A modular and flexible data acquisition system for a cosmic rays detector network

Guilherme Saito, Marco Leite,
Marcelo Munhoz, Marcel Kuriyama,
Ricardo Menegasso e Rodrigo de Paula
Instituto de Física da USP

27 de Abril de 2022

Proposal

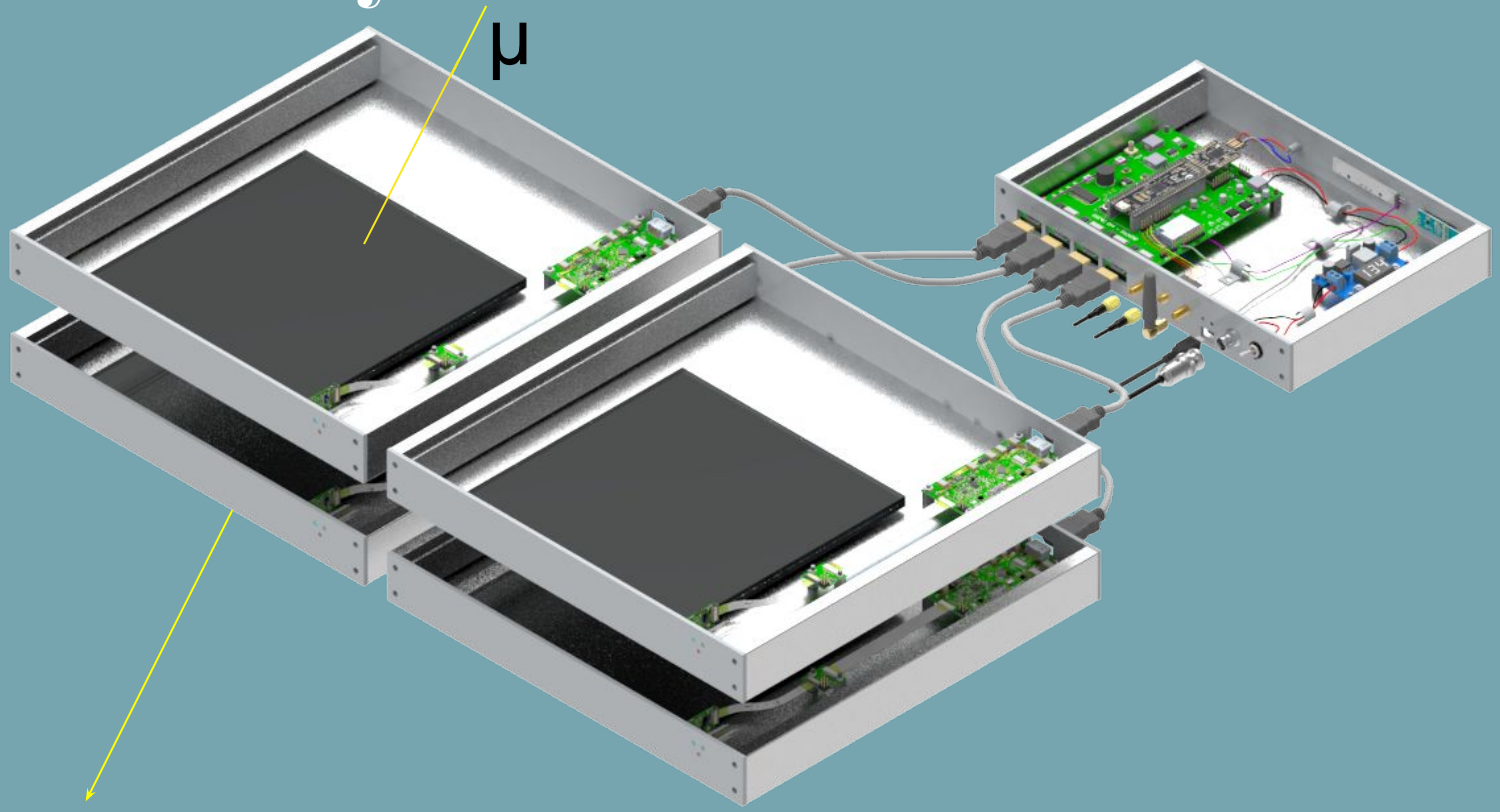
To create a cosmic rays detector network in both public and private schools involving students and teachers in the detector assembly and data analysis, motivating the study of particle physics topics and spin-off technologies.

- To collect data with the detectors built by the students themselves;
- Offer ready-to-use data analysis tools;
- Low cost, safe operation;
- Long term pedagogical proposal;
- Large geographical area coverage.

Station features

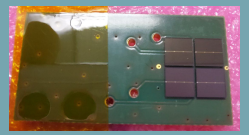
- 8 detection planes
- Plastic Scintillators
- Silicon Photomultipliers - SiPM (low operation voltage)
- GPS timestamping (accuracy of dozens of ns)
- Geometrical detection plane orientation (compass, gyro, accel)
- Station geographical location
- Wi-Fi connection
- Pressure, humidity, temperature and others
- Self-calibration
- Low cost and ease to use!

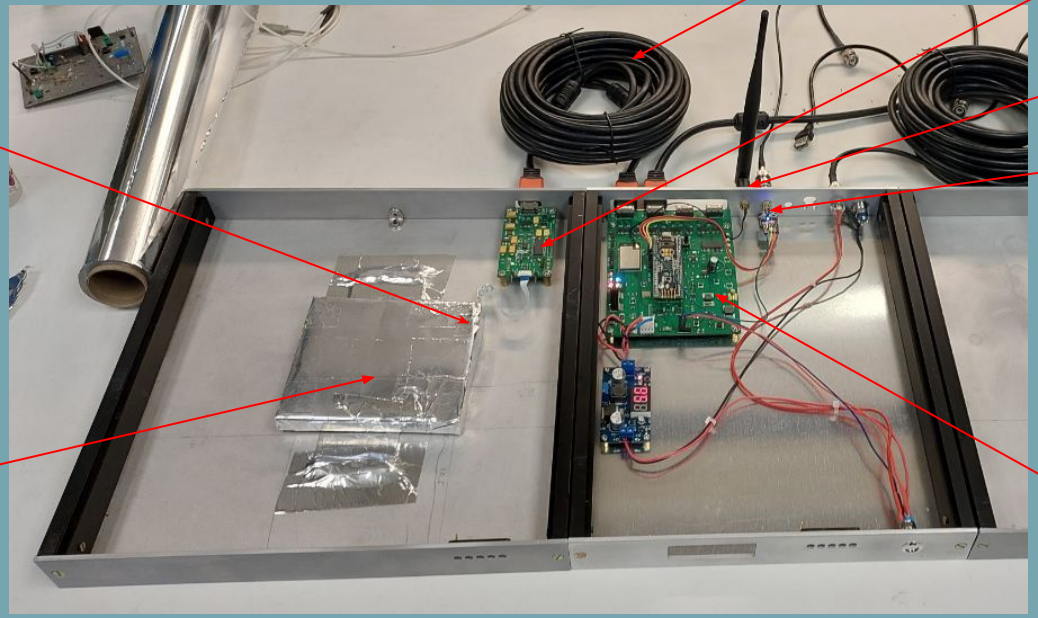
Cosmic Ray Detector Station



Station anatomy

SiPM





HDMI Cable

Frontend

Wi-Fi

GPS

Scintillator

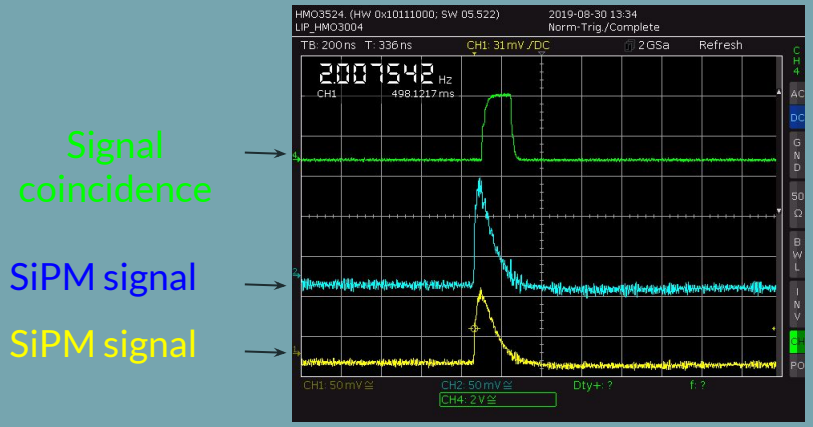
Backend

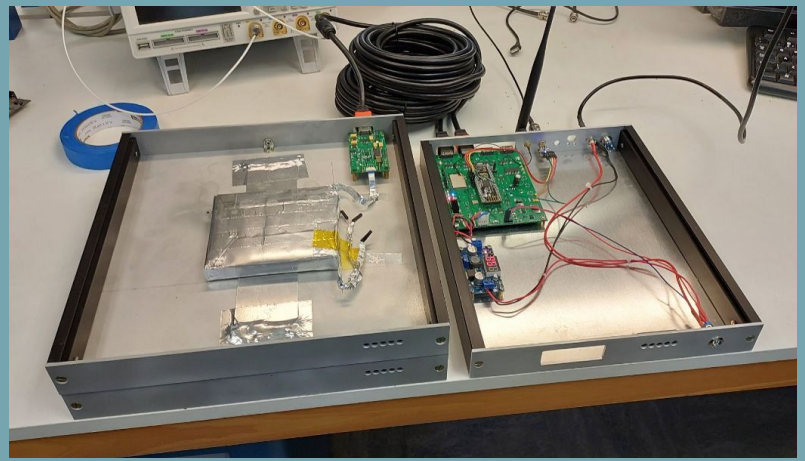
Prototype

Designed @USP

- Hardware project and assembly
- Firmware development
- Software implementation

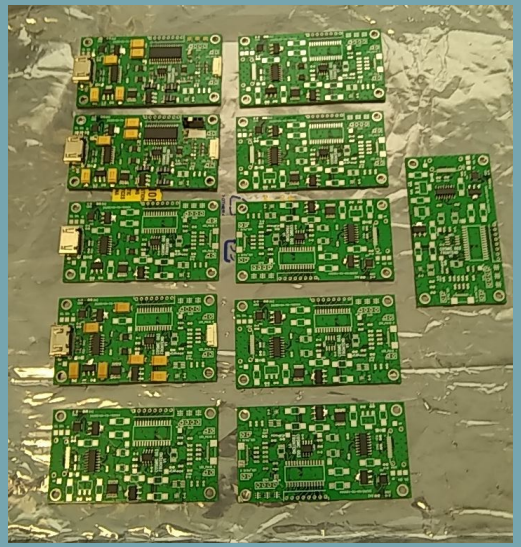
Prototype working since late 2019



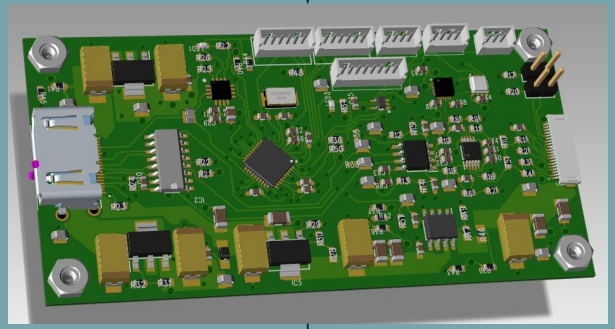


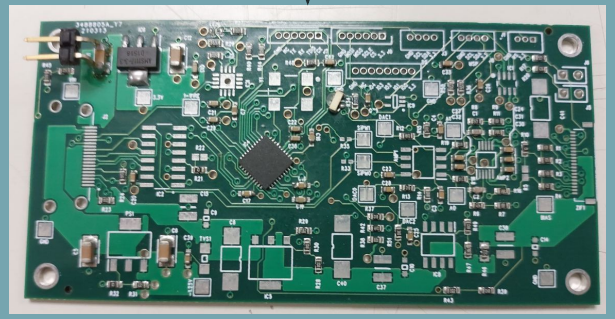
New Stations being assembled

Station for development



Version 2.0
10 Frontend manufactured



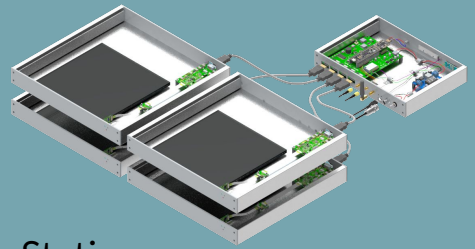


uC for display and slow control

Software

Wi-Fi Access Point

Internet



Station

All infrastructure can be deployed with 1 command only in any server

Reverse Proxy

MQTT Broker

Control Panel

Message Reconstruction

Database

Analysis Interface

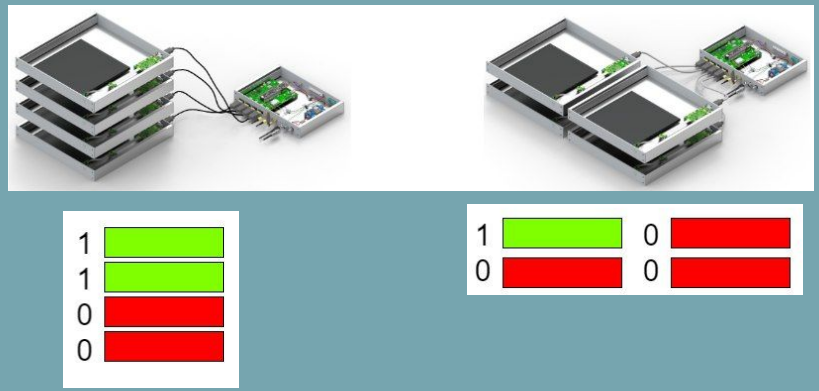
Continuous Monitoring

Data

Written to the Database:

- Event timestamp
- Trigger
- ID and station location
- Signal width
- RunNumber & CosmicBlock
- # event
- Bias and threshold voltages
- Environmental conditions
- Calibration
- GPS extra information

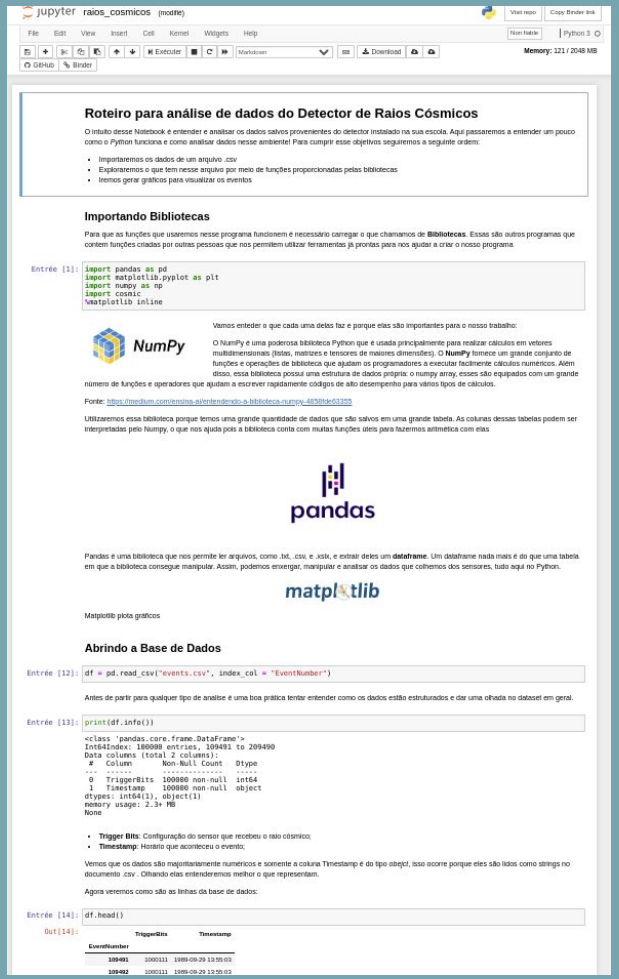
Trigger bits



Example of data being written to DB:
Which sensors triggered that event

Analysis interface

- Access via JupyterHub
- Python environment
- All usual data analysis tools
- Public data available on database



jupyter raios_cosmicos (modifié)

File Edit View Insert Cell Kernel Widgets Help

Executer Download GitHub Binder

Memory: 121 / 2048 MB

Roteiro para análise de dados do Detector de Raios Cósmicos

O intuito desse Notebook é entender e analisar os dados salvos provenientes do detector instalado na sua escola. Aqui passaremos a entender um pouco como o Python funciona e como analisar dados nesse ambiente! Para cumprir esse objetivos seguiremos a seguinte ordem:

- Importaremos os dados de um arquivo .csv
- Exploraremos o que tem nesse arquivo por meio de funções proporcionadas pelas bibliotecas
- Iremos gerar gráficos para visualizar os eventos

Importando Bibliotecas

Para que as funções que usaremos nesse programa funcionem é necessário carregar o que chamamos de **Bibliotecas**. Essas são outros programas que contem funções criadas por outras pessoas que nos permitem utilizar ferramentas já prontas para nos ajudar a criar o nosso programa

Entrée [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import cosmic
%matplotlib inline
```

NumPy

Vamos enteder o que cada uma delas faz e porque elas são importantes para o nosso trabalho:

O NumPy é uma poderosa biblioteca Python que é usada principalmente para realizar cálculos em vetores multidimensionais (listas, matrizes e tensores de maiores dimensões). O **NumPy** fornece um grande conjunto de funções e operações de biblioteca que ajudam os programadores a executar facilmente cálculos numéricos. Além disso, essa biblioteca possui uma estrutura de dados própria: o numpy array, esses são equipados com um grande número de funções e operadores que ajudam a escrever rapidamente códigos de alto desempenho para vários tipos de cálculos.

Fonte: https://medium.com/ensina-ai/entendendo-a-biblioteca-numpy-4858fde63355

Utilizaremos essa biblioteca porque temos uma grande quantidade de dados que são salvos em uma grande tabela. As colunas dessas tabelas podem ser interpretadas pelo Numpy, o que nos ajuda pois a biblioteca conta com muitas funções úteis para fazermos aritmética com elas

pandas

Pandas é uma biblioteca que nos permite ler arquivos, como .txt, .csv, e .xslx, e extrair deles um **dataframe**. Um dataframe nada mais é do que uma tabela em que a biblioteca consegue manipular. Assim, podemos envergar, manipular e analisar os dados que colhemos dos sensores, tudo aqui no Python.

matplotlib

Matplotlib plota graficos

Abrindo a Base de Dados

Entrée [12]:

```
df = pd.read_csv("events.csv", index_col = "EventNumber")
```

Antes de partir para qualquer tipo de analise é uma boa pratica tentar entender como os dados estão estruturados e dar uma olhada no dataset em geral.

Entrée [13]:

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100000 entries, 109491 to 209490
Data columns (total 2 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   TriggerBits  100000 non-null  int64
 1   Timestamp    100000 non-null  object
dtypes: int64(1), object(1)
memory usage: 2.3+ MB
None
```

- **Trigger Bits**: Configuração do sensor que recebeu o raio cósmico;
- **Timestamp**: Horário que aconteceu o evento;

Vemos que os dados são majoritariamente numéricos e somente a coluna Timestamp é do tipo *object*, isso ocorre porque eles são lidos como strings no documento .csv . Olhando elas entenderemos melhor o que representam.

Agora veremos como são as linhas da base de dados:

Entrée [14]:

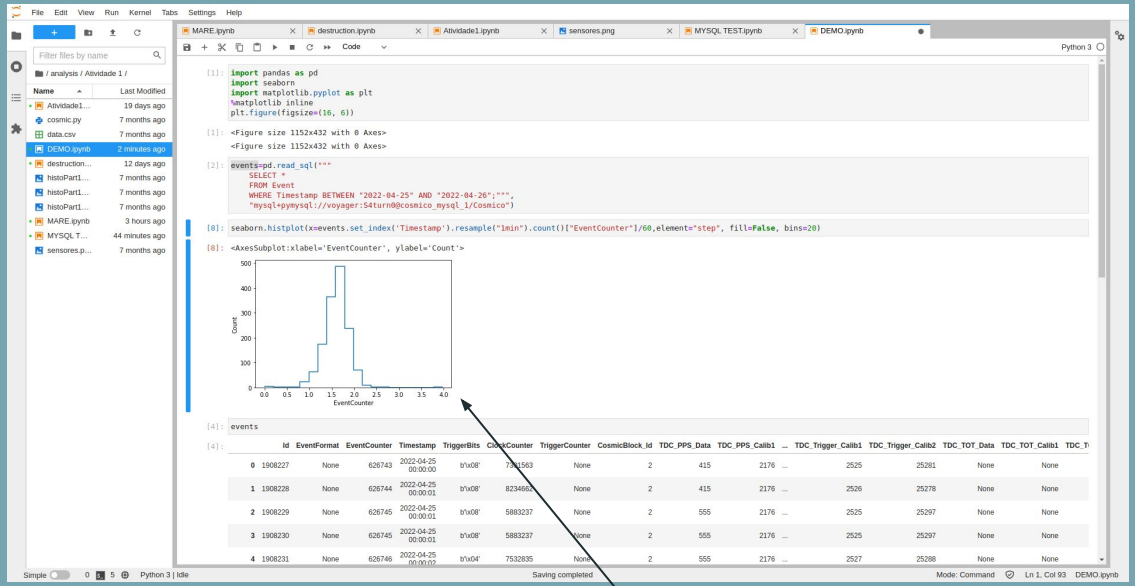
```
df.head()
```

Out[14]:

EventNumber	TriggerBits	Timestamp
109491	1000111	1989-09-29 13:55:03
109492	1000111	1989-09-29 13:55:03

Analysis

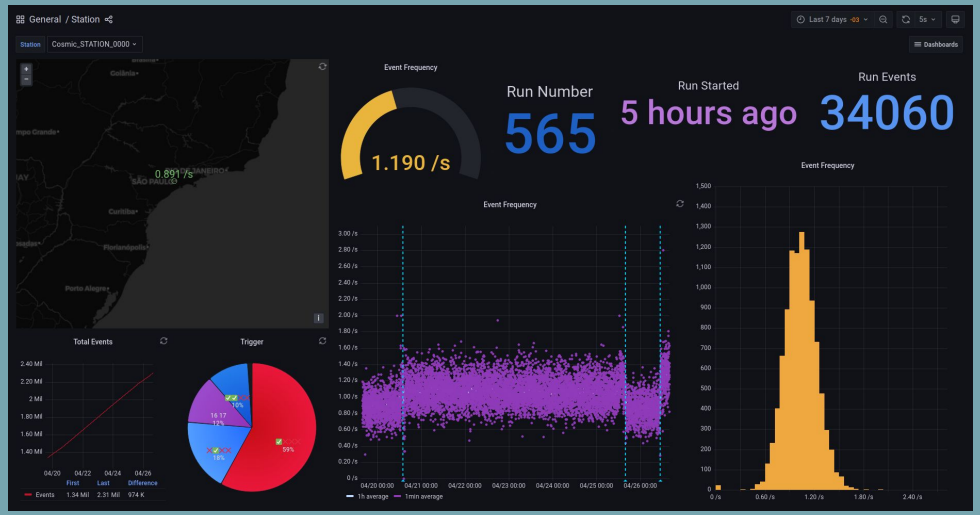
- Filtering
  - Timestamp interval
  - Position
  - Sensors
  - Trigger
  - Configuration
- Plots and histograms
  - Rate of events
- Statistics
- Stations timestamping correlation

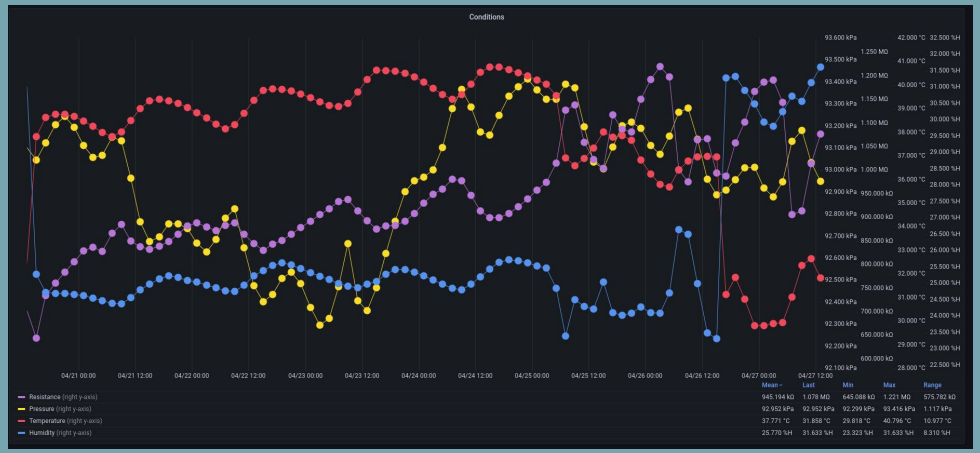


25 of April data

Monitoring

- Real time data
  - Event rate
  - Station location
  - Sensors
  - Trigger
- Parameter Control
  - SiPM Voltage
  - Threshold
  - Trigger
- Server infrastructure

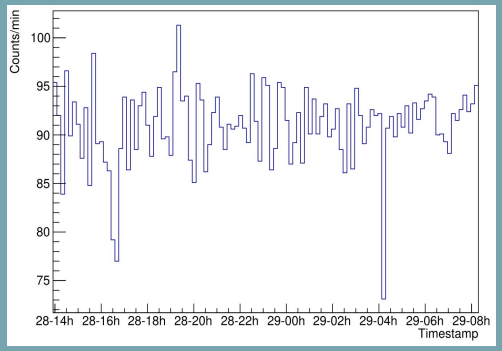




Public available on: raioscosmicos.if.usp.br/grafana

Results

Rate of events in 1 day analyzed from DB



- Functional Prototype
- Detector assembly for schools and USP Móvel
- Version 2.0 development
- Project in a mature state, with most aspects under control
- Paper about hardware/software already published on JINST

IOPscience Journals Books Publishing Support Login

Journal of Instrumentation

PAPER

A modular and flexible data acquisition system for a cosmic rays detector network

G.T. Saito[2,1], M.A.L. Leite[1], R. Menegasso[1], M.K. Kuriyama[1], M.G. Munhoz[1] and R.E. de Paula[1]

Published 21 April 2022 • © 2022 IOP Publishing Ltd and Sissa Medialab

Journal of Instrumentation, Volume 17, April 2022

Citation G.T. Saito *et al* 2022 *JINST* **17** C04026

Article PDF

2 Total downloads

Turn on MathJax

Get permission to re-use this article

Share this article

+ Article information

Abstract

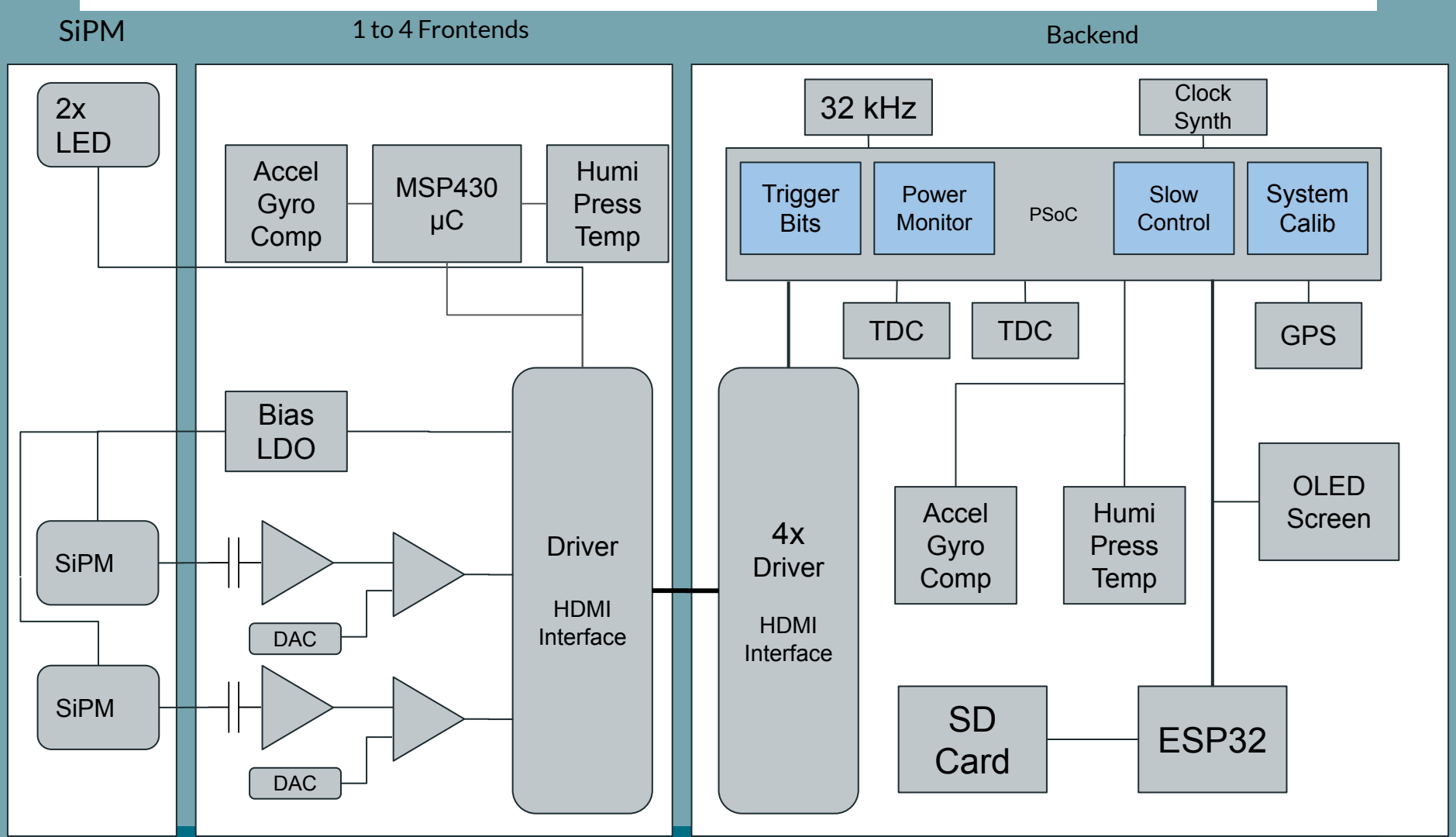
Abstract

In this paper, we describe a modular data acquisition system developed as the foundation of a cosmic ray detector network. Each detector setup (henceforth referred as a station) is composed of an independent hardware device that can be controlled and read-out through the Internet. This device is designed to acquire and process the signal of up to eight different detector planes. Each of these detector planes uses plastic scintillator slabs that are optically coupled to silicon photomultipliers (SiPM). Within a single station, different geometries and plane orientations are possible using the same baseline design. The main readout is based on a programmable system-on-a-chip (PSoC), a flexible and re-configurable commodity hardware that is used to implement the trigger and timing logic. A Time to Digital Converter (TDC) is used to determine the precise timing of the event relative to a GPS timing signal and to estimate the signal amplitude through the Time-over-Threshold (ToT) method. An auxiliary set of sensors provide environmental information and station detector planes orientation that, together with other operation data, are periodically sent to a server using the MQTT protocol. Data is cached using an in-memory database for online monitoring and further persisted into a SQL database for offline analysis. The server framework is based in software application containers allowing easy replication of the server infrastructure.

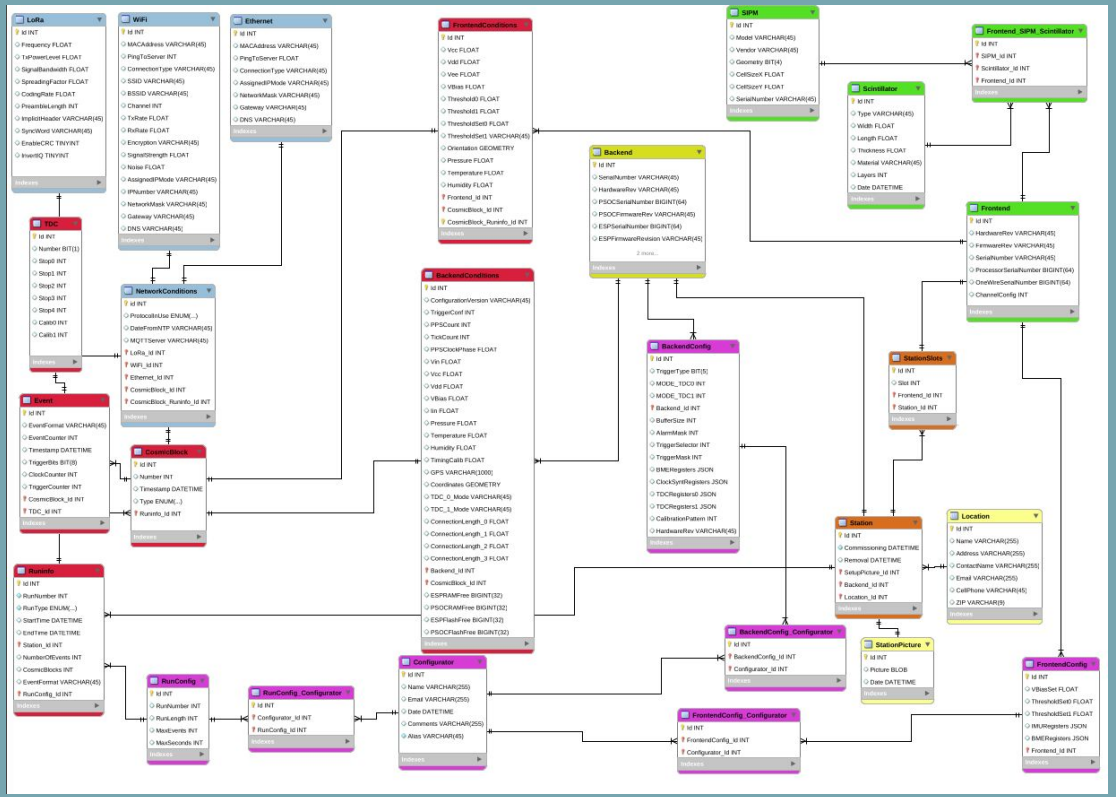
Source code and diagrams public available on gitlab.com/raioscosmicos

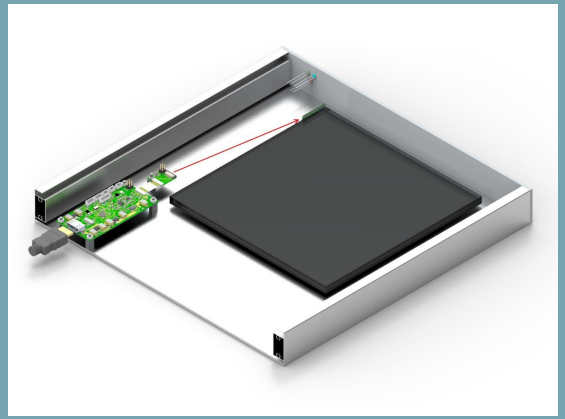
Thanks to RENAFAE for support on this project.

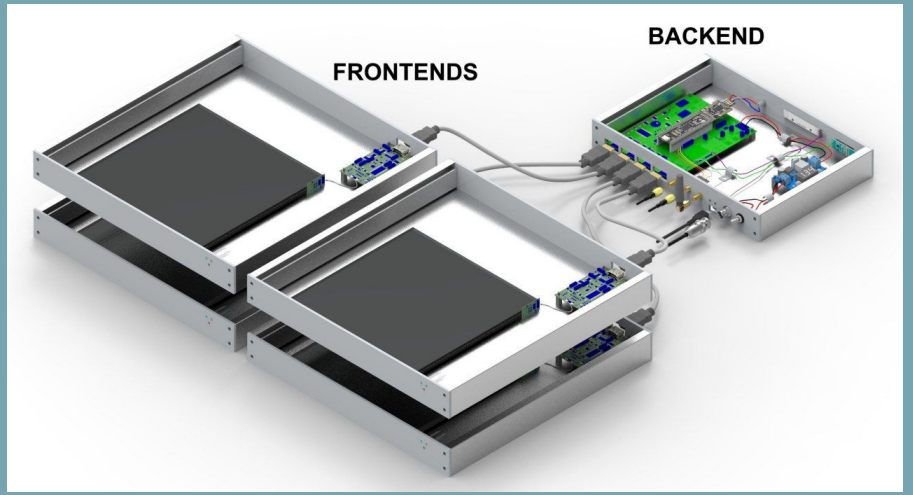
Backup


SiPM
1 to 4 Frontends
Backend
2x LED
Accel Gyro Comp
MSP430 µC
Humi Press Temp
32 kHz
Clock Synth
Trigger Bits
Power Monitor
PSoC
Slow Control
System Calib
TDC
TDC
GPS
Bias LDO
Driver
HDMI Interface
4x Driver
HDMI Interface
Accel Gyro Comp
Humi Press Temp
OLED Screen
SiPM
DAC
SiPM
DAC
SD Card
ESP32

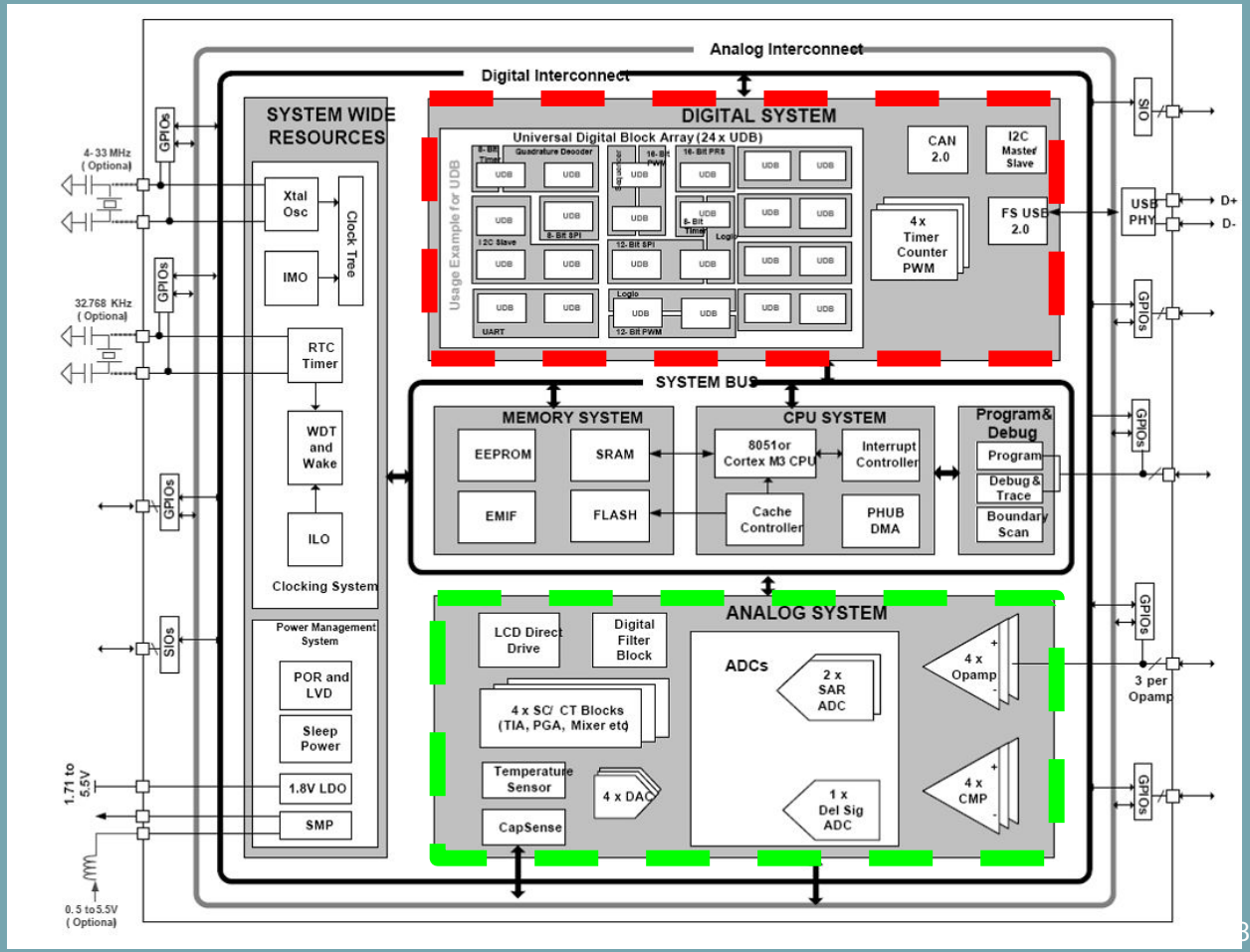
DB Schema

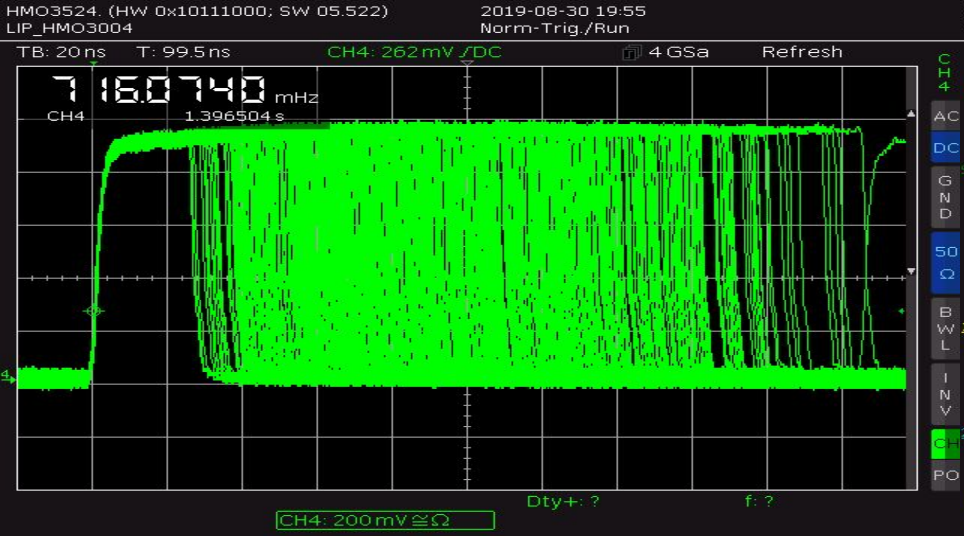

LoRa
Id INT
Frequency FLOAT
TxPowerLevel FLOAT
SignalBandwidth FLOAT
SpreadingFactor FLOAT
CodingRate FLOAT
PreambleLength INT
ImplicitHeader VARCHAR(45)
SyncWord VARCHAR(45)
EnableCRC TINYINT
InvertIQ TINYINT
WiFi
Id INT
MACAddress VARCHAR(45)
PingToServer INT
ConnectionType VARCHAR(45)
SSID VARCHAR(45)
BSSID VARCHAR(45)
Channel INT
TxRate FLOAT
RxRate FLOAT
Encryption VARCHAR(45)
SignalStrength FLOAT
Noise FLOAT
AssignedIPMode VARCHAR(45)
IPNumber VARCHAR(45)
NetworkMask VARCHAR(45)
Gateway VARCHAR(45)
DNS VARCHAR(45)
Ethernet
Id INT
MACAddress VARCHAR(45)
PingToServer FLOAT
ConnectionType VARCHAR(45)
AssignedIPMode VARCHAR(45)
NetworkMask VARCHAR(45)
Gateway VARCHAR(45)
DNS VARCHAR(45)
FrontendConditions
Id INT
Vcc FLOAT
Vdd FLOAT
Vee FLOAT
Vbias FLOAT
Threshold0 FLOAT
Threshold1 FLOAT
Threshold0to1 FLOAT
Threshold1to1 VARCHAR(45)
Orientation GEOMETRY
Pressure FLOAT
Temperature FLOAT
Humidity FLOAT
Frontend_Id INT
CosmicBlock_Id INT
CosmicBlock_RunInfo_Id INT
SiPM
Id INT
Model VARCHAR(45)
Vendor VARCHAR(45)
Geometry BIT(4)
CellSizeX FLOAT
CellSizeY FLOAT
SerialNumber VARCHAR(45)
Frontend_SiPM_Scintillator
Id INT
SiPM_Id INT
Scintillator_Id INT
Frontend_Id INT
Scintillator
Id INT
Type VARCHAR(45)
Width FLOAT
Length FLOAT
Thickness FLOAT
Material VARCHAR(45)
Layers INT
Date DATETIME
Backend
Id INT
SerialNumber VARCHAR(45)
HardwareRev VARCHAR(45)
PSOCSerialNumber BIGINT(64)
PSOCFirmwareRev VARCHAR(45)
ESPSerialNumber BIGINT(64)
ESPFirmwareRevision VARCHAR(45)
Frontend
Id INT
HardwareRev VARCHAR(45)
FirmwareRev VARCHAR(45)
SerialNumber VARCHAR(45)
ProcessorSerialNumber BIGINT(64)
OneWireSerialNumber BIGINT(64)
ChannelConfig INT
TDC
Id INT
Number BIT(1)
Stop0 INT
Stop1 INT
Stop2 INT
Stop3 INT
Stop4 INT
Calib0 INT
Calib1 INT
NetworkConditions
Id INT
ProtocolInUse ENUM(...)
DateFromNTP VARCHAR(45)
MQTTServer VARCHAR(45)
LoRa_Id INT
WiFi_Id INT
Ethernet_Id INT
CosmicBlock_Id INT
CosmicBlock_RunInfo_Id INT
BackendConditions
Id INT
ConfigurationVersion VARCHAR(45)
TriggerConf INT
PPSCount INT
TotCount INT
PPSClockPhase FLOAT
Vin FLOAT
Vcc FLOAT
Vdd FLOAT
Vbias FLOAT
Iin FLOAT
Pressure FLOAT
Temperature FLOAT
Humidity FLOAT
TimingCalib FLOAT
GPS VARCHAR(1000)
Coordinates GEOMETRY
TDC_0_Mode VARCHAR(45)
TDC_1_Mode VARCHAR(45)
ConnectionLength_0 FLOAT
ConnectionLength_1 FLOAT
ConnectionLength_2 FLOAT
ConnectionLength_3 FLOAT
Backend_Id INT
CosmicBlock_Id INT
ESPRAMFree BIGINT(32)
PSOCRAMFree BIGINT(32)
ESPFlashFree BIGINT(32)
PSOCFlashFree BIGINT(32)
BackendConfig
Id INT
TriggerType BIT(5)
MODE_TDC0 INT
MODE_TDC1 INT
Backend_Id INT
BufferSize INT
AlarmMask INT
TriggerSelector INT
TriggerMask INT
BMERegisters JSON
ClockSyntRegisters JSON
TDCRegisters0 JSON
TDCRegisters1 JSON
CalibrationPattern INT
HardwareRev VARCHAR(45)
StationSlots
Id INT
Slot INT
Frontend_Id INT
Station_Id INT
Event
Id INT
EventFormat VARCHAR(45)
EventCounter INT
Timestamp DATETIME
TriggerBits BIT(8)
ClockCounter INT
TriggerCounter INT
CosmicBlock_Id INT
TDC_Id INT
CosmicBlock
Id INT
Number INT
Timestamp DATETIME
Type ENUM(...)
RunInfo_Id INT
Station
Id INT
Commissioning DATETIME
Removal DATETIME
SetupPicture_Id INT
Backend_Id INT
Location_Id INT
Location
Id INT
Name VARCHAR(255)
Address VARCHAR(255)
ContactName VARCHAR(255)
Email VARCHAR(255)
CellPhone VARCHAR(45)
ZIP VARCHAR(8)
RunInfo
Id INT
RunNumber INT
RunType ENUM(...)
StartTime DATETIME
EndTime DATETIME
Station_Id INT
NumberOfEvents INT
CosmicBlocks INT
EventFormat VARCHAR(45)
RunConfig_Id INT
RunConfig
Id INT
RunNumber INT
RunLength INT
MaxEvents INT
MaxSeconds INT
RunConfig_Configurator
Id INT
Configurator_Id INT
RunConfig_Id INT
Configurator
Id INT
Name VARCHAR(255)
Email VARCHAR(255)
Date DATETIME
Comments VARCHAR(255)
Alias VARCHAR(45)
BackendConfig_Configurator
Id INT
BackendConfig_Id INT
Configurator_Id INT
StationPicture
Id INT
Picture BLOB
Date DATETIME
FrontendConfig_Configurator
Id INT
FrontendConfig_Id INT
Configurator_Id INT
FrontendConfig
Id INT
VBiasSet FLOAT
Threshold0Set FLOAT
Threshold1Set FLOAT
IMURegisters JSON
BMERegisters JSON
Frontend_Id INT

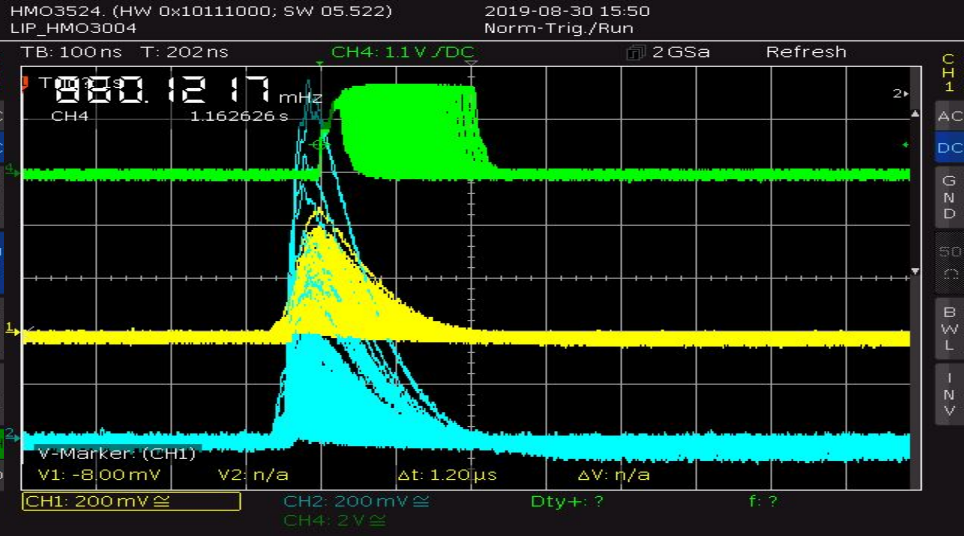


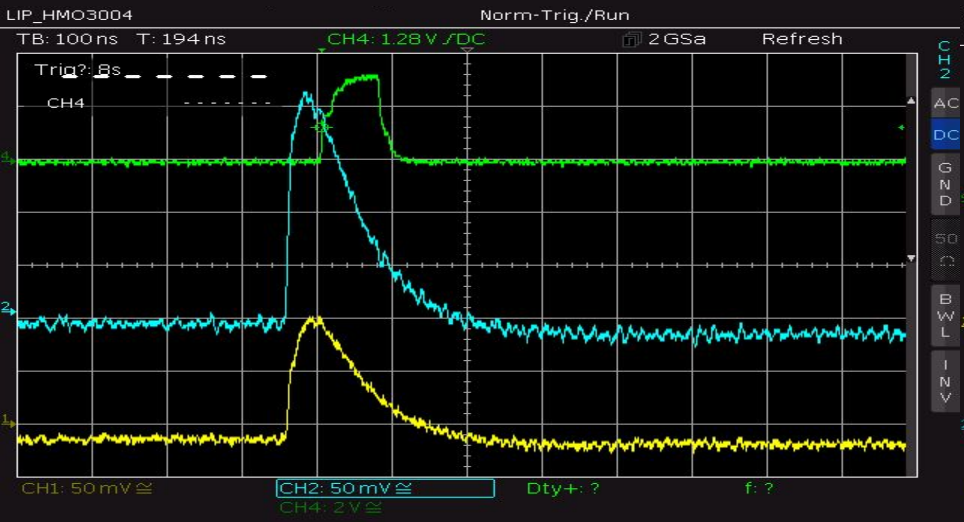

FRONTENDS
BACKEND

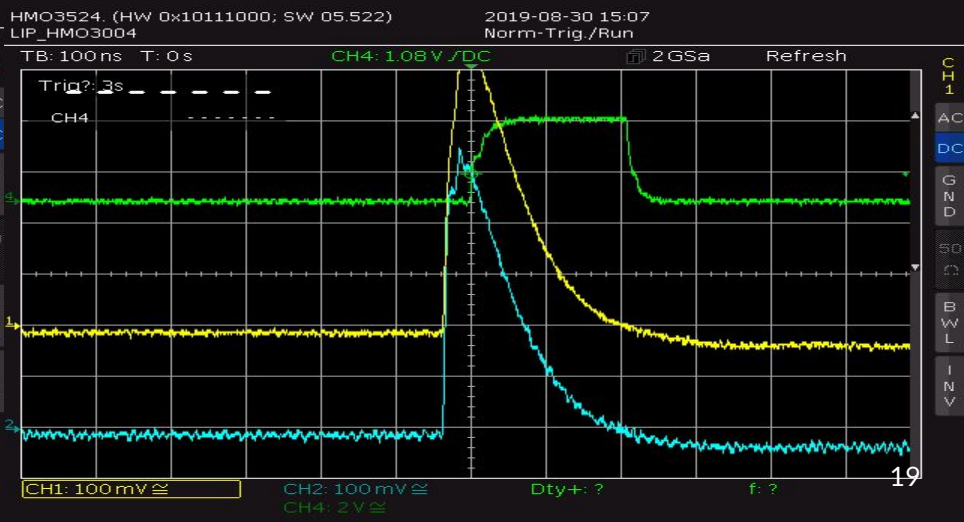
Baseado em um Cypress PSoC 5L




HMO3524, (HW 0x10111000; SW 05.522)
LIP_HMO3004
2019-08-30 19:55
Norm-Trig./Run
TB: 20 ns T: 99.5 ns
CH4: 262 mV /DC
4 GSa
Refresh
716.0740 mHz
CH4
1.396504 s
CH4: 200 mV≅Ω
Dty+: ?
f: ?


HMO3524, (HW 0x10111000; SW 05.522)
LIP_HMO3004
2019-08-30 15:50
Norm-Trig./Run
TB: 100 ns T: 202 ns
CH4: 1.1 V /DC
2 GSa
Refresh
880.1217 mHz
CH4
1.162626 s
V-Marker: (CH1)
V1: -8.00 mV
V2: n/a
Δt: 1.20 µs
ΔV: n/a
CH1: 200 mV≅
CH2: 200 mV≅
CH4: 2 V≅
Dty+: ?
f: ?


LIP_HMO3004
Norm-Trig./Run
TB: 100 ns T: 194 ns
CH4: 1.28 V /DC
2 GSa
Refresh
Trig?: 8s
CH4
CH1: 50 mV≅
CH2: 50 mV≅
CH4: 2 V≅
Dty+: ?
f: ?


HMO3524, (HW 0x10111000; SW 05.522)
LIP_HMO3004
2019-08-30 15:07
Norm-Trig./Run
TB: 100 ns T: 0 s
CH4: 1.08 V /DC
2 GSa
Refresh
Trig?: 3s
CH4
CH1: 100 mV≅
CH2: 100 mV≅
CH4: 2 V≅
Dty+: ?
f: ?

Product Overview

Digi-Key Part Number	516-4280-ND
Quantity Available	**268** Can ship immediately
Manufacturer	Broadcom Limited
Manufacturer Part Number	AFBR-S4N44C013
Description	SENSOR OPT 420NM UV
Manufacturer Standard Lead Time	12 Weeks
Detailed Description	Optical Sensor Ultraviolet (UV) 420nm

Price & Procurement

Quantity 1

516-4280-ND

Customer Reference

Add to Cart

All prices are in USD.

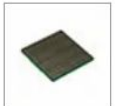
Price Break	Unit Price	Extended Price
1	26.41000	$26.41
10	23.33500	$233.35
100	17.80830	$1,780.83
500	16.33452	$8,167.26

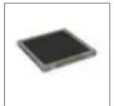
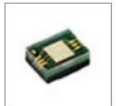
Submit a request for quotation on quantities greater than those displayed.

Documents & Media

Datasheets	AFBR-S4N44C013 Datasheet
Featured Product	AFBR-S4N44C013 NUV-HD Silicon Photomultiplier (SiPM)
Online Catalog	AFBR-S4N44C013

You May Also Be Interested In



TDC7200 (ACTIVE)

In English | Alert me

Time-to-digital converter for time-of-flight (ToF) applications for LIDAR and ultrasonic

DATASHEET

TDC7200 Time-to-Digital Converter for Time-of-Flight Applications in LIDAR, Magnetostrictive and Flow Meters datasheet (Rev. D)

View now | Download

Top purchased products for TDC7200

Part number	Buy from TI store	TI store inventory	Price | QTY	Package | Pins
TDC7200PW	Add to cart	3791	0.96 | 1ku	TSSOP (PW) | 14
TDC7200PWR	Add to cart	6236	0.80 | 1ku	TSSOP (PW) | 14

View all (2) >

Description & parametrics | Technical documents | Design & development | Order now | Quality & packaging | Support & training

Description | Features | Diagram | Design resources for you | Parametrics | Complete your design

Description

The TDC7200 is a Time-to-Digital Converter (TDC) for ultrasonic sensing measurements such as water flow meter, gas flow meter, and heat flow meter. When paired with the TDC1000 (ultrasonic analog-front-end), the TDC7200 can be a part of a complete TI ultrasonic sensing solution that includes the MSP430, power, wireless, and source code.

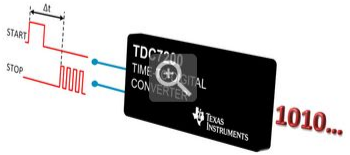
The Time to Digital Converter (TDC) performs the function of a stopwatch and measures the elapsed time (time-of-flight or TOF) between a START pulse and up to five STOP pulses. The ability to measure from START to multiple STOPs gives users the flexibility to select which STOP pulse yields the best echo performance.

View more

Features

- Resolution: 55 ps
- Standard Deviation: 35 ps
- Measurement Range:
  - Mode 1: 12 ns to 500 ns
  - Mode 2: 250 ns to 8 ms
- Low Power Consumption: 0.5 µA (2 SPS)
- Supports up to 5 STOP Signals
- Autonomous Multi-Cycle Averaging Mode for Low

Diagram



TDC7200 - Functional Diagram

Outras possibilidades

- Cherenkov
- Controlar elevação e direção
- Tempo de vida do muon
- Mais que 8 planos de detecção
- Correlacionar **E** da Terra