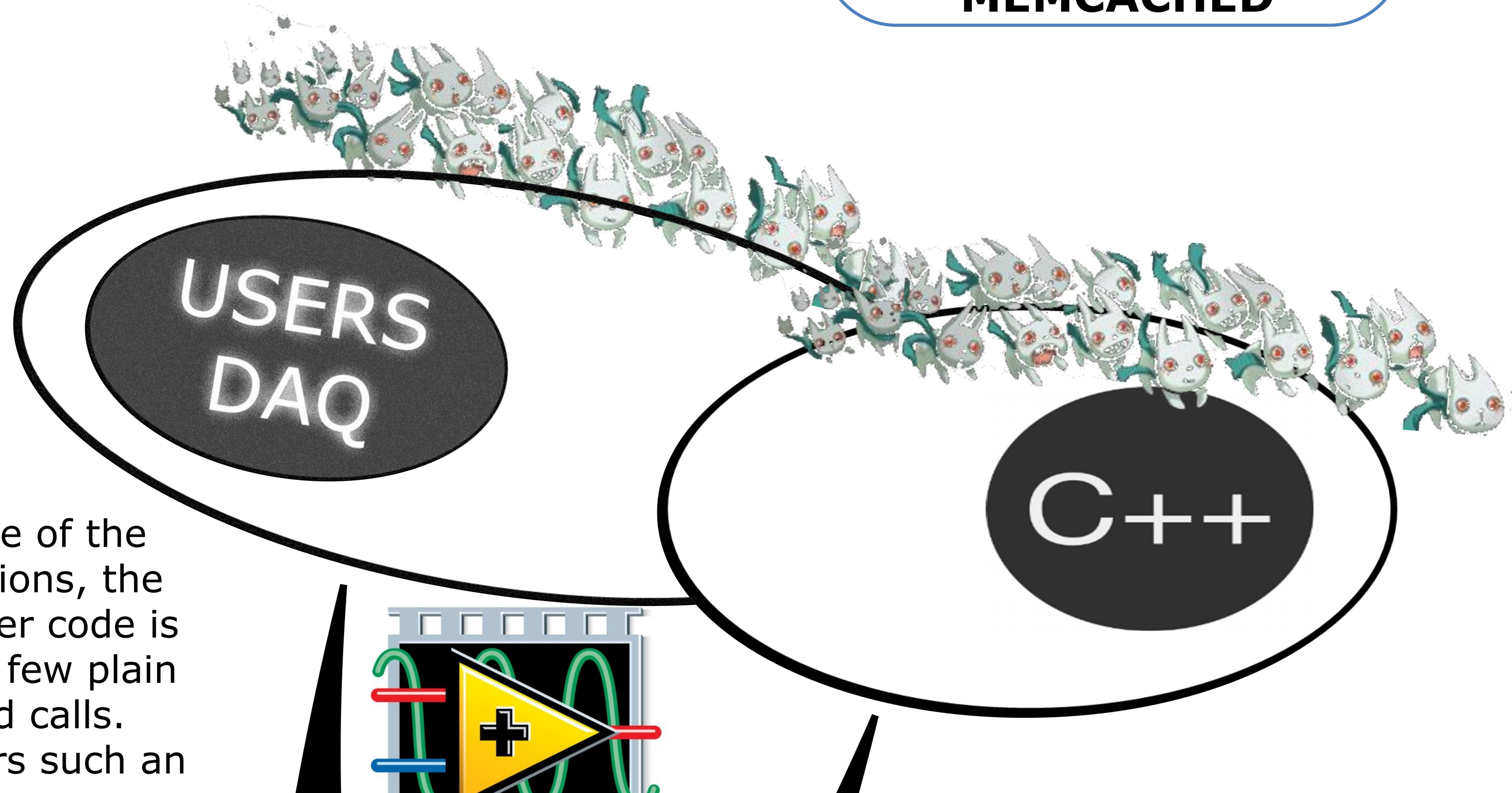


Table 1: Time Profiling PIXET Software with Python Scripting on MC

CODE (OS)	MC	Trigger	Trigger Type	Frame time [s]	N frames	N Rep	Acq. Rate [Hz]
PIXET(WIN) Py	Y	50 Hz	HW_Start	1.00E-05	1000	10	23.26
PIXET(WIN) Py	Y	50 Hz	HW_Start	1.00E-05	1	1000	24.61
PIXET(WIN) Py	N	50 Hz	HW_Start	1.00E-05	1000	10	48.48
PIXET(WIN) Py	N	50 Hz	HW_Start	1.00E-05	1	1000	24.90

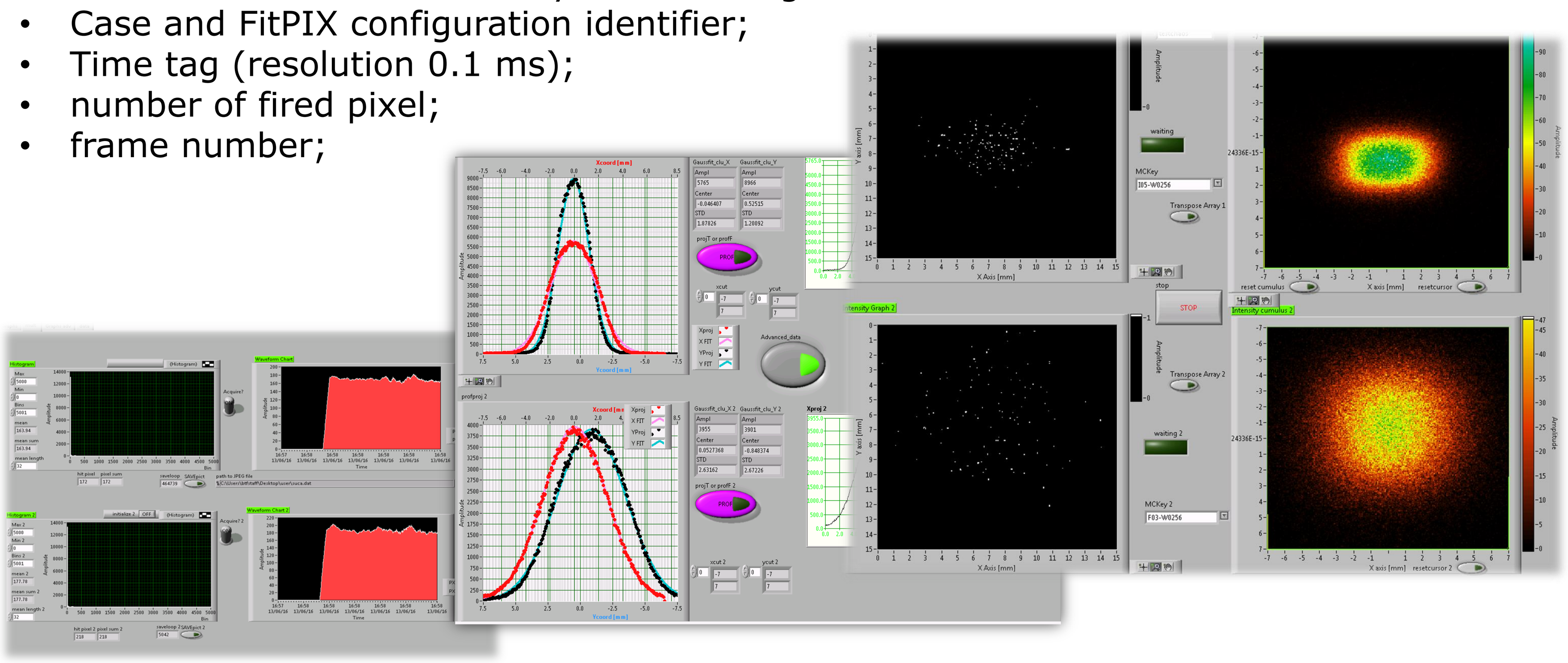
Thanks to the usage of the **MEMCACHED** functions, the integration in any user code is now just a matter of few plain C code include and calls. Allowing the BTF users such an easy integration in any, heterogeneous DAQ software, already during the first phases of the beam-time.

Users can easily implement this integration after having set NTP synchronization on their PC, before performing one of these actions: fetching the MC keys in a separate database for a delayed offline data-matching; implementing our example code in their own DAQ cycle, when providing a trigger signal to the FitPIXs; or, if the DAQ cycle is fast enough, even without any hardware synchronization, just matching the time of the two data streams (DAQ and FitPIX).

Develop of three possible data structures:
- array = send all the matrix data (65536 pixel value data, indirect pixel indexing),
- sparse mode 2dim array = a variable size bi-dimensional array [2,N]. Each slice is the couple of pixel data: pixel number and pixel value
- ultra sparse mode array = each array element stores the over threshold pixel number

All of these cases are headed by the following values:
- Case and FitPIX configuration identifier;
- Time tag (resolution 0.1 ms);
- number of fired pixel;
- frame number;

We have developed a LabVIEW runtime extracted beam parameters display (such as 3-d transverse image, beam centroid and Gaussian fit data), working both as shot by shot (instantaneous) and in cumulative mode. In addition, we have implemented a combined C/Root code, released freely, that fetches MC keys and integrates ROOT library to save ROOT trees, both in single and in multiple FitPIX configurations.



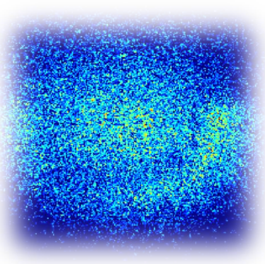


Table 5: Two detector, low data regime, consumer timing

Frame	Max Delay [s]	framerate [Hz]	Rejected
100000	0.010	49.98	0
100000	0.005	49.95	1
100000	0.002	49.95	6
100000	0.001	49.63	486

Table 2: Time Profiling BTF Compiled Code

Trigger	Trigger FitPIX	Frame time [s]	N frames	Rep	MC	Sparse file	Acq. Rate [Hz]
50 Hz	HW_Start	1.00E-05	1000	10	N	N	49.97
50 Hz	HW_Start	1.00E-05	1	1000	N	N	25.03
50 Hz	HW_Start	1.00E-05	1000	10	Y	N	31.00
50 Hz	HW_Start	1.00E-05	1	1000	Y	N	24.92
50 Hz	HW_Start	1.00E-05	1000	10	Y	Y	49.98
50 Hz	HW_Start	1.00E-05	1	1000	Y	Y	25.21
Auto	PXC_TRG_NO	1.00E-05	1000	10	Y	Y	41.80
Auto	PXC_TRG_NO	1.00E-05	1	1000	Y	Y	26.40
Auto	PXC_TRG_NO	1.00E-05	1000	10	Y	Y	78.64
Auto	PXC_TRG_NO	1.00E-05	1	1000	Y	Y	38.60

Table 3: Two Detector, Low Data, Timing Profile

Low	USB_0	USB_1	MC_0	MC_1
Mean[s]	5.9E-05	5.9E-05	8.62E-04	8.9E-04
Std[s]	1.7E-05	1.7E-05	1.79E-04	1.5E-04

Table 4: Two Detector, Full Matrix, Timing Profile

Full	USB_0	USB_1	MC_0	MC_1
Mean[s]	5.8E-05	6.0E-05	2.23E-02	2.33E-02
Std[s]	1.6E-05	1.5E-05	2.09E-03	1.37E-03