

EZcat with GPIO module AT command manual

Rev. 1 – 14/08/10



AT (PY) Command interface

Command	Reply	Parameters
PY#E=A	ok	A - AT commands ECHO: A = 0 > Disables command echo A = 1 > Enables command echo
PY#E?	#e: 0 ok	
PY#E=?	#e: (0-1) ok	
Description: This command enables/disables the command echo.		

Command	Reply	Parameters
PY#V=A	ok	A - AT commands reply format: A = 0 > Numeric A = 1 > Verbal
PY#V?	#v: 0 ok	
PY#V=?	#v: (0-1) ok	
Description: This command determines the command result format Numeric / Verbal .		
Example:	PY#V=1	PY#V=0
	OK	0
	BUSY	7
	ERROR	4
	READY	5
	CONNECT	2

Unit Firmware version

Command	Reply	Parameters
PY#FMR?	#fmr: CAT_V4 1.00 ok	None
<p>Description: This returns the firmware version of the unit micro controller</p>		

Default values & VSS Counters reset

Command	Reply	Parameters
PY#DEF=A	ok	<p>A - Default select: 0 > Restore deflate values.</p> <p>1 > Reset only VSS counters.</p> <p>2 > Restore deflate values & Reset VSS counters</p>
PY#DEF=?	#def: (0-2) ok	
<p>Description: This command, Restore the default setting values of the unit & Reset the VSS (Vehicle Speed Sensor) counters.</p>		

Connecting the external UART directly to the Modem UART

Command	Reply	Parameters
PY#PRG=A	ok	A - Mode: 0 > Restart modem before the UARTs connection. 1 > Connect the UARTs without modem restarting.
PY#PRG=?	#prg: (0-1) ok	
<p>Description:</p> <p>This command connects directly the external UART to the modem. Follow sending this command:</p> <ol style="list-style-type: none"> 1. The modem will turn off (at mode 0). 2. The Green, Yellow & Blue LED will blink fast for a 3 Sec. 3. At the end of the 3 Sec, the modem will restart (at mode 0). 4. To exit this mode - Push the button on the unit (SW). <p>Note: When using this command, all the function of the unit are disable.</p>		

Modem Python script WDT (Watch Dog Timer)

Command	Reply	Parameters
PY#WDT=A	ok	A - Python script WDT time out: 0 > Disable 1 - 5400 > x Sec
PY#WDT?	#wdt: 0 ok	
PY#WDT=?	#wdt: (0, 1-5400) ok	

Description:

This command allows to apply external WDT to the Modem Python script.

From sending this command:

1. From "py#wdt=X" (x > 0) the WDT is Enable.
The MCU will expect to receive any byte on the UART.
This will reload the WDT counter.

It recommended to send the value 0x0D (CR 'r') cause if it send alone,
the MCU AT CMD parser will ignore it.

2. if a byte is not send during the WDT time out follow will happen:
 - a. WDT will automatically Disable (py#wdt=0).
 - b. Modem will restart..

3. After the modem will restart the PY must send aging "py#wdt=X"
to Enable the WDT.

Note: When modem is Start / Restart the #wdt: 0 (WDT is Disable)

Sleep, Sleep gap, Wakeup

Command	Reply	Parameters
PY#SLEEP=A	ok	A - Sleep mode: 0 > Shutdown Modem, then Unit goes to sleep mode.
PY#SLEEP=?	#sleep: (0-0) ok	
<p>Description:</p> <p>This command is reducing the unit power consumption, by shut down the Modem and set the Unit to sleep mode.</p> <p>Wakeup form the sleep, can be done by one of the follow events (if the mode select =3 - Wakeup):</p> <ol style="list-style-type: none"> 1. Digital inputs (PY#INS) 2. Analog inputs (PY#ADCS) 3. Moving sensor 		
<p>Note: When the Unit is wakeup form sleep, is powers up the Modem. The wakeup event and other events that happen during the Modem power up, are accumulate in the Unit UART buffer (to get the events refer to PY#WEVENT?).</p> <p>During the sleep, the Yellow LED will blink for 50mS each 10Sec</p>		

Command	Reply	Parameters
PY#SLEEPDLY=A	ok	A - Defines a gap after activate sleep, to ignore wakeup events: 0 - 1200 (x ~1.024 Sec)
PY#SLEEPDLY=?	#sleepdly: (0-1200)	
<p>Description:</p> <p>This command is defines a gap to ignore wakeup events, when sleep is activate (PY#SLEEP=0).</p>		
<p>During the sleep gap, the Yellow LED will blink for 50mS each 1 Sec, at the end of the gap the Yellow LED will blink for 50mS each 10 Sec,</p>		

Command	Reply
PY#WEVENT?	<pre> py#wevent? #inl: 1,L,L #inl: 2,H,H #taxs: 1,INT . . ok </pre>
<p>Description:</p> <p>During wakeup from sleep all the unsolicited data events (input, ADC ..etc.) that send to the Modem are accumulate in the Unit UART buffer.</p> <p>if the first AT CMD from the Modem after the power up is PY#WEVENT? the unit will send to the modem all the events that was accumulate in the UART buffer during the Modem power up.</p> <p>If other AT CMD will be send as the first AT CMD after the Modem power up all the accumulated data will be erase.</p> <pre> py#wevent? #inl: 1,L,L #inl: 2,H,H #inl: 2,H,L ok </pre>	

Multi purpose LED

Command	Reply	Parameters
PY#LED=A,B,C	ok	A - LEDx Select: 1 > Green LED 2 > Blue LED
PY#LED?	py#led? #led: 1,0,0 #led: 2,0,0 ok	B - Interval ON (1 - 250) x 100mS C - Interval OFF (1 - 250) x 100mS
PY#LED=?	#led: (1-2) (0, 1-250) (0, 1-250) ok	

Description:

This command, controls the Green & Blue LED.

Example 1: PY#LED=1,0,0 - Green LED OFF

Example 2: PY#LED=2,1,0 - Blue LED ON

Example 3: PY#LED=1,5,10 - Green LED 0.5 (Sec) ON , 1 (Sec) OFF

Example 4: PY#LED=2,30,1 - Blue LED 3 (Sec) ON , 0.1 (Sec) OFF

Note: In sleep mode, if LED was constant ON before the sleep mode, it will stay ON during the sleep mode.

If the LED was blinking before the sleep mode, it will turn off during the sleep mode. The blinking will restore when the unit will wakeup.

The LED setting is not store in the internal EErom. On power up the LED will be OFF.

Open collector outputs

Command	Reply	Parameters
PY#OC=A,B,C	ok	A - CONOCx Select (Open collector output): 1 > CONOC1 (J31 pin#7) 2 > CONOC2 (J31 pin#8) B - Interval Hi-Z (1 - 250) x 100mS C - Interval LOW (1 - 250) x 100mS
PY#OC?	py#oc? #oc: 1,0,0 #oc: 2,0,0 ok	
PY#OC=?	#oc: (1-2) (0, 1-250) (0, 1-250) ok	

Description:

This command, controls the CONOC1 & CONOC2 open collector outputs.

Example 1: PY#OC=1,0,0 - CONOC1 Low

Example 2: PY#OC=2,1,0 - CONOC2 Hi-z

Example 3: PY#OC=1,50,50 - CONOC1 5 (Sec) Hi-z , 5 (Sec) Low

Example 4: PY#OC=2,10,100 - CONOC2 1 (Sec) Hi-z, 10 (Sec) Low

Note: In sleep mode, if Output was constant Low before the sleep mode, it will stay Low during the sleep mode.

If the Output was blinking before the sleep mode, it will set to Hi-z during the sleep mode. The blinking will restore when the unit will wakeup.

The Output setting is not store in the internal EErom. On power up the Outputs will be set to Hi-z.

Unit Push button (SW) - Setting

Command	Reply	Parameters
PY#SWS=A,B,C	ok	A - Mode Select: 0 > Solicited (By PY#SWL?) 1 > Unsolicited B - Button report (Unsolicited): 0 > Button push 1 > Button release 2 > Button push or release C - Debouncing interval (x10mS)
py#sws?	#sws: 0,0,5 ok	
Py#sws=?	#sws: (0-1) (0-2) (5-2000) ok	
Description: This command, configure the Push button (SW) report parameters. The state of the Push button (SW) can be read by PY#SWL?. The Unsolicited report is not using during the sleep mode.		
Example 1: py#sws=0,0,10 - Solicited report by PY#SWL?, Debouncing = 100mS		
Example 2: py#sws=1,0,15 - Unsolicited when the Button push (#swl: L,L), debouncing = 150mS		
Example 3: py#sws=1,1,5 - Unsolicited when the Button release (#swl: H,H), debouncing = 50mS		
Example 4: PY#SWS=1,2,5 - Unsolicited when the Button push / release (#swl: H,H)/(#swl: L,L), debouncing = 50mS		

Unit Push button (SW) - Status

Command	Reply	Parameters
py#swl?	#swl: A,B ok	'H' - (SW release) 'L' - (SW push) A = 'H' or 'L' - Current state. B = 'H' or 'L' - The level before SW state changing, during the debouncing.
<p>Description:</p> <p>This command, returns the push button (SW) state. Setting the SW in done by using PY#SWS command.</p>		
<p>Example 1: py#swl? - #swl: H,H (push button is release)</p>		
<p>Example 2: py#swl? - #swl: L,H (SW is push, if the debouncing test will pass, the result will be #swl: L,L)</p>		

Digital inputs - Setting

Command	Reply	Parameters
PY#INS=A,B,C,D	ok	<p>A - Digital input select: 1 > IN#1 (CONHVIO1 J31 pin#11) 2 > IN#2 (CONHVIO2 J31 pin#12)</p> <p>B - Mode select: 0 - Solicited 1 - Unsolicited 2 - Unsolicited + RTS 3 - Wakeup</p> <p>C - Transient: 0 > 'H' >> 'L' 1 > 'L' >> 'H' 2 > 'H' <> 'L'</p> <p>D - Debouching interval (x10mS)</p>
PY#INS?	#ins: 1,0,0,5 #ins: 2,0,0,5 ok	
PY#INS=?	#ins: (1-2) (0-3) (0-2) (5-6000) ok	
<p>Description:</p> <p>This command, configure the digital inputs setting: Report mode (Solicited, Unsolicited, Unsolicited + RTS, Wakeup). Level detect (H >> L, L >> H, H <> L). Define the detect Debouching interval (x10mS).</p> <p>After setting the input, report will be active when the input will be stable to 2 Sec.</p>		
<p>Example 1: PY#INS=1,0,0,5 - IN#1 (H >> L), Solicited report by PY#INL=1, Debouching = 50mS</p>		
<p>Example 2: PY#INS=2,1,1,10 - IN#2 (L >> H), Unsolicited, Debouching = 100mS</p>		
<p>Example 3: PY#INS=2,1,2,15 - IN#2 (L <> H), Unsolicited, Debouching = 150mS</p>		
<p>Example 5: PY#INS=2,2,1,5 - IN#2 (L >> H), Unsolicited + RTS, Debouching = 50mS</p>		
<p>Example 6: PY#INS=2,3,1,5 - IN#2 (L >> H), Unsolicited + sleep wakeup Debouching = 50mS</p>		

Digital inputs - Status

Command	Reply	Parameters
PY#INL=A	PY#INL=A #inl: A,B,C ok	A - Input reply: 1 > CONHVIO1 (J31 pin#11) 2 > CONHVIO2 (J31 pin#12)
PY#INL?	#inl: 1,B,C #inl: 2,B,C ok	H - Hi level or Hi-z L - Low level B = 'H' or 'L' - Current state C = 'H' or 'L' B - Indicates the current input logic level. C - Indicates the level before transient during the debouncing.
PY#INL=?	#inl: (1-2) ok	
<p>Description:</p> <p>This command, returns the status of the inputs.</p>		
<p>Example 1: PY#INL=1 - #inl: 1,L,H (in #1 'L' current logic level, 'H' the level before transient during the debouncing) .</p>		
<p>Example 2: PY#INL=2 - #inl: 2,H,H (in #2 'L' current logic level, 'H' the level at end of the debouncing) .</p>		

Moving sensor

Command	Reply	Parameters
PY#TAXS=A,B,C	ok	<p>A - Digital input select: 1 > IN#1 (CONHVIO1 J31 pin#11) 2 > IN#2 (CONHVIO2 J31 pin#12)</p> <p>B - Mode select: 0 - Solicited 1 - Unsolicited 2 - Unsolicited + RTS 3 - Wakeup</p> <p>C - Transient: 0 > 'H' >> 'L' 1 > 'L' >> 'H' 2 > 'H' <> 'L'</p> <p>D - Debouching interval (x10mS)</p>
PY#TAXS?	#taxs: 1,0,5 ok	
PY#TAXS=?	#taxs: (1-1) (0-3) (5-90) ok	
<p>Description:</p> <p>This command, configure the digital inputs setting: Report mode (Solicited, Unsolicited, Unsolicited + RTS, Wakeup). Level detect (H >> L, L >> H, H <> L). Define the detect Debouching interval (x10mS).</p> <p>After setting the input, report will be active when the input will be stable to 2 Sec.</p>		
<p>Example 1: PY#INS=1,0,0,5 - IN#1 (H >> L), Solicited report by PY#INL=1, Debouching = 50mS</p>		

Analog inputs - Setting

Command	Reply	Parameters
PY#INS=A,B,C,D, E,F,G,H,I	ok	<p>A - Analog input select: 0 > Vin (VCON J31 pin#13) 1 > ADC IN#1 (CONHVIO1 J31 pin#5) 2 > ADC IN#2 (CONHVIO2 J31 pin#4)</p> <p>B - Mode select: 0 - Solicited 1 - Unsolicited 2 - Unsolicited + RTS 3 - Wakeup</p> <p>C - Low pass filter sample: x 200mS (gap between the samples)</p> <p>D - Low Threshold alert (refer to diagram): 0 > No alert 1 > "L-TH" 2 > "L-TH" & back to "NR"</p> <p>E: "L-TH" x 1 mV (accuracy ??) F: "L-HY" x 1 mV (accuracy ??)</p> <p>G - Hi Threshold alert (refer to the diagram): 0 > No alert 1 > "H-TH" 2 > "H-TH" & back to "NR"</p> <p>H - "H-TH" x 1 mV (accuracy ?? resolution) I - "H-HY" x 1 mV (accuracy ?? resolution)</p> <p>Note: field D & G depends on the definition field B (mode select).</p>
PY#ADCS?	#adcs: 0,0,1,0,5,7,0,7,7 #adcs: 1,0,1,0,5,7,0,7,7 #adcs: 2,0,1,0,5,7,0,7,7 ok	
PY#ADCS=?	py#adcs=? #adcs: (0-0) (0-3) (1-36000) (0-2) (5-39998) (7-39998) (0-2) (7-40000) (7-39999) #adcs: (1-1) (0-3) (1-36000) (0-2) (5-28998) (7-28998) (0-2) (7-29000) (7-28999) #adcs: (2-2) (0-3) (1-36000) (0-2) (5-3298) (7-3298) (0-2) (7-3300) (7-3299) ok	

Description:

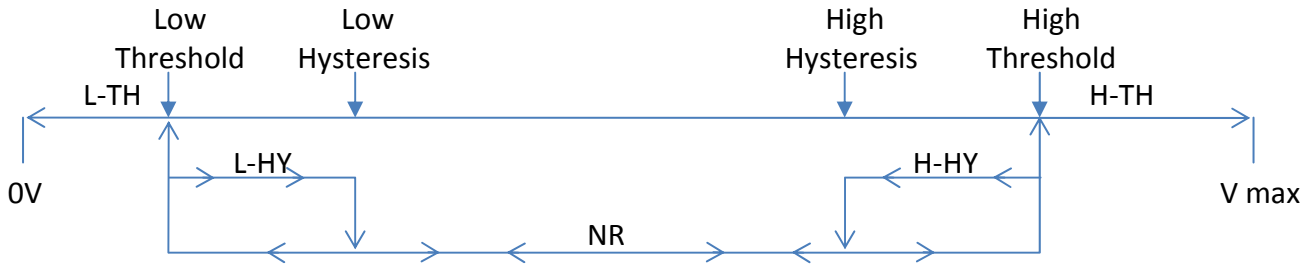
This command, configure the analog inputs setting:
Report mode (Solicited, Unsolicited, Unsolicited + RTS, Wakeup).

Low pass filter sample.

Hi, Low Threshold & Hysteresis - value
Hi, Low Threshold & back to normal -alert value

Note: During Sleep mode (0, 1) the ADC inputs are sample each ~1.024 Sec (~ x5 form the Normal mode).

Inputs range: ADC 0: 5mV - 40v
ADC 1: 5mV - 29v
ADC 2: 5mV - 3.3v



Example 1: PY#ADCS=0,2,1,1,7000,8000,2,20000,18000

0 - Measure Analog input 0 (Vin)
2 - Unsolicited alert
1 - **Low pass filter sample (x200 mS):**

1 - L-TH alert (Vin <= L-TH)
7000 - L-TH = 7v
8000 - L-HY = 7v

2 - H-TH alert (Vin => H-TH, Vin <=H-HY)
20000 - H-TH = 20v
18000 - H-HY = 18v

Example 2:PY#ADCS=1,0,10,0,6000,6500,0,8000,7000

1 - Measure Analog input 1
0 - solicited alert (By "py#adcl=1" or "py#adcl?" - refer to command explanation) .
6 - **Low pass filter sample (x200 mS):**

0 - L-TH (No alert)
6000 - L-TH = 6v
6500 - L-HY = 6.5v

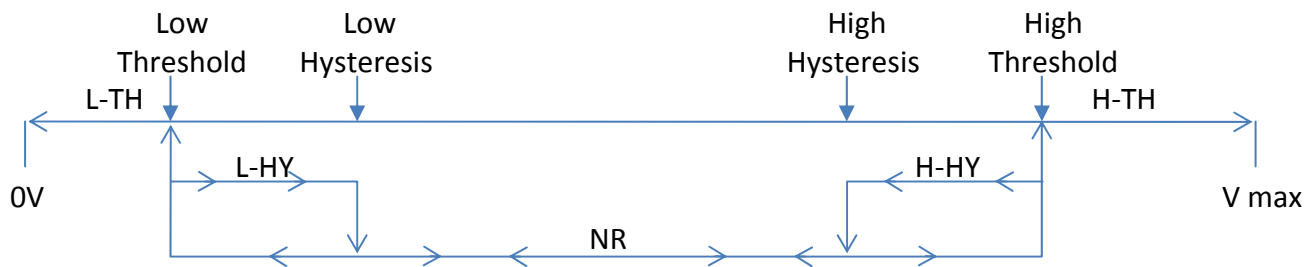
0 - H-TH (No alert)
8000 - H-TH = 20v
7000 - H-HY = 18v

Analog inputs - Status

Command	Reply	Parameters
PY#ADCL=A	PY#ADCL=A #adcl: A,B,C ok	A - Analog input select: 0 > Vin (VCON J31 pin#13) 1 > ADC IN#1 (CONHVIO1 J31 pin#5) 2 > ADC IN#2 (CONHVIO2 J31 pin#4)
PY#ADCL?	#adcl: 0,B,C #adcl: 1,B,C #adcl: 2,B,C ok	B - Measure result (mV). C - Level state on of the follow: L-TH or L-HY or H-TH or H-HY or NR (refer to the diagram)
PY#ADCL=?	#adcl: (0-2) ok	

Description:

This command, returns the analog input level (x 1mV) & the measure state (Threshold or Hysteresis or Normal).



Example: Input Setting "PY#ADCS=1,0,10,2,6000,7000,0,9000,8000"

PY#ADCL=0 available results (x - V measure)
(x)

#adcl: 0,5500,L-TH - (x < L-TH)

#adcl: 0,6500,L-HY - (L-TH > x < L-HY) , this result can get if the measure already pass the L-TH.

#adcl: 0,6500,NR - (L-TH > x < H-TH)