Making the gas readings retrospective.

When a gas sensor pulse is received it is evidence that .1 cubic metres of gas has been used since the previous gas sensor pulse, it is a post event signal, thus it is not possible to dynamically view real time gas usage, however it would be possible to post process the monthly file, record by record, to show the dynamic usage.

1. Stage 0.
   1. When a new day file is created, if the Last\_Gas\_Date\_Time\_Data fields, and the EEPROM fields, have not been updated, update them with the current Date and time. Write the Last\_Gas\_Date\_Time\_Data fields as the record immediately after the column headers record. If the day file already exists within Prefill\_Array update the Last\_Gas\_Date\_Time\_Data fields with the first record and whenever field 19 in the data file is >0 indicating a gas sensor interrupt was recorded in this record.
   2. Whenever a gas sensor interrupt is received update the Last\_Gas\_Date\_Time\_Data fields and the EEPROM fields with the current date and time..
   3. Subroutines which access the day file must therefore skip record 1, just like they skip record 0.
2. Stage 1.
   1. Each day file will contain the Last\_Gas\_Date\_Time\_Data fields, as record 1 (after the column totals).
   2. When a month end is detected:
      1. Generate two arrays containing the filenames (.csv and .txt) of all the ,csv and ,txt files on the SD drive..
      2. The two arrays should be sorted (earlier files first).
      3. Create a new file, “DateTimeTags” containing the first record from each .csv file (date and time) in date order, and the date and time whenever a gas sensor interrupt occurred (field 19 > 0) should be created.
      4. A monthly summary file should be created.
         1. Each .csv file should be read, skipping records 0 and 1.
         2. The startwritten to a If the daily file already exists, in the prefill\_array subroutine load the Last\_Gas\_Date\_Time\_Data fields and the eeprom with the saved Last\_Gas\_Date\_time\_Data fields.

1. Create a file containing the date and time information from all the daily files for records having a (19) record.
2. Creating the monthly file:
   1. Pass 1:
      1. Read the Last\_Gas\_Date\_Time\_Data.fields from each daily file.
      2. Read each record of the daily file until a gas sensor signal is detected (field 19)
      3. Using the date and time of b. calculate the “Per 5 second” gas usage between the 2 dates and times.
      4. Add the start Last\_Gas\_Date\_Time\_Data fields, the last Last\_Gas\_Date\_Time\_Data fields ad the gas usage into a array, one element per occurrence of field 19.
   2. Copy each record, missing the LastGasSensorSignal record) from the daily file to the monthly file with the calculated gas usage figure.
   3. Copy the current Date and Time of the b. record into the Write the last date and time of receiving the gas sensor signal into eeprom, so that it is preserved from day to day, or power on.
3. We know that records are made every 5 seconds
4. When a gas sensor signal is detected:
   1. Calculate the time difference between the current gas sensor signal and the saved gas sensor signal, divide this by 5. This will calculate the number of readings/records between signals.
   2. Calculate the gas volume per reading by dividing 0.1 cu m by the values calculated in 3.
   3. Create a new readings file
      1. Copy records from the current readings file until the date and time match the saved gas sensor signal date and time.
      2. From this point change the