Piera IPS Release Notes
Version 3.0.2
6/13/2024

This document covers Piera's Intelligent Particle Sensors (IPS-7100. IPS-5100, IPS3100) Manufactured after May, 2024 shipping with Firmware Version 3.1 or later. In this document these are referred to as second-generation IPS units.

Serial Numbers:
The Serial Number format remains the same for second-generation IPS Units with the following format:
IPS**X**100-**YYM######**
X = Model number, 7 = IPS7100, 5= IPS5100, 3=IPS3100
YY = Build Year, e.g. 24=2024
M = Build Month, from A-L, A = January - L=December.
= Serial Number of Build Month/Year

Second-generation IPS units are built after **May, 2024** and indicated by the following: **IPS7100-24E** or greater build year/month in the serial number. You can also verify this by monitoring bootup messages in UART as shown later in this document.

Hardware Changes:
These second-generation IPS Sensors utilize an ESP32-based MCU instead of the STM32-based MCU used in first generation IPS Sensors. This provides increased clock frequency (160mhz) and 2 CPU Cores which enables increased sampling rates and a dedicated CPU Core for sampling compared with the original units which used a single core operating at 64Mhz.

The units also feature fan-speed monitoring and control, A PWM signal from the fan monitors fan speed (RPM) and the fan duty cycle PWM is dynamically changed every second to keep fan RPM within the factory range of 3500 RPM + / 50 RPM to match the fan speeds they are calibrated. This ensures that as input voltage changes, or fan life degrades the units will operate at a steady fan RPM to ensure consistent airflow inputs.

Calibration Changes:
All second-generation IPS Sensors are more rigorously calibrated using 12-points of calibration steps compared to the first generation sensors which used 4-5 point calibrations. This ensures

linearity of sensor data across the entire operation range, especially focusing on low concentration regions (below 5ug/m3), greatly increasing the sensor performance in terms of reproducibility and accuracy. The increased calibration steps provide more precise calibrations improving the unit-to-unit variance.

Software Changes:
The firmware for these units is built using ESP-IDF. The codebase is similar to our Canaree I-Series Air Quality Monitors but without any Wifi or Bluetooth networking functionality as no Wifi/BLE radio is present on the ESP32-chip used in second-generation IPS versions.

I2C Mode Changes:
The second-generation IPS Sensors support the same I2C Command-sets as first-generation IPS sensors to ensure backward compatibility with OEMs products; The I2C address remains 0x4b and cannot be changed; However there are some minor timing-related changes outlined below:

- I2C Communication should be delayed until 5 seconds after power-on or soft reset
- There must be a minimum of 100 msec delay between I2C Commands.
- Customers should avoid using the soft reset command (0x2D) in I2C Mode to avoid additional delays. It is instead recommended to use factory-reset (0x2E) to ensure devices are using the factory configurations prior to I2C communication.
- Firmware versions prior to v1.3.7 may periodically output UART on I2C Bus when certain commands are sent. This behavior has been addressed in firmware versions v1.3.7 and above.

More information on I2C Mode can be found in this document on our support site:
https://pierasystems.com/wp-content/uploads/2021/06/Piera-I2C-Reference-Guide.pdf

Example Code for I2C Implementation can be found here:
https://github.com/PieraSystems/7100-I2C-example

UART Changes:
Sensor Output is available 5 seconds after power-on, which is output every second in the default mode. They are fully compatible with the current version of SenseiAQ App (v2.0.4) https://github.com/PieraSystems/SenseiAQ and the existing Python Tools provided from Piera Systems https://github.com/PieraSystems/python-tools (Logger.py and Monitor.py)

While UART output and commands are backward compatible with first generation IPS Units the output of $Dflash= command has additional values indicating calibration settings and fan speed controls as shown below

```
A: Dflash=
DAC1(Vgc):0,DAC2(Vref):1700
DAC3(Vb):1000,DAC4(Vth):2500
CK1:350273,CK2:175136,CK3:124562,CK4:59600,CK5:1012,CK6:123,CK7:49
```

```
MK1:10952,MK2:10952,MK3:10952,MK4:10952,MK5:2767,MK6:2963,MK7:2963
AVG1:32,AVG2:1500,AVG3:950,AVG4:4000,AVG5:87,AVG6:6794,AVG7:5438,AVG8:11646
STD1:120,STD2:161,STD3:3200,STD4:9978,STD5:300
VERSION_NUMBER:v3.1.0                # Firmware Version
SERIAL_NUMBER:IPS7100-24D000045      # Serial number
APP:                                 # Future Use
MAC_ADDRESS:000000000000             # Not Used on IPS
KEY_NUMBER:vfaQJEClIJfykFbHwbh2RA==  # IoT Key unique to every device
UART_SEND_TIME:1                     # Output Interval in Seconds
VSD:1,40,45,30,250                   # Vape/Smoke Detection Settings
CLEANING_INTERVAL:604800             # Cleaning mode in Seconds (Every 7 Days)
FAN_AUTO:1,3500                      # 1 = Fan speed Auto Active, 3500 RPM
LED_BRIGHT:50                        # Not Used on IPS
OUTPUT_UNIT:0                        # Default Output PC: #/Liter PM: ug/m^3
```

Additionally bootup messages contain additional information about the bootloader, memory space and partition tables. Below is an example output of the bootup messages.

```
rst:0xc (SW_CPU_RESET),boot:0x37 (SPI_FAST_FLASH_BOOT)
configsip: 188777542, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:6612
load:0x40078000,len:14788
load:0x40080400,len:3792
entry 0x40080694

I (28) boot: ESP-IDF v4.4.1-dirty 2nd stage bootloader
I (28) boot: compile time 13:28:25
I (28) boot: chip revision: 3
I (31) boot_comm: chip revision: 3, min. bootloader chip revision: 0
I (38) boot.esp32: SPI Speed      : 40MHz
I (43) boot.esp32: SPI Mode       : DIO
I (47) boot.esp32: SPI Flash Size : 4MB
I (52) boot: Enabling RNG early entropy source...
I (57) boot: Partition Table:
I (61) boot: ## Label            Usage          Type ST Offset   Length
I (68) boot:  0 nvs              WiFi data        01 02 00009000 00004000
I (76) boot:  1 otadata          OTA data         01 00 0000d000 00002000
I (83) boot:  2 phy_init         RF data          01 01 0000f000 00001000
I (91) boot:  3 user_nvs         WiFi data        01 02 00010000 00004000
I (98) boot:  4 ota_0            OTA app          00 10 00020000 00100000
I (106) boot:  5 ota_1            OTA app          00 11 00120000 00100000
I (113) boot: End of partition table
I (117) boot_comm: chip revision: 3, min. application chip revision: 0
I (125) esp_image: segment 0: paddr=00120020 vaddr=3f400020 size=0e644h (
58948) map
I (155) esp_image: segment 1: paddr=0012e66c vaddr=3ffb0000 size=019ach (
6572) load
```

```
I (157) esp_image: segment 2: paddr=00130020 vaddr=400d0020 size=2db04h
(187140) map
I (228) esp_image: segment 3: paddr=0015db2c vaddr=3ffb19ac size=00d3ch (
3388) load
I (230) esp_image: segment 4: paddr=0015e870 vaddr=40080000 size=0f71ch (
63260) load
I (260) esp_image: segment 5: paddr=0016df94 vaddr=50000000 size=00010h (
16) load
I (268) boot: Loaded app from partition at offset 0x120000
I (268) boot: Disabling RNG early entropy source...
I (280) cpu_start: Pro cpu up.
I (280) cpu_start: Starting app cpu, entry point is 0x400813bc
I (267) cpu_start: App cpu up.
I (295) cpu_start: Pro cpu start user code
I (295) cpu_start: cpu freq: 240000000
I (295) cpu_start: Application information:
I (299) cpu_start: Project name:     factory-app
I (304) cpu_start: App version:      1
I (309) cpu_start: Compile time:     May  3 2024 10:46:48
I (315) cpu_start: ELF file SHA256:  acca01749cc18dfa...
I (321) cpu_start: ESP-IDF:          v4.4.1-dirty
I (327) heap_init: Initializing. RAM available for dynamic allocation:
I (334) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM
I (340) heap_init: At 3FFB5810 len 0002A7F0 (169 KiB): DRAM
I (346) heap_init: At 3FFE0440 len 00003AE0 (14 KiB): D/IRAM
I (352) heap_init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM
I (359) heap_init: At 4008F71C len 000108E4 (66 KiB): IRAM
I (366) spi_flash: detected chip: generic
I (370) spi_flash: flash io: dio
I (374) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
I (1659) gpio: GPIO[35]| InputEn: 1| OutputEn: 0| OpenDrain: 0| Pullup: 0|
Pulldown: 0| Intr:1                                  # GPIO Status
The value is not initialized yet!
E (1709) i2c: i2c_driver_delete(403): i2c driver install errorart0_mode
Firmware version: v3.1.0                             # Firmware Version
Current partition: ota_1                             # Partition used for Boot
Current address: 0x120000                            # Boot Address used
10:46:48
I (1719) IPS: compilation_date:May  3 2024
I (1719) IPS: compilation_time:10:46:48
I (1729) IPS: secure_version:0
I (1729) IPS: project_name:factory-app
I (1729) IPS: git_version:1
I (1739) IPS: app_eld_sha256:3FFB7FEC                # Firmware hash
I (1739) IPS: idf_ver:v4.4.1-dirty
CalPoint = 12 set                                    # Calibration Steps = 12
```

The following error message is normal during UART Operation as the I2C Driver is disabled in this mode

```
E (1709) i2c: i2c_driver_delete(403): i2c driver install errorart0_mode
```

Firmware Updates:

A python-based Firmware Update tool is provided by Piera Systems which enables updating of the firmware in UART mode using the Piera supplied PEK Cable or a compatible (SiLabs CP2102 chipset) USB to UART TTL Adapter. For firmware updates, two partition tables are used. If the firmware fails to update the unit will boot the backup (previous version) partition table.

Power Consumption:

While both units are designed to operate at 4.5-5.5 VDC Previous IPS units using STM32-Chipset current draw was 75mA nominally, as a result of CPU changes these newer units require roughly 35% more power (115mA nominal) during normal operation but significantly less when using low power mode. Peak power consumption is seen immediately after bootup, with a maximum of 150mA @ 5V approximately 2 seconds after power-on, before dropping to 115mA for normal operation 3 seconds after bootup. Details of power consumption are provided below and measured at 5VDC.

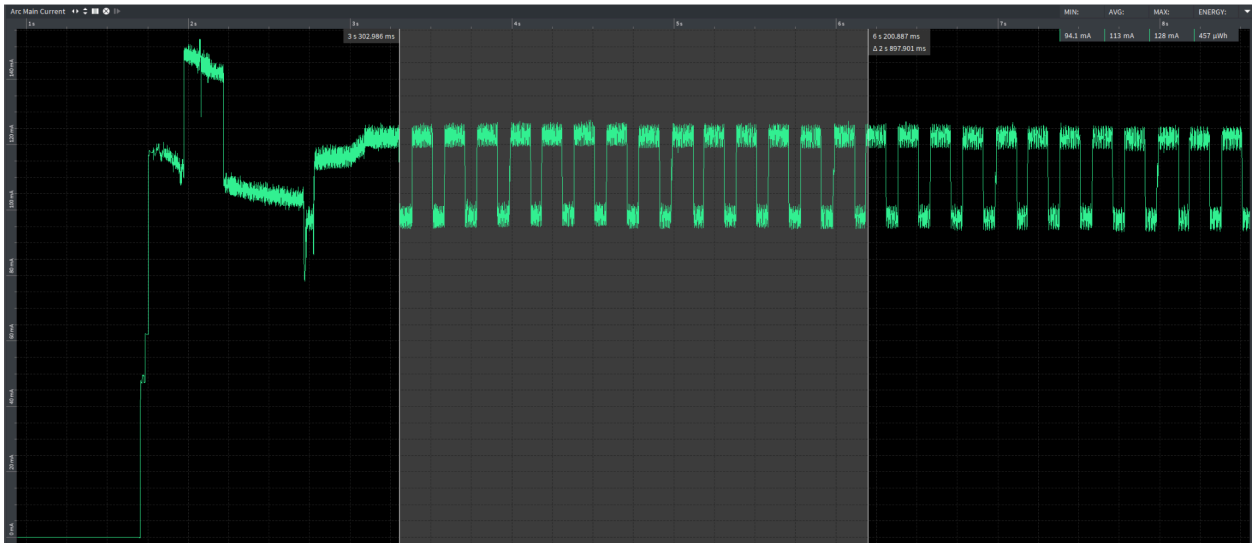

Figure 1 shows power consumption @ 5VDC for these IPS Units.

Low power mode:

A power-saving mode is enabled, which effectively puts the sensor to ‘sleep’ until it is given additional commands via UART or I2C. Power consumption in low power mode is reduced to 20mA within 1 second after entering this mode. In this mode the fan will shutdown and the sensor will wait for additional commands before booting again. The following command enters this mode in UART, after which no data is transmitted from the IPS device.

```
Wpsm=1
Entering light sleep
```

To enable sleep mode while operating in I2C mode the following command is used
`0x23`

Transmitting any UART data or I2C Commands to the sensor will remove it from light sleep mode. When leaving light sleep mode data is again available after 5 seconds.

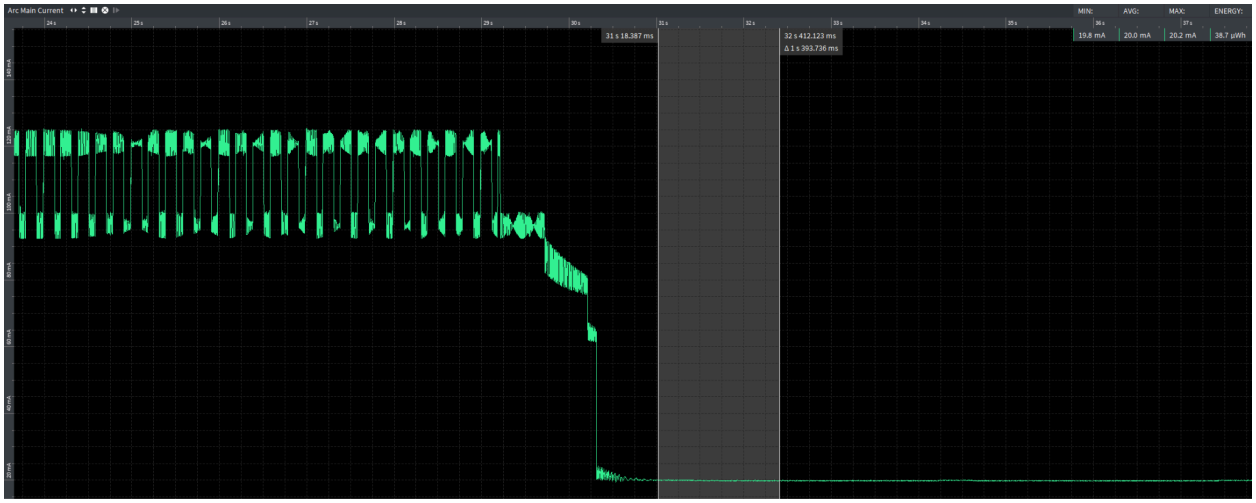

Figure 2: Shows Power consumption in normal mode and when activating light sleep mode.

For more information on IPS please visit https://www.pierasystems.com/support or feel free to contact us directly support@pierasystems.com