ICARUS connectivity test

Brenda Gómez [*a], Angela Fava [†b], Mark Convery [‡c], Gianluca Petrillo [§c], Yun-Tse Tsai [¶c], Somebody else [‖z], and adding the others is a to-do [**z]

[a]Universidad de Colima
[b]Fermi National Accelerator Laboratory
[c]SLAC National Accelerator Laboratory
[z]Someone's institution

January 2, 2019

Abstract

We present here the procedure and results of the connectivity tests done on the TPCs wire planes of the ICARUS detector. All three, Induction-1, Induction-2 and Collection planes were tested. A test pulse was injected to each wire and an output signal was obtained and measured. An off-site analysis of the output signal waveforms was carried out for all standard and non-standard chimneys. As a result of the tests, we present a detailed description and mapping of how the detector is done from inside.

Contents

[*]gomez.cortes14@gmail.com
[†]afava@fnal.gov
[‡]convery@slac.stanford.edu
[§]petrillo@slac.stanford.edu
[¶]yuntse@slac.stanford.edu
[‖]somebody@institution.edu
[**]of course

1 Introduction

Here goes an introduction to the document with a description of the structure and the content of each section. This may be the last part to be written, i.e. after all sections are completed.

2 T600 Detector – TPC (Time Projection Chamber)

detector done from inside.....

3 Wire Connectivity Tests

The main goal of the ICARUS TPC's connectivity tests here described has been to check the condition of the wires of the each anode plane of the detector; i.e., to look/check for wires not properly working or even disconnected.

The tests, as described in section 3.2, were done using a test box that sends a pulse of specific voltage, triggers the oscilloscope and receives back an output signal from the wires and sends it to oscilloscope where we can look and measure the expected peak and amplitude.

Since horizontal cables can't get a pulse injected, there was a slightly different procedure to test the 8 non-standard chimneys (See section 3.3).

3.1 Test box

The SLAC test box (fig. 1) was designated specially for the ICARUS TPC wire plane connectivity tests. It's a portable battery powered device with dimensions $20 \times 15 \times 10$ cm.

Figure 1: SLAC test box version 1

On the lid it contains:

- Built in pulser outputs on LEMO with both polarities,
- Connector for a single flat ribbon of twisted-pairs cable: TPC cables plug into this connector with all grounds connected together.

- An 8-position rotary switch that enables 4 channels at a time,
- The test pulse is a square wave of amplitude roughly 4.8 V, 100 µs long and with 1 µs rise time, emitted at a rate of 100 Hz (see section 3.1.1).

Two test boxes were designed by Mark Convery and produced at SLAC. The cable shields, although connected together, were not grounded to the box by design, not to disturb the grounding of the cable shields in the TPC. The consequences of this feature, in conjunction with the accidental grounding of wires to the detector ground in ICARUS TPC, is described in section 4.3. The second version of the box contains extra outputs that allowed us to ground the shields of the connectors (Fig. second box).

3.1.1 The test pulse

The test box emits a test pulse in the form of a square wave about 100 times every second (100 Hz). The test box is powered with three 1.5 V AA batteries, and no voltage regulator is used, resulting in the pulse being as high as the power batteries allow. This results to be roughly 4.8 V and, as described in section 4.3, subject to variations with time and the remaining charge in the batteries. The baseline of the pulse is set to half the maximum amplitude.

When connected to the TPC test capacitance, the derivative of the test signal is observed on the channels, resulting in a sharp negative dip followed after 100 µs by a sharp positive rise; both of them "decay" following a fast exponential trend (decay time τ is roughly 20 µs). These signals are on a baseline of about 2 V, and can reach an amplitude as large as 1 V. The time offset of the oscilloscope trigger is set manually on the oscilloscope itself so that the first peak happens around the middle of the acquired time window (usually around 400 µs).

3.2 Standard Chimneys

The TPC cables were pulled out from the detector through all the chimneys (Fig.) and the tests were performed in the following way:

Procedure:

1. The external trigger was connected from one of the test-pulse output of the box to the oscilloscope (at the back). Therefore, the setting for the trigger in the oscilloscope input was changed to: "EXT" (external).
2. 4 LEMO cables connected from the box to the four inputs of the oscillope correspond to four channels of the signal (Fig.).
3. A SMA-female coaxial cable goes connected on the "pulse-out" of the box via LEMO conection and the SMA-female side is connected to the pulser from the detector.

An important requirement before starting to take data was to test all 4 SMA cables (red tag) in the search for the highest pulse.

In order to verify that everything was working properly, the box was tested in-site by connecting the test-pulse output directly to one channel of the oscilloscope, with the settings changed to DC and 1 MΩ and looked for the 5 V square signal on the screen. In addition, we verified the SMA (from the detector) to LEMO (pulser cable from the box) connection by using a chain of LEMO to SMA-female plus SMA-male to LEMO to connect the test box to the oscilloscope.

4. For a systematic test, we chose to start from the cable S/V18 to S/V1. Since the test box injects and receives output signal from only 4 cables of a single flat ribbon connector, we had to swith through 8 positions with the black know on top of the box in order to test all 32 wires from 1 connector at a time.

5. The waveforms seen on the screen of the oscilloscope were downloaded using a Python script written by S. Castells [ref] and stored in an external disk for an offsite analysis.

Figure 2: Set up of the hardware components. The red circle shows the SMA to LEMO connectors through which the box injects the test pulse.

3.3 Non-standard chimneys

The non-standard chimneys are the 8 corners of both cryostats, see Fig. Apart from the difference in size between standard and non-standard chimneys, the latter ones contain the horizontal and corner wire connectors.

As described on section 2 the configuration of the horizontal wires does not allow to inject a pulse directly to the wires and therefore a different way to test them was developed.
The test pulse was injected to 32 wires of Induction-2 plane at once. From one standard chimney, two or three chimneys away from the corner, a connector between 2 and 9 was chosen and plugged in to a board (¡-name?). To pulse those 32 wires we use the same connection from the test box: a chain LEMO to SMA cables and to one of the 4 red-tagged pulser cables from the chimney.

3.4 Data acquisition

The test box output was monitored and sampled via Tektronix TDS 3054C oscilloscope. This oscilloscope can digitise and send out the input signals, providing 10^4 samples per channel, each sample with 8-bit precision[1]. The data acquisition code, written by Sergi Castells, reads a sequence of waveforms from each oscilloscope channel. Each waveform is stored in its own comma-separated value file (CSV), with a resolution is of 0.1 µs for the time and 10 mV for the signal amplitude. Therefore the waveform samples span 1 millisecond, and they are at baseline in roughly 60-80% of the time.

The testing procedure included cycling across all 8 switch positions of the test box for each connection, and recording 10 waveforms for each of the 4 channels monitored in the selected position. For each switch position, therefore, 40 waveforms are recorded. The total data size as stored in the final form is about 500 MiB per chimney.

3.5 Data sets

The original acquisition software by Sergi Castells grants the operator some options that affect the name of the waveform files. An acquisition-driving code was developed that reduces those options and enforces a file name pattern:

```
waveform_CH<channel>_CHIMNEY_<chimney>_CONN_<connector>_POS_<position>_<index>.csv
```

where:

- `<channel>` is the *oscilloscope channel* the waveform is read from, which ranges from `1` to `4`
- `<chimney>` is in the form `<module><row><index>`, with both module and row denoted by geographical position as `E` (east) or `W` (west), and the chimney number being between `01` and `20`; for example, `WE07`
- `<connector>` is the name of the connector plugged in the test box, made of a letter (`S` or `V`) and the zero-padded connection number in the range from `01` to `18`, e.g. `V05`.
- `<position>` is the setting of the test box switch selecting four channels among the 32 on the connector; it ranges from `1` to `8`
- `<index>` is a waveform counter; following Sergi's convention, this is a sequential number unique within waveforms on the same connector and oscilloscope channel. Assuming that 10 waveforms are taken per channel, position `POS_1` will have indices from `1` to `10`, `POS_2` from `11` to `20`, and so on, up to `POS_8` with indices `71` to `80`.

320 waveform files are expected per connector (10 waveforms for each of 32 channels, split in 4 oscilloscope channels and 8 test box switch positions), and 5760 per chimney. These files are stored in separate directories, one per chimney. The name of the directory follows the same convention as above, that is:

[1] The oscilloscope alternatively allows for 9 bit ADC conversion, at the cost of doubling

- CHIMNEY_<chimney> in the form <module><row><index>, with both module and row denoted by geographical position as E (east) or W (west), and the chimney number being between 01 and 20; for example, CHIMNEY_WE07

The whole dataset taken in the test session of August 2018 is currently stored in Fermilab dCache at the path:

```
/pnfs/icarus/persistent/commissioning/connectivity/201808
```

The script `checkChimneyFiles.sh` (currently stored in `/icarus/data/commissioning/connectivityTest` checks the presence of all the expected files for each chimney directory.

3.6 Data format

Each waveform is stored in its own comma-separated value file (CSV). The file name follows the pattern described above. Each line in the file corresponding to a sample, totaling 10000 lines for each waveform. Each sample is described by values in floating point text representation, with scientific notation: time in seconds, and signal level in volt. For example:

```
0.0,2.08
1e-07,2.08
2e-07,2.12
[...]
```

No metadata is saved in the file. This data format can be easily read directly by ROOT `TGraph`.

3.7 Streamlined interactive data acquisition

The original code by Sergi Castells required the operator to change parameters on the command line on each acquisition, plus an additional step at the end of the test each connector, that is to reset a counter stored in a text file.

After a few interactions with this approach, we wrote additional software (*test driver*) automating most of the steps required from the operator. This enforced a specific procedure both for testing and for software setup.

The procedure comprises testing one chimney at a time, and proceeding from connection 18 (V18 or S18) down to 1 (V01 or S01), and on each connector test the channel switching the test box to the position from 1 to 8 in strict sequence. The whole sequence must be strictly followed.

More details about the software used for data acquisition and its setup can be found in appendix A.

4 Waveform analysis

A fast data analysis was performed within hours or days from the data acquisition. Each waveform is analysed individually, through a few steps:

- baseline identification: average of the central 50% of the samples
- RMS: RMS of the samples used in the baseline determination
- extrema determination: amplitude of the positive and negative peaks along the whole waveform

For each channel, the parameters from all the available waveforms (10) are averaged to extract:

- baseline RMS (RMS of the 10 baselines)
- noise average (average of the 10 RMS, one for of each baseline)
- peak amplitude average and its uncertainty (the positive peak from each waveform are averaged)

This analysis is tuned for speed, and it can definitely be refined.

4.1 Baseline and RMS identification

The principle is to exclude the signal, intended as the response to the pulse, and to use the remaining of the waveform to estimate the baseline. The algorithm is designed not to require prior knowledge of the position or shape of the peaks.

The signal is shaped as two sharp-rising peaks, which including the decay tails are roughly 200 µs or shorter. The algorithm relies on the assumption that those two peaks are indeed roughly that narrow, although it does not assume anything on their size or shape. It flows as follows:

1. sort all the 10000 samples in increasing order; in this way, the samples taken during the negative peak will be at the left of the data, the ones at the positive peak will be at its right, and the middle of the data will be populated by long sequences of samples with similar values, from the baseline
2. select the samples from #2500 to #7499 from the sorted data for further analysis
3. take the average and RMS of the selected samples, which will represent the baseline and noise respectively

This algorithm may present some bias if the tail from the first peak overlaps the second peak. An alternative, simple algorithm would rely on the assumption that the peak is on the second half of the waveform, and would therefore use the first 40% or 50% of the waveform for baseline determination as in the last point of the algorithm.

An older version of the algorithm would rely of the knowledge of the peak position, and would fail when the waveform does not contain any actual signal.

4.2 Peak determination

Again a simple algorithm, the peak determination consists of two simple steps:

1. determine the absolute maximum and minimum of the waveform

2. subtract to both the baseline (obtained separately)

This algorithm is affected by the noise on the waveform, which can be quantified together with the baseline. Although this is less than ideal, the actual noise has been measured to be low enough to be not significant for the purpose of the fast analysis we performed[2].

4.3 Results

We summarize the results of the baseline, RMS, positive peak value, and the ratio of the positive peak value to the baseline in Figures 3 to 6. Each strip represents a single channel of the detector. The x-axis includes all the chimneys of the ICARUS detector, while the y-axis shows all the 18 cables, each with 32 channels, within a chimney. The color code in each figure indicates the values of the baseline, the RMS, the positive peak, or the ratio of the positive peak value to the baseline value.

The corner chimneys, Ch01 and Ch20 in each row (section 3.3), have only the channels connecting to horizontal wires, where the data was not recorded in this round of the connectivity test and therefore is not shown in the plots. Cables 1-9 in Ch02 and Ch03 and cables 10-18 in Ch18 and Ch19 are connected to the corner wires in the second induction and collection planes which are shorter in length. Owing to their ending points, the cables injecting the test pulse are not located at the same chimney. Without the information of the location of the pulse-injecting cables, we are not able to obtain a result of these wires, and Figures 5 and 6 show extremely low or overflown values.

In chimneys EE13, EE14, and EE17, the white strips in Figure 3 and the dark blue strips in Figures 4 and 5, indicating extremely low values, appear in every four channels. It is owing to the fact that the fourth channel in the oscilloscope happens to be set a larger scale in voltage, and, as a consequence, the relative values of the baseline and the pulse peak are small. The effect is canceled out in Figure 6, where the ratio of the peak to baseline values is plotted.

The fifth channel of the connector for the flat ribbon cables in a test box is damaged, and we are not able to read out the signals from the channel when using this test box. It thereby shows the dark blue strips in chimneys EE10-14, EW 03-07, WE02-12, and WW03-12 in Figures 5 and 6. Nonetheless, this round of the connectivity test aims to identify potential problems in terms of cables,

[2] In more recent versions of the library, the minimum and maximum are not evaluated from the single sample, but replaced by a running window average on 5 samples.

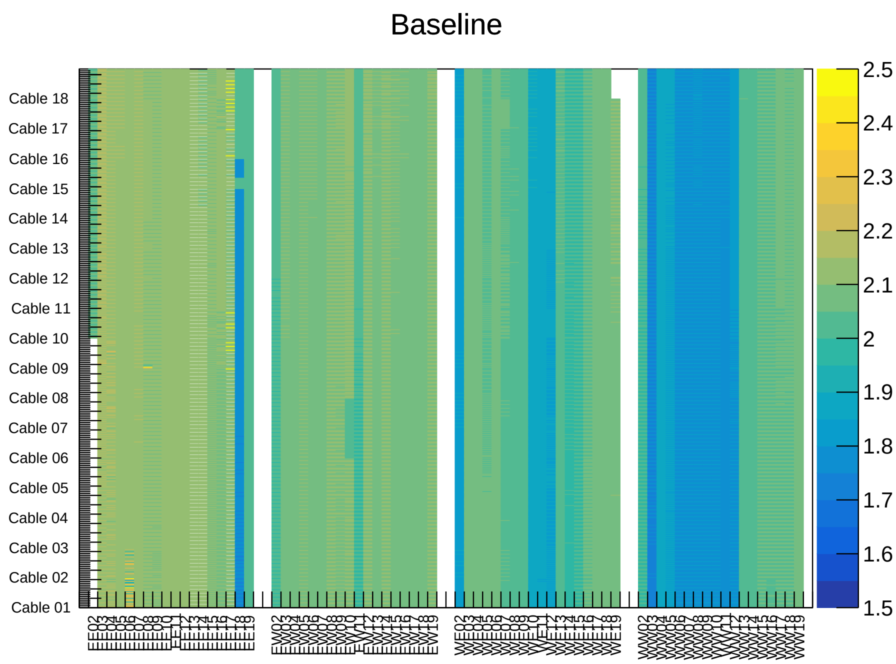

Figure 3: The map of the baseline values of all the channels. Each strip represents a single channel of the detector. The x-axis includes all the chimneys of the ICARUS detector, while the y-axis shows all the 18 cables, each with 32 channels, within a chimney. The color code in each figure indicates the values of the baseline.

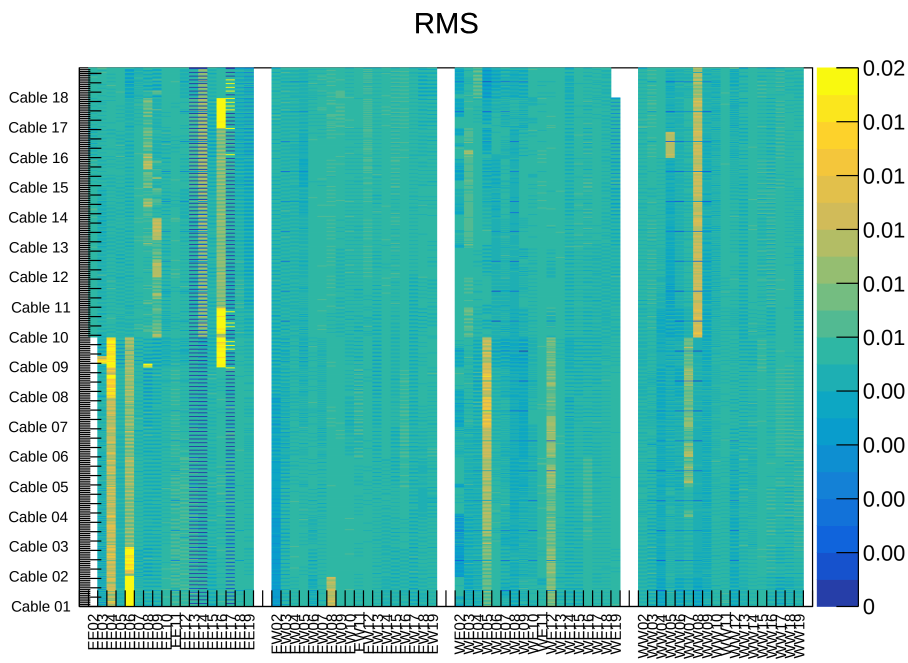

Figure 4: The map of the RMS of all the channels. Each strip represents a single channel of the detector. The x-axis includes all the chimneys of the ICARUS detector, while the y-axis shows all the 18 cables, each with 32 channels, within a chimney. The color code in each figure indicates the values of the RMS.

and a single damaged channel in the test box should not affect its effectiveness.

Since there is no voltage regulator for the battery powering the test boxes, the baseline value, which is set to the half of the voltage, decreases as the time being, as shown in Figure 3. Hence, Figure 6, which cancels out the baseline decrease, gives us a more uniform comparison across the whole detector.

In the EE and WE rows, cables 1-8, cable 9, cables 10-17, and cable 18 have separate pulse-injecting cables, while in the EW and WW rows, those groups are cable 1, cables 2-9, cable 10, and cables 11-18. During the tests, we did not have certain information on this grouping of wires, except that cables 1-9 and 10-18 are for induction and collection wires, respectively, and have different pulse-injecting cables. Consequently, for some chimneys, cables 1-9 are tested with the same pulse-injecting cable, and so are cables 10-18; for other chimneys, cables 1-8, 9, 10-17, and 18 are tested with different pulse-injecting cables. With the cross talk effect discussed in the next paragraph, it is not straightforward to distinguish whether the measured signal corresponds to the injected pulse or the cross talk signal in the special cables, cables 9 and 18 in the EE and WE chimneys and cables 1 and 11 in the EW and WW chimneys. In addition, it is raised that the grouping of the channels for the same pulse-injection cables might be cables 1-8, 9, 10-17, 18 for the chimney EE02, but cables 1, 2-9, 10-17, 18 for the next chimney, EE03, and similar patterns for other chimneys. However, we have not reached a conclusive statement by analyzing the current data, which convolved several issues including the cross talk, potentially wrong cables for pulse-injection, etc. We defer the conclusion to the next iteration of connectivity test where we expect to have equipment disentangling those issues.

The wires involved in the connectivity test should normally not be connected to the detector ground, but it was observed during the overhaul at CERN that some of them are. The design of the test box in this iteration does not connect the wire cables to the detector ground, and therefore in the normal case of the wire grounding, that is ground wires *not* being connected to the detector ground, the cross talk from the adjacent twisted wires contributes a significant portion to the signal. Roughly speaking, the amplitude of the cross talk signal is similar to to that of the injected pulse, which severely complicates the signal interpretation. In the normal case, where the wire is correctly connected but not grounded to the detector, the signal will be the superposition of the injected pulse and the cross talk signal. On the other hand, in the cases that the wire is not connected and not grounded, and that the wire is connected and also grounded, the signal will contain, respectively, only the cross talk and only the injected pulse, both of which will have roughly half the amplitude of the signal in the normal condition. By analyzing all the possible scenarios listed in Table 1, we ensure the connectivity of the channels with the full size amplitude in Figure 6. Further, to distinguish the connected wires among those with the half size amplitude, such as cable 8 in chimney WW06, cables 4 and 17 in chimney WW12, we explicitly check their grounding. The check confirms those

	Shield grounded	Shield isolated
Cable connected	1/2 size signal	Full size signal
Cable disconnected	No signal	1/2 size signal

Table 1: The qualitative signal amplitudes of all the possible scenarios of shield grounding and wire connection.

wires are grounded and we are reading the injected pulse but not the cross talk from adjacent wires.

While the a grounded, connected channel and the one of a disconnected channel both yield a half-amplitude signal, they have different characteristics which can in principle be used to resolve them. The idea is that the signal measured on a "standard" channel is compound of its own pulsed signal, accounting according to table 1 for half the amplitude, and of the signal induced by the cross talk. This induced signal is contributed half from channels at one side of the considered channel, which therefore contribute one forth of the amplitude, and half from the channels on the other side, supplying the remaining one forth of the signal. Leveraging this model, it can be conceived that if a channel is connected and grounded, in which case we observe on it half-amplitude signal from its own pulse, the grounded wire acts as a shield, and therefore does not induce crosstalk into adjacent channels, but rather suppresses crosstalk. This would suggest that the adjacent channels are induced cross talk only from one side, and their total signal would be their own pulsing signal plus the cross talk from only one of the two sides, making up for a signal of three quarters of the amplitude. On the other hand, if the channel is disconnected (or, "dead"), it is free, it acts like an antenna capturing the half-amplitude signal, but neither contributes cross talk nor it shields it: for the adjacent channels, it simply does not exist. In that case, the adjacent channels will receive the full cross talk from the channels beyond the dead one. The expectation would be to have almost full amplitude, barred second order effects. In practice, though, it proved very hard to resolve these two cases from the available data, in part because the amplitude fluctuations make the decision arbitrary, and in part because, as reported later, we did not discover any dead channel to test the hypothesis. This model is mildly corroborated by the observation that the first channel in each cable has a slightly lower amplitude, but that is in no way close to the postulated 25%. Also, no evidence of that effect is visible on the *last* channel of the cables. It should be noted though that the last channel does have adjacent cables on both sides, since the cable ribbon hosts 34 wire pairs, and this complicates the comparison. There are hints suggesting the shielding effect on channels adjacent to a grounded, connected one, but the observed decrease is in no way conclusive. As a consequence, we could not, with the simple analysis we performed, confirm or repel this model, which remains just a suggestive hypothesis.

Figure 6 shows slight non-uniformity in the signal amplitude of some cables, such as cables 10-13 in chimney EE09. Looking into the values of those cables, we clarify those cables do not have particularly non-uniform signal amplitudes,

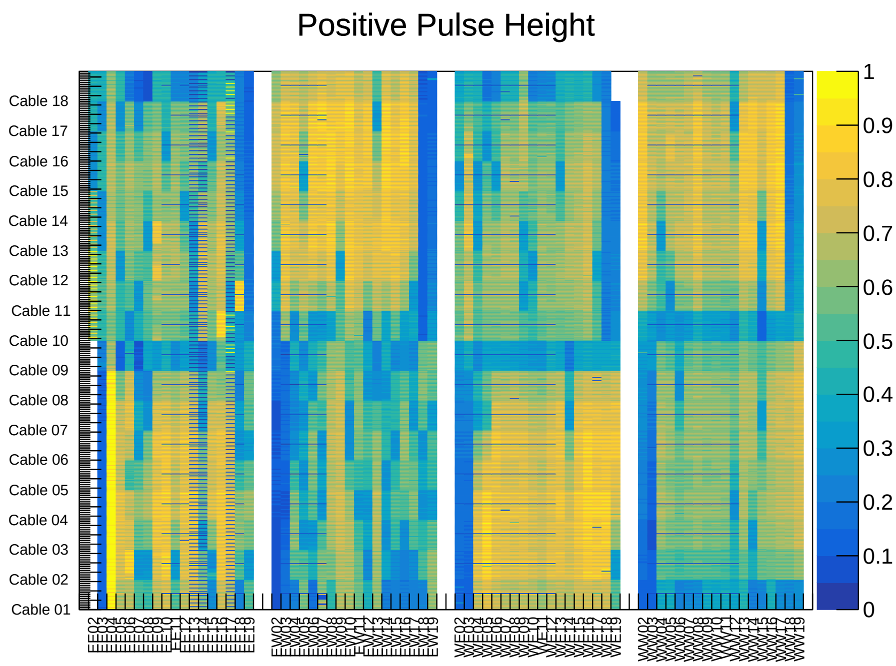

Figure 5: The map of the positive peak values of all the channels. Each strip represents a single channel of the detector. The x-axis includes all the chimneys of the ICARUS detector, while the y-axis shows all the 18 cables, each with 32 channels, within a chimney. The color code in each figure indicates the values of the positive peak.

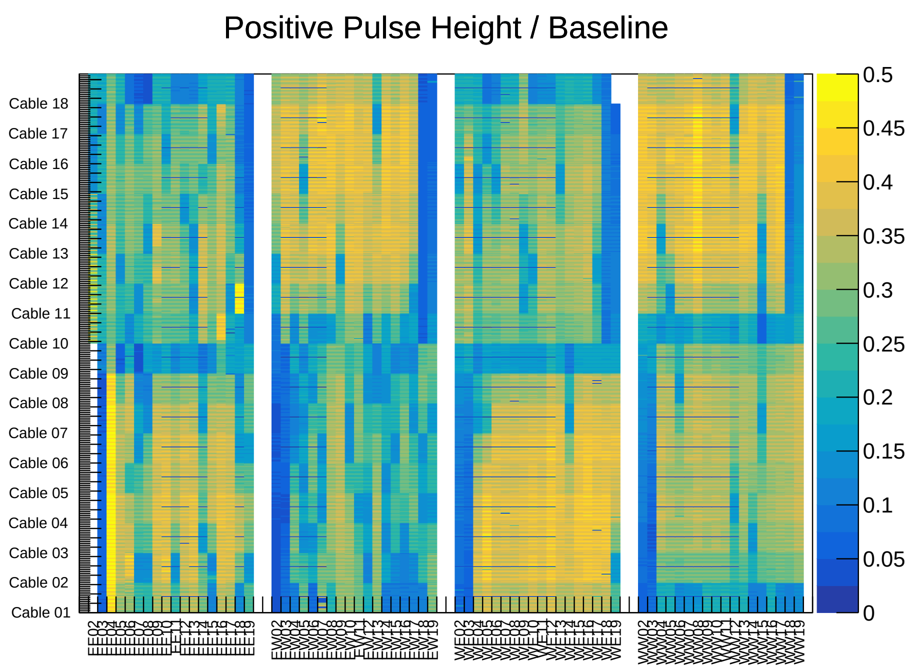

Figure 6: The map of the positive peak to baseline ratios of all the channels. Each strip represents a single channel of the detector. The x-axis includes all the chimneys of the ICARUS detector, while the y-axis shows all the 18 cables, each with 32 channels, within a chimney. The color code in each figure indicates the values of the ratio.

and the effect shown in Figure 6 is rather owing to the color scale.

We conclude that there is no disconnected cable, and the source of all the cables with lower signal amplitude is confirmed to be the grounding.

5 Interpretation of the results

Histograms, shielded/non-shielded cable problems, cross-talk, etc....

A Setup of acquisition code

The data acquisition pattern followed in this test has been described in section 3.7. In this section, technical details are described for the set up of the software for that acquisition.

All the original code written by Sergi Castells and the following extensions are published in a public GIT repository. A copy of that repository is currently available in GitHub as `PetrilloAtWork/ICARUSconnectivityTest`[3]. The software used for data acquisition in September 2018 was being developed as the need arose. At the beginning, Sergi's code, roughly equivalent to the tag `v01_00` of the repository above, was used directly. By the end of the data acquisition, version `v03_00` was used, and code was drawn from that version for the analysis as well. In the following connectivity test, planned for December 2018, the starting version will be `v04_00` (see appendix A.1).

The test driver is unfortunately hard to correctly set up the first time. The software is all written in python 2, and it requires both the python Visual Instrument Software Architecture (PyVISA, exposing the module `visa`) and CERN ROOT (PyROOT, exposing the module `ROOT`) interfaces. It proved hard to get to a configuration where python would load both at the same time on OSX, while no Linux laptop was tried. The pattern that seemed more likely to succeed was to install PyVISA following online instructions found at `https://pyvisa.readthedocs.io/en/master/index.html`, which included installing National Instrument backend software[4], that we used in its release 18.0. After having tested that the `visa` module correctly work, again as described in the linked documentation, we compiled ROOT (6.14) from source *in the same environment* (that is, in the same shell).

The setup of the test instrumentation is the same as for the software from Sergi. The data acquisition session is run within a single `python` session. Its setup includes importing the test driver module (`testDriver.py`), setting the IP address of the oscilloscope in use, and finally instantiate the test driver. A possible setup sequence is:

```
from testDriver import *
loadScopeReader('192.168.230.29')
reader = ChimneyReader()
```

After this per-session setup, the data acquisition is driven by choosing the chimney (e.g. `reader.start("EW09")`) and repeatedly call the acquisition function (`reader.next()`). It is also possible to skip to remove the last acquisition and repeat it, or skip to a different one.

Note that for up-to-date instructions the file `README.md` in the GIT repository supersedes the information provided here.

[3]URL: `https://github.com/PetrilloAtWork/ICARUSconnectivityTest`.

[4]Registration to National Instruments web site is required in order to download the software, but no fee payment was required to download the driver package.

A.1 Improvement implemented for December 2018 test

This section shortly lists improvements that are implemented and will be tested for the following connectivity test, the last without the official ICARUS DAQ, planned for December 2018.

- single script working for all oscilloscopes
- configuration of the script driven by configuration file(s)
- integrated verification of the correctness of the data format at the end of the acquisition from a chimney
- generation of a data transfer script
- single connection to the oscilloscope for the whole session (more error prone but *much* faster; `ChimneyReader.scope().reconnect()` may help to recover connection issues
- ability to jump to a chosen cable and position at any time (`ChimneyReader.jumpTo()`)

The expected acquisition flow is outlined as follows:

1. preparation of the data acquisition: start of the python interactive shell, initialisation of ‘testDriver‘ helper object with the proper configuration file, selection of the first chimney to read out;
2. sequential acquisition of waveforms from all cables, with the same pattern as in September 2018;
3. after completion, verification of the data written on the local disk; this may take 5-10 minutes, while the operators may “close” the tested chimney and prepare for the acquisition of the next one (but note that if data is found incomplete, more acquisition might need to take place);
4. after data is validated, generation of a data transfer script
5. execution of the generated script into a separate shell, going in parallel with the rest of the activity

The remote destination of the data is specified in the configuration file.

B Supporting library software

The analysis is supported by a utility library which is available as GIT repository, with the main repository being stored by GitHub at github.com:PetrilloAtWork/ICARUSconnectivityTest.git. It is possible to download it with:

```
git clone git@github.com:PetrilloAtWork/ICARUSconnectivityTest.git
```

or equivalent.

The analysis library used for the online analysis is tagged as `v03_00`. In that version, both waveform parsing and analysis code are in `drawWaveforms.py`. Utility classes are provided:

- `WaveformSourceInfo`: a data structure describing a waveform by its parameters: chimney, connection, test box switch position, oscilloscope channel and index
  - `channel` attribute is provided with the number of connection channel, computed from the switch position and oscilloscope channel and ranging between `1` and `32`
  - `increaseIndex()` is a shortcut to increase the index attribute
  - `firstIndexOf(position)` is a static method returning the index of the first waveform on a given switch position
- `WaveformSourceParser`: a class that represents the group of waveform files which contains a specified one, the "triggering file":
  - the triggering file is specified at construction or with `parse()` method
  - the parameters of the triggering files are accessible via `sourceInfo` (a `WaveformSourceInfo` instance)
  - the pattern of the file path is stored as `sourceFilePattern`, and a full file name can be obtained by `sourceFilePattern % sourceInfo` (with `sourceInfo` *any* valid `WaveformSourceInfo`)
  - the list of expected names for all `N` files at the current (or the specified) oscilloscope channel can be obtained via `allChannelSources()`
  - the list of expected names for all `4N` files at the current test box switch position can be obtained via `allPositionSources()`

Plotting functions:

- `plotWaveformFromFile()` creates and returns a single `TGraph` with the specified waveforms;
- `plotAllPositionWaveforms()` returns a `TCanvas` split in as many pads as there are oscilloscope channels (`4`), and plots in each pad all the waveforms from a channel

  - the first argument is a `WaveformSourceParser` object which determines which waveforms are plotted (equivalent to `allPositionSources()`)
  - the vertical range of all graphs is fixed to be at least between 1 and 3 volt
  - statistics are extracted and printed in each pad, as documented below

- `plotAllPositionsAroundFile()` plots all the waveforms for the same connector and test box switch position as the one of the specified waveform file; the plots are described with `plotAllPositionWaveforms()` above.

The statistics box of the plots includes:

waveforms the number `N` of waveforms in the plot (typically 10);

baseline the average of the `N` baselines of the `N` waveforms, each one computed with `extractBaseline()` algorithm;

RMS the average of the `N` RMS on the baseline of the `N` waveforms; this is a quantification of the average noise;

maximum for each waveform, the largest value is found; the *maximum* is the average of the distribution of maxima from the `N` waveforms, and the uncertainty is its error;

peak for each waveform, the positive and negative peaks are found over the baseline, and the largest is used; the *peak* is the average of the distribution of single waveform peaks;

RMS is the RMS of that peak value distribution.

Additionally, a small statistics library is provided:

- `MinAccumulator`, `MaxAccumulator`, `ExtremeAccumutator`, `MinAccumulatorN`, `MaxAccumulatorN` and `ExtremeAccumutatorN` remember the N largest or smallest (or both) of the values that are `add()`'ed to them

- `StatAccumulator` gives average, error, RMS and related quantites, of the sample given to it element by element (via `add()` call)

The fundamental analysis algorithms applied on a single waveform are:

- `findMaximum()`, `findMinimum()` and `findExtremes()` return the position respectively of the maximum, of the minimum and of both, within the list of values in argument;

- `extractBaselineFromPedestal()` extracts the baseline and its RMS and uncertainty, as average of the samples in the start of the waveform;

- `extractBaseline()` extracts the baseline of the waveform and its RMS and uncertainty, as described in the previous section;

- `extractPeaks()` extracts the value and location of the two peaks of the waveform, with uncertainties, as described in the previous section;
- `extractStatistics()` runs many of the previous algorithms, and also select the absolute peak.