9. Measure pico farads - New Cap Measurement Approach

Purpose: Get the ATTiny84 to measure pico-farad values.

Goal: The approach I've been using for cap measurement to this point has failed for the pF level of capacitance of a moisture sensor - my home made one or, apparently, any such sensor. For an 'Arduino' to measure these low values a different approach is required. So the goal of this milestone is to implement this new approach on the ATTiny to see if I can get it to work.

Key References --

New Approach - Cap Divider Network. This is explained @ https://wordpress.codewrite.co.uk/pic/2014/01/21/cap-meter-with-arduino-uno/

To Modify RPi side .cpp code --

Use Kate on my desktop.

Then copy to the RPi using below command:

```
scp /home/jroc/Dropbox/projects/MoistureSensor/CapSensor/RPi/RPi_CapDataReceive.cpp pi@192.168.1.118:/home/rf24libs/RF24/examples_linux/
```

Then on the RPi, cd into the build directory and execute 'make'

BUT if this is the first iteration of the program, you need to first add it to CMakeLists.txt in the examples_linux directory (above the build directory).

NOTE: All the code is under git version control. So the executables for both the ATTiny and RPi are on my desktop.

Raspberry Pi Login --

Pi user is: pi. So SSH login is:

```
ssh pi@192.168.1.118
```

password: `pi9012`

And then once in, get into the working directory for the nRF24 utilities and programs:

```
cd /home/rf24libs/RF24/examples_linux/build
```

The run the RPi receiving program:

```
./RPi_CapDataReceive
```

06/05/2023 --

Update: **Success!!** The new approach is reading my home-made sensor, giving consistent readings between 8-9 pF in free-air. And it responds nicely to placing my fingers over the copper stripes. Not testing in water as it is not water-tight. So now I can experiment with making a decent, and water-tight, sensor.

Prep --

Committed some minor comment changes in tiny84_CapMeasureAndTx.ino to git in prep for coding up the new approach.

Opened `tiny84_CapMeasureAndTx.ino` and did a 'Save As' into **`tiny84_pfCapMeasureAndTx.ino`**, which will be the code for the new measurement approach.

Software --

RPi: No change, was able to use the 'data-receiver' program as -is.

ATTiny: Created a new function to implement the new measurement approach: `void capacitorMeasurement(TxPayloadStruct * pLoad)`. Which, amazingly enough, worked the first time.

Git Commit: Made a commit, named: "06/05/2023: Successful proof of concept for pF measurement by ATTiny."

Hardware --

Crazy simple, per the blog post describing this approach. Simply connect the capacitor to be measured between the two pins of the ATTiny that are used to (1) Charge the cap; then (2) to measure the voltage. So per how I implemented it in code in this proof of concept version:

Physical Pin #5 = "`chargePin`"

Physical Pin #13= "`voltReadPin`"

No additional connections, or components were needed. Including no connection to ground, which I was initially uncertain about. But the 'virtual' capacitor, C!, in the circuit diagram that is formed by the stray capacitance also has a 'virtual' ground.

Next Step: Fabricate water-tight sensor.