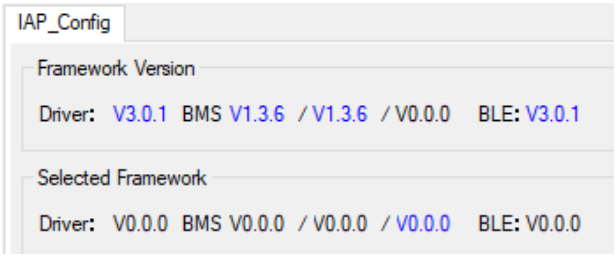
Application Note: Segway-Ninebot ES4 Sharing Kick-Scooter

Overview

Astra Telematics AT240 and AT200 devices can be used with the Segway Ninebot ES4 Kick-Scooter "sharing" version. The solution allows control of the scooter through the device, using Astra $ commands over RS232, SMS, TCP socket or BLE (where available on device hardware) to control the scooter and access any of the configuration options made available through the Ninebot UART protocol. Data from the scooter is read and reported, as described later in this document.

Scooter Requirements

Segway-Ninebot ES4 Sharing version kick-scooter must be loaded with Ninebot sharing firmware, as below:



Utilities, instructions and binary files for the above firmware can be found here:

https://astratelematics-my.sharepoint.com/:f:/g/personal/phil_gps-telematics_co_uk/EnTqlD2zd7NJoRuRzCyEQ_QBcaeOjRNMHBDaOIqhLPD4-g?e=7fjKx2

Astra Device Firmware:

The features in this application note are applicable from Astra firmware version E7.0.30.xx with NINEBOT build option enabled. Please discuss with Astra Telematics support team for more details.

Astra Device Hardware Options:

Note that this application requires that the AT240/AT200 device serial port operates at TTL levels, rather than the standard RS232 levels, hence devices must be ordered with that option:

AT240-TTL
AT200-TTL

There is no extra charge for this option, but it requires a hardware change and therefore must be configured by Astra prior to shipping.

Astra Device Configuration:

$SRAL,2,12,115200
$IGNM,4
$PROT,16,2348812399

Version 1.5 31/03/2019

Data Format and Communication Protocol

Data is read from the ES4 scooter at all time, as in IoT mode it is never switched off, and is reporting using Module 32 of our Modular Communication Protcol X:

32	**2147483648**	**0**	**SEGWAY NINEBOT ES4 SHARING:**	current speed	whilst scooter is turned on	0x55 heartbeat	uint	kmh x 10	1
				internal battery voltage		0x55 heartbeat	uint	volts	1
				external battery voltage		0x55 heartbeat	uint	volts	1
				dry battery voltage		0x55 heartbeat	uint	volts	1
				internal battery SoC		0x55 heartbeat	uint	%	1
				external battery SoC		0x55 heartbeat	uint	%	1
				overall battery SoC		0x55 heartbeat	uint	%	1
				internal battery temperature 1		0x55 heartbeat	uint	degrees C	1
				internal battery temperature 2		0x55 heartbeat	uint	degrees C	1
				external battery temperature 1		0x55 heartbeat	uint	degrees C	1
				external battery temperature 2		0x55 heartbeat	uint	degrees C	1
				status		0x55 heartbeat	bitfield		1
								module total:	**12**

Documentation for Astra Modular Communication Protocol X can be found here:

https://astratelematics-my.sharepoint.com/:f:/g/personal/phil_gps-telematics_co_uk/EkFBHuzBPVBChXIgVKhC74EBcJU9x5fWcX-cKiiUFUtShA?e=gSmSUA

Installation Process and Scooter IoT Features

After updating the scooter, connect the Astra device and power on the scooter using the button or $NBPW,1 command. Once the Astra device is connected to the scooter and both are turned on, the scooter will detect the presence of the IoT device and enter “IoT mode”, which is basically ready for sharing applications, as follows:

1. Button disabled for powering scooter ON/OFF
2. Button disabled for switching headlight ON/OFF
3. Button disabled for changing riding modes
4. On/off control wire disabled
5. Auto-off disabled, scooter can be left on indefinitely
6. LOCK mode - brake activates, lights flash and beeper sounds if attempted to ride or push
7. Whilst LOCKed, screen and lights are off
8. UNLOCK mode – read to ride
9. If the IoT device is disconnected, the scooter will indicate error 32 (3 long beeps, followed by 2 short beeps)

Astra OTA Commands:

$NBLK,1 ninebot lock
$NBLK,0 ninebot unlock
$FIND ninebot find – flash lights and sound beeper (3 times, 2 seconds approx.)

NOTE: these commands use the same control wire as the button, and hence don’t work in IoT mode:
$NBPW,1 ninebot power ON
$NBPW,0 ninebot power OFF

Version 1.5 31/03/2019

Astra-Ninebot Generic Configuration Command:

Allows any command and value to be set in the Ninebot memory control table. See Ninebot ES Scooter Protocol 917 for details of commands and options.

$NBSC,<address>,<value>

For example, see the extract from the above document, showing the NB_CTL_REBOOT command:

Address	Name	Command item	Type	Permission[1]	Initial value
0x77	NB_CTL_ENGINE	Start or shut down the engine	S16	R/W	0
0x78	NB_CTL_REBOOT	Restart system [6], write 1 for restart	S16	R/W	0
0x79	NB_CTL_POWEROFF	Shutdown [6], write 1 for shutdown	S16	R/W	0

To send a REBOOT using the Astra Ninebot Generic Command $NBSC, take the address 0x78 and the value 1, convert to decimal, 120 and 1, then use with the command as follows:

$NBSC,120,1

DEVICE STATUS CHECK:

$TEST returns ES4 master controller firmware version, as follows:

TEST:AT240V8
7.0.30.36
357520074225570
PWR:41.0V (100%)
BAT:98%
GPS:N/A (0%)
GPRS:ERR (0%)
APN:N/A
SKT:N/A
ACK:N/A
IGN:ERR (ON)
IMOB:OFF
REPQ:0
NBFW:3.0.1