Michael Blunt

sps30 writeup

This is a complex python program that utilizes many advanced packages, functions, and communications. However, the function names in conjunction with the comments throughout added by myself and the original devs (UnravelTEC) are very helpful. I will not provide a specific line by line analysis of the code, rather, I will provide a general overview of functionality and intended purpose.

**Initial Setup:**

* The program starts by defining variables and basic functions.
  + “eprint” is a custom function for printing error messages
  + pigpio and I2C static variables are set
* I2C detect is called and run
  + If SPS30 is found, returns 0
  + Else exit(1)
* Exit gracefully
  + This function is never used in our program, as the SPS30 program should never have to terminate itself.

**PigPio:**

* This is more into the realm of UnravelTEC’s work, I did not write or edit much of this at all. No documentation on these specific functions is available (see their writeup) so I will do my best to interpret.
* Starts by checking if pigpio daemon is running on localhost
  + If not prints an error message
* Closes the previous “checking” connection
  + Checks and handles error to this function
* Opens a connection and assigns it to variable “h”
  + Lots of commented functions here, I believe I received this code with these functions commented out.

**Communication and Data Gathering:**

* This is also entirely UnravelTEC’s work, but I will document it as well as I can. Everything is defined as a function and called later on in the program.
* Read Bytes:
  + Attempts to read bytes from I2C connection
* Write Bytes:
  + Attempts to write bytes to I2C connection, returns true if you can write to the I2C connection
* Read From Address:
  + Not sure the specifics of this function, appears to attempt to read from a range of addresses 3 separate times.
* Read Article Code:
  + Reads the article code from SPS30 from specific address and prints it
* Read Serial Number
  + Reads the serial number from SPS30 from specific address and prints it
* Read Cleaning Interval
  + Reads the cleaning interval and prints it

**Measurements:**

* Start/stop measurement
  + Self explanatory, resets SPS30 if error is thrown
* Reset
  + Tries a “nice” reset, sends reset request to SPS30
* Data ready
  + Throws errors if data is wrong or did not return anything
* Calculate Integer/Float
  + No idea what is happening in the internals but it returns a variable apiece that is used later in outputs
* Print Prometheus
  + Returns data to a log file
* Print Human
  + Prints a human readable version of the data
* Read PM Values
  + Reads values from specified addresses, passes data to the Prometheus and Human (if DEBUG=true) functions
* Initialize
  + starts the measurements or exists if false
* Big Reset
  + Resets and closes the pigpio connection

**Team Function (run):**

* This is the section of code most modified
* Steps:
  + Makes log file
  + Resets
  + Initialize
  + Checks to make sure data is available
  + Calculates the data manually,
    - Borrowed method from printHuman
  + Creates a json formatted file
  + Defines filename as the current UNIX timestamp
  + Places files in /home/pi/
  + Takes timestamp from transfer\_log.txt
    - Writes that into its own log (sensor\_log.txt)