iEdge 4.0

Management Guide

Polysense Technologies Inc.

2024-03-14

INFORMATION

This document will describe iEdge 4.0 management, including reporting management, sensor management, board (PIN) management and remote management. Most can be implemented through the USB console or remote server.

HISTORY

1) 2023-04-24

- Update document format
- Add “vocb” (0x6b) for VOC PPB report type
- Modify Sensor-11 (SHT40, HPA, VOC … I2C) default power to PWR3 pin

2) 2023-05-06

- Fix some typoes
- Add sensor 19 (SO2) and 20 (NH3)

3) 2023-05-08

- Redefine sensor 19 as ZE03 for all ZE03 sensors (like SO2, NH3, HCL)

4) 2023-05-11

- Remove Chinese from FOOT and HEAD
- Fix some typoes for 15..23 → 16..23 in EPD face judgement
- Fix some typoes for EPD Chinse/English switching

5) 2023-05-14

- Re-Fix some typoes for EPD Chinse/English switching

6) 2023-05-16

- Support Light sensor (bitmap Bit[4] of I2C/SHT30 sensor) and related payload type

7) 2023-05-23

- Fix some typoes
- Support Current report (under 33-MV sensor type)
- Support DTU mode in ctrl field
- Support OTA command

8) 2023-05-31

- Support new OTA command for EVENT/ACT
- Support more controlling mode for FASTCNT sensor type (3)
- Support Angle/Acceleration/Ohm data types

9) 2023-06-01

- Correct PLSS DTU format descriptions

10) 2023-06-08

- Correct UV power unit to uw/cm^2

11) 2023-06-14

- Add new EPD bit definition for 485/temp sensors format
- Support tmp as single temp and etmp as array of temp

12) 2023-06-15

- Support sensor type 22 – laser distance

13) 2023-06-19

- Support sensor type 23 – ZPHS multi sensor
- Support sensor type 64 – Counter12
- Support report type BG95 (0x4185/0x4385)

14) 2023-07-07

- Support H2S payload type
- Support sensor type 24 – DGM10 multi sensor
- Add description for DTU/MQTT format

15) 2023-07-13

- Support sensor type 25 – wind speed sensor and related payload
- Support sensor type 26 – wind dir sensor and related payload

16) 2023-07-19

- Support sensor type 27 – water/liquid flow speed
- Support sensor type 56 – generic IO duty report
- Support configuration command – loragroup

17) 2023-07-22

- Support sensor type 34 – water pressure
- Add more field description for type 56

18) 2023-08-10

- Modify the directory structure of Chapter 5
- Remove sensor manufacturer information

- Modify Style Line Spacing

19) 2023-08-31

- Fix plss flow speed type value mistake (0x6e)
- Add GPS sensor type and related payload type

20) 2023-09-05

- Add PSS-321917~PSS-32191M

21) 2023-09-22

- Add PSS-318712/PSS-21R0P3/PSS-21R0P4/PSS-21R0P5

22) 2023-09-26

- Support type-29 light

23) 2023-10-10

- Support payload type "o3" - 0x1f

24) 2023-11-03

- Support new sensor type 66 - Soil#2 with multiple data types
- Add new co data type

25) 2023-11-18

- Support new sensor type 67 - PH/ORP
- Support new sensor type 0x1001 - Pack
- Add more description for set io[n] 0xAABB for power control

26) 2023-11-21

- Correct the parameters' description for CO2NS (type 9)

27) 2023-11-30

- Support new sensor type 68 - TB600
- Support new sensor type 69 - XDIO
- Change dir unit to 0.01 degree

28) 2024-03-14

- Update various missing PLSS report types
- Add more description for VBAT/EPD

INDEX

Section 1 Management Console

Polysense iEdge4.0 products have a management console. Most products of the iEdge 4.0 family is using a USB (type-A or type-C) cable to connect from the board to the PC computer's USB/Serial ports. The PC computer can operate the iEdge4.0 products by a Terminal/Serial software (like SecureCRT or others). The serial port setup by default is:

- 9600 Baudrate
- 8 Data bits
- 1 Stop bits
- No Parity Check

The operating command are all text based. Some quick command (like '0' -reset) can support typing just one letter, for example: just input a letter of '0' in the console to trigger system reset. But most commands are ended by a CR (\r or 0x0d in hex) line ending. All examples and introductions listed in the document will omit to describe the CR line ending. But it is indeed required. See below example for choosing LoRa-US902 as the reporting type:
set report 0x4000

Commands are case insensitive. Data numbers support both hex or decimal format. For the above example, any of below command will all work same:
SET REPORT 0x4000
SeT Report 0x4000
Set report 16384

Please note:

- *1 All letters of one command should be sent to the console all together.

If SecureCRT is being used, please turn on the Command Window and typing from it. Commands typing at regular interactive window will be treated as multiple bursting commands and can not be parsed correctly.

- *2 Space character is sensitive

Do NOT use multiple space characters for one stop point. Following commands will show the wrong commands for "set report 0x4000":

- " set report 0x4000"
  Error reason: there's a SPACE before "set"
- "set report 0x4000"
  Error reason: there're two SPACEs between "set" and "report"

Section 2 Quick Commands

Polysense iEdge4.0 products support several quick commands. These commands are just one letter (for example: 0 or 9). With or without the CR line-ending will both be correct for these quick commands. See below for detailed:

0

System reset. After typing ‘0’, the console will prompt a message and wait for another ‘y’ (YES) input within several seconds. If ‘y’ is chosen before the confirming timeout, system will reset immediately.

1

List all the quick commands it can support.

2

Show running information. Like:

```
[Running Mode]
                POS: v1.0.2.3
           Free Mem: 19552
        Mininal Mem: 18256
     Running Ticks: 36690937
   Sleeping Ticks: 24623
           OS Stack: 848
          APP Stack: 904
         IDLE Stack: 64
     Startup Stack: 0x4078
          Reserved: 110000007c0000007b000000002003000000000
          Exp Sleep: 239641
```

POS – Polysense OS. Here, it prints the kernel binary firmware’s version.

Free Mem – Current free memory in unit of bytes

Minimal Mem – Minimal free memory in history

Running Ticks – Number of ms (1 tick = 1ms) from last bootup or reset

Sleeping Ticks – Number of ms (1 tick = 1ms) system sleeping

Others – Internal debug purpose

3

4

5

6

7

8

NOT recommend to use these commands. Using “xupg” commands for firmware upgrading is recommended.

9

Sometime the system is abnormal (for example always reset for unknown reason), using this command during bootup can prevent the system from taking more actions so that the console will still be stable/available for debugging purpose. Please note: this command will only take effect during the short delay of bootup section: “Waiting for remote access …”.

… (other commands)…

Internal commands. Not recommended.

Section 3 System Commands

Polysense iEdge4.0 products support following system management commands:

del *

Delete all driver firmware from flash.

del XXXX

Delete a certain driver firmware from flash. XXXX could be either an driver type or driver filename got by “list” command. For example:

del 0x4000 – Delete LoRa module firmware
del lora – Delete LoRa module firmware (same as “del 0x4000”)
del 2 – Delete DS18B20 sensor driver firmware
del ds18b20 – Delete DS18B20 sensor driver firmware
del 0x2 – Delete DS18B20 sensor driver firmware

For more detailed about the driver type/filename, please use or refer to “list” command.

help

Show all help information including all supported command list. Below is an example:

```
[APP v4.0.1.21 Help]
Type following commands in chat window:
 ADC PXn          : Show PIN (pa0~pf15) ADC Input
 CFGP             : Show PIN/CFG file
 CFGR             : Show CFG file
 DBG XXXX         : Debug commands or help info
 DEL XXXX or * : Delete Driver Type/Name (* for all)
 DUMP             : Show all EEPROM CFG (including all zero fields)
 HELP             : Show this help information
 HELP DBG/PIN/SET : Show DBG/PIN/SET help information
 INFO             : Show running information
 INP PXn          : Show PIN (pa0~pf15) Input level
 LIST             : List driver directory
 MOD PXn xxx      : Set PIN (pa0~pf15) mode
 OUT PXn 0/1      : Ouput PIN (pa0~pf15) 0/1
 PIN AAA[n] BBB: Set PIN Field(AAA[n]) to BBB
 PWD PXn xxx      : PWM ouput PIN (pa0~pf15) Duty (1~10000)
 PWM PXn xxx      : PWM ouput PIN (pa0~pf15) Hz
 PINS             : Show all PINs information
 SET AAA BBB      : Set Field(AAA) to BBB
 SHOW             : Show none-zero EEPROM CFG
```

SRUN : Set FSM State RUN
STOP : Set FSM State STOP
XUPG : Xmodem upgrade (can be any firmware, configurations and drivers)

Note:
1) BBB can be number or string according to each field
2) BBB can be * to clear this field to all ZERO/EMPTY
3) n is ZERO based

help dbg

help pin

help set

Show detailed help information for dbg/pin/set commands.

info

Show system running information, software version and running sensor drivers

list

List all driver firmwares programmed on flash. It includes both external communication driver (like LoRa, NBIOT, WIFI …) and various sensor drivers. See below example:

[DRIVER DIRECTORY]

Type	Filename	Class	Size	Disk Version
0x4000	LORA	Module	1100	1536 1.7
0x4180	BC28	Module	1628	2048 1.4
0x4181	ESP	Module	1256	1536 1.2
0x4182	N58	Module	1500	1536 1.2
0x4183	A780	Module	1272	1536 1.1
0x4184	N306	Module	2008	2048 1.7
0x42FF	CONSOLE	Module	176	512 1.0
0x0001	VBAT	Sensor	380	512 1.9
0x0002	DS18B20	Sensor	1632	2048 1.11
0x0003	FASTCNT	Sensor	528	1024 1.4
0x0004	PT100	Sensor	760	1024 1.4
0x0005	PH	Sensor	592	1024 1.4
0x0006	MTLCD	Sensor	9080	9216 1.11
0x0007	CO2	Sensor	1476	1536 1.3
0x0009	CO2NS	Sensor	3500	3584 1.4
0x000A	EPD	Sensor	11616	11776 1.7
0x000B	SHT30	Sensor	2204	2560 1.4
0x000C	O2ZE03	Sensor	676	1024 1.0

0x000D	PM	Sensor	828	1024 1.0
0x000E	UART	Sensor	620	1024 1.0
	Total		42832	48128

Total Remaining: 9216 Bytes

ota X.X.X.X

OTA download an image from remote OTA server by given versions. The system image is controlling by different major version numbers. All images are located at:

https://ota.polysense.online/iEdge4.0/history/

1.0.x.x – POS kernel image. The basic images for kernel running. For example:

1.0.2.9 for https://ota.polysense.online/iEdge4.0/history/pos_v1.0.2.9.bin

4.0.x.x – Main application image. For example:

4.0.2.4 for https://ota.polysense.online/iEdge4.0/history/main_v4.0.2.4.bin

XX.XX.x.x – Module/sensor driver images. XX.XX refers to the report type or sensor type. X.x refers to driver version. For example:

https://ota.polysense.online/iEdge4.0/history/drv4000_lora_v1.9.bin

The above lora module driver's report type is 0x4000. So XX.XX is 64.0 (0x4000 is split into 40 and 00 in hex, and corresponding to 64.0 in decimal). V1.9 is this image version. So the final version for OTA indicating this image is: 64.0.1.9

"ota 64.0.1.9" will initiate an OTA session and finally downloaded the above LoRa driver from OTA site. During this OTA session, the system might reset several times. After it's done and check the driver version by "list" command, you will see:

< list before OTA>

0x4000	LORA	Module	1124	1536 1.8

.....

< list after OTA>

0x4000	LORA	Module	1124	1536 1.9

.....

srun

Resume system from STOP to RUNNING state. Sometime, when operating the system flash through "del" or "xupg", it requires to stop the system running. After the driver upgrading or deleting operations, we can resume the system to be RUNNING again. In this case, "srun" can be used to resume system running immediately. Please note for some drivers (especially those with interrupt or very kernel combined operations) "srun" might not always work. In any case, a system reset after driver upgrading is always the best way to use. Here, "srun" is just a debug command for fast response.

stop

Stop the system running. Especially when upgrading any driver firmware, system should be stop in advance than driver flash programming. After all operation is done, a “srun” command can be called to resume system normal running or a system reset can also work for any cases.

xupg

Upgrade any firmware through XMODERM protocol. Please note for safe upgrading operation, following sequence should be used:

1) Issue command “stop”
2) Wait and check system running status by command “info” to make sure system is in STOP state
3) Issue command “xupg” and choose the firmware downloaded from https://ota.polysense.online/iedge4.0/
4) Wait until upgrading is done
5) Some upgrading will automatically reset after upgrading. Some will just leave the system as “stop”, in this case, issue command “srun” to re-enable the system or a system reset.

Section 4 Generic Configuration

Polysense iEdge4.0 products support following generic configuration commands. Besides these, there’re also a group of sensor related commands. See below about the generic configuration commands:

cfgr

Show the current configuration file. The content could be saved as a file (removing [] headers, leaving only “set xxxx” commands and making sure the final command having a CR line-ending) and upgraded to boards through “xupg”commands. Below is an example of “cfgr” content:

```
set Magic 0x92747613
set Report 0x4001
set CTRL 0x5
set EUI 7a20bdfffebc963c
set LoRaEUI 7a20bdfffebc963c
set LoRaAPP 2017060000000000
set LoRaKey 11223344556677889900aabbccddeeff
set LoRaAckLmt 0x40
set LoRaAckDly 0x20
set LoRaNBTrans 0x1
```

set LoRaTX 2
set LoRaRX 222
set ResetModule 8
set ResetSystem 24
set Cycle[0] 300
set Type[0] 0x1
set Cycle[1] 300
set Type[1] 0xe
set IO[1] 0x2
set Publish /iot/00000000/up/7a20bdfffebc963c
set Subscribe /iot/00000000/dn/7a20bdfffebc963c
set MUSR RDdemo
set MPWD polysense_0379
set MCID plss_7a20bdfffebc963c
set DNS 8.8.8.8
set Port 1560
set URL ota.polysense.online
set BAND 5,8
set APN cmnbiot
set SSID plss_wifi
set WPWD polysense

dump

show

Obsolte. Please use “cfgr” for configuration showing.

set magic XXXX

Internal use. Do NOT use it.

set report XXXX

Setup uplink report type. Command “list” for Class Module will display all current supported reporting types. Please note if 0x41xx is supported, 0x43xx is also supported by same driver firmware. Below is an overall list:

0x4000 – LoRa US902
0x4001 – LoRa CN470
0x4002 – LoRa EU868
0x4003 – LoRa EU779
0x4004 – LoRa EU433
0x4005 – LoRa EU470 (Private, not for public)
0x4006 – LoRa AS923
0x4007 – LoRa RU864
0x4008 – LoRa AU915

0x40ff – LoRa various regions
0x4180 – BC28 family NBIOT/4G/LTE/CAT-1 MQTT report
0x4181 – ESP family WIFI MQTT report MQTT report
0x4182 – N58 family NBIOT/4G/LTE/CAT-1 MQTT report
0x4183 – A780 family NBIOT/4G/LTE/CAT-1 MQTT report
0x4184 – N306 family NBIOT/4G/LTE/CAT-1 MQTT report
0x4185 – BG95 family NBIOT/4G/LTE/CAT-1 MQTT report
0x42ff – Local console report (debug purpose)
0x4380 – BC28 family NBIOT/4G/LTE/CAT-1 Polysense IVIEW report
0x4381 – ESP family WIFI MQTT Polysense IVIEW report
0x4382 – N58 family NBIOT/4G/LTE/CAT-1 Polysense IVIEW report
0x4383 – A780 family NBIOT/4G/LTE/CAT-1 Polysense IVIEW report
0x4384 – N306 family NBIOT/4G/LTE/CAT-1 Polysense IVIEW report
0x4385 – BG95 family NBIOT/4G/LTE/CAT-1 Polysense IVIEW report

set ctrl XXXX

Bitmap controlling for different functions:
Bit[0x8000] – LoRa Report also using JSON format
Bit[0x4000] – Using module power-saving mode instead of power-off mode
Bit[0x2000] – LED long (internal debug purpose)
Bit[0x1000] – DNS strict TTL time (internal debug purpose)
Bit[0x0800] – DTU working mode
Bit[0x0008] – Module AT debug printing
Bit[0x0004] – Driver debug printing
Bit[0x0002] – System running debug printing
Bit[0x0001] – Report data console log printing
For example:
set ctrl 0x4001 - Turn ON Bit[0x0001] for console log printing and NBIOT PSM
set ctrl 0x400f - Turn ON all debug and NBIOT PSM
set ctrl 0x4000 – Just turn ON NBIOT PSM

Note: For LoRa/WIFI(ESP)/4G modules, please use 0 (set ctrl 0) for this field.

set eui XXXX

Set the device EUI by giving number. For example:
Set eui 7a20bdfffebc963c

WARNING: DO NOT CHANGE THIS FIELD.

set loraeui XXXX

Set the LoRa module EUI used by LoRa OTA. Typically, this value should be same as the device EUI set by “set eui XXXX” commands.

set loraapp XXXX

Set the LoRa OTA Application EUI. For example:

set LoRaAPP 2017060000000000

set loractrl XXXX

Set the LoRa common working conditions. This is a bitmap definition. See below:

Bit[0] – Use fixed OTA DEVNOCNE: 0-Auto, 1-Fixed DEVNONCE

Bit[1] – Set ADR enable: 0-ADR OFF, 1-ADR ON

Bit[2-7] – Reserved

For example:

set loractrl 2

ADR ON and Auto DEVNONCE.

set loractrl 0

ADR OFF and Auto DEVNONCE.

set loractrl 3

ADR ON and fixed DEVNONCE.

set loraflags XXXX

Set the LoRa working flags. This is a bitmap definition. See below:

Bit[0] – Set Confirm/Unfirm: 0-Unfonfirm MSG, 1-Confirm MSG

Bit[1-7] – Reserved

set loragroup XXXX

Set the LoRa frequency group for each band. This is a bitmap definition. See below:

Bit[0..3] – Group From

Bit[4-7] – Group to

For example, 0x11 for Group #1 to #1; 0x00 for Group #0 to #0.

For some regions like AS923, only Group From is used:

Group 0 – Regular AS923

Group 1 – Regular AS923-2 (OFFSET= -1.8M)

Group 2 – Regular AS923-3 (OFFSET= -6.6M)

set lorakey XXXX

Set the LoRa OTA Application key. For example:

set LoRaKey 11223344556677889900aabbccddeeff

set loranonce XXXX

Set the LoRa OTA JOIN DEVNONCE. For example:

Set loranonce 0x1234

Please note this field is only valid when Bit[1] of “loractrl” field is 1.

set loraaskey XXXX

set loranskey XXXX

set loraaddr XXXX

set loramodel XXXX

set loraacklmt XXXX

set loraackdly XXXX

set loranbtrans XXXX

Internal use.

set loratx XXXX

Set the LoRa TX FPORT. All uplink LoRa payload will be sent through this FPORT number.

set lorarx XXXX

Set the LoRa RX FPORT. Only content received from this FPORT number will be received.

set resetmodule XXXX

Set the module reset timeout times. If continuous X times of ACK is not received, the external communication module will be power reset.

set resetsystem XXXX

Set the whole system reset timeout times. If continuous X times of ACK is not received, the whole system will be reset.

set gsrv XXXX

Internal reserved field.

set console XXXX

Internal reserved field.

set fixed XXXX

Set the number of DS18B20 ROM IDs saved on flash. DS18B20 will only report the sensor temperature with matching ROM IDs. If this number is ZERO, DS18B20 will scan how many sensors now and save those ROM ID/number onto flash. So if you replace the DS18B20 sensor, please remember to clear its saving ROM IDs by – “set fixed 0”

set rom[N] XXXX

Set the saved #N (N=0..15) DS18B20 ROM ID to given value. For example:
set ROM[0] 288e02380d000020
set ROM[1] 2821ac360d0000dc
Usually, “fixed” and “rom[N]” field will be auto-updated by first time DS18B20 is scanned. In case, a new group of DS18B20 sensors are installed/replaced; use command “set fixed 0” to clear all previous saving ROM IDs. When “fixed” is zero, “rom[N]” will all be ignored.

Set rom[0] Set the number of DS18B20 ROM IDs saved on flash. DS18B20 will only report

the sensor temperature with matching ROM IDs. If this number is ZERO, DS18B20 will scan how many sensors now and save those ROM ID/number onto flash. So if you replace the DS18B20 sensor, please remember to clear its saving ROM IDs by – "set fixed 0"

set publish XXXX

Set the MQTT publishing topic. For example:

set Publish /iot/00000000/up/7a20bdfffebc963c

set subscribe XXXX

Set the MQTT subscribing topic. For example:

set Subscribe /iot/00000000/dn/7a20bdfffebc963c

set musr XXXX

Set the MQTT username. For example:

set MUSR RDdemo

set mpwd XXXX

Set the MQTT user password. For example:

set MPWD polysense_1234

set mcid XXXX

Set the unique MQTT Client ID. For example:

set MCID plss_7a20bdfffebc963c

Note: to prevent from using two same MCID, a string containing device EUI is a good choice.

set qosp XXXX

set qoss XXXX

Set the MQTT publishing/subscribing QOS level number. 0 is recommended.

set dns X.X.X.X

Set the DNS server IP. For example:

set DNS 8.8.8.8

set port XXXX

Set the MQTT Server or Polysense IVIEW Server port. For example:

set Port 1560

Note: Polysense IVIEW uses 1560 port and most MQTT server uses 1883 port. Be aware of using the correct port number corresponding to the current report type.

set URL XXXX

Set the MQTT Server or Polysense IVIEW Server IP or URL address. For example:

set URL ota.polysense.online

set URL 47.104.85.77

Note: when IP address is using, DNS will be skipped and report time can be a little bit faster than URL.

set BAND XXXX

Set the NBIOT/CAT-1 working frequency band. For example:
set BAND 5,8

set APN XXXX

Set the NBIOT/CAT-1 APN. For example:
set APN cmnbiot

set SSID XXXX

Set the WIFI SSID used for WIFI report type. For example:
set SSID plss_wifi

set WPWD XXXX

Set the WIFI password used for WIFI report type. For example:
set WPWD pwd12345

set NRSV XXXX

set WRSVD XXXX

set retry XXXX

set delay XXXX

Reserved field. Do NOT change these fields.

Section 5 Sensor Configuration

Polysense iEdge4.0 products support 8 (#0~#7) sensor controlling slots. Each slot can choose one sensor type and related reporting cycle, IO and power delay parameters. Thus, the products can support at most 8 sensors. Please note:

1) A slot is disabled when its cycle is 0
2) Two or multiple slots can have same sensor type ID. In this case, this sensor driver will be executed/reported twice (no matter its IO/parameters are same or not).

All sensor configuration (except for “type” field) can have two kinds of provisions:

1) Set AAA[N] XXXX

   Set slot #N (N=0..7)’s sensor field AAA to XXXX
   For example:
   Set type[0] 1
   Set cycle[0] 60
   Set type[1] 2
   Set cycle[1] 60

2) Set AAA XXXX

   Set all slots’ sensor field AAA to XXXX
   For example:
   Set type[0] 1
   Set type[1] 2
   Set cycle 60
   The above “set cycle 60” will work same as two commands: “set cycle[0] 60” and “set cycle[1] 60”.

Content in this section will describe using the format of “set AAA[N] XXX”. See below:

- set cycle[N] XXXX

  Set the reporting cycle to XXXX. For example:
  set cycle[0] 86400

- set type[N] XXXX

  Set the sensor type XXXX. Use “list” command to read what kinds of sensor types supporting now. For example:
  set type[1] 14

- set pid[N] XXXX

  Internal use to record the internal Sensor Product ID.

- set power[N] XXXX

  Set this slot#N’s power heating time in unit of ms before data collecting. For example:
  Set power[1] 1000

The number of power will have some special meanings. See below:

0 – No heating time and operating data collecting immediately after power on

1~0xffef – Power heating time in unit of ms

0xfff0~0xfffd – 2/4/8/16/… minutes of heating time

0xfffe – Never turn ON the senors’ s power

0xffff – Never turn OFF the senors’ s power

- set io[N] XXXX

Set this slot#N's IO channel and Power control ID. This field XXXX is manupulated by two secionts. AA and BB. For example set io[n] 0xAABB. 0xBB is using an ID from 0~N, indicating its IO slot channel. For UART sensors, this will mean UART0~UARTn. For I2C sensors, this will mean I2C0~I2Cn. 0xAA is also using an ID from 0~0x3f, indicating its POWER control. Below is the power control ID definitions:

0: Default power control ID. Sensor driver will overwrite this to each driver's default power control.

1: Power control 3.3V

2: Power control for PT100

3: Power control for WIRE1

4: Power contol for external module

5: Power control for Analog

6: Power control for PWRH and PWRH_EN

7: Power control of 3.3V OpenDrain

8: Power control of 5V OpenDrain

9: Power control of LED

10: Power control of VBAT sampling

11~15: User defined sensor/modle power control

16: Power control for PWRH_EN

61: Power control for PWRH only

62: Power control for PWRH only and 3.3V

63: Power control for PWRH and PWRH_EN and 3.3V

Example 1), if a PM2.5 sensor is going to use 12V power and also turning on 3.3V for UART2. It should be set by:

set io[n] 0x3f02 (0x3f is 63 for turnning on 12V and 3.3V, 2 is for UART2).

Example 2), if a sensor is going to use 12V only and 485 (UART3), it should be set by:

set io[n] 0x603 (0x06 is for 12V and 3 is UART3/485)

Example 3), if a sensor driver will use Power 3.3V by default and I2C#1 is be chosen, it should be set by:

set io[n] 1 (Leave the higher bits as ZERO for default power control, only lowest bits 1 is set indicating I2C1)

- set srsv[N] XXXX
- set arg[N] XXXX
- set d0~2[N] XXXX
- set thw0[N] XXXX
- set thw1[N] XXXX
- set thld[N] XXXX

These parameter fields will be used differently by each sensor type. Blow is the detailed:

PSS-21 gas sensor

PSS-21E0J1

Sensor Type – 7 (CO2)

Collect Sensor CO2 PPM. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect CO2 from UART2.

"set io[N] 0"

Collect CO2 from UART0.

◆ SRSV – Bit[0..7]: Calibration control

Specify the sensor calibration action.

0 – Not calibration

1 – Do Calibration in next duty and the driver will clear this field to 0 automatically after calibration.

For example:

"set srsv[N] 1"

CO2 calibration will be performed on next duty and then auto clear this field to zero.

◆ SRSV – Bit[8..31]: Sensor UART baudrate

If it's 0 (default), 9600 will be used, or else use the specified Baudrate. For example:

"set srsv[N] 0x01c20000"

Collect CO2 using 0x01c200 (115200) baudrate .

PSS-21E0J2

Refer to PSS-21E0J1.

PSS-21E0J3

Refer to PSS-21E0J1.

PSS-21E0J4

Refer to PSS-21E0J1.

PSS-21E0J5

Refer to PSS-21E0J1.

PSS-21E0J6

Refer to PSS-21E0J1.

PSS-21E0J7

Sensor Type – 9 (CO2NS)

Collect CO2NS sensor CO2 PPM. This sensor always uses 9600 baudrate and CTRL5 to capture the sensor signal ready input. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect CO2 from UART2.

"set io[N] 0"

Collect CO2 from UART0.

- SRSV – Bit[0..7]: Calibration control

Specify the sensor calibration action.

0 – Not calibration

1 – Do Calibration in next duty and the driver will clear this field to 0 automatically after calibration.

For example:

"set srsv[N] 1"

CO2 calibration will be performed on next duty and then auto clear this field to zero.

Note: DO NOT use this manual calibration and always leave this field as 0; this sensor will supporting auto-calibration by given times of reporting.

- SRSV – Bit[8..15]: Averaging time in unit of seconds

If it's 0 (default), 10s will be used, or else use the specified averaging time. The sensor will try to sample as much as possible during given averaging time. Final reported number is calculated by the AVG of all sampling data made in this duration (MIN/MAX are removed before AVG). For example:

"set srsv[N] 0x1400"

Collect CO2 using 20s averaging time.

- SRSV – Bit[16..23]: Sleeping time after each time of collecting

If it's 0 (default), 0s will be used, or else use the specified sleeping time. During each

averaging duration, the sensor will collect multiple times; this field defines the sleeping time between two collectings. Making this field longer will reduce the collecting times for each average duration. So 0 is recommended here.

- SRSV – Bit[24..28]: Moving filter ratio in unit of 5 percentage

If it's 0 (default), 50 (50%) will be used, or else use the specified moving ratio x 5%. Each round the sensor get a final value, it will be filtered as following formula:

MOVING_RATIO = Bit[24..28] * 5

NEW_FINAL = OLD + (NEW_COLLECT – OLD) * MOVING_RATIO / 100

For example, when 10 is used, NEW_COLLECT=600 PPM, OLD=500 PPM, the final value collected on this new duty would be : 500 + (600-500) * 0.5 = 550 PPM.

Below is an example of changing this ratio to 75% (using 0xf or 15, 15 x 5% = 7%):

"set srsv[N] 0x0f000000"

- SRSV – Bit[31]: User button for calibration
  - 0 - Do NOT response user button
  - 1 - When user button is pushed, sensor will be forced to do a calibration in the nearest time
- DO – Bit[0..15]: Sampling offset adjust
- DO – Bit[16..31]: Sampling ratio

Each time a raw data is collected by the sensor, these two fields will be applied. The formula is:

NEW_COLLECT = (RAW + SAMPLING_OFFSET) * SAMPLING_RATIO / 100

Note: 1) SAMPLING OFFSET is a signed integer, a value of 0xffff will be -1 for this case.

NEW_COLLECT is still not the new FINAL data. It still needs to be calculated by moving filter ratio. The new final data would be calculated by:

NEW_FINAL = OLD + (NEW_COLLECT – OLD) * MOVING_RATIO / 100 = OLD + ((RAW + SAMPLING_OFFSET) * SAMPLING_RATIO / 100 – OLD) * MOVING_RATIO / 100

For example:

"set d0[N] 0x16FE00"

Use offset=-200 (0xFE00) ratio=110% (0x16 * 5 = 110). to get the collecting data. In this case, if sensor raw data is 746 PPM, the new collecting data would be: (746-200)*1.10 = 600 PPM. If moving filter ratio is 50%, OLD=500, the new final data reported in this duty would be: 500 + (600-500) * 0.5 = 550 PPM.

- THW0 – Enable long sleep when CO2PPM is bigger than this value
- THW1 – Enable long sleep when delta of CO2PPM is bigger than this value
- THW2/THW3 – Enable long sleep when time is NOT within this range
  - THW2 - Working time range FROM (for example 0x081e for 08:30)
  - THW3 - Working time range TO (for example 0x1200 for 18:00)
- THW4 – Long sleep time in unit of seconds. When long sleep is abled by any of the above criteria, the system will perform the given sleeping time for power saving.

PSS-21E0J8

Refer to PSS-21E0J8.

PSS-21E0J9

Refer to PSS-21E0J8.

PSS-21H061

Sensor Type – 19

Collect various data from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWRH pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

PSS-21H062

Refer to PSS-21H061.

PSS-21H063

Refer to PSS-21H061.

PSS-21H091

Refer to PSS-21H061.

PSS-21H092

Refer to PSS-21H061.

PSS-21H093

Refer to PSS-21H061.

PSS-21H0A1

Refer to PSS-21H061.

PSS-21H0A2

Refer to PSS-21H061.

PSS-21H0A3

Refer to PSS-21H061.

PSS-21H0B1

Sensor Type – 12 (O2)

Collect Sensor O2 concentration. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Reserved
- SRSV – Bit[8..31]: Sensor UART baudrate

If it's 0 (default), 9600 will be used, or else use the specified Baudrate. For example:

"set srsv[N] 0x01c20000"

Collect using 0x01c200 (115200) baudrate .

PSS-21H0B2

Refer to PSS-21H0B1.

PSS-21H0B3

Refer to PSS-21H0B1.

PSS-21H0C1

Refer to PSS-21H061.

PSS-21H0C2

Refer to PSS-21H061.

PSS-21H0C3

Refer to PSS-21H061.

PSS-21H0D1

Refer to PSS-21H061.

PSS-21H0D2

Refer to PSS-21H061.

PSS-21H0D3

Refer to PSS-21H061.

PSS-21H0E1

Refer to PSS-21H061.

PSS-21H0E2

Refer to PSS-21H061.

PSS-21H0E3

Refer to PSS-21H061.

PSS-21H0F1

Refer to PSS-21H061.

PSS-21H0F2

Refer to PSS-21H061.

PSS-21H0F3

Refer to PSS-21H061.

PSS-21H0H1

Refer to PSS-21H061.

PSS-21H0H2

Refer to PSS-21H061.

PSS-21H0H3

Refer to PSS-21H061.

PSS-21H0P1

Refer to PSS-21H061.

PSS-21H0P2

Refer to PSS-21H061.

PSS-21H0P3

Refer to PSS-21H061.

PSS-21H0U1

Refer to PSS-21H061.

PSS-21H0U2

Refer to PSS-21H061.

PSS-21H0U3

Refer to PSS-21H061.

PSS-21H0V1

Refer to PSS-21H061.

PSS-21H0V2

Refer to PSS-21H061.

PSS-21H0V3

Refer to PSS-21H061.

PSS-21H0W1

Refer to PSS-21H061.

PSS-21H0W2

Refer to PSS-21H061.

PSS-21H0W3

Refer to PSS-21H061.

PSS-21H611

Refer to PSS-21H061.

PSS-21H612

Refer to PSS-21H061.

PSS-21H613

Refer to PSS-21H061.

PSS-21S131

Sensor Type – 13 (PM)

Collect PM1.0/2.5/10 Sensor. This driver uses 9600 baudrate. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR6 (5V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x800"

Turn ON/OFF PWR8 (5V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0.

PSS-21R090

//todo

PSS-21R091

Sensor Type – 68 (TB600)

Collect TB600 multi sensor. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control and turn on the PWR3 for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x600"

Turn ON/OFF PWRH as the power control.

- ◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- ◆ D0 – Bit[0..7]: Bitmap control for sub-data reporting. For each bit, '0' will skip and '1' will report this sub-data. Refer to below:
  - Bit[0] – Main sensor data (varied for each installed GAS module)
  - Bit[1] – Temperature
  - Bit[2] – Humidity

  Note, if this D0-Bit[0..7] is zero, the driver will used 0x07 for default, meaning Main/Humidity/Temperature will all be reported.

PSS-21R093

Refer to PSS-21R091.

PSS-21R094

Refer to PSS-21R091.

PSS-21R095

Refer to PSS-21R091.

PSS-21R096

Refer to PSS-21R091.

PSS-21R0H1

Refer to PSS-21R091.

PSS-21R0H2

Refer to PSS-21R091.

PSS-21R0H3

Refer to PSS-21R091.

PSS-21R0H4

Refer to PSS-21R091.

PSS-21R0H7

//todo

PSS-21R0H8

//todo

PSS-21R0H9

//todo

PSS-21R0HA

//todo

PSS-21R0P3

Refer to PSS-21R091.

PSS-21R0P4

Refer to PSS-21R091.

PSS-21R0P5

Refer to PSS-21R091.

PSS-22 humidity sensor

PSS-22B011

//Todo

PSS-22B021

Senor type is 0x11, Refer to PSS-234021.

PSS-23 ion sensor

PSS-232072

Sensor Type – 18 (NPK)

Collect NPK index from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control and turn on the PWR3 for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

◆ D0 – Bit[0..7]: Bitmap control for sub-data reporting. For each bit, ‘0’ will skip and ‘1’ will report this sub-data. Refer to below:

- Bit[0] – N Index
- Bit[1] – P Index
- Bit[2] – K Index

Note, if this D0-Bit[0..7] is zero, the driver will used 0x07 for default, meaning N/P/K will be reported.

“set d0[N] 1”

Collect only N Index

PSS-234011

Sensor Type – 67 (PH_OPR)

Collect PH/OPR from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control and turn on the PWR3 for high voltage; or else, use the specified PWR control. For example,

“set io[N] 0x600”

Turn ON/OFF PWRH as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

“set io[N] 3”

Collect from UART3.

“set io[N] 0”

Collect from UART0.

◆ SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

“set srsv[N] 0x01”

Collect using device address 01.

PSS-234021

Sensor Type – 5 (PH)

Collect PH index from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

“set io[N] 0x700”

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect PH from UART2.

"set io[N] 0"

Collect PH from UART0.

- SRSV – Bit[0..7]: MODBUS Device Address (0~0xff)

Specify the sensor's MODBUS device address. For example:

"set srsv[N] 1"

Collect PH using MODBUS address 1 and default baudrate.

- SRSV – Bit[8..31]: Sensor UART baudrate

If it's 0 (default), 9600 will be used, or else use the specified Baudrate. For example:

"set srsv[N] 0x01c20003"

Collect PH using MODBUS address 3 and 0x01c200 (115200) baudrate .

PSS-234022

Refer to PSS-234021.

PSS-31 mechanical sensor

PSS-311121

Sensor Type – 34 (Water Pressure)

Collect analog water pressure. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: User defined AIN pin

If io (bit0..7) is 0 (default), it will use AIN4 collecting; or else, use the specified AIN pin. For example:

"set io[N] 2"

Collect from AIN2.

- D0 – Bit[0..15]: Result calculation ration. When zero, 1280 will be used for this field.
- D0 – Bit[16..31]: Result calculation offset. When zero, 100 will be used for this field.

Final result is calculated with:

(AnalogMV – D0[16..31]) * D0[0..15]

With default 1280/100, it refers to 100~1350ms for 1.6MPa final result

PSS-311122

Refer to PSS-311121.

PSS-311123

Refer to PSS-311121.

PSS-311124

Refer to PSS-311121.

PSS-311125

Refer to PSS-311121.

PSS-311126

Refer to PSS-311121.

PSS-311127

Refer to PSS-311121.

PSS-311128

Refer to PSS-311121.

PSS-315511

Sensor Type – 11 (SHT30, HPA, VOC, … I2C sensors)

Collect SHT30/HPA/VOC/… I2C sensors. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x100"

Turn ON/OFF PWR1 (3.3V) as the power control.

- IO – Bit[0..7]: I2C ID (0..3)

Specify the I2C ID (0..7). For example:

"set io[N] 1"

Collect from I2C1.

"set io[N] 0"

Collect from I2C0.

- SRSV – Bit[0..7]: Sensor selecting

Bitmap definition for different sensors.

Bit[0] – Temperature

Bit[1] – Humidity

Bit[2] – HPA

Bit[3] – VOC

Bit[4] – Light

Bit[5..7] – Reserved

For each bit, 0 – Not selected; 1 – Selected.

If all these fields are zero, the driver will treat it as 0xff and try to scan/collect all the above sensors.

- SRSV – Bit[8..31]: Reserved field

Leave it as all zero.

PSS-316631

Refer to PSS-316691.

PSS-316691

Sensor Type – 27 (Water/liquid flow speed)

Collect water (liquid) flow speed (m3/H). Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

PSS-3166A1

Refer to PSS-316691.

PSS-3166B1

Refer to PSS-316691.

PSS-3166C1

Refer to PSS-316691.

PSS-3166D1

Refer to PSS-316691.

PSS-3166E1

Refer to PSS-316691.

PSS-3166F1

Refer to PSS-316691.

PSS-3166G1

Refer to PSS-316691.

PSS-3166H1

Refer to PSS-316691.

PSS-3166J1

Refer to PSS-316691.

PSS-3166K1

Refer to PSS-316691.

PSS-3166L1

Refer to PSS-316691.

PSS-3166M1

Refer to PSS-316691.

PSS-3166N1

Refer to PSS-316691.

PSS-3166P1

Refer to PSS-316691.

PSS-3166Q1

Refer to PSS-316691.

PSS-317611

//Todo

PSS-318711

Sensor Type – 65 (GPS)

Collect GPS position data. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0

- SRSV – Bit[0..7]: Collecting time

Specify the sensor's collecting time in unit of 5s. For example:

"set srsv[N] 2"

Collect at most 10s.

- SRSV – Bit[8..31]: Sensor UART baudrate

If it's 0 (default), 9600 will be used, or else use the specified Baudrate. For example:

"set srsv[N] 0x01c20003"

Collect at most 15s and 0x01c200 (115200) baudrate .

PSS-318712

Refer to PSS-318711.

PSS-319211

//Todo

PSS-319212

Sensor Type – 26 (Wind Dir)

Collect wind direction from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWRH pin for power control and turn on the PWRH_EN for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

PSS-319311

Refer to PSS-315511.

PSS-32 thermal sensor

PSS-321111

Sensor Type – 2 (DS18B20)

Collect DS18B20 temperatures. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 (WIRE1 PWR) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x100"

Turn ON/OFF PWR1 (3.3V) as the DS18B20 power control.

- IO – Bit[0..7]: User defined DS18B20 WIRE1 PIN

Internal used. By default, it will use AIN3 (WIRE1) pin. Please use it as 0 (default).

PSS-321112

Refer to PSS-321111.

PSS-321113

Refer to PSS-321111.

PSS-321911

Sensor Type – 4 (PT100)

Collect PT100 temperature. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR2 (VCC PT100) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x100"

Turn ON/OFF PWR1 (3.3V) as the power control.

- IO – Bit[0..7]: User defined PT100 pin

If io (bit0..7) is 0 (default), it will use AIN6 (AIN PT100) for temperature collecting; or else, use the specified AIN pin. For example:

"set io[N] 2"

Collect PT100 temperature from AIN2.

PSS-321912

Refer to PSS-321911.

PSS-321913

Refer to PSS-321911.

PSS-321914

Refer to PSS-321911.

PSS-321915

Refer to PSS-321911.

PSS-321916

Refer to PSS-321911.

PSS-321917

Refer to PSS-321911.

PSS-321918

Refer to PSS-321911.

PSS-321919

Refer to PSS-321911.

PSS-32191A

Refer to PSS-321911.

PSS-32191B

Refer to PSS-321911.

PSS-32191C

Refer to PSS-321911.

PSS-32191D

Refer to PSS-321911.

PSS-32191E

Refer to PSS-321911.

PSS-32191F

Refer to PSS-321911.

PSS-32191G

Refer to PSS-321911.

PSS-32191H

Refer to PSS-321911.

PSS-32191J

Refer to PSS-321911.

PSS-32191K

Refer to PSS-321911.

PSS-32191L

Refer to PSS-321911.

PSS-32191M

Refer to PSS-321911.

PSS-33 optical sensor

PSS-331011

Sensor Type – 22 (Laser Distance)

Collect laser distance from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

PSS-331021

Refer to PSS-331021.

PSS-331022

Refer to PSS-331021.

PSS-331023

Refer to PSS-331021.

PST-333043

Sensor Type – 64 (Counter12)

Collect bi-dir IRQ based counters. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use NO power control; or else, use the specified PWR control. For example,

"set io[N] 0x603"

Turn ON/OFF PWRH as the power control.

- IO – Bit[0..7]: IRQ#1 DIO index

Specify the DIO ID (0..15) as the IRQ#1 source. For example:

"set io[N] 3"

- SRSV – Bit[0..15]: IRQ#2 DIO index

Specify the DIO ID (0..15) as the IRQ#2 source. For example:

"set srsv[N] 4

- SRSV – Bit[16]: Single, or Dual IRQ polarity trigger mode
  - 0 – Both RISING and FALLING Trigger
  - 1 – Single Polarity trigger
- SRSV – Bit[17]: Single IRQ polarity
  - 0 – Falling trigger
  - 1 – Rising trigger

For example:

"set srsv[N] 0x10004 (Single Falling Trigger with DIO#4 is IRQ#2 source)

PST-333045

Refer to PSS-333043.

PSS-333051

Sensor Type – 48 (Generic IO Control)

This sensor will works as same as sensor type – 3 (Fastcnt), except that Type -3 (Fastcnt) will report cnt8 through PLSS_PL – 56 and this sensor will use PLSS_PL – 108. For more detailed, please refer to Sensor Type – 3 (Fastcnt).

PSS-333052

Refer to PSS-333051.

PSS-333053

Refer to PSS-333051.

PSS-334062

Sensor Type – 32 (UV)

Collect UV power. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: User defined AIN pin

If io (bit0..7) is 0 (default), it will use AIN4 collecting; or else, use the specified AIN pin. For example:

"set io[N] 2"

Collect from AIN2.

- D0 – Multi-ratio from ADC to final result in unit of 0.0001. For example, 10000 for 1.0. By default when this field is zero, 20151 (2.0151) will be used for this sensor.

"set d0[N] 20000"

Set sensor ADC to result ratio as 2.0000. If ADC is 1000mv, final result will be 4000 mw/cm2.

PSS-33B011

Refer to PSS-315511.

PSS-33M011

Refer to PSS-333051.

PSS-34 magnetic sensor

PSS-341011

Refer to PSS-333051.

PSS-35 electrical sensor

PSS-351020

Sensor Type – 33 (MV)

Collect analog voltage (unit of mv). Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

◆ IO – Bit[0..7]: User defined AIN pin

If io (bit0..7) is 0 (default), it will use AIN4 collecting; or else, use the specified AIN pin. For example:

"set io[N] 2"

Collect from AIN2.

◆ D0 – Multi-ratio from ADC to final result in unit of 0.0001. For example, 10000 for 1.0. By default when this field is zero, 10000 (1.0000) will be used for this sensor.

"set d0[N] 13300"

Set sensor ADC to result ratio as 1.3300. If ADC is 2000mv, final report value will be 2660mv.

◆ SRSV – Bit[0]: MV or current report

If SRSV (bit0) is 0 (default), it will report mv voltage; or else, report as current. For example:

"set srsv[N] 1"

Report current.

PSS-351021

Refer to PSS-351020.

PSS-351022

Refer to PSS-351020.

PSS-351023

Refer to PSS-351020.

PSS-351024

Refer to PSS-351020.

PSS-351025

Refer to PSS-351020.

PSS-351026

Refer to PSS-351020.

PSS-351027

Refer to PSS-351020.

PSS-351028

Refer to PSS-351020.

PSS-351029

Refer to PSS-351020.

PSS-35102A

Refer to PSS-351020.

PSS-3530C1

Sensor Type – 3 (FASTCNT)

Collect PIN interrupt counter. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will NOT use any power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: User defined DIO pin for interrupt collecting

If io (bit0..7) is 0 (default), it will use DIO0 for interrupt trigger; or else, use the specified DIO pin. For example:

"set io[N] 2"

Collect interrupt from DIO2.

Note: as INTTERUPT is always turn ON, related power control should be stable during the background interrupt working . So this sensor's power heating time should be 0xffff (Always ON) in most cases, which means that "set power[N] 0xffff" is recommended for this kind of sensor unless it does NOT require any power control.

- SRSV – Bit[0]: Counter Read control, 0-Read/Keep, 1-Read/Clear
- SRSV – Bit[1]: Counter report mode, 0-Counter+State, 1-Counter only
- SRSV – Bit[2]: Counter size, 0-8 bit, 1-32 bit
  - 8Bit counter will use PLSS_PL – 56 (Leakage)
  - 32Bit counter will use PLSS_PL -19
- SRSV – Bit[3]: Event report control, 0-Event report is enabled, 1-Disbled
- SRSV – Bit[4]: IRQ triggering, 0-Falling+Rising trigger, 1-Single polarity trigger (using Bit[5] definitions)
- SRSV – Bit[5]: IRQ single trigger polarity, 0-Falling trigger, 1-Rising trigger. (only used when Bit[4] is 1)
- SRSV – Bit[6]: OTA EVENT/ACT command response, 0-Response EVENT/ACT command, 1-Not Response
- SRSV – Bit[8..15]: Power ID for EVENT/ACT response (When EVENT/ACT OTA command is issued, the driver will ENABLE this power until IO is ZERO and then DISBALE this power)

PSS-3530D1

Refer to PSS-3530C1.

PSS-355031

//todo

PSS-355032

Sensor Type – 25 (Wind Speed)

Collect wind speed from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWRH pin for power control and turn on the PWRH_EN for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

PSS-36 acoustic sensor

PSS-362011

Refer to PSS-362013.

PSS-362012

Refer to PSS-362013.

PSS-362013

Sensor Type – 16 (Sound)

Collect sound/noise level from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWRH pin for power control and turn on the PWRH_EN for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

PSS-362014

Refer to PSS-362013.

PSS-362021

Refer to PSS-362013.

PSS-362022

Refer to PSS-362013.

PSS-362023

Refer to PSS-362013.

PSS-363021

Sensor Type – 15 (Ultra-Sonic Distance)

Collect ultra-sonic distance from UART (RS232 or RS485) ports. The driver firmware will always choose 9600 to communicate with this sensor. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0.

PSS-363022

Sensor Type – 20 (Multi-Sensor Ultra-Sonic Distance)

Collect multi-sensor ultra-sonic distance from UART (RS232 or RS485) ports. The driver firmware will always choose 9600 to communicate with this sensor. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01

- D0 – Bit[0..7]: Number of sensors attached. This can link multiple sensors with different address like: N, N+1, N+2, ···. Only the first sensor's address N should be set.

PSS-4X composite sensor

PSS-403011

Refer to PSS-315511.

PSS-403012

Refer to PSS-315511.

PSS-403013

Refer to PSS-315511.

PSS-403014

Refer to PSS-315511.

PSS-403015

Refer to PSS-315511.

PSS-403016

Refer to PSS-315511.

PSS-403017

Refer to PSS-315511.

PSS-403018

Refer to PSS-315511.

PSS-403019

Refer to PSS-315511.

PSS-40301A

Refer to PSS-315511.

PSS-40301B

Refer to PSS-315511.

PSS-40301C

Refer to PSS-315511.

PSS-403021

Refer to PSS-315511.

PSS-403023

Refer to PSS-315511.

PSS-403031

Refer to PSS-315511.

PSS-403091

//todo

PSS-423031

Sensor Type – 23 (ZPHS Multi Sensor)

Collect multiple data from ZPHS multi sensor. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- D0 – Bit[0..14]: Report bitmap control. If this field is ZERO, 0xff will be used. For each bit, 0-Not Report, 1-Report
  - Bit[0]: CO2/PPM
  - Bit[1]: VOC/CH2O (Refer to D0-Bit[15])
  - Bit[2]: Humidity
  - Bit[3]: Temperature
  - Bit[4]: PM2.5
- D0 – Bit[15]: The above Bit[1] control, 0-VOC, 1-CH2O
- D0 – Bit[16..31]: CO2 Calibration
  - 0: Not Calibration
  - 400 (or others): Calibrate CO2 PPM to 400 (or other values specified)

Note: mote will do calibration when detecting this field is not zero and reset this field to zero after calibration.

PSS-423041

Refer to PSS-423031.

PSS-423051

Refer to PSS-423031.

PSS-423062

Sensor Type – 17 (Soil)

Collect soil humidity, temperature, salinity, and EC level from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control and turn on the PWR3 for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

- D0 – Bit[0..7]: Bitmap control for sub-data reporting. For each bit, '0' will skip and '1' will report this sub-data. Refer to below:
  - Bit[0] – Soil Humidity
  - Bit[1] – Soil Temperature
  - Bit[2] – Soil Salinity
  - Bit[3] – Soil EC

  Note, if this D0-Bit[0..7] is zero, the driver will used 0x0b for default, meaning Humidity/Temperature/EC will be reported.

"set d0[N] 1"

Collect only Soil Humidity

PSS-423081

// todo

PSS-4230A1

Sensor Type – 24 (DGM10 Multi Sensor)

Collect multiple data from EC – DGM10 multi sensor. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the Modbus/485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01

- D0 – Bit[0..14]: Report bitmap control. If this field is ZERO, 0xff will be used. For each bit, 0-Not Report, 1-Report
  - Bit[0]: Temperature
  - Bit[1]: Humidity
  - Bit[2]: Gas#0
  - Bit[3]: Gas#1
- D0 – Bit[16..31]: Calibration
  - Bit[0]: Gas#0 calib, 0-Start ZERO recalib, 1-Load ZERO default
  - Bit[1]: Gas#0 zero, 0-Keep current, 1-Write current as ZERO
  - Bit[2]: Gas#1 calib, 0-Start ZERO recalib, 1-Load ZERO default
  - Bit[3]: Gas#1 zero, 0-Keep current, 1-Write current as ZERO
  - Bit[4]: Gas#0 calib control, 0-Do nothing, 1-Set
  - Bit[5]: Gas#0 zero control, 0-Do nothing, 1-Set
  - Bit[6]: Gas#0 calib control, 0-Do nothing, 1-Set
  - Bit[7]: Gas#0 zero control, 0-Do nothing, 1-Set

Note: mote will do calibration when detecting this field is not zero and reset this field to zero after calibration.

PSS-RX dual gas sensor

PSS-R0C0A1

Refer to PSS-4230A1.

PSS-R0E091

Refer to PSS-4230A1.

PSS-other

Sensor Type – 0 (Duty report)

Report internal duty counter. Not use any parameter fields

Sensor Type – 1 (VBAT)

Collect battery voltage. This sensor will always turn ON/OFF PWR10 (VBAT) and PWR1 (VCC_AIN). Below is other parameter fields used by this sensor:

- IO – Bit[8..14]: Additional power control

If io (bit8..14) is not ZERO, it will also turn ON/OFF the corresponding PWR controlling. For example,

"set io[N] 0x100"

Turn ON/OFF PWR1 (3.3V) during batter voltage collecting.

Note: Refer to this document's Section PIN Configuration about each PIN mapping.

- IO – Bit[0..7]: User defined ADC sampling PIN

If io (bit0..bit7) is not ZERO, it will use this given AIN number for voltage sampling. Or else (by default, 0), it will use AIN5 (AIN VBAT) pin for voltage sampling. For example,

"set io[N] 2"

Collect voltage from AIN2 pin.

"set io[N] 0x102"

Control PWR1 and collect AIN2 as battery voltage.

- SRSV – Bit[0]: Report RSSI siginal control, 0-Not Report, 1-Report
- SRSV – Bit[1]: Mute battery voltage, 0-Report, 1-Muted (not report)
- SRSV – Bit[2]: Report battery pecentage, 0-Not Report, 1-Report
- D0 – Bit[0..15]: Battery voltage caculate ratio
- D0 – Bit[16..31]: Battery voltage caculate offset
  Final battery voltage = ADC_voltage * ratio / 1000 + offset
- D1 – Bit[0..15]: Battery voltage for 0% percentage
- D1 – Bit[16..31]: Battery voltage for 100% percentage
  By default, will use 3300~3600mv for percentage 0~100%

Sensor Type – 6 (MTLCD)

MTLCD sensor is a LCD screen driver with some special control logic for RFID scanning. It uses following PINs as hardcoding:

- DIO5 (USER BUTTON): RFID scanning and LCD ON controlling
- DIO0: DC charging input
- CTRL1: RFID reset control
- CTRL2: LCD RESET
- CTRL3: LCD DC (data/command) control
- SPI0: LCD SPI
- PWR9 (LCD): LCD power control
- LED RED: Warning LED
- 2700 Hz: Beep frequency

Below is the parameters used by this sensor for other functions:

- D0 – Bit[0..7]: LCD ON button time in unit of 100ms

If D0 (bit0..7) is 0 (default), it will 100ms as the minimal button time to turn ON LCD; or else uses the specified button time. For example,

"set d0[N] 0x2"

Turn ON LCD after 200ms button pressed.

- D0 – Bit[8..15]: RFID scanning button time in unit of 100ms

If D0 (bit8..15) is 0 (default), it will 2000ms as the minimal button time to scan RFID; or else uses the specified button time. For example,

"set d0[N] 0x300"

Scan RFID after 3000ms button pressed.

- D0 – Bit[16..23]: RFID erasing button time in unit of 100ms

If D0 (bit16..23) is 0 (default), it will 10000ms as the minimal button time to erase the current RFID; or else uses the specified button time. For example,

"set d0[N] 0xf0000"

Scan RFID after 15000ms button pressed.

- D0 – Bit[24..31]: LCD ON time in unit of 100ms

If D0 (bit24..31) is 0 (default), it will 10000ms as the LCD ON time; or else uses the LCD ON time. LCD will turn OFF after this setting ON time for power saving. For example,

"set d0[N] 0x08000000"

Turn ON LCD for 8000ms for each button press.

- SRSV: RFID scanned last time

The last scanned RFID value will be stored in this field. This number will be updated after each button pressing (over RFIC scanning delay).

Sensor Type – 10 (EPD)

EPD sensor is an ultra power saving LCD screen driver. Its refreshing time is long (~4s), but its picture can still be kept even when it's in power off state. It uses following PINs as hardcoding:

- DIO5 (USER BUTTON): Language/unit switching button

The screen will switch the display according to below format by each button pushing:

English + Celsius

Chinese + Celsius

English + Fahrenheit

Chinese + Fahrenheit

- CTRL2: LCD RESET
- CTRL3: LCD DC (data/command) control
- CTRL4: EPD refreshing busy check
- CTRL6: DC/Charging control (PIN1)
- CTRL7: DC/Charging control (PIN2)

The above CTRL6/7 will be combined as a state value = CTRL7*2 + CTRL6. Driver will use following strategy to process this value:

0: NO VBAT

1: CHARGED

2: CHARGING

3: VBAT

- SPI1: LCD SPI
- PWR9 (LCD): LCD power control

Below is the parameters used by this sensor for other functions:

- D0 – Bit[0..7]: Good humidity range/from
- D0 – Bit[8..15]: Good humidity range/to

These fields are in unit of 1% humidity. If it's valid (to > from), use this specified range to check humidity for good; or else, use default good range, which is 40% ~ 70%. For example:

"set d0[N] 0x4614"

Use good humidity range 20% (0x14) ~70% (0x46)

- D0 – Bit[16..23]: Normal humidity range/from
- D0 – Bit[24..31]: Normal humidity range/to

These fields are in unit of 1% humidity. If it's valid (to > from), use this specified range to check humidity for normal; or else, use default normal range, which is 30% ~ 80%. For example:

"set d0[N] 0x500a0000"

Use normal humidity range 10% (0x0a) ~80% (0x50)

"set d0[N] 0x500a4614"

Use normal humidity range 10% (0x0a) ~80% (0x50) and use good humidity range 20% (0x14) ~70% (0x46).

- D1 – Bit[0..7]: Good CO2 range/from
- D1 – Bit[8..15]: Good CO2 range/to

These fields are in unit of 100 PPM. If it's valid (to > from), use this specified range to check CO2 for good; or else, use default good range, which is 0 ~ 800 PPM. For example:

"set d1[N] 0x0a00"

Use good CO2 range 0 ~1000 (0xa=10, 10 x 100 = 1000) PPM.

- D1 – Bit[16..23]: Normal CO2 range/from
- D1 – Bit[24..31]: Normal CO2 range/to

These fields are in unit of 100 PPM. If it's valid (to > from), use this specified range to check CO2 for normal; or else, use default normal range, which is 0 ~ 1600 PPM. For example:

"set d1[N] 0x14000000"

Use normal CO2 range 0 ~ 2000 (0x14=20, 20 x 100 = 2000) PPM

"set d1[N] 0x140000a00"

Use normal CO2 range 0 ~ 2000 (0x14=20, 20 x 100 = 2000) PPM

and use good CO2 range 0 ~1000 (0xa=10, 10 x 100 = 1000) PPM.

- D2 – Bit[0..7]: Good temperature range/from
- D2 – Bit[8..15]: Good temperature range/to

These fields are in unit of 1 Celsius degree. If it's valid (to > from), use this specified range to check temperature for good; or else, use default good range, which is 18 ~ 26 Celsius degree. For example:

"set d2[N] 0x1e0a"

Use good temperature range 10 (0x0a) ~30 (0x1e) Celsius degree.

- D2 – Bit[16..23]: Normal temperature range/from
- D2 – Bit[24..31]: Normal temperature range/to

These fields are in unit of 1 Celsius degree. If it's valid (to > from), use this specified range to check temperature for normal; or else, use default normal range, which is 18 ~ 28 Celsius degree. For example:

"set d2[N] 0x20000000"

Use normal temperature range 0 ~32 (0x20) Celsius degree.

"set d2[N] 0x20001e0a"

Use normal temperature range 0 ~32 (0x20) Celsius degree and use good temperature range 10 (0x0a) ~30 (0x1e) Celsius degree.

- SRSV – Bit[0]: Language control

This field will decide the language for: 0 – English, 1 – Chinese

- SRSV – Bit[1]: Unit control

This field will decide the temperature unit for: 0 – Celsius, 1 – Fahrenheit

For example:

"set srsv[N] 1"

Use Chinese + Celsius.

"set srsv[N] 2"

Use English + Fahrenheit.

- SRSV – Bit[2]: VBAT Percentage range

This field will decide the VBAT percentage calculation range for: 0 – 3.6V batter, 1 – 4.2V battery

For example:

"set srsv[N] 4"

Use 4.2V percentage translation logic.

"set srsv[N] 0"

Use 3.6V percentage translation logic.

- SRSV – Bit[3]: Reserved
- SRSV – Bit[4..5]: Face control

By default , when this field is 0, the driver will display different faces: BAD/NORMAL/GOOD by corresponding temperature, humidity and CO2 ranges. However, this face could be forcedly chosen by following number of this field:

0 – Auto

1 – BAD

2 – NORMAL

3 – GOOD

For example:

"set srsv[N] 0x31"

Force to display GOOD face with Chinese + Celsius.

- SRSV – Bit[6..31]: Reserved (leave it all zero)
- THW0 – Bit[0..1]: Logo display control

- 0 – Display regular CO2/PPM in the first panel
- 1~3 – Display vendor log

◆ THW0 – Bit[2]: History bar displaying
- 0 – Display regular GOOD/BAD face
- 1 – Display history CO2 PPM bar

◆ THW0 – Bit[3]: LED sync
- 0 – No LED sync
- 1 – Sync GOOD/NORMAL/FACE for LED-BLUE/YELLOW/RED

◆ THW0 – Bit[4..7]: Service model
- 0 – CO2/PPM with single temperature/humidity logic (Good/Bad Face)
- 1 – Three temperature/humidity logic
- 2 – Two temperature/humidity logic

◆ THW0 – Bit[8]: Title EUI displaying
- 0 – Report time displaying
- 1 – EUI displaying

For example:

"set thw0[N] 0x11" – Choose three temperature/humidity logic with logo displaying

"set thw0[N] 0x20" – Choose two temperature/humidity logic without logo displaying

Sensor Type – 14 (UART)

Collect UART raw input. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0

◆ SRSV: Sensor UART baudrate

If it's 0 (default), 9600 will be used, or else use the specified Baudrate. For example:

"set srsv[N] 0x01c200"

Collect using 0x01c200 (115200) baudrate .

Sensor Type – 21 (Multi Modbus/RS485 Temperature)

Collect multi temperature sensor from UART (RS232 or RS485) ports. The driver firmware will always choose 9600 to communicate with this sensor. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR1 (3.3V) pin for power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 2"

Collect from UART2.

"set io[N] 0"

Collect from UART0.

◆ SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01

◆ D0 – Bit[0..7]: Number of sensors attached. This can link multiple sensors with different address like: N, N+1, N+2, …. Only the first sensor's address N should be set.

Sensor Type – 29 (485 Light sensors)

Collect light from 485/RS232 ports. Below is the parameters used by this sensor:

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control. For example,

"set io[N] 0x603"

Turn ON/OFF PWRH (5V) as the power control.

◆ IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

◆ SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

Sensor Type – 56 (Generic IO Duty Report)

This sensor will report the IO state on each duty cycle. Unlike the type – 3 or type – 48, this sensor will not use IRQ/INTERRUPT and only report the state by duty cycle..

◆ IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will NOT use any power control; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: User defined DIO pin for interrupt collecting
- SRVS – Bit[0..14]: Waiting/polling time until a signal is detected. When zero, no waiting
- SRVS – Bit[15]: The above waiting/polling time unit: 0-s, 1-ms
- SRVS – Bit[16..30]: Sleeping time in unit of ms during waiting/polling. When zero, 100ms will be used; or else, use the specified duration. This field is always in unit of ms.
- SRVS – Bit[31]: Waiting until given polarity detected. 0-Waiting until high, 1-Waiting until low

Sensor Type – 66 (Soil#2)

Collect soil#2 humidity, temperature, EC, N, P, K and PH index from UART (RS232 or RS485) ports. Below is the parameters used by this sensor:

- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use PWR3 pin for power control and turn on the PWR3 for high voltage; or else, use the specified PWR control. For example,

"set io[N] 0x700"

Turn ON/OFF PWR7 (3.3V OpenDrain) as the power control.

- IO – Bit[0..7]: UART ID (0..3)

Specify the UART ID (0..7). For example:

"set io[N] 3"

Collect from UART3.

"set io[N] 0"

Collect from UART0.

- SRSV – Bit[0..7]: Device Address

Set the 485 Device address. For example:

"set srsv[N] 0x01"

Collect using device address 01.

- D0 – Bit[0..7]: Bitmap control for sub-data reporting. For each bit, '0' will skip and '1' will report this sub-data. Refer to below:
  - Bit[0] – Soil Humidity
  - Bit[1] – Soil Temperature
  - Bit[2] – Soil EC
  - Bit[3] – N index
  - Bit[4] – P index
  - Bit[5] – K index

- Bit[6] – PH index

  Note, if this D0-Bit[0..7] is zero, the driver will used 0x07 for default, meaning Humidity/Temperature/EC will be reported.

"set d0[N] 0x40"

Collect only Soil PH

"set d0[N] 0x38"

Collect only Soil N/P/K

Sensor Type – 69 (XDIO)

Collect multi DIO interrupts and IO state. Below is the parameters used by this sensor:

- IO – Bit[15]: Special IO conrol.
  - 0 - Bit[0..7] is used by either PIN/DIO or PIN/RSVD
  - 1 - Bit[0..7] is used as directly first DIO PIN. (0x00/0x20/0x40 for PA0/PB0/PC0...)
- IO – Bit[8..14]: Power control

If io (bit8..14) is 0 (default), it will use none power control. For example,

"set io[N] 0x600"

Turn ON/OFF PWRH as the power control.

- IO – Bit[0..7]: DIO/RSVD ID or Direct PIN number if Bit[15] is 1.
- SRSV – Bit[0]: When IO-Bit[15] == 0, this will control how to map IO-Bit[0..7]. 0 - IO Bit[0..7] is RSVD#N, 1 - IO Bit[0..7] is DIO#N. When IO-Bit[15] == 1, IO-Bit[0..7] is always mapped to PIN number directly.

For example,

"set io[n] 0x8040" - Use PC0 as the first DIO pin
"set io[n] 1", "set srsv[n] 0" - Use PIN/RSVD[1..N] for DIO pin
"set io[n] 3", "set srsv[n] 1" - Use PIN/DIO[3..N] for DIO pin

- D0 – Bit[0..7]: Number of DIO pins. When zero, 10 will be used.
- D0 – Bit[8..15]: Number of settling down time in unit of 0.1s. When zero, 2.5s will be used.
- D0 – Bit[16..23]: Number of additional settling down cycle. When zero, no additional settling down cycle.

For example:

"set d0[n] 0x030000" will check until no additional interrupts for at most (3+1)*2.5s = 10s. This does NOT mean that each interrupt processing will take 10s. It means when first interrupt is triggered, it will check for 2.5s and if no additional interrupts raised, raise and stop; if there's other interrupts hanppening during this 2.5s, it will check another 2.5s. 3 means it will check at most another three 2.5s. Once any 2.5s is no new interrupts, it will stop and raise the DIO status immediately.

Sensor Type – 0x1001 (Packer)

Pack report data and group multiple cycles into one big packet. Below is the parameters used by this sensor:

- SRSV – Bit[0]: Control whether report timestamp or not, 0-report, 1-not report. Note, some modules like LoRa does NOT have correct timestamp. Reporting timestamp for this module will report a strange timestamp.
- SRSV – Bit[1]: Control whether report threshold checking bitmap or not: 0-not report, 1-report. Threshold checking will check whether each data is according the THW0..7 or not. And finally, use a "bmp" field to return its status: a zero in bit indicting not crossing, 1 for crossing. For example, bmp0=0x0f, indicating the first four times of data is crossing, the others are not crossing.
- D0 – Bit[0..15]: Number of duty cycles will be packed into one big packet
- THW0~THW7: Threshold value for each kind of data (eight data types at most)

For example,

"set d0[N] 0x3"

Group cycle #0, #1 and #2 into one big report packet. Finally send this packet on #2, #5 (grouping 3~5) and #8 (groupping 6..8) duty cycles.

Section 6 DBG Commands

Polysense iEdge4.0 products support various debug functions in local console. See below about the debug commands:

dbg 1

Debug and operate the external communication module directly. This can be used to debug those modules' AT command and check module status. After entering this debug state, the regular DUTY/collection will be stopped. Typing ESC twice (with 1s delay between these two ESC) can end this debug state and return to normal state.

dbg 1 1

Same as "dbg 1". While this command will issue command "+++" to stop the module transparent mode and then enter debug state. It's useful if the module is sometime working in transparent mode and ignore all AT command. Also, typing two ESC can quite to normal state.

dbg 2 XXX

Same as "dbg 1". However, this command will use a specified Baudrate. For example:

"dbg 2 115200"

Debug the external AT module using 115200 baudrate. Note that whether the module is in

115200 or not is not controlled by this command; this command only set itself to 115200 and try to communicate with the external module.

dbg 3

Backup current configuration (content form “cfgr”) to factory default area. Next time of rebooting, if the current configuration is corrupted (for example magic number is wrong), system will load from factory default and overwrite the current wrong configuration by this backup content.

dbg 4

Same as “dbg 1”. However, during the debug state, the module will be power reset after 15s. This command will be useful for module bootup debugging.

dbg 5 xxxx

Re-Schedule duty for given XXXX ms. The XXXX can be omitted or 0. In this case, the next duty will happen immediately. For example:

“dbg 5”

Call next duty and start to collect/report data immediately

“dbg 5 6000”

Call next duty and start to collect/report data after 6s from now.

dbg 6 xxxx

Allow/dis-allow system sleeping. XXXX can be 0-Allow, 1-DisAllow. For example:

“dbg 6 1”

Do NOT sleep anymore.

“dbg 6 0”

Regular system sleep is allowed.

dbg 7 xxxx

Require/not-require system confirming. XXXX can be 0-Require, 1-Not Require. If Not Require, reboot and firmware upgrading will happen immediately instead of waiting for typing ‘Y’. For example:

“dbg 7 1”

Do NOT require confirmation

“dbg 7 0”

Require confirmation

dbg 8~11 xxxx

Debug UART (8 for UART0, 9 for UART1, 10 for UART2, 11 for UART3) with XXXX Baudrate.

XXXX can be regular Baudrate number like 9600 or 115200. The system will print "===>" to the specified UART and echo back all content received on this UART. If no operation in this UART for 30s, the debug state will quit automatically. For example:

"dbg 10 9600"

Debug UART2 with 9600 baudrate.

dbg 0x2xxxxxxx

Read 32-Bit memory content from address 0x2xxxxxxx. For example:

"dbg 0x20000000"

Read memory at 0x20000000.

Note this operation is dangerous, using wrong address will cause system abnormal.

dbg 0x2xxxxxxx XXXX

Write 32-Bit memory content at address 0x2xxxxxxx with value XXXX. For example:

"dbg 0x20000100 0"

Write memory at 0x20000100 with value 0.

Note this operation is dangerous, using wrong address/value will cause system abnormal.

adc PXX

ADC sampling from given PIN PXX. The PXX could be PA0, PA1, …, PF15. The result collected by ADC sampling is in unit of 1 mv. For example:

"adc pb13"

ADC sampling from PB13.

mod PXX XXXX

Change PIN PXX mode to XXXX. The XXXX is a bit definition:

Bit[0..3] – Alternative mode 0~15

Bit[10] – OpenDrain enable

Bit[11] – Pulldown enable

Bit[12] – Pullup enable

Bit[13] – Drive low enable (1 for drive low, 0 for drive high)

Bit[14] – Input mode

Bit[15] – Analog mode

Bit[16] – IRQ High level trigger

Bit[17] – IRQ Low level trigger

Bit[18] – IRQ Rising trigger

Bit[19] – IRQ Falling trigger

For example,

"mode pa0 0xc000"

Set PA0 as Anlog+Input mode.

“mode pa0 0”

Set PA0 as Output mode

“mode pa0 0x5000”

Set PA0 as Input+Pullup mode.

“mode pa0 0x4800”

Set PA0 as Input+Pulldown mode.

“mode pa0 0x400”

Set PA0 as Output+OpenDrain.

out PXX XXXX

Output XXXX to PIN PXX. For example:

“out pa0 0”

Output LOW to PA0.

“out pa0 1”

Output HIGH to PA0.

pwd PXX XXXX

PWM output to PIN PXX with duty XXXX (in unit of 0.01%). For example:

“pwd pb7 5000”

PWM output PB7 with duty=50%.

Section 7 PIN Configuration

Polysense iEdge4.0 products support PIN assignment changing in local console. See below about the PIN assignment commands:

cfgp

Show current PIN assignment. The content could be saved as a file (removing [] headers, leaving only “pin xxxx” commands and making sure the final command having a CR line-ending) and upgraded to boards through “xupg” commands. Below is an example of “cfgp” content:

pin Magic 0x37b72348

pin Version 0x4000002

pin LEDR !PF5

pin LEDG !PF4

pin PullUp PC9

pin PullDn PC8

pin SetH[0] !PF6

pin PWR[0] !PB5

pin PWR[1] !PB5
pin PWR[2] !PC13
pin PWR[3] !PC12
pin PWR[4] !PB2
pin PWR[5] !PA15
pin PWR[6] PB6
pin PWR[7] PB3
pin PWR[10] PF7
pin Wire1[0] PA8
pin AIN[0] PB13
pin AIN[1] PB13
pin AIN[2] PB14
pin AIN[3] PB15
pin AIN[4] PC4
pin AIN[5] PB1
pin AIN[6] PC5
pin DIO[0] PD2
pin DIO[1] PB13
pin DIO[2] PB14
pin DIO[3] PB15
pin DIO[4] PC4
pin DIO[5] PA0
pin PWM[0] PB7
pin PWM[1] PB3
pin PWM[2] PB4
pin PMODE[0] 0x23
pin PMODE[1] 0x2
pin PMODE[2] 0x36
pin CTRL[1] PB9
pin UART0[0] PA9
pin UART0[1] PA10
pin UART1[0] PA2
pin UART1[1] PA3
pin UART2[0] PC3
pin UART2[1] PC2
pin UART3[0] PC6
pin UART3[1] PC7
pin UART3[2] PA1
pin UART3[3] !PF6
pin LPUART0[0] PB10
pin LPUART0[1] PB11
pin LPUART1[0] PC10

pin LPUART1[1] PC11
pin I2C0[0] PB8
pin I2C0[1] PB9
pin I2C1[0] PA11
pin I2C1[1] PA12
pin SPI0[0] PA4
pin SPI0[1] PA5
pin SPI0[2] PA6
pin SPI0[3] PA7
pin SPI1[0] PB12
pin SPI1[1] PB13
pin SPI1[2] PB14
pin SPI1[3] PB15

Each item of the configuration is in format of "pin AAA[N] BBB". This command can also be typed in console to change the PIN assignment. A reboot is required to make the change taking effect. AAA[N] is the field name. Some field is an array and thus requires to use array index #N. BBB is the PIN name or special mode numbers for some cases. See below about all supporting PIN commands.

pin Magic XXX

Internal use. Do NOT change it.

pin Version XXX

Set the PIN configuration file version. Usually, this file version is coherent to a certain PCB version. For example:

"pin version 0x04000003"

Set PIN version to 4.0.0.3.

pin LEDR [!]PXX

pin LEDG [!]PXX

pin LEDB [!]PXX

pin LEDY [!]PXX

Set LED-RED/GREEN/BLUE/YELLOW to give PIN PXX. A '!' can be given in front of the PXX. In this case ('!' is given), outout LOW to turn it ON and HIGH to turn it OFF; or else, LOW for OFF, HIGH for ON). For example:

"pin LEDG pa0"

Set LED-GREEN through PA0 with 0-OFF, 1-ON.

"pin LEDR !pa1"

Set LED-RED through PA1 with 1-OFF, 0-ON.

pin pullup PXX

pin pulldn PXX

Set PULL-UP / PULL-DOWN PIN assignment (especially for WRIE-1 DS18B20 operations). Or example:

"pin pullup PB7"

Set PB7 as the PULLUP PIN.

"pin pulldn PB8"

Set PB8 as the PULLDN PIN.

- pin pullup2 PXX
- pin pulldn2 PXX

Set second group PULL-UP / PULL-DOWN PIN assignment especially for WRIE-1 DS18B20 operations). For example:

"pin pullup2 PB7"

Set PB7 as the second PULLUP PIN.

"pin pulldn2 PB8"

Set PB8 as the second PULLDN PIN.

pin seth[N] PXX

pin setl[N] PXX

Set the bootup PIN groups to output HIGH or LOW state. 'N' is: 0..7 (Each group has at most 8 PIN items). "seth" is to control HIGH group and "setl" is to control LOW group. In case some PINs needs to be HIGH or LOW, set it through this command. For example:

"pin seth[0] pa7"

"pin seth[1] pb2"

"pin setl[0] pc1"

"pin setl[1] pc3"

Output PA7, PB2 to HIGH and output PC1, PC3 to LOW during bootup.

Note that the driver firmware or DBG command could still change these PINs to any state after bootup; this command controls only the bootup state.

pin pwr[N] [!]PXX

Set Power #N PIN assignment. 'N' is: 0..31. A '!' can be given in front of the PXX. In this case ('!' is given), outout LOW to turn it POWER ON and HIGH to turn it POWER OFF; or

else, LOW for POWER OFF, HIGH for PPOWER ON). For example:

“pin pwr[1] !pb5”

Set Power #1 PIN to PB5 with LOW=ON, HIGH=OFF.

“pin pwr[7] pb3”

Set Power #7 PIN to PB3 with LOW=OFF, HIGH=ON.

Below is a recommended power ID usage. Please note that this is not a fixed mapping; each driver could choose any power ID and thus get power control by any PINs.

- PWR[1] – VCC3.3V
- PWR[2] – VCC PT100
- PWR[3] – VCC WIRE1 (DS18B20)
- PWR[4] – VCC External Module
- PWR[5] – VCC ADC Sampling
- PWR[6] – VCC 5V
- PWR[7] – 3.3V OpenDrain
- PWR[8] – 5V OpenDrain
- PWR[9] – LED/LCD Power control
- PWR[10] – VCC VBAT
- PWR[11]..PWR[31] – User defined for expansion

pin wire1[N] PXX

Set WIRE1#N PIN assignment for DS18B20 sensors. ‘N’ is: 0..7 for at most eight WIRE1 bus PINs. For example:

“pin wire1[0] pa8”

Set PA8 as WIRE1#0 PIN.

pin ain[N] PXX

Set Analog INPUT (AIN#N) PIN assignment. ‘N’ is: 0..7 for at most eight analog input PINs. For example:

“pin ain[1] pb13”

“pin ain[2] pb14”

Set PB13 as AIN#1, PB14 as AIN#2.

Below is a recommended AIN ID usage. Please note that this is not a fixed mapping; each analog driver could choose any AIN ID and thus get analog sampling by any PINs.

- AIN[1]..AIN[4] – AIN#1 .. AIN#4
- AIN[5] – VBAT analog input
- AIN[6] – PT100 analog input
- AIN[7] – User defined for future expansion

pin dio[N] PXX

Set digital IO (DIO#N) PIN assignment. ‘N’ is: 0..7 for at most eight digital IO PINs. For example:

“pin dio[4] pc4”

“pin dio[5] pa0”

Set PC4 as DIO#4, PA0 as DIO#5.

Below is a recommended DIO ID usage. Please note that this is not a fixed mapping; each IO driver could choose any DIO ID and thus get digital control by any PINs.

- DIO[1]..DIO[4] – DIO#1 .. DIO#4
- DIO[5] – User button digital input
- DIO[6]..DIO[7] – User defined for future expansion

pin pwm[N] PXX

pin pmode[N] XXX

Set PWR#N PIN assignment and its mode number. ‘N’ is: 0..3 for at most 4 PWM PINs. The PWM mode number is defined as following bitmap definitions:

- PMODE[N] Bit[0..3] – GPIO AF Mode
- PMODE[N] Bit[4..7] – HW PWM ID (HW TIMER ID)

For example:

“pin pwm[0] pb7”

“pin pmode[0] 0x23”

Set PB7 as PWM#0 PIN using HW TIMER ID#2 and GPIO AF#3.

“pin pwm[1] pb3”

“pin pmode[1] 0x2”

Set PB3 as PWM#1 PIN using HW TIMER ID#0 and GPIO AF#2.

Below is a recommended PWM ID usage. Please note that this is not a fixed mapping; each PWM driver could choose any PWM ID and thus get digital control by any PINs.

- PWM[0] – On-board BEEP buzzer
- PWM[1]..PWM[2] – PWM#1, PWM#2
- PWM[3] – User defined for future expansion

pin ctrl[N] PXX

Set general control (CTRL#N) PIN assignment. ‘N’ is: 0..31 for at most 32 any control purpose PINs. For example:

“pin ctrl[1] pb9”

Set PB9 as CTRL#1 PIN.

Note that only corresponding driver will initialize and touch the related control PINs. The system kernel firmware will leave all control PINs as ANALOG by default.

pin UART0~3[0] PXX

pin LPUART0~3[0] PXX

Set UART0~3/LPUART0~3 TX PIN.

- pin UART0~3[1] PXX
- pin LPUART0~3[1] PXX

Set UART0~3/LPUART0~3 RX PIN.

- pin UART0~3[2] PXX
- pin LPUART0~3[2] PXX

Set UART0~3/LPUART0~3 RS485 RD/WR control PIN.

pin UART0~3[3] PXX

pin LPUART0~3[3] PXX

Set UART0~3/LPUART0~3 power control PIN.

For example:

“set lpuart0[0] pb10”

“set lpuart0[1] pb11”

Set LPUART0 as TX/PB10, RX/PB11.

“set uart3[0] pc6”

“set uart3[1] pc7”

“set uart3[2] pa1”

“set uart3[3] !pf6”

Set UART3 as TX/PC6, RX/PC7, RS485 RDWR/PA1 (1-Write, 0-Read), Power Control/PF6 (0-ON, 1-OFF).

pin I2C0~3[0] PXX

Set I2C0~3 SCL PIN.

pin I2C0~3[1] PXX

Set I2C0~3 SDA PIN.

For example:

“pin i2c0[0] pb8”

“pin i2c0[1] pb9”

Set PB8 as I2C0 SCL, PB9 as I2C SDA.

pin SPI0~3[0] PXX

Set SPI0~3 CS PIN.

pin SPI0~3[1] PXX

Set SPI0~3 CLK PIN.

pin SPI0~3[2] PXX

Set SPI0~3 MISO PIN.

pin SPI0~3[3] PXX

Set SPI0~3 MOSI PIN.

For example:

“pin spi0[0] pa4”

“pin spi0[1] pa5

“pin spi0[2] pa6”

“pin spi0[3] pa7”

Set PA4 as SPI0 CS, PA5 as CLK, PA6 as MISO, PA7 as MOSI PINs.

Section 8 Report Format

Polysense iEdge4.0 products support following kinds of reporting format:

- JSON Format
  Used when the report type is 0x41xx.
- PLSS IVIEW Format
  Used when the report type is 0x43xx or 0x40xx.
- PLSS DTU Format
  Used when the ctrl field 0x0800 is defined.

JSON Format

Polysense JSON format is using the regular text based JSON format. It’s using all lowercase for TEXT. Below is an example:

{“vbat”:3.56, “co2”:523, “tmp”:[25.6, 25.5,17.2], “uart”:”1122aabbccee”}

Please note:

1) There’s no SPACE character within JSON message
2) Each keyword of data field is surrounded by “, like the above “vbat” and “tmp”
3) If a certain data contains multiple values (like the above “tmp”), all these data will be grouped as an array and surrounded by []

Binary raw data like the UART original payload will be reported as heximal string format and also

surrounded by “

See below list about all data keywords used in JSON format:

“angx”:

X axis angle in unit of degree

“angy”:

Y axis angle in unit of degree

“angz”:

Z axis angle in unit of degree

“accx”:

X axis acceleration in unit of mg

“accy”:

Y axis acceleration in unit of mg

“accz”:

Z axis acceleration in unit of mg

“co”:

CO in unit of 1PPM

“co2”:

CO2 in unit of 1PPM

“cnt”:

Counter used for IO and interrupt related countings

“current”:

Electronic current in unit of 1Ampire

“dir”:

Direction: 0~359.00 in unit of degree.

“distance”:

Distance in unit of 1mm

“duty”:

Internal duty counter

“ec”:

Soil EC in unit of 1us/cm

“ehum”:

Array of humidity in unit of 1%

“etmp”:

Array of temperature in unit of 1 celsius degree

“lati”:

GPS latitude

“long”:

GPS longitude

“h2s”:

H2S in unit of 1 PPM

“hpa”:

Atmosphere pressure in unit of 100PA

“k”:

K (Potassium) index in unit of 1

“light”:

Light in unit of 1lux

“mv”:

Analog voltage in unit of 1mv

“n”:

N (Nitrogen) index in unit of 1

“nh3”:

NH3 in unit of 1 PPM

“o3”:

O3 in unit of 1 ug/m^3

“ohm”:

Ohm in unit of 1 ohm

“p”:

P (Phosphorus) index in unit of 1

“per”:

Percentage in unit of 1%

“ph”:

PH index in unit of 1

“pm1”:

PM1.0 in unit of 1

“pm2_5”:

PM2.5 in unit of 1

“pm10”:

PM10 in unit of 1

“pt”:

PT100 temperation in unit of 1 celsius degree

“rfid”:

RFID (decimal number)

“salinity”:

Salinity level in unit of 1 mg/L

“so2”:

SO2 in unit of 1 ug/m^3

“sound”:

Sound (noise) level in unit of DB

“state”:

Sensor IO state: 0 – IO low, 1 – IO high.

“tmp”:

Regular temperature in unit of 1 celsius degree

“uart”:

UART original binary content as “hexadecimal string” format

“uv”:

UV power in unit of 1uw/cm^2

“vbat”:

Battery voltage in unit of 1V

“voc”:

VOC in unit of 1ug/m^3

“vocb”:

VOC in unit of 1 PPB

“water”:

Water pressure in unit of 1 KPA

“wind”:

Wind speed in unit of 1m/s

PLSS IVIEW Format

PLSS (Polysense) IVIEW format is a binary reporting format. In those Internet based networking style (NBIOT/CAT1/4G/WIFI/…), it’s a regular UDP packet with a small header (also binary format) and then followed by the reporting payload in binary. For LoRa network, the PLSS IVIEW payload is used just as the LoRa WAN MAC payload directly following the LoRa WAN FPort field. See below for detailed:

- Internet PLSS IVIEW Packet Format

[MAC Layer Header…]
[Standard IP Header]
[Standard UDP Header]
[PLSS IVIEW UDP Header]
[PLSS IVIEW PAYLOAD]

From MAC to UDP header, those are standard UDP packets format. The “PLSS IVIEW UDP Header” is a 11 bytes small header used as the first part of UDP Payload. These 11 bytes is defined as:

Byte[0] – 0x25

Byte[1] – 0x00~0xff (Sequence ID, changed on each report)

Byte[2]..Byte[9] – Device EUI

Byte[10] – 0x03

For example, below is an example of report for a device EUI=0x7a20bdfffebc963c:

25007a20bdfffebc963c03d77e070e10

They "d77e070e10" is the PLSS IVIEW payload ("vbat=3.6" in this case).

- LoRa WAN PLSS IVIEW Packet Format

[LoRa WAN MSG Type]

[LoRa WAN DevAddr]

[LoRa WAN FCtrl]

[LoRa WAN FCnt]

[LoRa WAN FOpts]

[LoRa WAN FPort]

[PLSS IVIEW PAYLOAD]

[LoRa WAN MIC]

In fact, this is just a standard LoRa WAN packet. The PLSS IVIEW PAYLOAD is just inserted as LoRa WAN MAC payload, which can be parsed by any Network Servers.

No matter it's LoRa or Internet, the PLSS IVIEW payload format definition is identical. The packet will use any of following format: UART Raw Format or TYPE/VALUE format. It will NEVER use two of such format in one packets. One packet if it's TYPE/VALUE format, can contain multiple TYPE/VALUE pairs. See below for detailed:

- PLSS IVIEW - UART Raw Format

d7-86-Original UART content in binary

So far, only Sensor Type – 14 (UART) will use this format to collect the raw payload from UART and report. All other sensors are using following TYPE/VALUE format. If a system contains both UART and TYPE/VALUE formats, it will be reported separately, for example: report UART format in a packet and then report TYPE/VALUE format in another packet.

- PLSS IVIEW – TYPE/VALUE Format

d7-7e-Type1-Value1-Type2-Value2-·····-TypeN-ValueN

Please note that: 1) Type is always one byte field; 2) Value can be any bytes, which depends on the type field in front of the value. If value field is more than 1 byte, network order byte sequence is used. For example, the battery voltage = 3600 (0x0e10) ms will be: 07 – 0x0e – 0x10.

See below for all PLSS IVIEW Type/Value definitions:

Type – 0x01: Temperature (internal)

- Value Length: 2 Bytes
- Value Unit: 0.1 Celsius

Type – 0x02: Humidity

- Value Length: 1 Byte
- Value Unit: 1%

Type – 0x03: Three Axes Acceleration

- Value Length: 6 Byte.

(Including following three axes, each axis is a 2 Byte value:)

  - X Acceleration: 2 Byte
  - Y Acceleration: 2 Byte
  - Z Acceleration: 2 Byte

- Value Unit: 1 mg

Type – 0x04: Three Axes Angle (Inclination)

- Value Length: 6 Byte.

(Including following three axes, each axis is a 2 Byte value:)

  - X Angle: 2 Byte
  - Y Angle: 2 Byte
  - Z Angle: 2 Byte

- Value Unit: 0.01 degree

Type – 0x06: Light

- Value Length: 2 Bytes
- Value Unit: 1 lux

Type – 0x07: Voltage of Battery

- Value Length: 2 Bytes
- Value Unit: 1 mv

Type – 0x08: Sound (Noise) Level

- Value Length: 2 Bytes
- Value Unit: 0.1 DB

Type – 0x09: GPS position

- Value Length: 8 Bytes
  - Latitude: 4 Bytes
    - Bit[31]: 0-North, 1-South
    - Bit[0..30]: Position data in unit of 0.001 second
  - Longitude: 4 Bytes
    - Bit[31]: 0-East, 1-West
    - Bit[0..30]: Position data in unit of 0.001 second

Type – 0x0a: CO

- Value Length: 2 Bytes
- Value Unit: 1 PPB

Type – 0x0b: Voltage of ADC Sampling

- Value Length: 2 Bytes
- Value Unit: 1 mv

Type – 0x0d: Atmosphere Pressure

- Value Length: 4 Bytes
- Value Unit: 1 PA

Type – 0x0e: Temperature (external)

- Value Length: 2 Bytes
- Value Unit: 0.1 Celsius

Type – 0x13: Numeric Value (RFID or other counters)

- Value Length: 4 Bytes
- Value Unit: 1

Type – 0x15: NH3

- Value Length: 2 Bytes
- Value Unit: 1 PPB

Type – 0x1a: H2S

- Value Length: 2 Bytes
- Value Unit: 1 PPB

Type – 0x1f: O3

- Value Length: 2 Bytes
- Value Unit: 1 ug/m^3

Type – 0x21: SO2

- Value Length: 2 Bytes
- Value Unit: 1 ug/m^3

Type – 0x26: Distance

- Value Length: 2 Bytes
- Value Unit: 1 mm

Type – 0x28: CO2 PPM

- Value Length: 2 Bytes
- Value Unit: 1 PPM

Type – 0x29: Current

- Value Length: 2 Bytes
- Value Unit: 10 mA

Type – 0x2a: Water pressure

- Value Length: 2 Byte
- Value Unit: 1 KPA

Type – 0x2b: PM2.5/PM10/PM1.0

- Value Length: 6 Bytes (2 Bytes x 3)
  - PM2.5 (2 Bytes)
  - PM10 (2 Bytes)
  - PM1.0 (2 Bytes)
- Value Unit: 1

For example:

2b – 00 – e0 – 00 – c0 – 01 – 01

PM2.5=224, PM10=192, PM1.0=257

Type – 0x30: Direction

- Value Length: 1 Bytes
- Value Unit: 0~15 (0 North, 4 East, 8 South, 12 West)

Type – 0x31: Wind Speed

- Value Length: 2 Bytes
- Value Unit: 1mm/s

Type – 0x38: IO State (Leakage)

- Value Length: 1 Byte
- Value Unit: None
- Bit[7]: IO State
- Bit[0..6]: IO state change counter

For example:

38 – 83a

State=1, Counter=3 (Leakage sensor is 1)

Type – 0x3b: VOC

- Value Length: 2 Byte
- Value Unit: 1 ug/m^3

Type – 0x3d: O2 Concentration

- Value Length: 2 Byte
- Value Unit: 0.1%

Type – 0x4c: CO (any Unit)

- Value Length: 2 Byte
- Value Unit: Not defined (various for different sensors)

Type – 0x4d: RF signal

- Value Length: 4 Byte
- Value Unit: 1%

Type – 0x4f: UV power

- Value Length: 2 Byte
- Value Unit: uw/cm2

Type – 0x50: High Resolution Humidity

- Value Length: 2 Byte
- Value Unit: 0.1%

Type – 0x51: Dissolved Oxygen

- Value Length: 2 Byte
- Value Unit: 0.01%VOL

Type – 0x52: NTU

- Value Length: 2 Byte
- Value Unit: 0.1 NTU

Type – 0x53: EC

- Value Length: 4 Byte (float)
- Value Unit: 1 mS/cm

Type – 0x54: CH2O PPB

- Value Length: 2 Byte
- Value Unit: 1 PPB

Type – 0x55: Dissolved Oxygen

- Value Length: 4 Byte (float)
- Value Unit: 1 mg/L

Type – 0x56: CHL

- Value Length: 4 Byte (float)
- Value Unit: 1 ug/L

Type – 0x57: BGA

- Value Length: 4 Byte (float)
- Value Unit: 1 ug/L

Type – 0x58: COD

- Value Length: 4 Byte (float)
- Value Unit: 1

Type – 0x59: TOC

- Value Length: 4 Byte (float)
- Value Unit: 1

Type – 0x5a: NH3_N

- Value Length: 4 Byte (float)
- Value Unit: 1 mg/L

Type – 0x5b: K+

- Value Length: 4 Byte (float)
- Value Unit: 1 mg/L

Type – 0x5c: NH4

- Value Length: 4 Byte (float)
- Value Unit: 1 mg/L

Type – 0x5d: BMP64

- Value Length: 4 Byte
- Value Unit: 1

Type – 0x5e: Radiation

- Value Length: 2 Byte
- Value Unit: 1 w/m^2

Type – 0x5f: C2H4

- Value Length: 2 Byte
- Value Unit: 1 PPB

Type – 0x62: NTU

- Value Length: 4 Byte (float)
- Value Unit: 1 NTU

Type – 0x63: Salinity

- Value Length: 4 Byte (float)
- Value Unit: 1

Type – 0x64: PH (float)

- Value Length: 4 Byte (float)
- Value Unit: 1

Type – 0x65: PH Index

- Value Length: 2 Byte
- Value Unit: 0.01

Type – 0x66: Salinity

- Value Length: 2 Byte
- Value Unit: 1 mg/L

Type – 0x67: Soil EC

- Value Length: 2 Byte
- Value Unit: 1 us/cm

Type – 0x68: N (Nitrogen) Index

- Value Length: 2 Byte
- Value Unit: 1

Type – 0x69: P (Phosphorus) Index

- Value Length: 2 Byte
- Value Unit: 1

Type – 0x6a: K (Potassium) Index

- Value Length: 2 Byte
- Value Unit: 1

Type – 0x6b: VOC PPB

- Value Length: 2 Byte
- Value Unit: 1 PPB

Type – 0x6c: IO State (Generic)

- Value Length: 1 Byte
- Value Unit: None
- Bit[7]: IO State
- Bit[0..6]: IO state change counter

For example:

38 – 83

State=1, Cntr=3 (Sensor IO is 1)

Type – 0x6d: Ohm

- Value Length: 4 Byte
- Value Unit: 0.001 Ohm

Type – 0x6e: Flow speed

- Value Length: 4 Byte
- Value Unit: 0.001 m^3/h

Type – 0x6f: Percentage (VBAT or others)

- Value Length: 1 Byte
- Value Unit: 1%

PLSS DTU Format

PLSS (Polysense) DTU format is a binary reporting format. In those Internet based networking style (NBIOT/CAT1/4G/WIFI/…), it’s a regular UDP packet with a small header (also binary format) and then followed by the reporting payload in binary. For LoRa network, the PLSS DTU payload is used just as the LoRa WAN MAC payload directly following the LoRa WAN FPort field. For MQTT report style, the binary data will be reported as heximal string, for example:

Binary: 607a20bdfffebc963cff000101

MQTT Payload: “607A20BDFFFEBC963CFF000101”

See below for detailed:

- Internet PLSS DTU Packet Format

[MAC Layer Header…]

[Standard IP Header]

[Standard UDP Header]

[PLSS DTU UDP Header]

[PLSS DTU PAYLOAD]

From MAC to UDP header, those are standard UDP packets format. The "PLSS IVIEW UDP Header" is a 9 bytes small header used as the first part of UDP Payload. These 9 bytes is defined as:

Byte[0] – **0x60**

Byte[1]..Byte[8] – Device EUI

For example, below is an example of report for a device EUI=0x7a20bdfffebc963c:

607a20bdfffebc963cff000101

They "ff000101" is the PLSS DTU payload.

No matter it's LoRa or Internet, the PLSS DTU payload format definition is identical. The packet will use any of following format:

PLSS_DTU_PAYLOAD =

[CHANNEL1 … 1Byte]

[LENGTH1 … 2Bytes in **Network order**]

[DATA1 … Length of bytes]

[CHANNEL2 … 1Byte]

[LENGTH2 … 2Bytes in **Network order**]

[DATA2 … Length of bytes]

…… n CHANNEL/LENGTH/DATA blocks ……

[CHANNELn … 1Byte]

[LENGTHn … 2Bytes in **Network order**]

[DATAn … Length of bytes]

The CHANNEL field is just one byte. 0~0xfe for IO/UART channel ID. 0xff for Management channel ID.

The LENGTH field defines the number of DATA following this LENGTH field. For example:

03 00 02 31 32 ff 00 01 01

The above will indicate:

- UART#3 reports 2 bytes of data: 0x31 0x32. (The serial UART3/485 port has received two chars: "12" in this case)
- Management Channel reports 1 byte of data: 0x01

The data reported in management channel is the PLSS DTU management commands. The management command is defined as one byte of command, followed by specified parameters according to each command. All parameters are defined as Network order in DTU. See below for all DTU management commands:

0x00 – Command Reset

Reset the system. Required parameters:

U32 – Four bytes of mode. (Always use 0 now)

0x01 – Command Ping

Ping to remote side. No required parameters.

0x02 – Command Get Image

Get a piece of image content from the remote server side. Required parameters:

U32 – Expected image version

U32 – Image reading offset in this step.

U16 – Image reading length in this step. This length is usually short and limited by the extern module's MTU limitation.

U16 – Reserved field. Typically it's all zero.

0x03 – Command Get Image Response

Response a piece of image content corresponding to the above Command Get Image. The remote server will send this command to the DTU and return its expecting image content. Required parameters:

U32 – Returned image version

U16 – Image reading offset in this step.

U16 – Flag field.

Bit[0] – 0: No more content, 1: more content should be GET/RESPONSEed.

Bit[1..15] – Reserved (leave it all zero)

… – Image content bytes. The length is control the LENGTH field in this clause.

LENGTH = Length(ImageContent) + 8 (U32+U16+U16)+ 1 (Command byte)

8 indicating the above U32, U16 and U16 parameters length.

0x04 – Command Get Image Info

Get information of an image from the remote server side. Required parameters:

U32 – Expected image version

0x05 – Command Get Image Info Response

Response the image information corresponding to the above Command Get Image Info. The remote server will send this command to the DTU and return its expecting image information. Required parameters:

U32 – Returned image version

U32 – Image length.

U16 – CRC16 of the image.

This CRC16 is using MODBUS CRC and the value is store in this field as Network order.

U16 – Flag field.

Bit[0] – 0: No more content, 1: more content should be GET/RESPONSEed.

Bit[1..15] – Reserved (leave it all zero)

0x06 – Command OTA Image

Tell the DTU to start OTA with given controlling field and version list. The remote server will send this command to the DTU to trigger a series of OTA operations. Required parameters:

U16 – Control field

Bit[0] – 0: OTA through current DTU channel, 1: OTA through PLSS OTA SITE

Bit[1] – 0: Skip same version, 1: OTA image even when version is same

Bit[2..15] – Reserved

U32 – Image version1

U32 – Image version2

…

U32 – Image versionN

The number of image version to be OTAed is controlled through the clause LENGTH field:

LENGTH = N(ImageVersion) x 4 + 2 (U16 – Control Filed)

For example:

FF 00 0B 06 00 01 01 00 02 09 04 00 02 04

This command refers to: Mgmt Channel, 11 Bytes, 06 OTA Command, 00-01 PLSS SITE OTA, v1.0.2.9 POS kernel and v4.0.2.4 MAIN app version.

Section 9 OTA Management

Polysense iEdge4.0 products support remote OTA management. The network remote server can change the configuration through OTA channel (NBIOT or WIFI, 4G…).

PLSS IVIEW Format

The OTA management command is manipulated through the similar UDP PLSS IVIEW Format (described in Section 8 Report Format / PLSS IVIEW Format), see below:

Byte[0] – 0x25

Byte[1] – 0x00~0xff (Sequence ID, changed on each report)

Byte[2]..Byte[9] – Device EUI

Byte[10] – 0xdf

Byte[11…] – OTA Management Command

For example, below is an example of RESET (OTA Command = 000000) a device with EUI=0x7a20bdfffebc963c:

25007a20bdfffebc963cdf000000

For LoRa applications, the OTA management command is inserted directly as LoRa Payload (only 000000 for the RESET example) with FPORT=222 by default. For other internet based, the command including the 25xxxxxx is all used as the UDP payload.

Each command is defined as Type/Length/Payload format. For example, 00 for reset, 0000 is Payload Length (Little endian, as Reset command does not requires payload so this field is 0000). Please note that: one OTA command list can issue multiple OTA commands, for example:

000000010200123402 01ff

This command list includes three commands: 00, 01 and 02. See below:

00 (Command0) 0000 (Command0 Length)

01 (Command1) 0200 (Command1 Length = 0x02 bytes) 1234 (Command 1 Payloads)

02 (Command2) 0100 (Command1 Length = 0x01 bytes) ff (Command2 Paylods)

00 - Reset Device

Reset the device immediately. Length has to be zero. For example:

000000

01 – Reserved (Not Support)

Check the network reachable. Length has to be zero. For example:

010000

02 – Reserved (Not Support)

Get the configuration payload by offset. Length has to be four: two for offset, two for length For example:

02040000021000

Get the configuration payload from offset = 0x0200, length=0x0010 (16 Bytes).

03 – Set the Configuration

Set the configuration payload by offset. Length has to be >=3: two for offset, 1+ for length of payload to be set:

For example:

0304000002abcd

Set the configuration from offset = 0x0200 to: 0xab 0xcd.

For more configuration fields and offset. Please refer to:

https://ota.polysense.online/iEdge4.0/doc/structma__eeprom__cfg__t.html

The above URL will describe each field with ECFG#0xXXX description. This 0xXXX is the just offset that can be used for this configuration SET command. For example report type is 0x0004. If we change report type to 0x4001, the command would be: 03040004000140:

03 – Command Set

04 00 – Length = 0x0004

04 00 – Offset = 0x0004

01 40 – Report Type = 0x4001

Please note all fields is in little endian, so a 0x4001 will be 01 40 in command.

04 – Reserved (Not Support)

05 – Reserved (Not Support)

06 – Set the Cycle

Set the all sensors cycle to given seconds. Length has to be 4.

For example:

0604000a000000

Set all sensors' cycle to 0x0000000a (10s).

10 – OTA images

OTA images. Length has to be 2+4*N. N refers to number of versions to be OTAed.

For example:

0a0a000100010020904000204

OTA with

length=0x000a (Length(Type) + 2 x Length(Version)=2 + 2 x 4 = 10),

type=0x0001 (PLSS OTA SITE),

Image Verion1 = 1.0.2.9

Image Verion2 = 4.0.2.4

After the remote device has received this command, it will initiate a PLSS SITE OTA session and finally OTA image with kernel = 1.0.2.9 and main = 4.0.2.4. The system will resume back after this OTA session is done.

11 – Event/action trigger

Trigger the device to generate a special event. Some sensors supports to trigger a special action by this command. For example, if FASTCNT (type=3) is enabled, this command when issued to device side will trigger the FASTCNT driver to Power UP the PWRH (configurable) PIN until the DIO input signal is recovered from 1 to 0. This can be used by some remote operations like remote locker locking/unlocking. Length has to be zero now.

For example:

0b0000

Trigger the device to perform special actions.

Example – Change NS Port

Set NS port to 1560 (0x618) and reset:

0304005e031806000000

Example – Change NS Port/URL

1) Set NS port to 1560 (0x618)
2) Set NS URL to "1.2.3.4"

String must contains \0 ending. So this would be: 312e322e332e3400 in hex binary

3) Reset

Commands:

030c005e031806312e322e332e3400000000

Example – Change to PLSS IVIEW/N306 NB Report (0x4384)

1) Set Report to 0x4384
2) Set NS port to 1560 (0x618)
3) Set NS URL to “ota.polysense.online”

String must contains \0 ending. So this would be: 6f74612e706f6c7973656e73652e6f6e6c696e6500 in hex binary

4) Reset

Commands:

030400040084430319005e0318066f74612e706f6c7973656e73652e6f6e6c696e6500000000

Example – Change to MQTT IVIEW/N306 NB Report (0x4184)

Call:

1) Set Report to 0x4184
2) Set NS port to 1883 (0x75b)
3) Set NS URL to “ota.polysense.online”

String must contains \0 ending. So this would be: 6f74612e706f6c7973656e73652e6f6e6c696e6500 in hex binary

4) Reset

030400040084410319005e035b076f74612e706f6c7973656e73652e6f6e6c696e6500000000