RS232 Communication protocol for AAG_CloudWatcher (part 4c[1], 1 Jun 2021) including pocketCW

Errata in some of Antonio's original documents:

- "B!" results in "!V" (instead of "!N", doc 1.0)
- the value returned by the rain sensor, "!R xxxx" is NOT a value in the range 0..1025, but the actual frequency read (0 .. 6000, approximately) – doc 1.0
- "m!" Results in "!m" (instead of "!M", doc 1.2)

Relevant hardware changes since last (1.2) document:

- units may have a RH sensor, either internally (limited edition, sensors were given away for free for testing purposes) or externally installed.
- Similarly, units may have a dual sensor, atmospheric pressure + relative humidity
- New anemometer model: the original, gray one, became unfortunately unavailable a time ago, new one is black and differs slightly in the speed calculation
- there's a new compatible product, the pocket CloudWatcher offering the option of PC connection (via USB).

Change in wind speed calculation:

The user should be given the option to choose what model of anemometer is using . Options are:

- original (gray) model. Speed as reported by the "w" command
- current (black) model.
  - If reported value = 0, speed = 0
  - If reported value > 0, speed = (value * 0.84) + 3 (Km/h)

Pocket CloudWatcher support:

- A slight delay of 2 seconds should be added when the port is openend before sending any data. This can be done regardless of the hardware, no harm
- brightness will be reported in mLux – this can be left as is, and properly supported in the unsafe alarms etc, or converted to the older range of values (0 meaning very bright, 50000 or above very dark).

[1] Included atmospheric pressure sensor, updated to firmware 5.8.x

Commands added since document Rs232_Comms_v120:

From firmware versions 5.6 onwards:

Units may have a relative humidity sensor (also providing an ambient temperature value), and our software allows the user to select where the ambient temperature should be taken from, being the one coming from the external RH sensor the best option.

- "**h!**" - > relative humidity, it can have 2 possible responses
  - "**!h xxxx**" (same format as other older commands), with xxxx being a 0..100 value, as a percentage of duty of a PWM out signal of the Si7007 (or a compatible value from the Si7021). A value of 100 (xxxx = 100) means sensor error, else:
    rh = xxxx * 125 / 100 - 6
  - "**!hhxxxx**" (the second "h" meaning high resolution) – A value of 100 is ok here
    rh = xxxx * 125 / 65536 - 6

  Values slightly off (above 100 or below 0 RH) are ok and should be limited to the 0..100 range.

- "**t!**" → temperature as read by the RH sensor (specially for the external one, Si7021, much accurate than the one coming from the IR sensor). Also 2 possible responses
  - "**!t xxxx**" (same notes as above apply)
    temp = (xxxx * 1.7572) – 46.85
  - "**!thxxxx**" (same...)
    temp = (xxxx * 175.72) / 65536) – 46.85

  If the result of the ambient temperature calculation results in an impossible temperature (I use the [-40..80]ºC range), then the RH sensor should be marked as invalid, both for temperature and rh readings.

From firmware 5.71 onwards:

- "**B!O!O!T!**" → will reboot the unit, thus allowing firmware update

From firmware versions 5.8 onwards:

Units may have a dual atmospheric pressure and relative humidity sensor (also providing two ambient temperature value). The RH and RH-based temperature values are compatible with the commands above, for pressure:

- **"p!"** → atmospheric pressure
  - **"!p xxxx"**.

Absolute pressure = xxxx / 16 (Pa)

To convert to the more common relative pressure, the following formula is being used:

rel pres = abspres*(1-(0.0065*HASL/(Temp+0.0065*HASL+273.15)))^(-5.275)

Where:

- HASL: height above sea level (in meters)
- Temp: ambient temperate (in degrees celsius)

- **“q!”** → temperature of atm. press sensor
  - **“!q xxxx”**

    Temperature = xxxx / 100 (ºC)