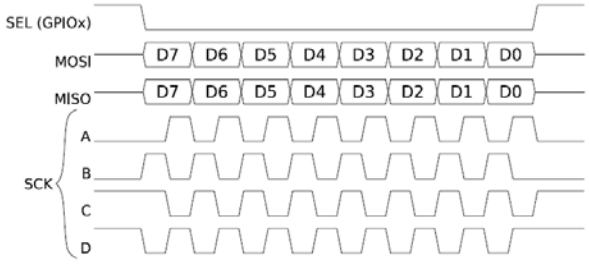


- Key switch
 - ① Public: switch to public key [AT\$410=1]
 - ② Private: switch to private key [AT\$410=0]

- AT commands

Command	Name	Description												
AT	Dummy Command	Just returns 'OK' and does nothing else. Can be used to check communication.												
AT\$SB=bit[,bit]	Send Bit	Send a bit status (0 or 1). Optional bit flag indicates if AX-SFEU should receive a downlink frame.												
AT\$SF=frame[,bit]	Send Frame	Send payload data, 1 to 12 bytes. Optional bit flag indicates if AX-SFEU should receive a downlink frame.												
AT\$SO	Manually send out of band message	Send the out-of-band message.												
AT\$TR?	Get the transmit repeat	Returns the number of transmit repeats. Default: 2												
AT\$TR=?	Get transmit range	Returns the allowed range of transmit repeats.												
AT\$TR=uint	Get transmit repeat	Sets the transmit repeat.												
AT\$uint?	Get Register	Query a specific configuration register's value. See chapter "Registers" for a list of registers.												
AT\$uint=uint	Set Register	Change a configuration register.												
AT\$uint=?	Get Register Range	Returns the allowed range of transmit repeats.												
AT\$IF=uint	Set TX Frequency	Set the output carrier macro channel for Sigfox frames.												
AT\$IF?	Get TX Frequency	Get the currently chosen TX frequency.												
AT\$DR=uint	Set RX Frequency	Set the reception carrier macro channel for Sigfox frames.												
AT\$DR?	Get RX Frequency	Get the currently chosen RX frequency.												
AT\$CW=uint,bit[,uint_opt]	Continuous Wave	<p>To run emission tests for Sigfox certification it is necessary to send a continuous wave, i.e. just the base frequency without any modulation. Parameters:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Range</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Frequency</td> <td>800000000–999999999, 0</td> <td>Continuous wave frequency in Hz. Use 868130000 for Sigfox or 0 to keep previous frequency.</td> </tr> <tr> <td>Mode</td> <td>0, 1</td> <td>Enable or disable carrier wave.</td> </tr> <tr> <td>Power</td> <td>0–14</td> <td>dBm of signal Default: 14</td> </tr> </tbody> </table>	Name	Range	Description	Frequency	800000000–999999999, 0	Continuous wave frequency in Hz. Use 868130000 for Sigfox or 0 to keep previous frequency.	Mode	0, 1	Enable or disable carrier wave.	Power	0–14	dBm of signal Default: 14
Name	Range	Description												
Frequency	800000000–999999999, 0	Continuous wave frequency in Hz. Use 868130000 for Sigfox or 0 to keep previous frequency.												
Mode	0, 1	Enable or disable carrier wave.												
Power	0–14	dBm of signal Default: 14												
AT\$CB=uint_opt,bit	Test Mode: TX constant byte	<p>For emission testing it is useful to send a specific bit pattern. The first parameter specifies the byte to send. Use '-1' for a (pseudo-)random pattern. Parameters:</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Range</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Pattern</td> <td>0–255, -1</td> <td>Byte to send. Use '-1' for a (pseudo-)random pattern.</td> </tr> <tr> <td>Mode</td> <td>0, 1</td> <td>Enable or disable pattern test mode.</td> </tr> </tbody> </table>	Name	Range	Description	Pattern	0–255, -1	Byte to send. Use '-1' for a (pseudo-)random pattern.	Mode	0, 1	Enable or disable pattern test mode.			
Name	Range	Description												
Pattern	0–255, -1	Byte to send. Use '-1' for a (pseudo-)random pattern.												
Mode	0, 1	Enable or disable pattern test mode.												
AT\$T?	Get Temperature	Measure internal temperature and return it in 1/10 th of a degree Celsius.												
AT\$V?	Get Voltages	Return current voltage and voltage measured during the last transmission in mV.												

Command	Name	Description																						
AT\$I=uint	Information	<p>Display various product information:</p> <p>0: Software Name & Version Example Response: AX-SFEU 1.0.6-ETSI</p> <p>1: Contact Details Example Response: support@axsem.com</p> <p>2: Silicon revision lower byte Example Response: 8F</p> <p>3: Silicon revision upper byte Example Response: 00</p> <p>4: Major Firmware Version Example Response: 1</p> <p>5: Minor Firmware Version Example Response: 0</p> <p>7: Firmware Variant (Frequency Band etc. (EU/US)) Example Response: ETSI</p> <p>8: Firmware VCS Version Example Response: v1.0.2-36</p> <p>9: SIGFOX Library Version Example Response: DL0-1.4</p> <p>10: Device ID Example Response: 00012345</p> <p>11: PAC Example Response: 0123456789ABCDEF</p>																						
AT\$P=uint	Set Power Mode	<p>To conserve power, the AX-SFEU can be put to sleep manually. Depending on power mode, you will be responsible for waking up the AX-SFEU again!</p> <p>0: software reset (settings will be reset to values in flash)</p> <p>1: sleep (send a break to wake up)</p> <p>2: deep sleep (toggle GPIO9 or RESET_N pin to wake up; the AX-SFEU is not running and all settings will be reset!)</p>																						
AT\$WR	Save Config	<p>Write all settings to flash (RX/TX frequencies, registers) so they survive reset/deep sleep or loss of power.</p> <p>Use AT\$P=0 to reset the AX-SFEU and load settings from flash.</p>																						
AT:Pn?	Get GPIO Pin	<p>Return the setting of the GPIO Pin n; n can range from 0 to 9. A character string is returned describing the mode of the pin, followed by the actual value. If the pin is configured as analog pin, then the voltage (range 0...1 V) is returned. The mode characters have the following meaning:</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Pin drives low</td> </tr> <tr> <td>1</td> <td>Pin drives high</td> </tr> <tr> <td>Z</td> <td>Pin is high impedance input</td> </tr> <tr> <td>U</td> <td>Pin is input with pull-up</td> </tr> <tr> <td>A</td> <td>Pin is analog input (GPIO pin 0...3 only)</td> </tr> <tr> <td>T</td> <td>Pin is driven by clock or DAC (GPIO pin 0 and 4 only)</td> </tr> </tbody> </table> <p>The default mode after exiting reset is U on all GPIO pins.</p>	Mode	Description	0	Pin drives low	1	Pin drives high	Z	Pin is high impedance input	U	Pin is input with pull-up	A	Pin is analog input (GPIO pin 0...3 only)	T	Pin is driven by clock or DAC (GPIO pin 0 and 4 only)								
Mode	Description																							
0	Pin drives low																							
1	Pin drives high																							
Z	Pin is high impedance input																							
U	Pin is input with pull-up																							
A	Pin is analog input (GPIO pin 0...3 only)																							
T	Pin is driven by clock or DAC (GPIO pin 0 and 4 only)																							
AT:Pn=?	Get GPIO Pin Range	<p>Print a list of possible modes for a pin. The table below lists the response.</p> <table border="1"> <thead> <tr> <th>Pin</th> <th>Modes</th> </tr> </thead> <tbody> <tr> <td>P0</td> <td>0, 1, Z, U, A, T</td> </tr> <tr> <td>P1</td> <td>0, 1, Z, U, A</td> </tr> <tr> <td>P2</td> <td>0, 1, Z, U, A</td> </tr> <tr> <td>P3</td> <td>0, 1, Z, U, A</td> </tr> <tr> <td>P4</td> <td>0, 1, Z, U, T</td> </tr> <tr> <td>P5</td> <td>0, 1, Z, U</td> </tr> <tr> <td>P6</td> <td>0, 1, Z, U</td> </tr> <tr> <td>P7</td> <td>0, 1, Z, U</td> </tr> <tr> <td>P8</td> <td>0, 1, Z, U</td> </tr> <tr> <td>P9</td> <td>0, 1, Z, U</td> </tr> </tbody> </table>	Pin	Modes	P0	0, 1, Z, U, A, T	P1	0, 1, Z, U, A	P2	0, 1, Z, U, A	P3	0, 1, Z, U, A	P4	0, 1, Z, U, T	P5	0, 1, Z, U	P6	0, 1, Z, U	P7	0, 1, Z, U	P8	0, 1, Z, U	P9	0, 1, Z, U
Pin	Modes																							
P0	0, 1, Z, U, A, T																							
P1	0, 1, Z, U, A																							
P2	0, 1, Z, U, A																							
P3	0, 1, Z, U, A																							
P4	0, 1, Z, U, T																							
P5	0, 1, Z, U																							
P6	0, 1, Z, U																							
P7	0, 1, Z, U																							
P8	0, 1, Z, U																							
P9	0, 1, Z, U																							
AT:Pn=mode	Set GPIO Pin	<p>Set the GPIO pin mode.</p> <p>For a list of the modes see the command AT:Pn?</p>																						

Command	Name	Description															
AT:ADC Pn[-Pn[(1V 10V)]]?	Get GPIO Pin Analog Voltage	Measure the voltage applied to a GPIO pin. The command also allows measurement of the voltage difference across two GPIO pins. In differential mode, the full scale range may also be specified as 1 V or 10 V. Note however that the pin input voltages must not exceed the range 0..VDD_IO. The command returns the result as fraction of the full scale range (1 V if none is specified). The GPIO pins referenced should be initialized to analog mode before issuing this command.															
AT:SPI[(A B C D)]=bytes	SPI Transaction	<p>This command clocks out bytes on the SPI port. The clock frequency is 312.5 kHz. The command returns the bytes read on MISO during output. Optionally the clocking mode may be specified (default is A):</p> <table border="1"> <thead> <tr> <th>Mode</th> <th>Clock Inversion</th> <th>Clock Phase</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>normal</td> <td>normal</td> </tr> <tr> <td>B</td> <td>normal</td> <td>alternate</td> </tr> <tr> <td>C</td> <td>inverted</td> <td>normal</td> </tr> <tr> <td>D</td> <td>inverted</td> <td>alternate</td> </tr> </tbody> </table>  <p>Note that SEL, if needed, is not generated by this command, and must instead be driven using standard GPIO commands (AT:Pn=0 1).</p>	Mode	Clock Inversion	Clock Phase	A	normal	normal	B	normal	alternate	C	inverted	normal	D	inverted	alternate
Mode	Clock Inversion	Clock Phase															
A	normal	normal															
B	normal	alternate															
C	inverted	normal															
D	inverted	alternate															
AT:CLK=freq,reffreq	Set Clock Generator	Output a square wave on the pin(s) set to T mode. The frequency of the square wave is $(\text{freq} / 2^{16}) \times \text{reffreq}$. Possible values for reffreq are 20000000, 10000000, 5000000, 2500000, 1250000, 625000, 312500, 156250. Possible values if freq are 0...65535.															
AT:CLK=OFF	Turn off Clock Generator	Switch off the clock generator															
AT:CLK?	Get Clock Generator	Return the settings of the clock generator. Two numbers are returned, freq and reffreq.															
AT:DAC=value	Set $\Sigma\Delta$ DAC	Output a $\Sigma\Delta$ DAC value on the pin(s) set to T mode. Parameter value may be in the range -32768...32767. The average output voltage is $(1/2 + \text{value} / 2^{17}) \times \text{VDD}$. An external low pass filter is needed to get smooth output voltages. The modulation frequency is 20 MHz. A possible low pass filter choice is a simple RC low pass filter with R = 10 k Ω and C = 1 μF .															
AT:DAC=OFF	Turn off $\Sigma\Delta$ DAC	Switch off the DAC															
AT:DAC?	Get $\Sigma\Delta$ DAC	Return the DAC value															

Command	Name	Description
AT\$TM=mode,config	Activates the Sigfox Testmode	<p>Available test modes:</p> <p>0. TX BPSK Send only BPSK with Synchro Bit + Synchro frame + PN sequence: No hopping centered on the TX_frequency. Config bits 0 to 6 define the number of repetitions. Bit 7 of config defines if a delay is applied or not in the loop</p> <p>1. TX Protocol: Tx mode with full protocol with Sigfox key: Send Sigfox protocol frames with initiate downlink flag = True. Config defines the number of repetitions.</p> <p>2. RX Protocol: This mode tests the complete downlink protocol in Downlink only. Config defines the number of repetitions.</p> <p>3. RX GFSK: RX mode with known pattern with SB + SF + Pattern on RX_frequency (internal comparison with received frame ⇔ known pattern = AA AA B2 27 1F 20 41 84 32 68 C5 BA AE 79 E7 F6 DD 9B. Config defines the number of repetitions. Config defines the number of repetitions.</p> <p>4. RX Sensitivity: Does uplink + downlink frame with Sigfox key and specific timings. This test is specific to SIGFOX's test equipments & softwares.</p> <p>5. TX Synthesis: Does one uplink frame on each Sigfox channel to measure frequency synthesis step</p>
AT\$SE	Starts AT\$TM=3,255 indefinitely	Convenience command for sensitivity tests
AT\$SL[=frame]	Send local loop	Sends a local loop frame with optional payload of 1 to 12 bytes. Default payload: 0x84, 0x32, 0x68, 0xC5, 0xBA, 0x53, 0xAE, 0x79, 0xE7, 0xF6, 0xDD, 0x9B.
AT\$RL	Receive local loop	Starts listening for a local loop.