FUNCTIONS USED FOR COMMUNICATION WITH A6 MODULE ON CRODUINO ETA BOARD DEFINED IN ETA_AT LIBRARY

Function name	Return value	Function variables	Function description
`updateSerial()`	`void`		Function enables the input of AT commands via Serial Monitor and forwards modules responce to Serial Monitor
`powerOn()`	`void`		Function powers the A6 module on
`powerOff()`	`void`		Function powers the A6 module off
`infoGenral`	`void`		Lists general info about Croduino ETA A6 board
`Test()`	`void`		Function initializes all AT commands required for communication to begin.
`SMSsend(char *addr, char*msg)`	`void`	addr-number to which we want to send the SMS msg-message we're sending	Function sends an SMS message to the given number
`SMSread(int x)`	`String`	x-index on which SMS is written	Function reads the SMS message which is located on the given index(x) alongside with all the inforation that comes with an SMS (number of sender, date of reception) or, in case of an error, returns the number of said error
`SMSdelete(int x)`	`String`	x- index on which SMS that we want to delete is written	Function deletes the SMS which is located on the given index(x) or, in case of an error, returns the number of said error
`SMSformat()`	`void`		Function deletes all SMS messages that were stored to SIM card.
`SMSlistAll()`	`void`		Function lists all SMS messages stored on SIM to Serial Monitor
`SMSlistUnread()`	`void`		Function lists all unread SMS messages stored on SIM to Serial Monitor

`SMSlistRead()`	`void`		Function lists all read SMS messages stored on SIM to Serial Monitor
`SMSnumber()`	`int`		Function returns the number of SMS messages stored on sim
`SMSnumberOfSender(String x)`	`String`	x-String we got from SMSread()	From the String we got from SMSread() we get number of the sender and return it as a String variable
`SMStext(String x)`	`String`	x- String we got from SMSread()	From the String we got out of SMSread() we get message text of the SMS message and return it as a String variable
`callMake(char *addr)`	`void`	addr-number we want to dial	Function dials the given number
`callAnswer()`	`void`		Function answers the incoming call
`callHangUp()`	`void`		Function declines the incoming call
`autoAnswerSettings(int x, int y)`	`void`	x-number of „rings“ y=1(if we want to answer the incoming calls) ili y=2 2(if we want to decline incoming calls)	Function sets automatc call answering/declining after defined number of „rings“.
`switchToSpeakers()`	`void`		Function sets the speaker as a default communication device
`switchToHeadphones()`	`void`		Function sets headphones as a default communication device
`enableTextMode()`	`void`		Function switches from default PDU mode to an easier to read TEXT mode
`enablePDUMode()`	`void`		Function switches from TEXT mode to default PDU mode

`begin(unsigned long speed)`	`void`	Speed-value of baud rate we wish to set	Function sets baud rate for serial communication with A6 module
`getSignalQuality()`	`int`		Function returns the value of signal quality in 0-100 range
`SIMgetID()`	`String`		Function returns SIM ID
`availableOperators()`	`String`		Function returns available mobile networks as a String
`currentOp()`	`String`		Function returns mobile network code on which we are currently connected
`requestHTTP(const char *a, const char *b, const char *c, const char *d);`	`String`	a – APN, b – Server we connect, c – API link, d - Host address	HTTP request – connects to web page with TCP/IP communication and returns data you want to read with GET method
`closeHTTP`	`void`		Close HTTP communication
`ETA_AT()`			Constructor for the class

If you'd wish for functions to write responces from A6 module to Serial Monitor you'll have to go to:

Arduino/libraries/ETA_AT/src/ETA_AT.h

and in that file uncomment line 7, #define DEBUG.

Function currentOp() returns mobile network code which you can find and check to which mobile network it belongs on:

https://clients.txtnation.com/hc/en-us/articles/218719768-MCCMNC-mobile-country-code-and-mobile-network-code-list-