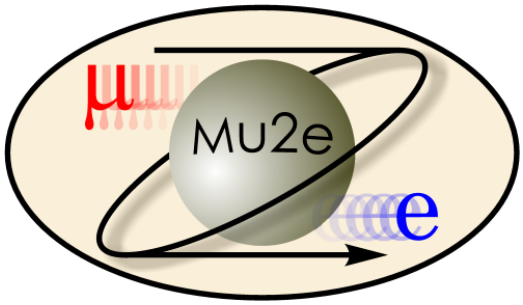

Standard Operating Procedure: Automated Straw Resistance Measurements

Cole Kampa

University of Minnesota

November 2, 2017

kampa041@umn.edu

1 Goal

Immediately after removing the paper and before gluing in the CO_2 end pieces, the resistance of each straw will be measured and recorded. In an effort to diagnose any issues with the metalization in each straw, four resistances will be measured, all from the ends of the straws: inside-inside (i-i), inside-outside (i-o), outside-inside(o-o) (figure 1). Thus for each pallet 96 measurements must be taken. We strive to do this in a safe, efficient, and reproducible manner.

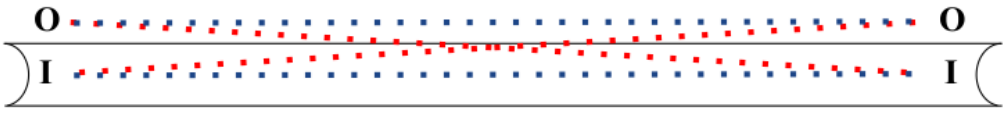

Figure 1: Diagram of different resistance measurements on each straw. Note that the cross measurements in red, inside-outside & outside-inside, should have an infinite resistance.

2 Equipment

- Full pallet of 24 Mylar straws (incremental numbering, st##### format)
- Resistance Meter v1.0 (Arduino Uno & PCB in acrylic box) (figure 2)
- 9V power supply with barrel connector
- USB A to USB B cable
- Left and right resistance connectors and ribbon cables
- Aluminum bar covered in electrical tape
- Small, stainless steel scissors
- Computer with Python 3 and Mu2e-Factory repository cloned to Desktop (see readme.md for information on necessary Python libraries)
- Barcode scanner connected to computer via USB

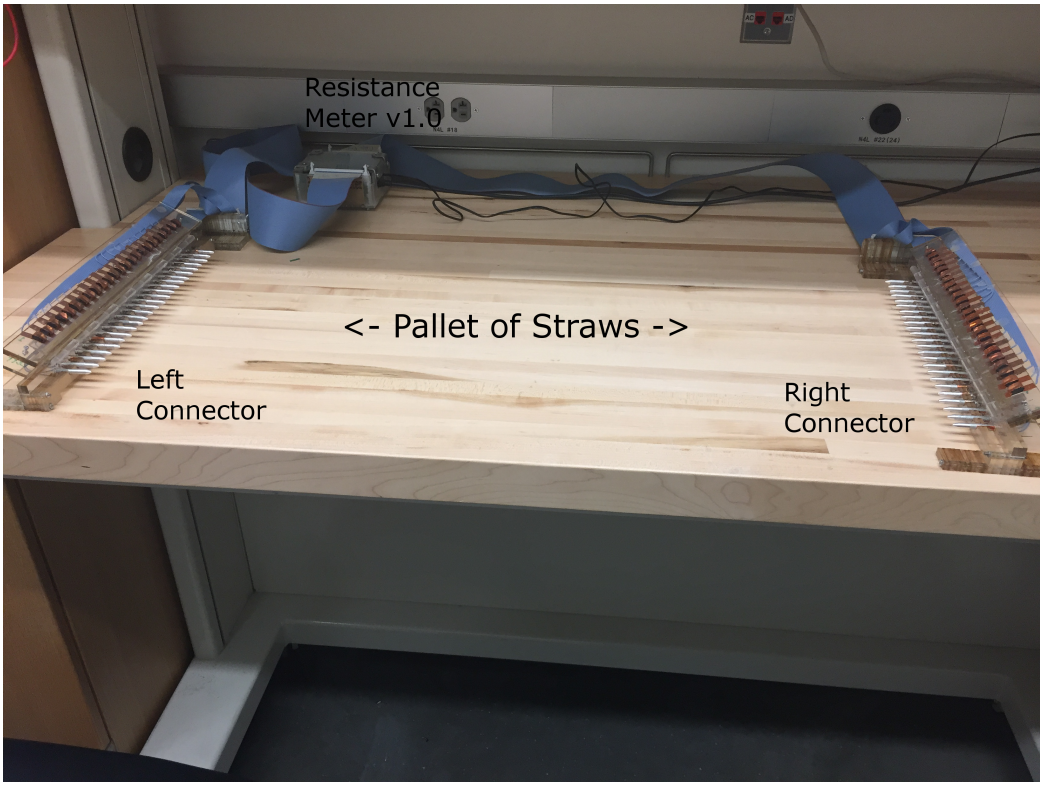

Figure 2: Full resistance meter setup labeled.

3 Important Files

The following files are located in the Mu2e-Factory github repository (https://github.com/ckampa13
Factory), which is cloned to the computer used with this process on the desktop. All files are within the Mu2e-Factory/Resistance_Measurement/Automated/ directory.

- measurement.py (main script, measures straw resistance; shortcut located next to folder on desktop: Resistance Measurement)
- calib.csv (Directory: .../Calibration/ ; calibration data stored in this file)
- StrawResistance_YYYY-MM-DD_HHMMSS.csv (Directory: /Resistance_Data/; measurement data stored using this naming convention)

4 Risks and Dangers

The primary risk associated with this measurement is the risk of damaging the straws. It is very important to follow the SOP carefully, otherwise straw kinks are likely to occur. Since the connector slides into the straws and pressure is on both the inside and outside metalization, there is also the possibility of scraping the straws. This should be avoided at all times.

5 Resistance Measurement Procedure

5.1 Resistance Meter v1.0 Setup

1. Plug USB cable: USB B into Resistance Meter v1.0 box Arduino Uno USB serial port and USB A into computer USB port (figure 3).
2. Plug power supply (9V) into Arduino power port and outlet (figure 3).
3. Plug connector labeled Right into matching ribbon cable. The other end of the ribbon cable should be plugged into the top of the Resistance Meter box ribbon cable connector labeled Right.
4. Repeat (3) for Left connector. Note that the left cable has a given orientation to be plugged in and is labeled (Left Box and Left Connector).

5.2 Connection to Straws

1. Put on nitrile gloves.
2. Place pallet of straws on middle of desk.
3. Using aluminum bar with electrical tape, gently push on the right ends of the straws (ie in direction to slide straws to the left in the pallet troughs) so that they are all aligned on the right side. This will likely cause the left side to be very uneven.
4. Place right connector on far right side of pallet, with probes facing the straws. The connector should nest over the aluminum base of the pallet.

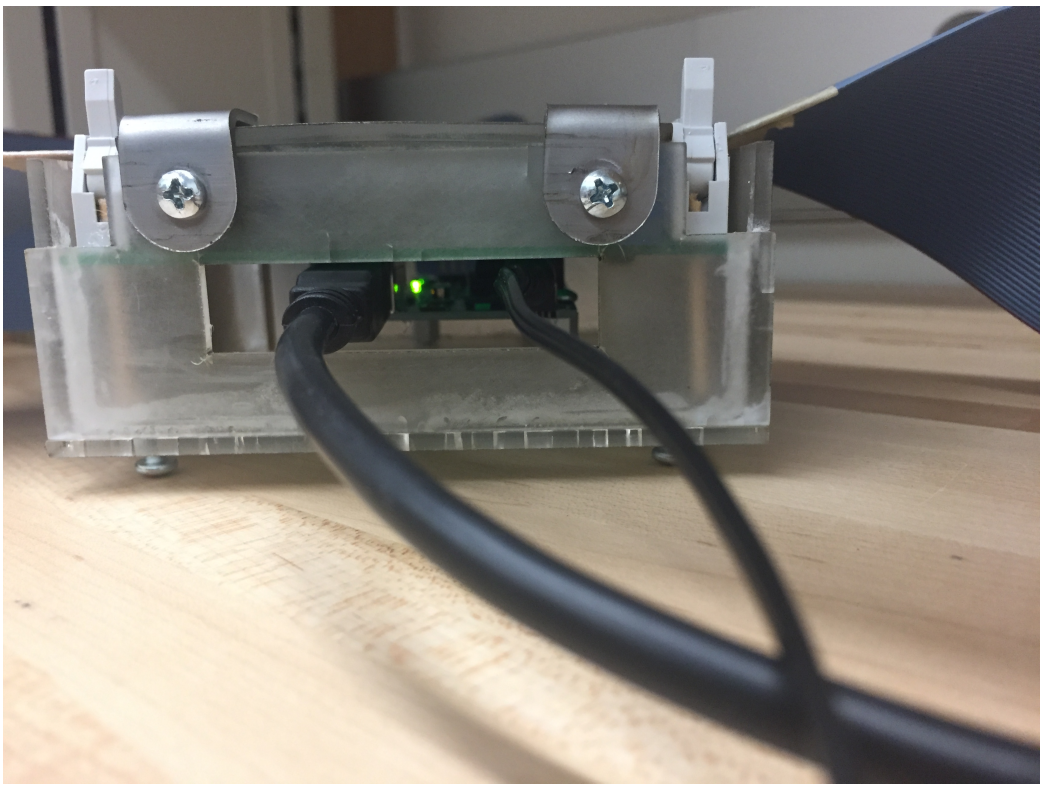

Figure 3: Plugged in Resistance Meter: USB cable on left, power supply cable on right.

5. Open top hinged half of the connector and slowly slide towards straws. As aluminum probes slide in the straws, it may be necessary to put light pressure on the center of the straws to prevent them from sliding. Slide probes all the way in.

6. Using the scissors, cut the left end of the straws to as even a length as possible, while still allowing some of the straws to stick out of the troughs (this allows for better connections).

7. Using the method described above, attach the left connector to the straws.

5.3 Collect Data

1. Open resistance test GUI from shortcut, named `RUN_RESISTANCE_TEST`, on desktop. A screenshot of the GUI is shown in Figure 5.

2. Using barcode scanner at computer, scan worker ID and workstation ID

3. Scan in the straw barcode for the lowest-numbered straw on the pallet into the box in the GUI.

4. Have one person stand at the left connector. They should use their forearm to apply even pressure to the top of the left connector, while the operator applies even pressure to the right connector (figure 4).

Figure 4: Putting pressure on the right connector. Make sure to wear gloves when connector is actually attached to straws.

5. Click on "generate" in the GUI to measure the resistances.

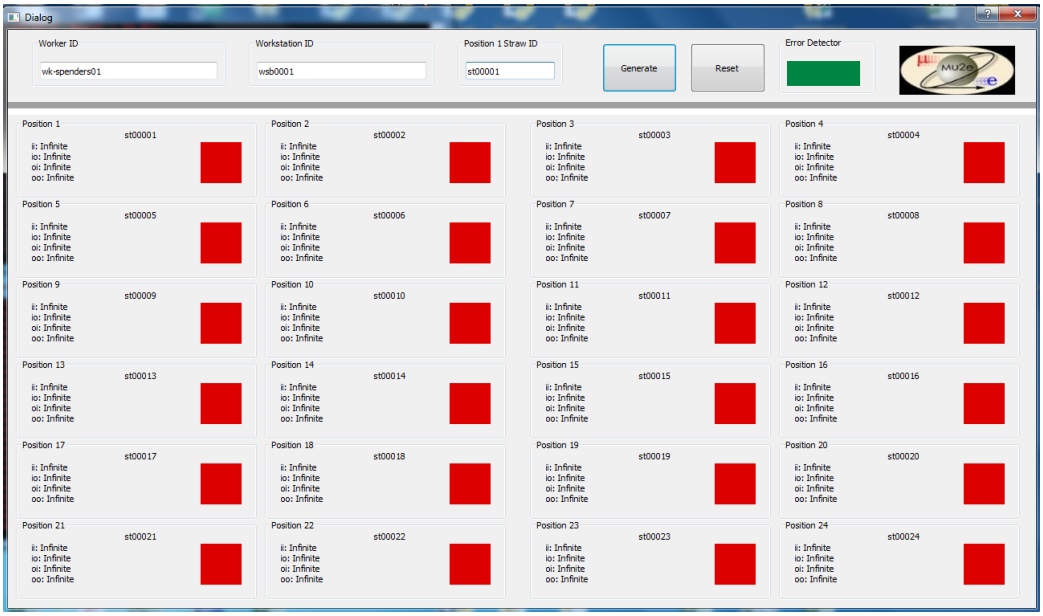

Figure 5: Resistance testing GUI.

6. The display will show measurements with pass or fail ratings, with green for pass and red for fail. To retest the straws that failed, simply hit "generate" again.

7. The data saves automatically. When finished, either exit out of the program, or click on "reset" to test another pallet of straws.

5.4 Cleanup

1. Lift right connector top plate

2. Slowly slide right connector out of straws, and store on back portion of desk

3. Repeat for left connector

4. Dispose of any small straw fragments from cutting process into trash bin

6 Troubleshooting

If anything is unplugged, plug it in and try again (i.e. power, USB, ribbon cables on box end or connector end). If this does not work, immediately alert the workstation manager, who will replace entire setup with new resistance meter box and connectors (should have 2 sets at all times). When replacing, check serial port (using Arduino software) and adjust calib.py and measurement.py to reflect the change in port number. Then, run calib.py (see Calibration SOP), and proceed with production.

At a test computer, the manager/troubleshooter should plug malfunctioning box in to power and USB to the computer, and connect a 150Ω calibration pallet to the ribbon connectors on the box. Run the script to see if each channel reads approximately 150. If they do, there is likely a bad connection somewhere within one or both of the connectors.

Some other possible culprits are listed below.

1. Script Crashes:
    - First, run a command prompt (to prevent script from automatically closing) in the directory that the script is located (Shift+RClick on Windows 7 in File Explorer, select Open Command Window Here), then run the command $*pythonmeasurement.py*$
    - Take note of any errors that follow
2. Hardware Errors:
    - Crashes immediately upon running:
        - Verify Arduino is plugged in to power and USB to computer
        - Using Arduino software, check port of board (COM## on Windows, dev/... on Linux) using Tools-Port (should be listed as Arduino Uno)
        - make sure `com_port` variable in `measurement.py` line 42 matches port
3. Typographical Errors:
    - Fails at Temp or Humid:
        - Make sure to enter int or float values

  - Fails after entering start and end straws
  - Verify format of st##### where # are digits 0-9
  - Difference between start and end straw should be 23 (i.e. full pallet of 24 straws, in order)

4. Bad Measurements:

   - Check last time of calibration (kept in log sheet on desk above resistance meter)
   - Try redoing calibration (see SOP for calibration)
   - Make sure connections are solid
   - One channel always reads zero ohms or inf when it shouldnt:
     - Use handheld multimeter to measure resistance from connection point at straw to cable end (see pinout diagrams). If resistance is high, check connection points along the way: ribbon connector (upside down), solder points (copper tape), copper tape to aluminum bullet in bottom half.
     - Isolate problem and resolder if necessary
     - Connect calibration pallet 1 (150 Ohm resistors) to resistance meter and run measurement.py, if channel still measures improperly, there might be a problem with the PCB (likely a failed multiplexer or a broken trace, which are not easy to fix. Put to side to be inspected by electronics specialist)
     - Multiple lines going to the same analog input of Arduino (see pinout diagrams) are not working, test PCB with a new Arduino
     - Test a new PCB if necessary