Titan MCU Software Command

Version 0.08

Reversion History

Version	Description/Changes	Date
0.01	Initial version	Dec/10/2020
0.02	Add FAN/PWM	Jan/4/2021
0.03	Add UNVID and OW search	Jan/15/2021
0.04	Add LED	Mar/05/2021
0.05	Add VERSION	Jun/03/2021
0.06	Add ALLGPIO	Dec/13/2021
0.07	Update OW format	Dec/20/2021
0.08	Add WDG	Dec/28/2021

Command list

Function code	Description	Status
VERSION	Read MCU version (BL & AP)	Done
PLID	Read Panel Board ID	Done
UNVID	Read UNV board ID	Done
GPIO	Digital Input and Output control	Done
ALLGPIO	ALL GPIO	Done
INFT	Interrupt flag register	Done
ITEMP	Read On-board environment sensor	Done
IACCE	Read On-board accelerometer sensor	Done
IEEPROM	Read/Write EEPROM	
OW	Read OW sensor	Done
LED (XB)	LED matrix control	Done
ADC	Read ADC input	Done
FAN	FAN (PWM) control	Done
WDG	Watchdog control	Done

1.0.1
Get MCU Software Version Command

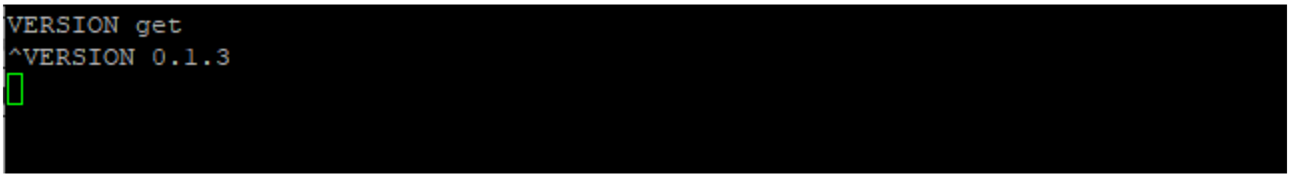
Function Code	VERSION	
Command Code	get : get MCU version code	
End Code	\n	\n for Linux, \r\n for Windows
Return	^VERSION 0x0.0.0	

Execution Command:

Function, Command code,\n;

Response

Function, Command code, Version numbers,\n



```
VERSION get
^VERSION 0.1.3
```

1.0.2
Get Panel_ID Command

Function Code	PLID	
Command Code	get : get Panel_ID setting	
End Code	\n	\n for Linux, \r\n for Windows

Return	Panel_ID = (1 byte)	PANEL_ID = 0x0 (Titan XA)
		PANEL_ID = 0x1 (Titan XB)
		PANEL_ID = 0x2 (Titan XB03)

Execution Command:

Function, Command code,\n;

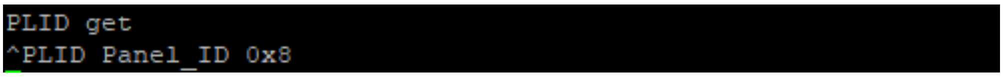
Response

^Function, Panel_ID, Data (1 byte),\n

EX.

PLID get\n;

PLID Panel_ID 0xA\n;



```
PLID get
^PLID Panel_ID 0x8
```

1.0.3
Get UNV_ID Command (Phase-in XB)

Function Code	UNVID	
Command Code	get : get UNV_ID setting	
End Code	\n	\n for Linux, \r\n for Windows

Return	UNV_ID = (1 byte)	UNV_ID = 0x0 (Titan XA)
		UNV_ID = 0x1 (Titan XB)
		UNV_ID = 0x2 (Titan XB03)

Execution Command:

Function, Command code,\n;

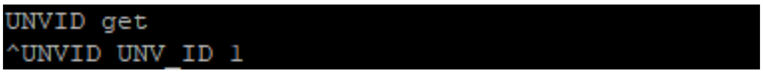
Response

^Function, UNV_ID, Data (1 byte),\n

EX.

UNVID get\n;

^UNVID UNV_ID A\n;



```
UNVID get
^UNVID UNV_ID 1
```

1.0.4 Force enter BOOTLOADER

1.1.1

GPIO Configuration Command

Function Code	GPIO	
Command Code	set : Config GPIO as High clear : GPIO as Low get : Get GPIO State	
Value (IO number)	0 ~ 51 (Decimal)	OUTPUT 0- 23 INPUT 24-51
End Code	\n	\n for Linux \r\n for Windows
Return	Pass Fail	

Execution Command:

Function code, Command Code, Value, End code,\n

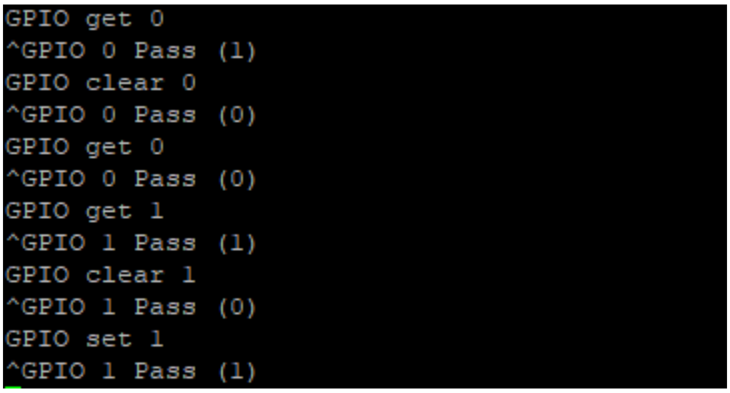
Response Value

^Function code, Value, Pass/Fail,\n

EX.

GPIO Set 0\n;

^GPIO 0 Pass (1)\n;



```
GPIO get 0
^GPIO 0 Pass (1)
GPIO clear 0
^GPIO 0 Pass (0)
GPIO get 0
^GPIO 0 Pass (0)
GPIO get 1
^GPIO 1 Pass (1)
GPIO clear 1
^GPIO 1 Pass (0)
GPIO set 1
^GPIO 1 Pass (1)
```

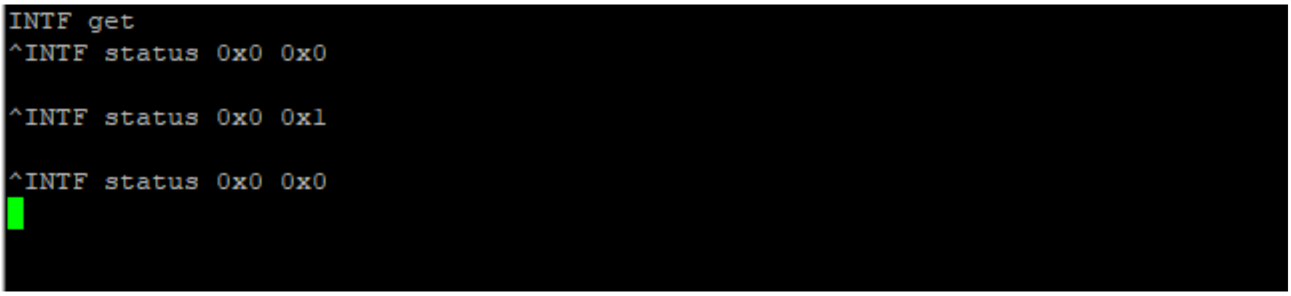
1.1.2

Interrupt flag register

Function Code	INTF	(MCU_STATUS toggle 10ms for interrupts)
Command Code	get : get INTF status	
End Code	\n	\n for Linux, \r\n for Windows
Return	INTF status (2 bytes) Bit15-4: Define for priority GPI Bit3: RBP_STATUS (TBD) Bit2: Door_DET (PF8/GPIO26) Bit1: Emergency Event (TBD) Bit0: Case_Open (PF4/GPIO52)	The INTF will be cleared after read from host.

EX.

INTF get\n;
^INTF status 0x0 0x1\n; 1st read (Case Open)
^INTF status 0x0 0x0\n; 2nd read

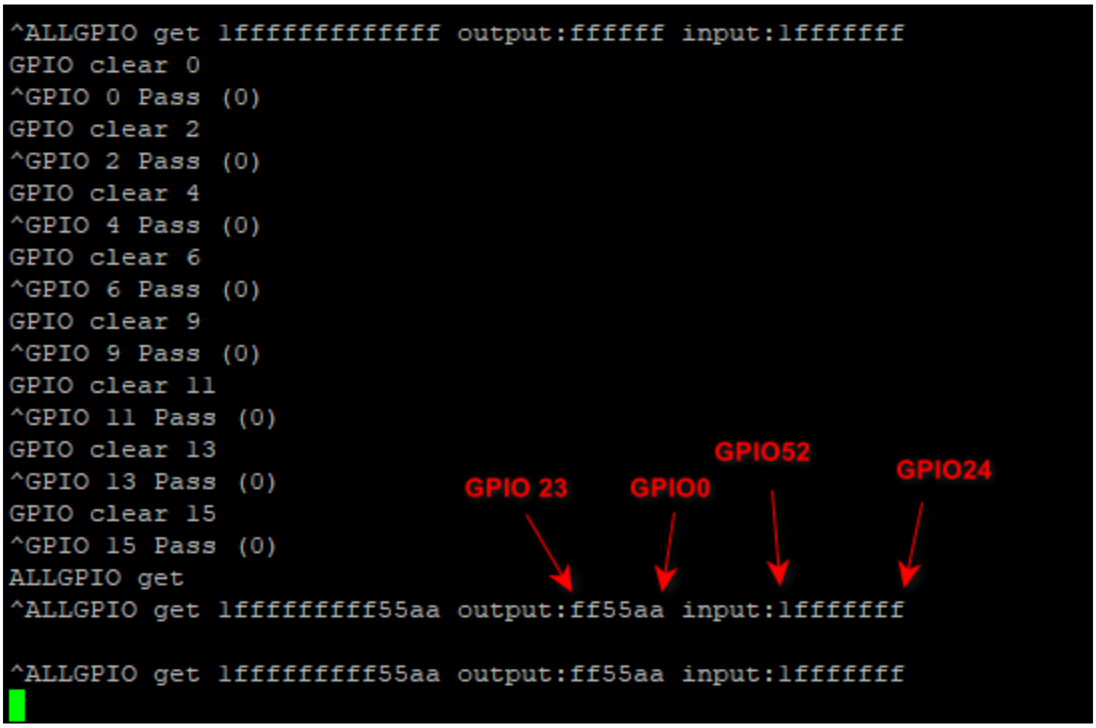


```
INTF get
^INTF status 0x0 0x0

^INTF status 0x0 0x1

^INTF status 0x0 0x0
```

1.1.3

Get ALL GPIO state Command

Function Code	ALLGPIO	
Command Code	get : Get GPIO State	
End Code	\n	\n for Linux \r\n for Windows
Return	1st string GPIO52(MSB) to GPIO0(LSB) 2nd string OUTPUT GPIO23(MSB) to GPIO0(LSB) 3rd string INPUT GPIO52(MSB) to GPIO24(LSB)	



```
^ALLGPIO get 1fffffffffffff output:ffffff input:1fffffff
GPIO clear 0
^GPIO 0 Pass (0)
GPIO clear 2
^GPIO 2 Pass (0)
GPIO clear 4
^GPIO 4 Pass (0)
GPIO clear 6
^GPIO 6 Pass (0)
GPIO clear 9
^GPIO 9 Pass (0)
GPIO clear 11
^GPIO 11 Pass (0)
GPIO clear 13
^GPIO 13 Pass (0)
GPIO clear 15
^GPIO 15 Pass (0)
ALLGPIO get
^ALLGPIO get 1fffffffff55aa output:ff55aa input:1fffffff

^ALLGPIO get 1fffffffff55aa output:ff55aa input:1fffffff
```

1.2.1

Get ITEMP Command

Function Code	ITEMP	
Command Code	get : get internal Temperature raw	
End Code	\n	\n for Linux, \r\n for Windows
Return	product: Sensor info Temp Reg: raw data (2 bytes) convert value: convert result Humidity Reg: raw data (2 bytes) convert value: convert result	

Execution Command:

Function, Command code,\n;

Response

^Function, product, temp, reg, convert value\n;

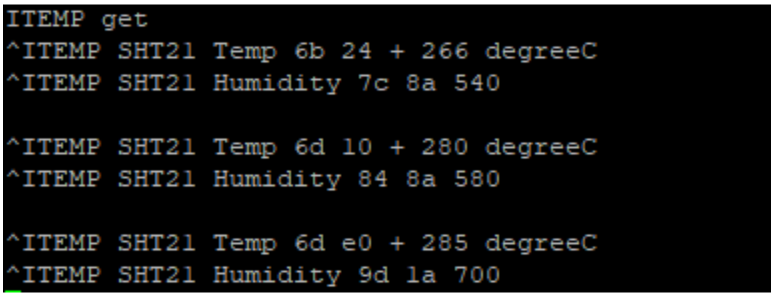
^Function, product, humidity, reg, convert value\n;

EX.

ITEMP get\n;

^ITEMP SHT21 temp 6f 44 + 295 degreeC\n;

^ITEMP SHT21 humidity 7a 9e 530\n;



```
ITEMP get
^ITEMP SHT21 Temp 6b 24 + 266 degreeC
^ITEMP SHT21 Humidity 7c 8a 540

^ITEMP SHT21 Temp 6d 10 + 280 degreeC
^ITEMP SHT21 Humidity 84 8a 580

^ITEMP SHT21 Temp 6d e0 + 285 degreeC
^ITEMP SHT21 Humidity 9d 1a 700
```

1.2.2
Get IACCE Command

Function Code	IACCE	
Command Code	get : get internal accelerometer raw	
End Code	\n	\n for Linux \r\n for Windows
Return	product: Sensor info convert value: convert result (mg)	

Execution Command:

Function, Command code,\n;

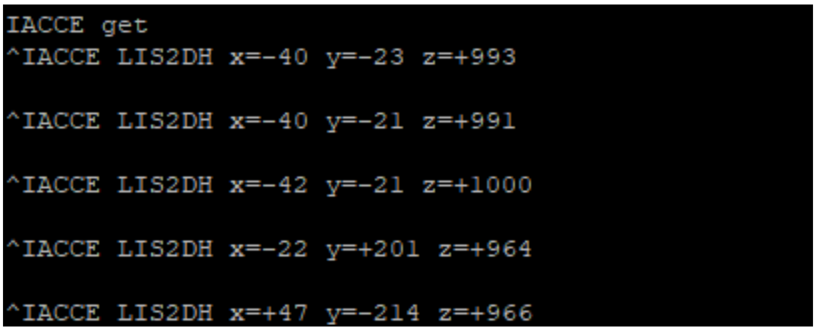
Response

^Function, product, Acce x= , y=, z= \n

EX.

IACCE get\n;

^IACCE LIS2DH x=0.0, y=0.0, z=9.8\n;



```
IACCE get
^IACCE LIS2DH x=-40 y=-23 z=+993

^IACCE LIS2DH x=-40 y=-21 z=+991

^IACCE LIS2DH x=-42 y=-21 z=+1000

^IACCE LIS2DH x=-22 y=+201 z=+964

^IACCE LIS2DH x=+47 y=-214 z=+966
```

1.2.3
IEEPROM Command

Function Code	IEEPROM	
Command Code	read: read EEPROM write: write EEPROM	
End Code	\n	\n for Linux \r\n for Windows
Return		

Execution Command:

Function, Command code,\n;

Response Command code, Version numbers,\n

1.3.1
Get OW Command

Function Code	OW	
Command Code	get : get 1-wire Temperature raw search: search OW device	
End Code	\n	\n for Linux \r\n for Windows
Return	product: Sensor info SN: Sensor SN (2 bytes) convert value: convert result	

Execution Command:

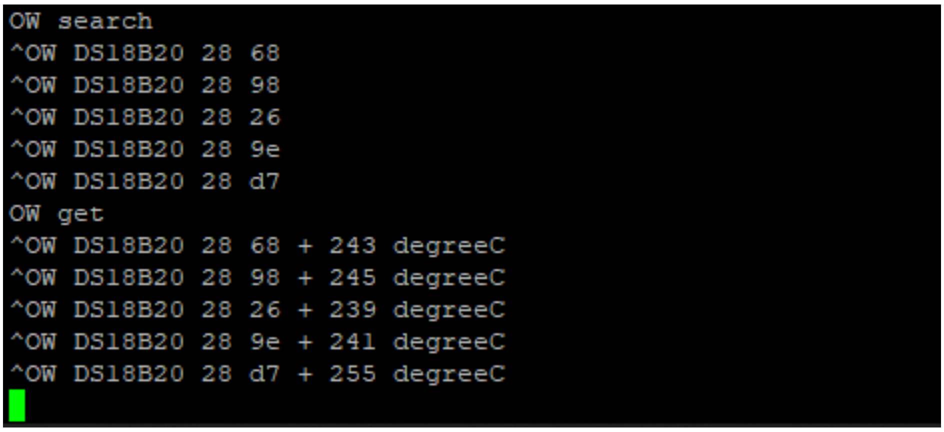
Function, Command code,\n;

Response

^Function, product, SN, converter value\n

Ex:

```
OW get\n;
^OW DS18B20 C9 28 + 228 degreeC\n; 228 => 22.8
^OW DS18B20 32 28 + 231 degreeC\n; 231 => 23.1

OW search\n;
^OW DS18B20 28 f6\n;
OW get\n;
^OW DS18B20 28 f6 + 223 degreeC\n;
```



```
OW search
^OW DS18B20 28 68
^OW DS18B20 28 98
^OW DS18B20 28 26
^OW DS18B20 28 9e
^OW DS18B20 28 d7
OW get
^OW DS18B20 28 68 + 243 degreeC
^OW DS18B20 28 98 + 245 degreeC
^OW DS18B20 28 26 + 239 degreeC
^OW DS18B20 28 9e + 241 degreeC
^OW DS18B20 28 d7 + 255 degreeC
```

Titan support OW up to 5 devices

OW device search time

OW devices	Time(Second)
1	3.28
2	6.6
3	9.8
4	12.9
5	16.3

1.3.2
LED Command (Phase-in XB)

Function Code	LED	
Command Code	set : Turn-on LED clear : Turn-off LED get : LED status	
Channel number	0 – 39 255 (all on or off)	0 – 19 OUTPUT 20 – 39 INPUT
End Code	\n	\n for Linux \r\n for Windows
Return	Pass Fail	

Execution Command:

Function, Command code, CH#,\n;

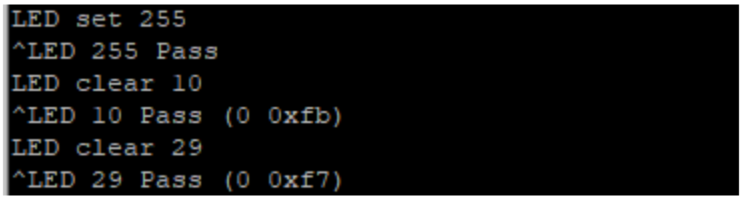
Response

Function, CH#, Pass/Fail,\n;

EX.

LED Set 0\n;

^LED 0 Pass (1)\n;



```
LED set 255
^LED 255 Pass
LED clear 10
^LED 10 Pass (0 0xfb)
LED clear 29
^LED 29 Pass (0 0xf7)
```

1.4.1
Get ADC Command

Function Code	ADC	
Command Code	get : get ADC raw	
Channel number	0 - 15	
End Code	\n	\n for Linux \r\n for Windows
Return	Product: Sensor info CH#: Channel 0 - 15 Reg : raw data (0-1023 Decimal) convert value: convert result	

Execution Command:

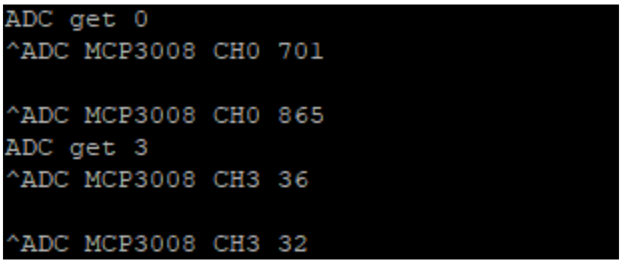
    Function, Command code, Channel number,\n;

Response

    ^Function, product, CH0, reg\n
    ^Function, product, CH1, reg\n

EX.

    ADC get 0\n;
    ^ADC MCP3008 CH0 1016\n;



```
ADC get 0
^ADC MCP3008 CH0 701

^ADC MCP3008 CH0 865
ADC get 3
^ADC MCP3008 CH3 36

^ADC MCP3008 CH3 32
```

1.5.1 FAN Command

Function Code	FAN	
Command Code	set : set FAN’s PWM duty get : get FAN’s PWM duty	
Channel number	1 ~ 3	
Value	0 ~ 100 (Decimal)	PWM duty cycle Default value is 0%
End Code	\n	\n for Linux \r\n for Windows
Return	Pass Fail	

Execution Command:

Function, Command code, Channel, Value \n;

Response

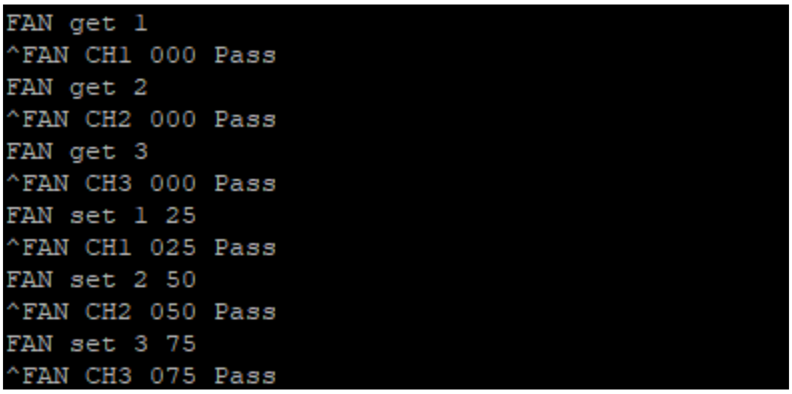
^Function, CH#, duty, Pass/Fail \n

Ex:

FAN get 1\n;

^FAN CH1 050 Pass

^FAN set 1 75\n (FAN1 duty set to 75%)



```
FAN get 1
^FAN CH1 000 Pass
FAN get 2
^FAN CH2 000 Pass
FAN get 3
^FAN CH3 000 Pass
FAN set 1 25
^FAN CH1 025 Pass
FAN set 2 50
^FAN CH2 050 Pass
FAN set 3 75
^FAN CH3 075 Pass
```

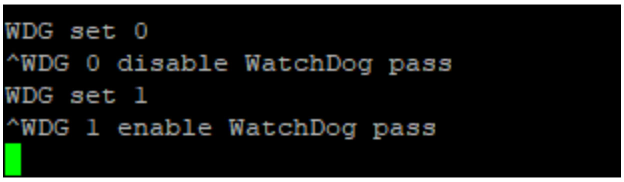
1.6.1 WDG Command

Function Code	WDG	
Command Code	set : enable/disable watchdog 0: Disable 1: Enable	
End Code	\n	\n for Linux \r\n for Windows
Return	Pass Fail	

Execution Command:

Function, Command code \n;

Response



```
WDG set 0
^WDG 0 disable WatchDog pass
WDG set 1
^WDG 1 enable WatchDog pass
```