Eccel Technology Ltd
Embedded RFID made simple

B1 Client User Guide

v1.2
28/08/2017

Table of contents

1 Introduction 2

2 Quick start guide 3

2.1 Minimum software requirements 3

2.2 Software installation 3

2.3 Connecting the reader 3

3 Interface description 4

3.1 Settings panel 4

3.2 Commands panel 5

3.2.1 Simple commands 5

3.2.2 UART Config 8

3.2.3 IO 12

3.2.4 RFID Commands 16

3.2.5 RFID Memory 58

3.2.6 RFID Info 60

3.3 Output panel 61

4 Error codes 62

1 Introduction

This document describes the functionality of the graphical user interface which is used to operate the RFID B1 module and other B1-based readers e.g. the USB-B1. The B1 client is a very useful tool which allows simple testing of all features of the B1 module. Thanks to its simplicity, the B1 client can be used both by experienced and inexperienced RFID engineers. This software aids better understanding of the memory organization of different tags, available commands and the results of command execution.

Chapter 2 includes instructions on how to get started with the B1 client and how to connect it to the B1 based module family of products.

The B1 client is divided into 3 panels: Settings panel, Commands panel and Output panel. Each of these are described in detail with multiple examples in chapter 3. Section 3.1 describes the Settings panel, section 3.2 describes the Commands panel and section 3.3 describes the Output panel.

Chapter 4 contains information about possible problems the user may encounter when using the B1 client and testing B1 based modules.

All examples described in this document were done using the USB-B1 reader.

2 Quick start guide

2.1 Minimum software requirements

The application is designed to be used in a Microsoft® Windows® environment. The B1 Client requires Microsoft .NET 4.0 Framework. This can be found at:

https://www.microsoft.com/en-US/download/details.aspx?id=17851

2.2 Software installation

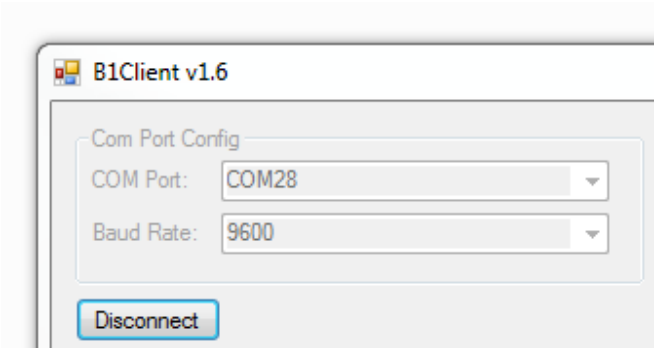
To install the B1 Client, download the latest version from http://www.eccel.co.uk/wp-content/uploads/B1Client.zip, extract the zip file and start the B1 Client.

Connect the B1-based reader, e.g. the USB-B1 to the PC via the USB interface, make sure that the jumpers selecting antenna type are properly set up. All required drivers should install automatically. Once the driver is installed, it will need to find out the number of the COM Port that has been set up.

Open the Device Manager, look in *Ports (COM & LTP)* and find the COM Port which was created by the USB driver e.g. *USB Serial Port (COM 28).*

2.3 Connecting the reader

In the main window of the B1 Client chose the proper COM port, select the Baud Rate to 9600 and click *Connect* button. 9600 is the default Baud Rate during the initial power up of the B1 based module but it can be changed. For more information please see section 3.2.2. When everything is properly set up the *Connect* button will change into the *Disconnect* button.



3 Interface description

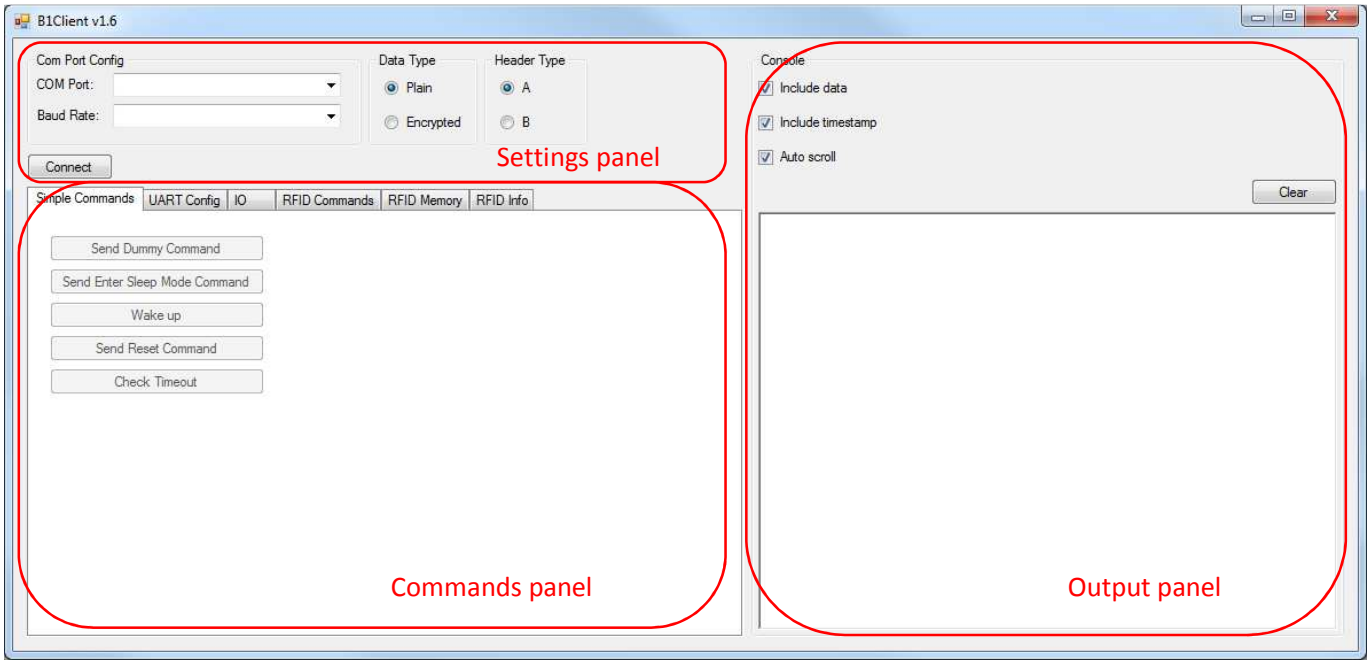


Figure 1. B1 Client User Interface

3.1 Settings panel

In the **COM port config** field there are two drop-down lists: **COM Port** and **Baud Rate**. From the first list the user can select the proper COM port. From the second list the user can select the proper Baud Rate. All new B1-based modules have Baud Rate at 9600 by default. Baud Rate can be changed after connecting.

In the **Data Type** field the user can configure the data type. The check box by Plain data is checked by default.

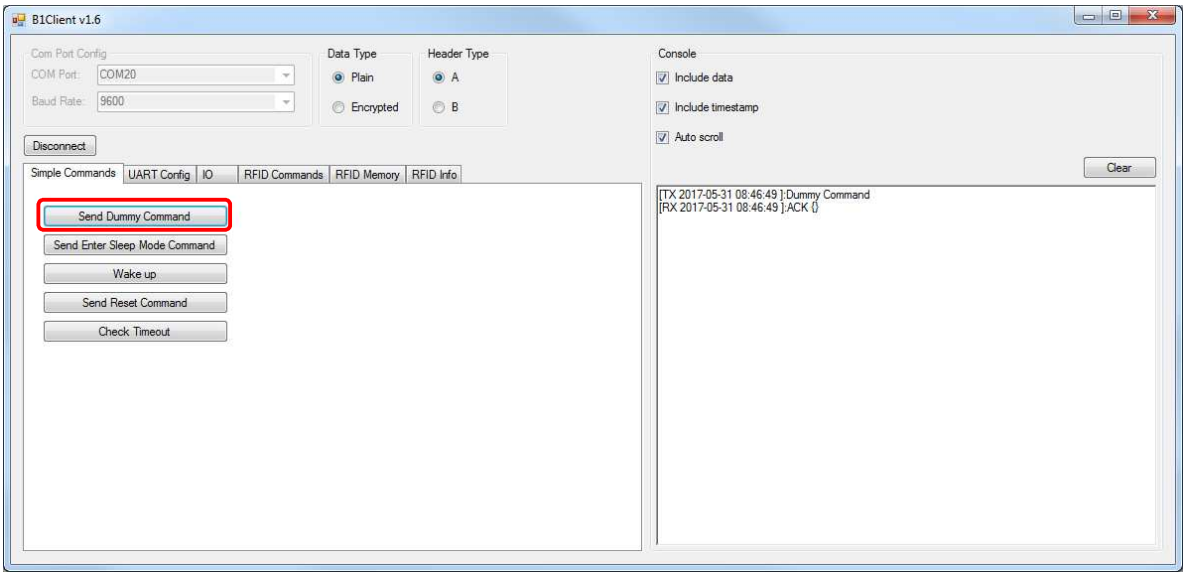
In the **Header Type** field the user can select between the A or B Header type. More information about header types can be found in the B1 User Manual.

3.2 Commands panel

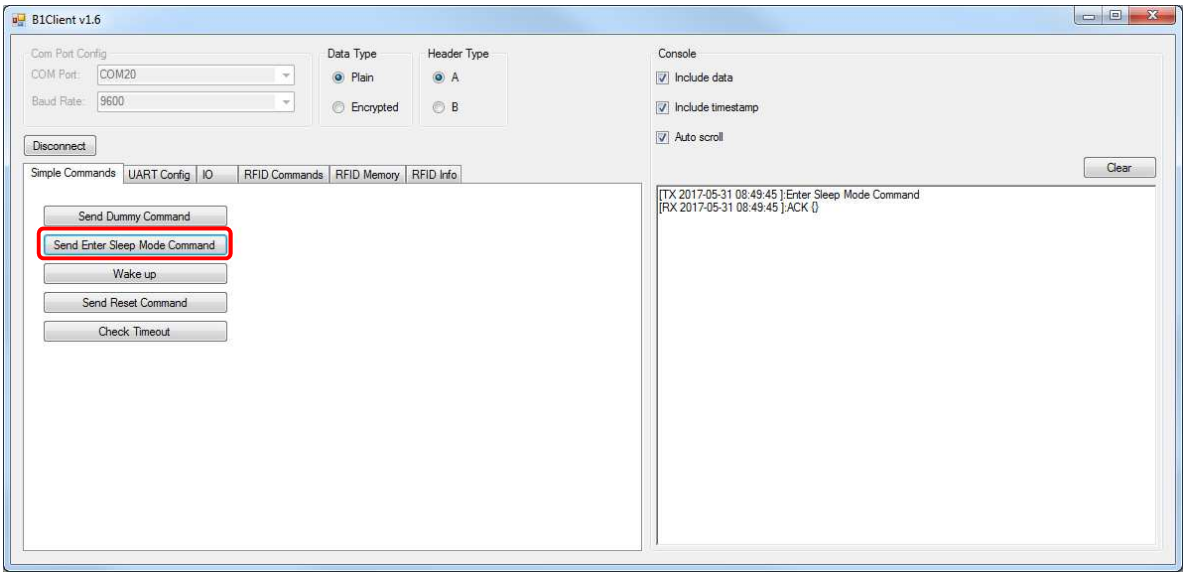
The Commands panel contains many predefined commands grouped into 6 tabs which allow full testing of all features of the B1-based readers.

3.2.1 Simple commands

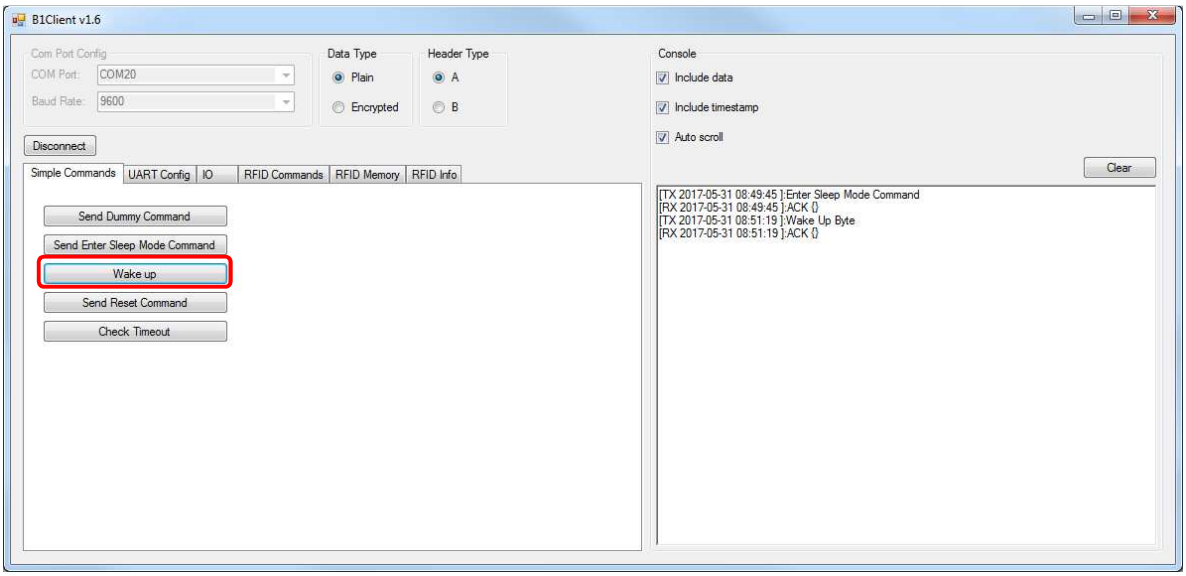
Send Dummy Command – This command is used to check that the module works and communication settings are properly set up. The module replies to this command with an ACK response.



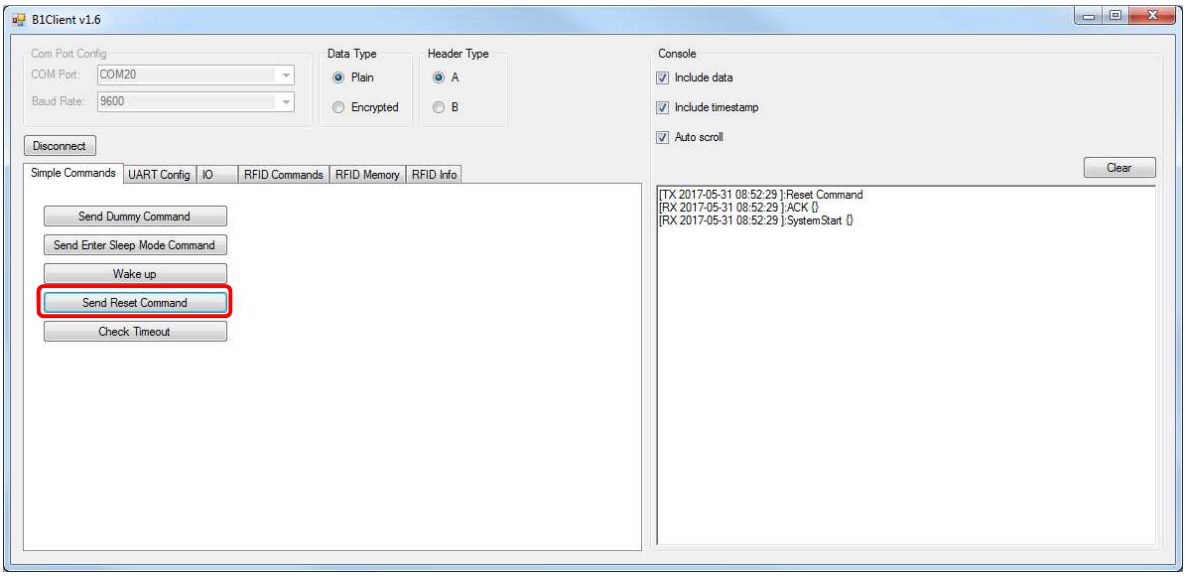
Send Enter Sleep Mode Command – After receiving this command the module replies with an ACK response and enters Sleep Mode.



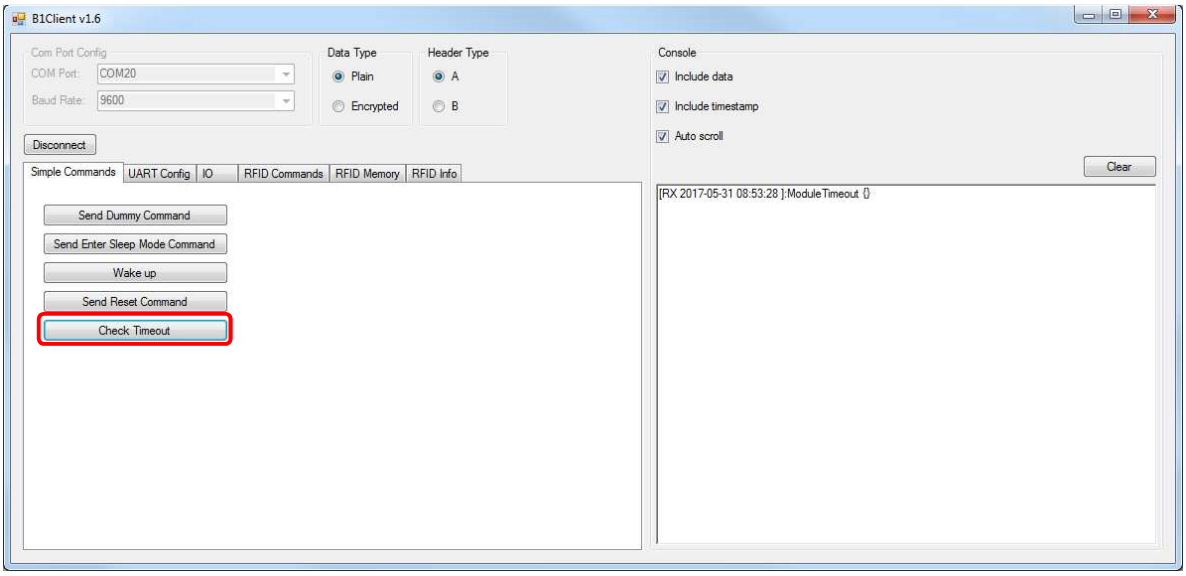
Wake up – This command wakes the module up from the Sleep Mode. After waking up the module sends an ACK response.



Send Reset Command – Reset the B1-based module. The module should reply with an ACK response and after reset the initialization procedure should start. The user will be informed by receiving the SystemStart response.

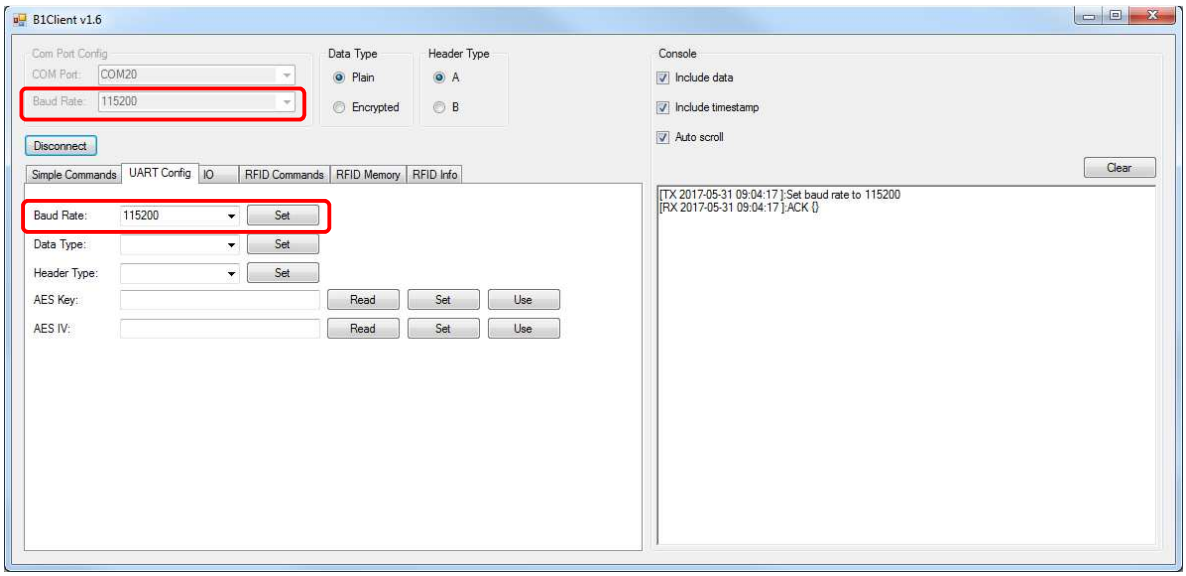


Check Timeout – This command is used to check if the time delay between bytes in a packet is under 100 ms. A timeout is implemented in the system to avoid the scenario that a lost byte during transmission could cause an infinite time delay for the module to respond. When the module starts receiving (which means it detects a STX character) it sets up a timer which is reset upon receipt of each successive character and turned off at the end of the transmission. When the timer reaches 100ms a timeout packet is sent and the RX buffer is cleared. In practice this places a restriction upon the master controller that the time delay between any two bytes sent in a packet must be less than 100ms.

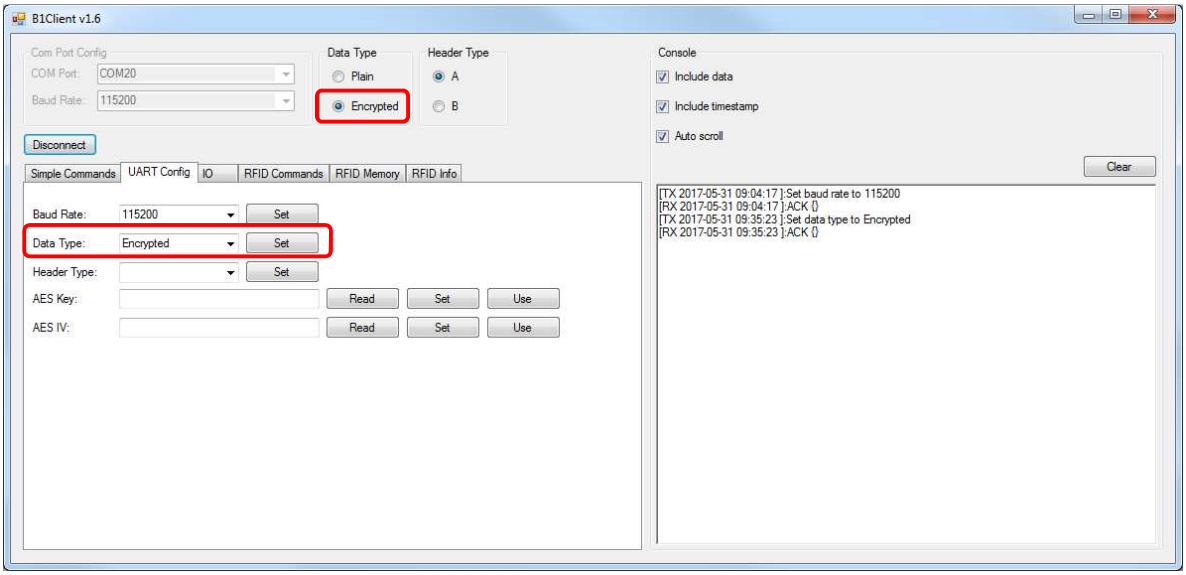


3.2.2 UART Config

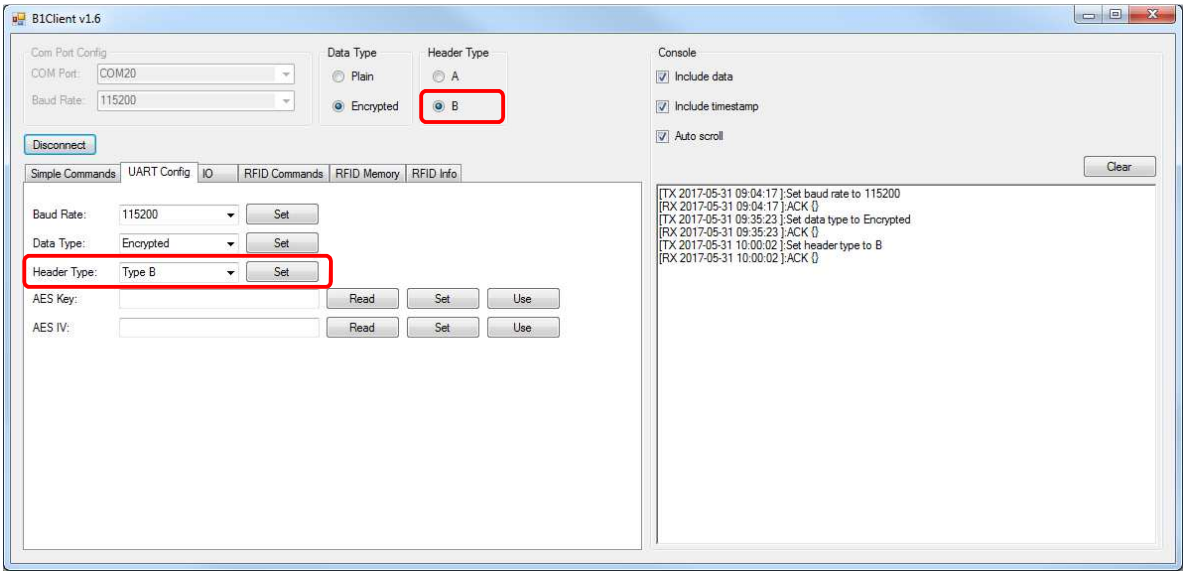
Baud Rate – The user can change UART Baud Rate by choosing one of the available speeds: 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200. After changing Baud Rate in the UART Config tab the user should also click the "Disconnect" button, change Baud Rate in the Settings panel to the same value, chose the proper COM Port and connect with the module. A new value of Baud Rate will be saved in the module's memory even after disconnecting the power supply from a module. During the next attempt to connect a module with the B1 client the user should select a predefined Baud Rate instead of default 9600. To restore a module to the default settings, disconnect a module from the power supply, insert a jumper between the 'PDWN' (Power down) and the 'GND' pin, connect the power supply for a few seconds, disconnect the power supply and then remove the jumper. After all these operations, the module is restored to its default settings ie. Baud Rate is 9600.



Data Type – Before executing this command the user should set the AES Key and AES IV (by default they equal '0'). The user can select between the Plain or Encrypted Data Type. The default value is 'Plain'. When changing to 'Encrypted' the user should also change the Date Type in the Settings Panel.



Header Type – The user can select between the A or B Header Type. More information about Header Types can be found in the B1 user manual. The default type is 'A'. While changing to 'B' the user should also change the Header Type in the Settings Panel.



REMARK: The procedure of restoring the default settings of the UART is described in the Baud Rate section.

AES Key – The Read AES-128 Key command takes no arguments. When the module receives a valid command, it replies with an ACK and the bytes of the AES Key that are to be used for the encryption of data. By default the AES Key 0 and AES Key 1 are 0x0000000000000000(h).

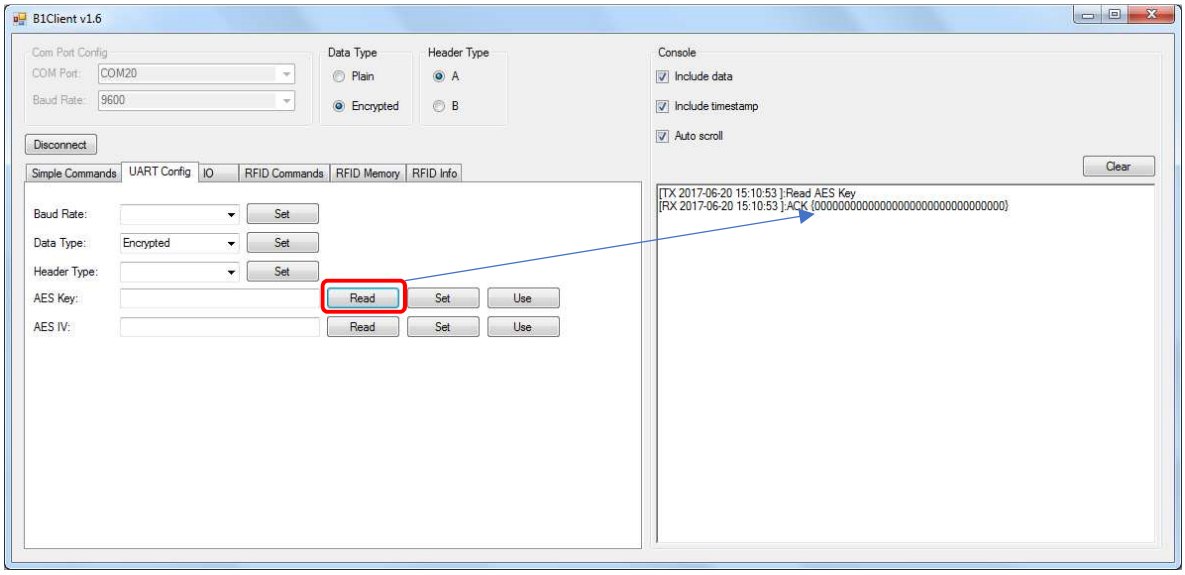


Figure 1. The Read AES Key command (default value is '0')..

The Set AES Key command takes as an argument an AES Key which is a 16-byte long encryption key used for encrypting and decrypting data. When the module receives this command, it changes the AES key and replies with an ACK response and no parameters. The reply is encrypted with the new key if Encrypted Data is selected. To Set a new AES Key the user should write it in the text box, press 'Set' button and then press 'Use' button.

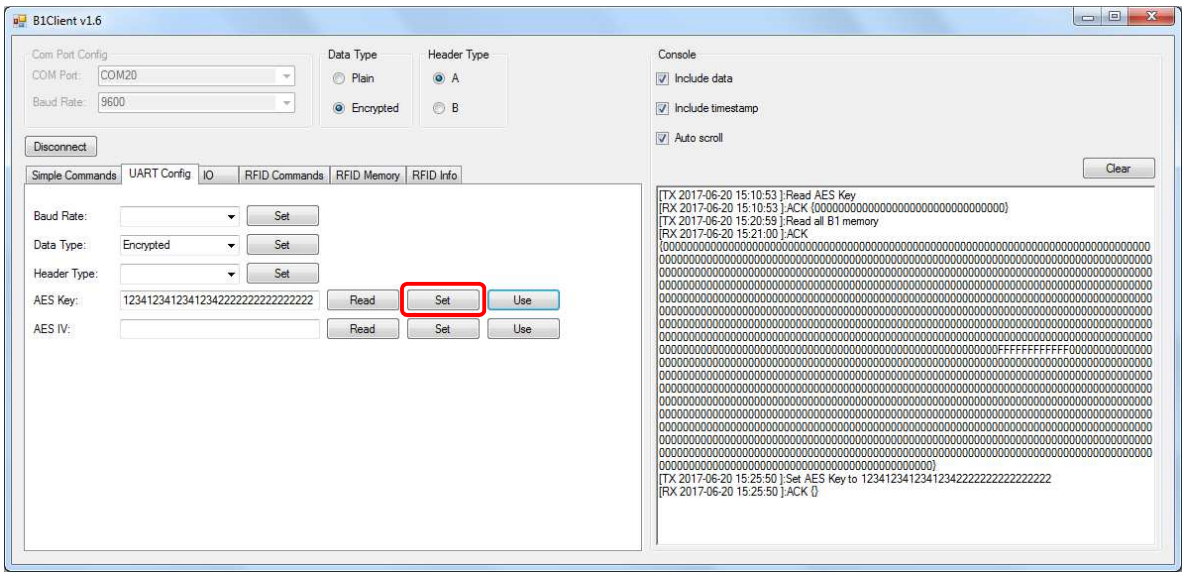


Figure 2. Setting a new AES Key.

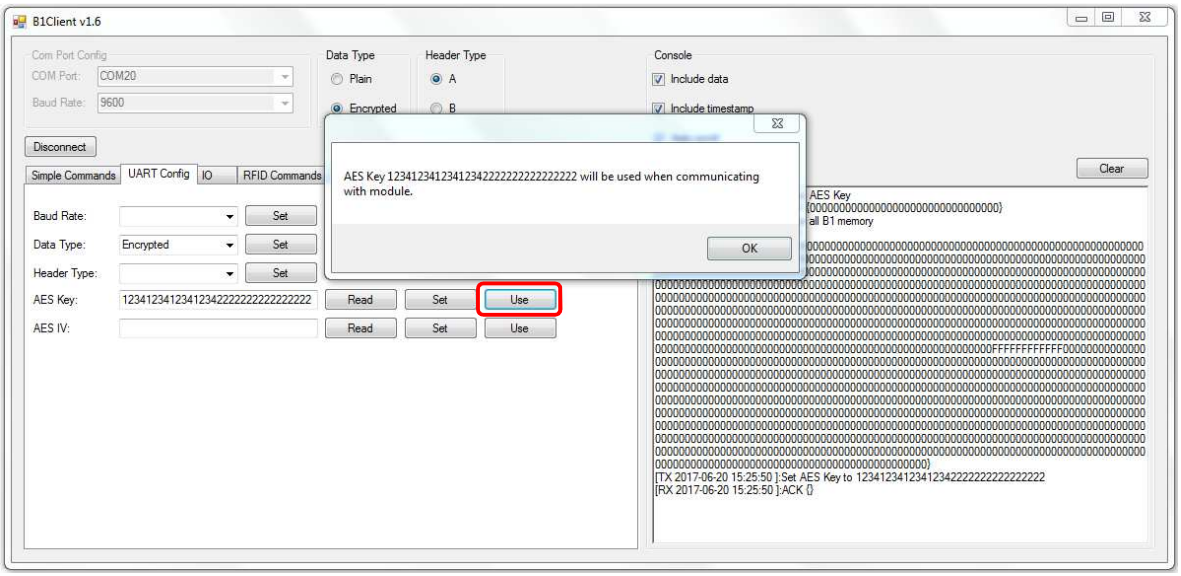


Figure 3. Information after clicking the 'Use' button.

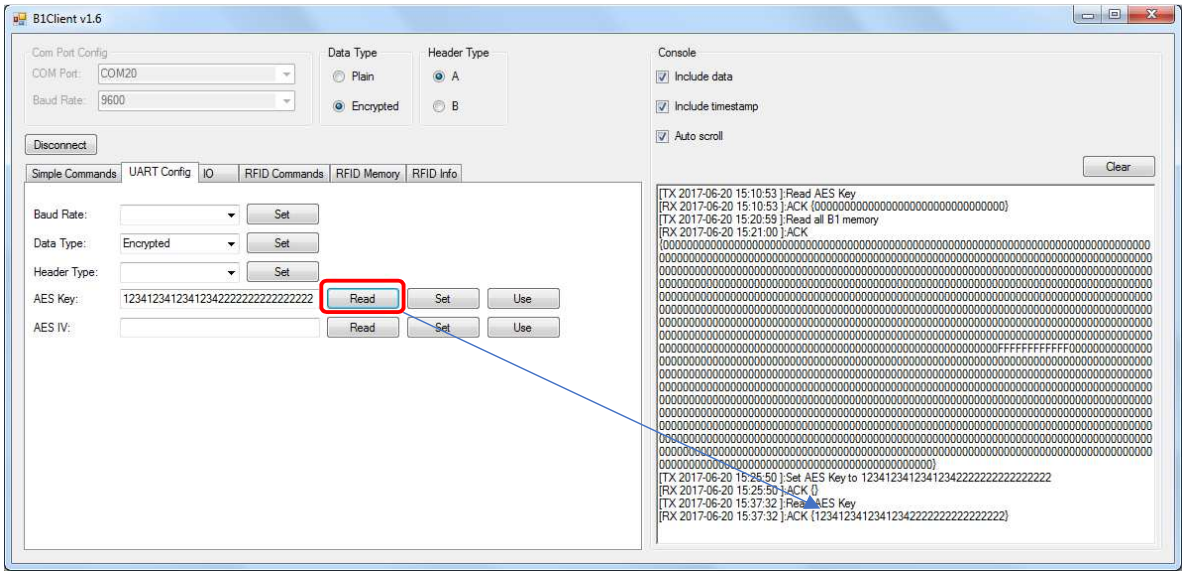


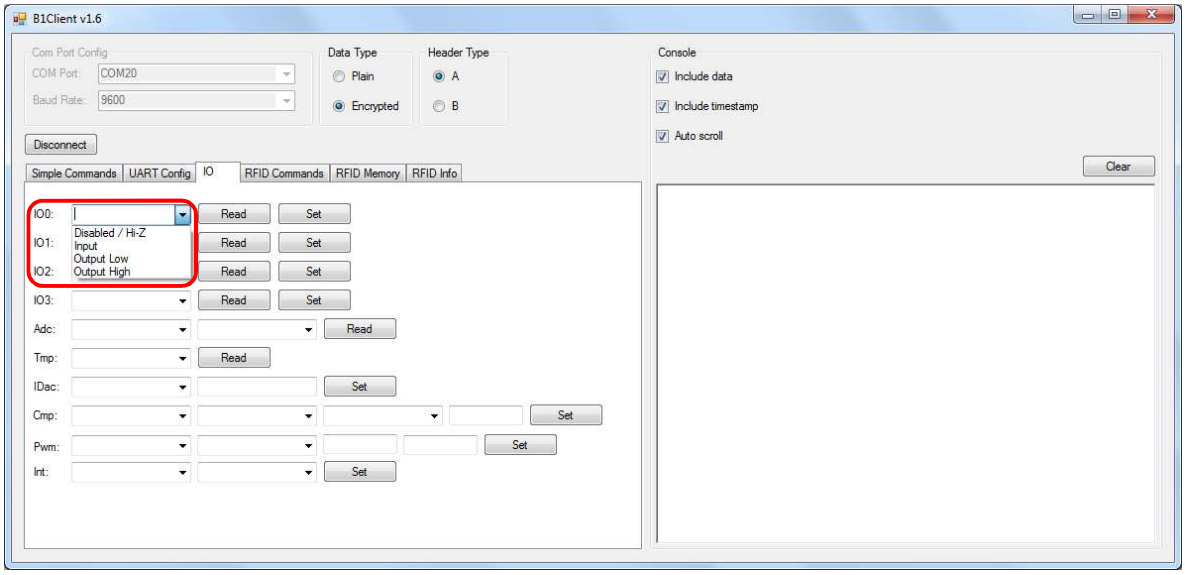
Figure 4. The new AES Key after clicking the 'Read' button.

AES IV – The Read AES Init Vector command takes no arguments. When the module receives a valid command, it replies with an ACK and the 16 byte AES Init Vector that is to be used for the encryption of data.

Set AES Init Vector takes as an argument an Initialization Vector, which is a 16-byte long vector used to initialize the encryption. When the module receives this command, it changes the initialization vector and replies with an ACK response and no parameters. The reply is encrypted using the new initialization vector if Encrypted Data is selected. The procedure of reading and setting the new AES IV is the same as described above in the AES Key section.

3.2.3 IO

IO0 – The user can read the actual state of the GPIO0 of the module or set one of four defined states: Disabled/Hi-Z, Input, Output Low, Output High. Max. output current is 6mA. If the User sends e.g. the Read IO0 state command a module should answer with ACK{00} or ACK{01} which means low/Hi-Z and high state respectively. Changing the state to the Disabled/Hi-Z does not disable the interrupts on this GPIO.

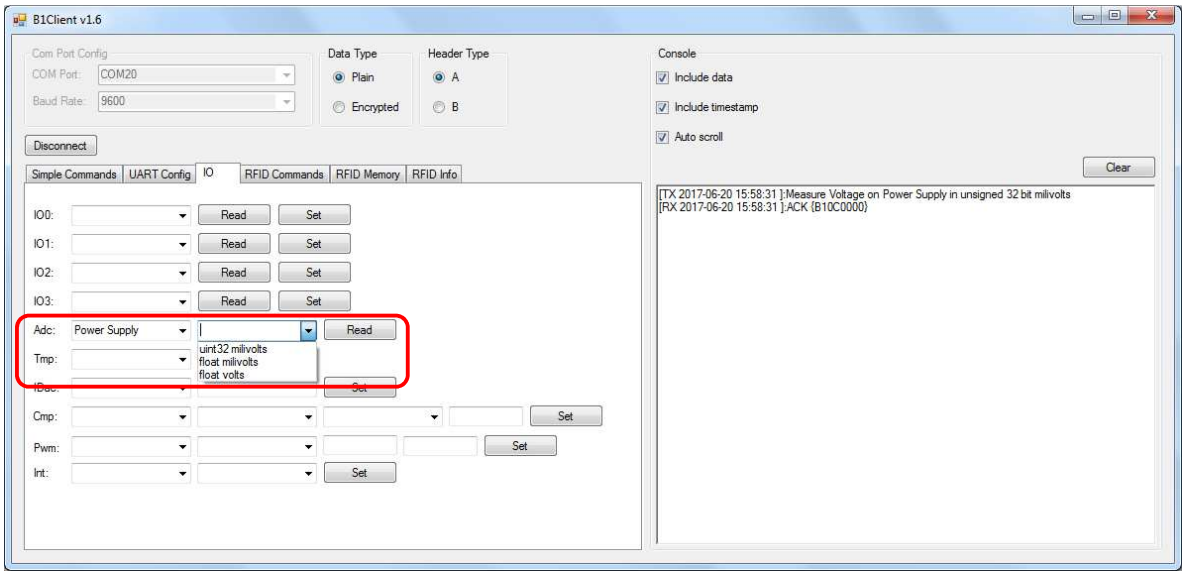


IO1 – The user can read the actual state of the GPIO1 of the module or set one of four defined states: Disabled/Hi-Z, Input, Output Low, Output High. Max. output current is 6mA.

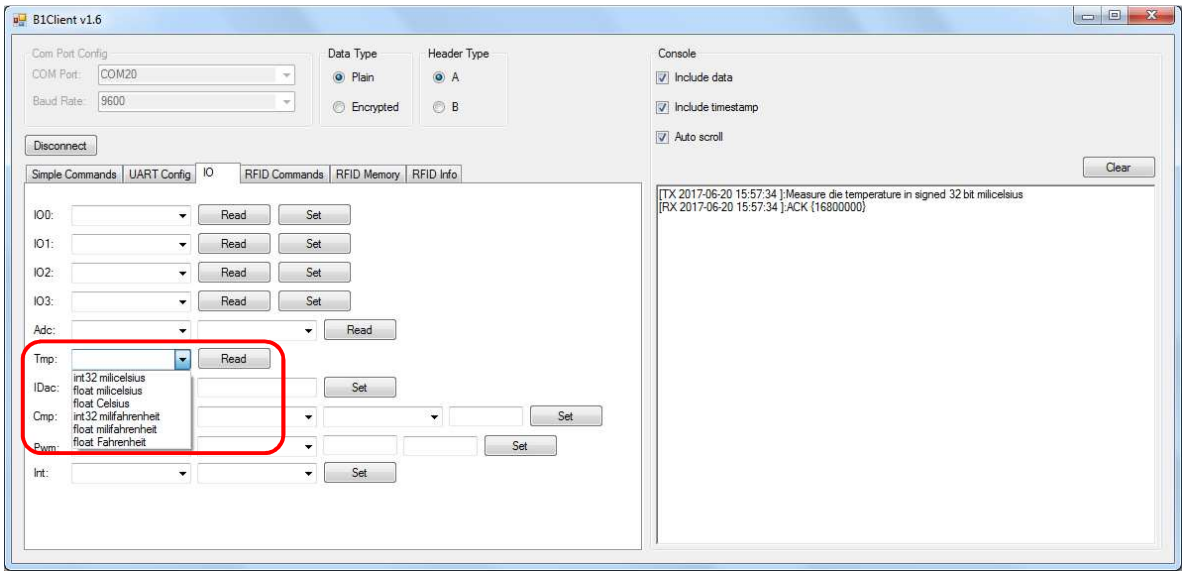
IO2 – The user can read the actual state of the GPIO2 of the module or set one of four defined states: Disabled/Hi-Z, Input, Output Low, Output High. Max. output current is 6mA.

IO3 – The user can read the actual state of the GPIO3 of the module or set one of four defined states: Disabled/Hi-Z, Input, Output Low, Output High. Max. output current is 6mA.

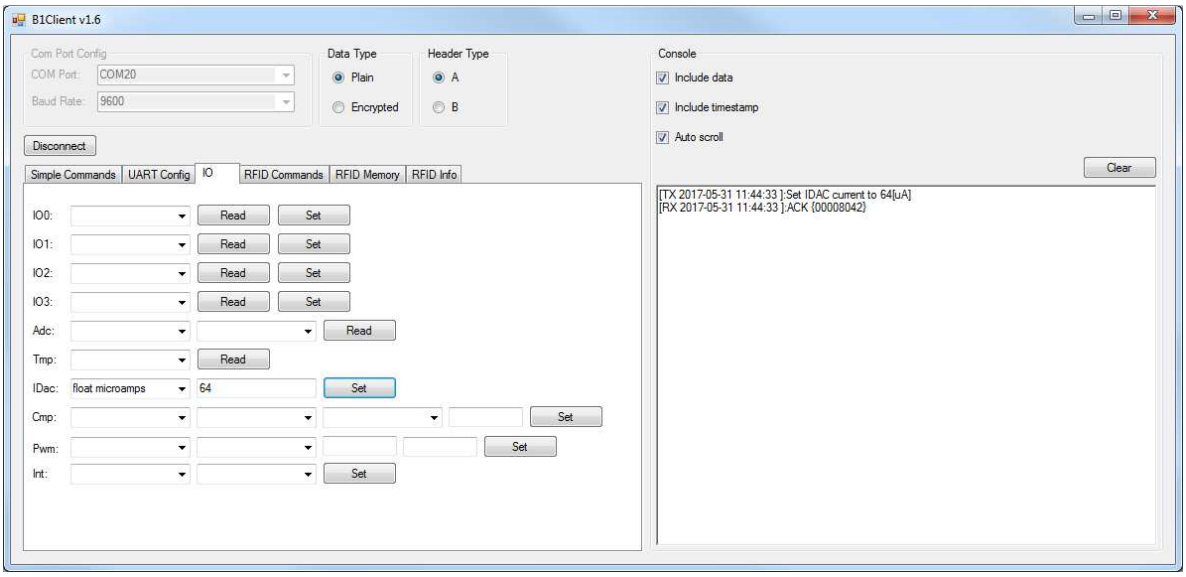
Adc – In the first drop-down list the user can select a source of voltage to convert to the digital value. There are two possibilities: the Power Supply and the ADC pin in the B1 module. In the second drop-down list there is an option to choose the binary representation type (`uint_32` milivolts, `float` milivolts and `float` volts).



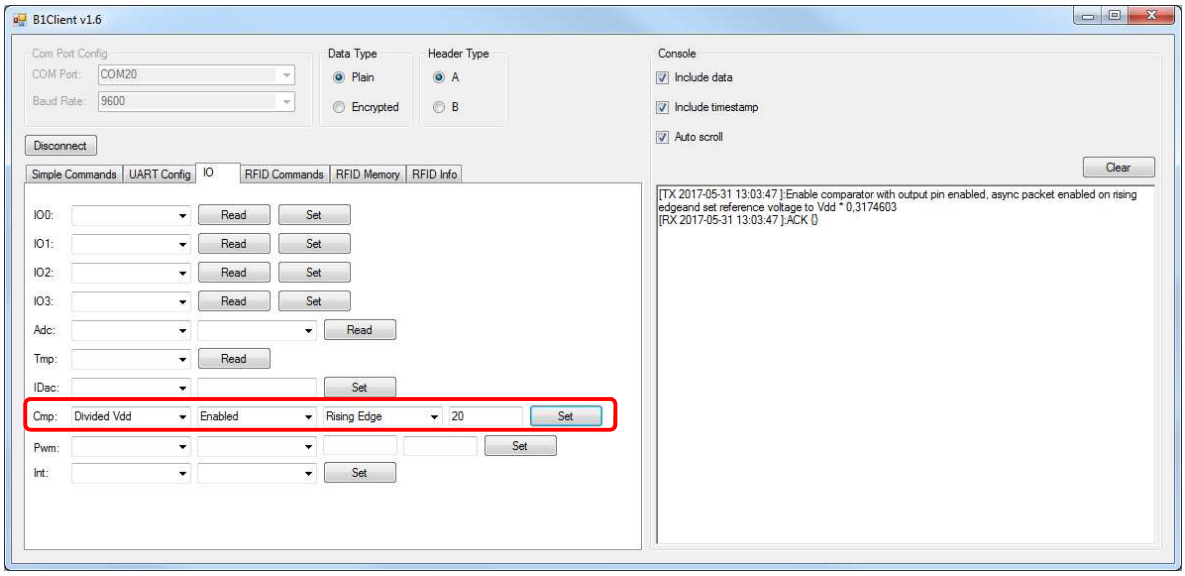
Tmp – In the drop-down list the user can select the binary representation type from `int32` and `float` as well as type of the temperature scale. When the module receives this command, it measures the die temperature and replies with an ACK response and 4 bytes of data which represents the temperature in the requested format and units. The die temperature is the microcontroller's chip temperature.



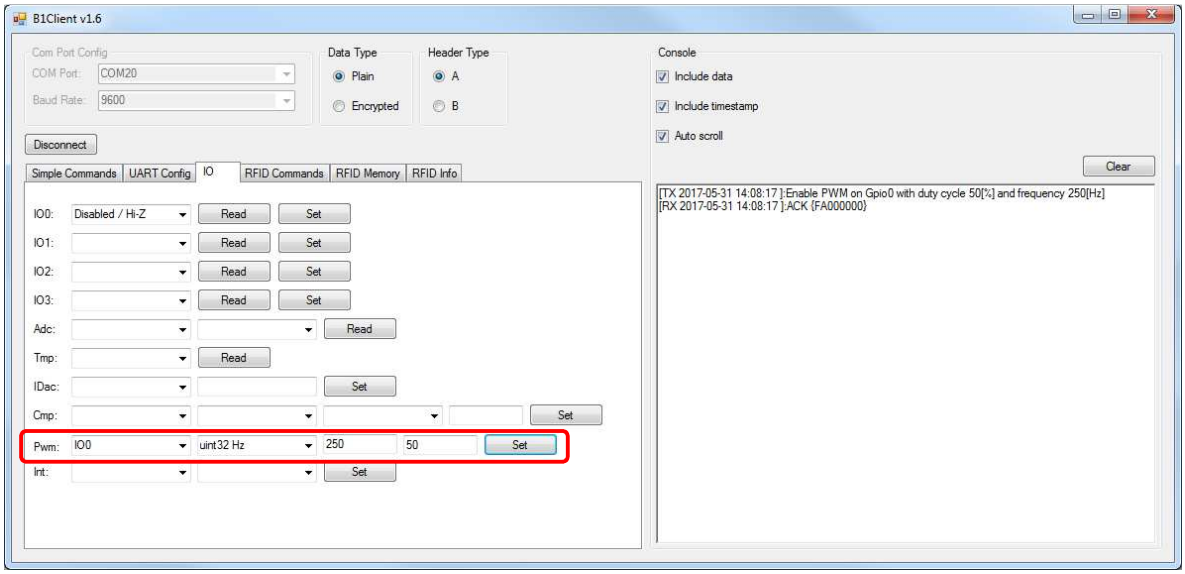
IDac – In the drop-down list the user can select the binary representation type from `int32` and `float` values. In the text box control the user can type a value (in decimal representation) to be written into the DAC. The maximum acceptable value is 64000 nA or 64 µA. The module responds with the actual value that is set up.



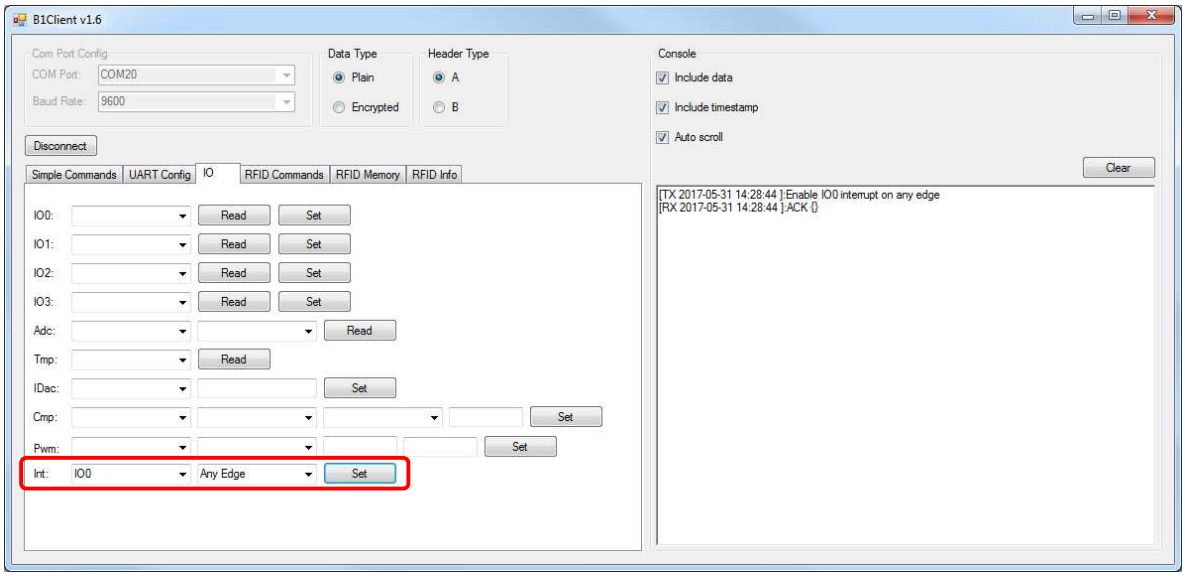
Cmp – In the first drop-down list the user can select the Negative Input Connection between 3 possibilities: 1.25V, 2.5V or Divided Power Supply. The Power Supply is divided as per the following formula: $V_{ref} = V_{dd}\frac{Dividend}{63}$. In the second drop-down list the user can define the Output Pin Configuration. There are three possibilities: enabled, disabled, enabled and negated. The comparator can generate system interrupts and send asynchronous UART packets if there is a change of its output state. In the third drop-down list the user can select the UART Asynchronous Packet Edge Sensitivity from: Falling Edge, Rising Edge, Any Edge or Disabled. At the last text box control the user can write the Power Supply Divider Value. Available range is between 0 and 63.



Pwm – In the first drop-down list the User can select the IO Number, upon which the PWM signal will be generated. In the second drop-down list, the User can select the Value Format of frequency in Hz or period in seconds. The following formats are available: `uint32` and `float`. In the third text box control the User can set the frequency or period. Available values for frequency are from 0.313 Hz to 207.9 kHz and period from 4.81 µs to 3.19 s. In the last text box control the User can set the Duty Cycle from 1 to 99%.

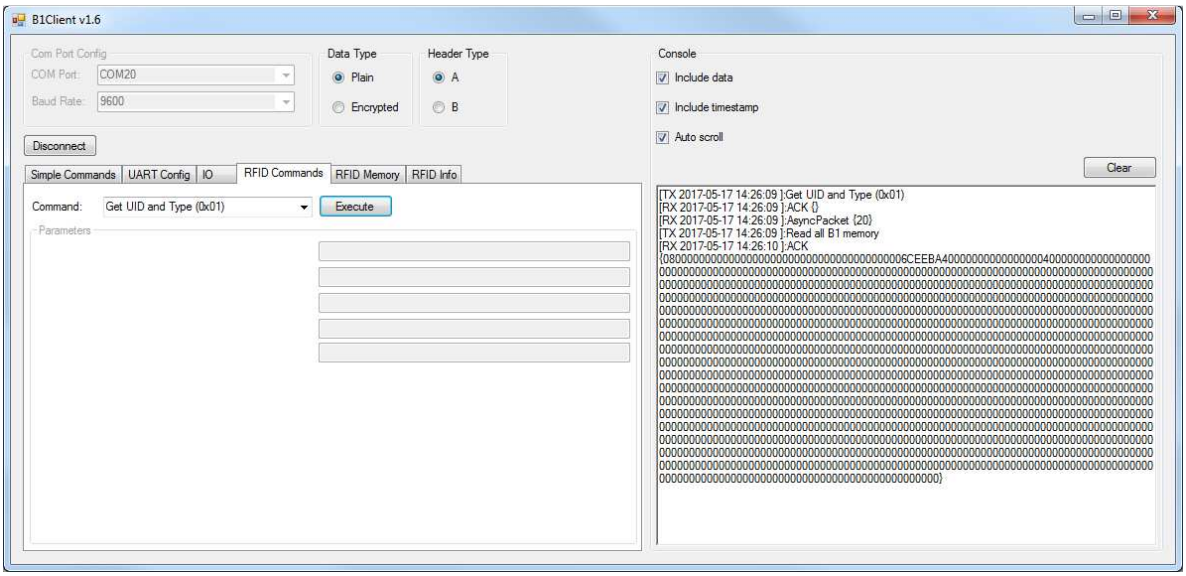


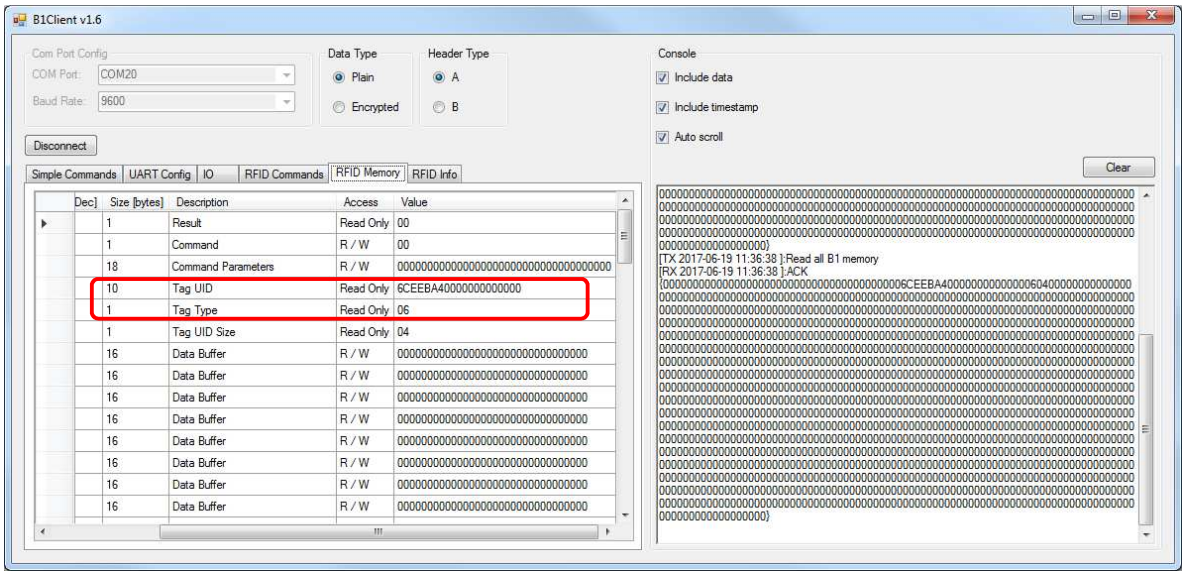
Int – In the first drop-down list the user can select the IO Number of the GPIO to which the interrupt setting is to be applied. At the second drop-down list the User can select one of four available configurations: Falling Edge, Rising Edge, Any Edge or Disabled (the last parameter will disable all interrupts). The module will be generate an interrupt when any edge will appear in IO0 (see the image below).



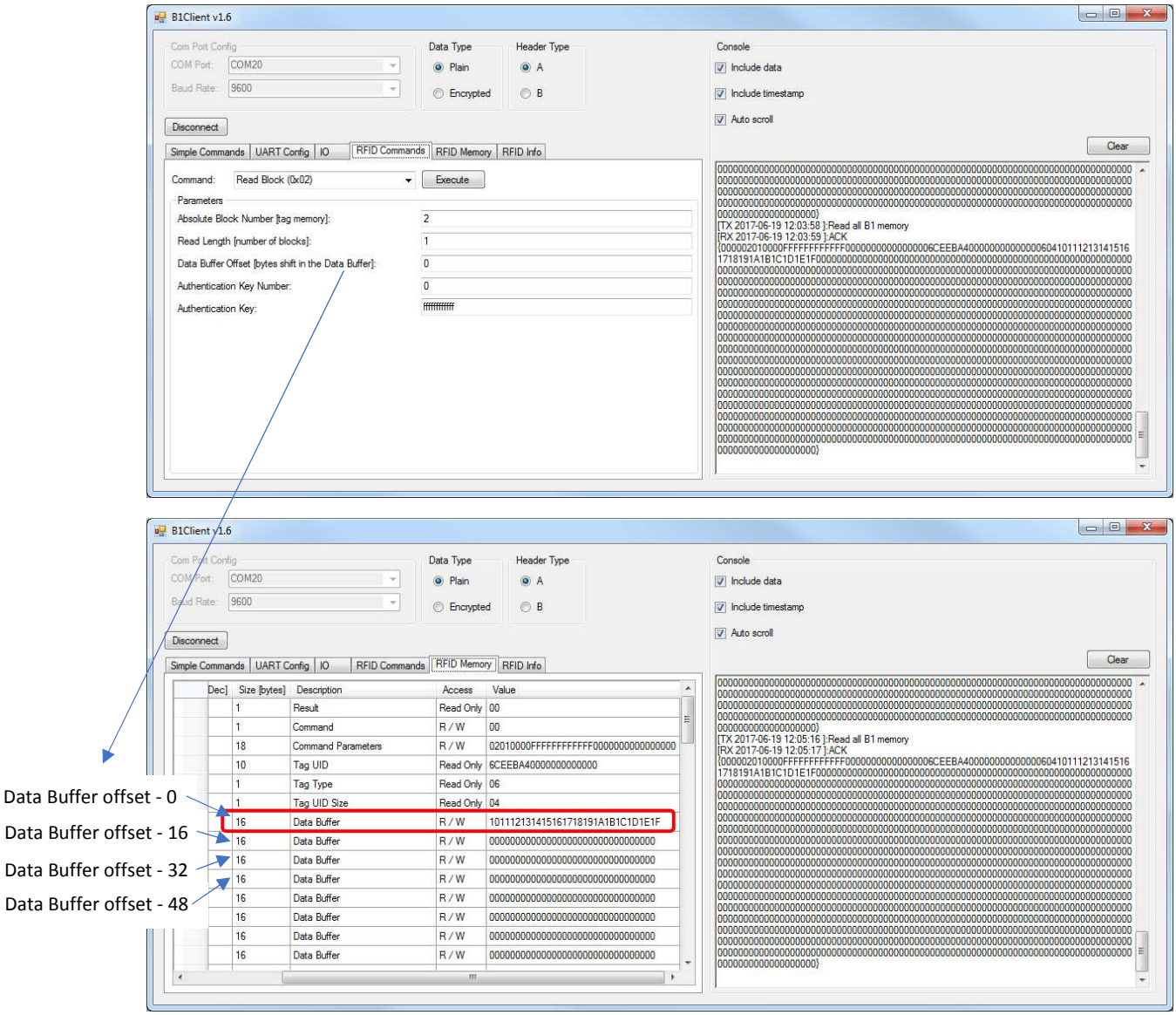
3.2.4 RFID Commands

Get UID and Type (0x01) – After receiving this command, the B1 module checks for any tag presence in the field. If there is no tag in the field, it returns 'No tag' value. If there is a tag in the field, it reads its UID and Type and writes it to Tag UID and Type registers. This command must always be executed first before any other command to turn on and initialize the tag. Additionally, this command must be executed after any error when doing any operations on a tag to reset the tag and to enable it to be able to respond to further command requests. This command has no parameters. In the RFID Info tab there is more information about the read tag.





Read Block (0x02) – (only Mifare Classic) The Read Block command takes as arguments block number of the first block to read (Absolute Block Number - It is counted from 0. It is not a byte address. For example, the block 0x00 in Mifare memory at sector 0x01 would have an Absolute Block Number of 0x04), the number of blocks to read (Read Length), the byte offset in the Data Buffer (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. After a successful authentication, the first block is read and copied to the Data Buffer. The user can see the read sector in the RFID Memory tab. The whole Block is in the Data Buffer at offset '0'.



Write Block (0x03) – (only Mifare Classic) The Write Block command takes as arguments block number of the first block to write (Absolute Block Number), the number of blocks to write (Write Length), the byte offset in the Data Buffer (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. The desired data byte array which the user wants to store in the tag memory must be in the first place stored in the Data Buffer starting from the Data Buffer Offset (byte index).

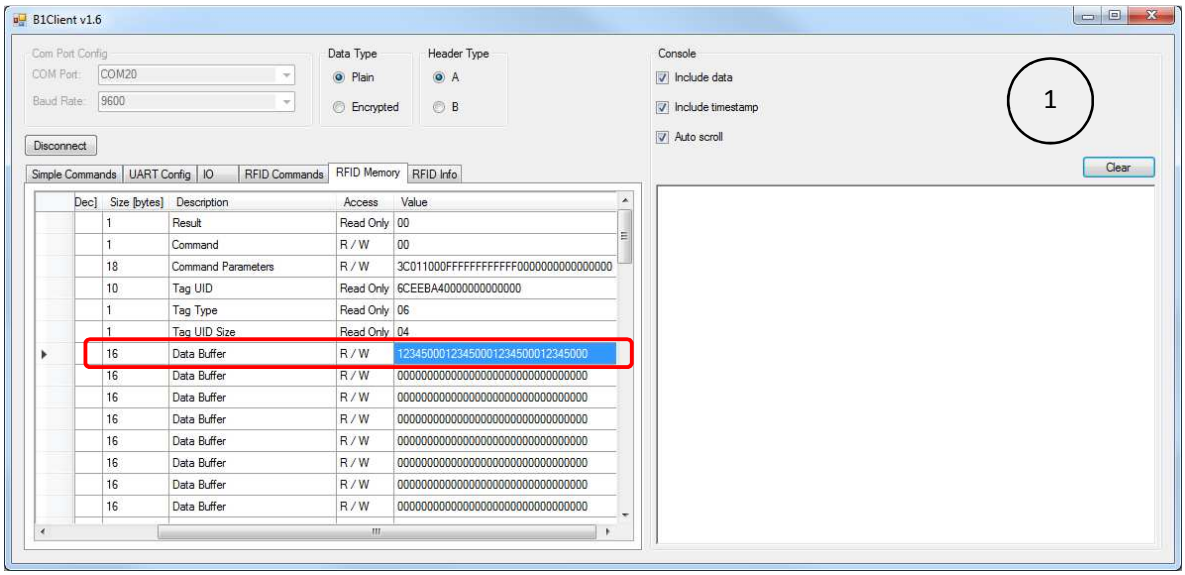


Figure 5. Writing data to the Data Buffer.

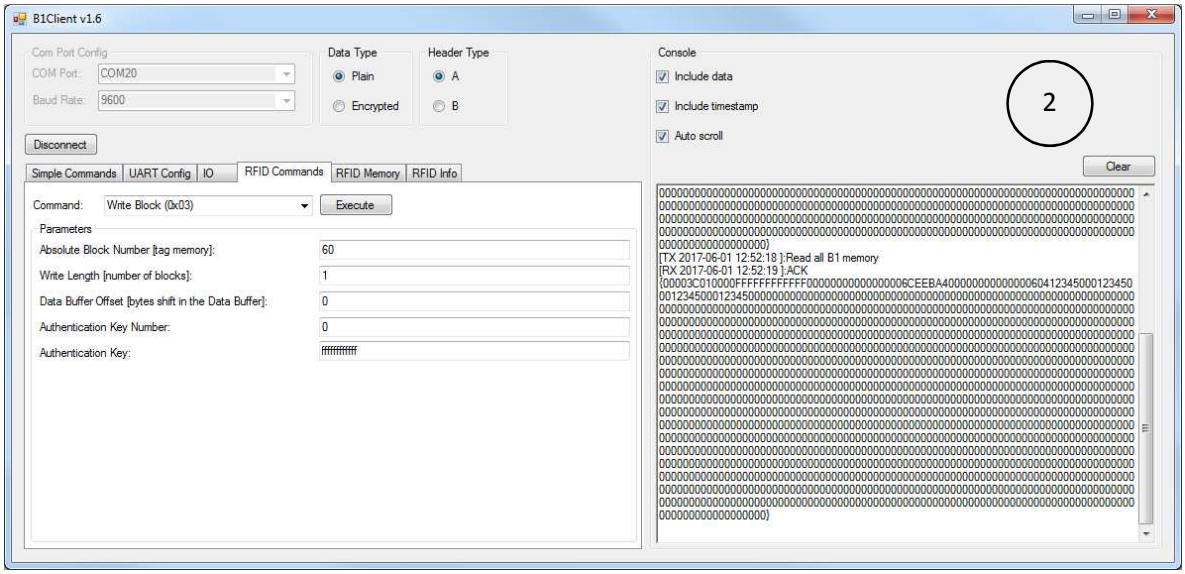
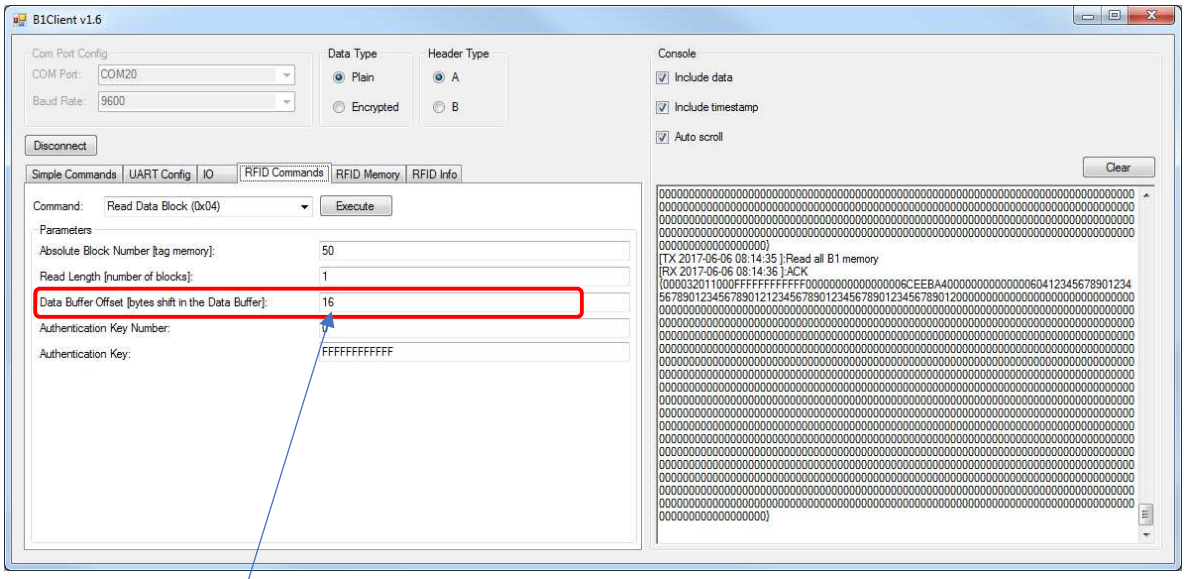
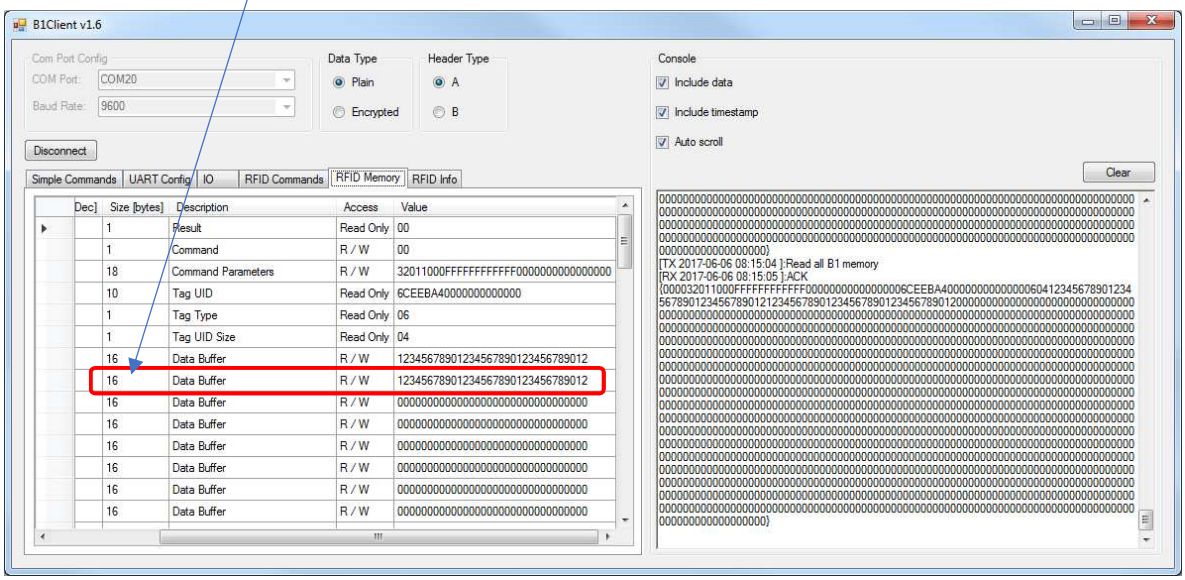


Figure 6. Writing parameters and executing the command.

Read Data Block (0x04) – (only Mifare Classic) The Read Data Block command takes as argument block number of the first block to read (Absolute Block Number), the number of blocks to read (Read Length), the byte offset in the Data Buffer (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. The difference between the Read Data Block command and the Read Block command is that the first one omits blocks containing authentication keys and lock bits. It only reads the data blocks from a tag memory. If the next blocks fall into another tag memory sector, this sector is authenticated using the same key.





Write Data Block (0x04) – (only Mifare Classic) The Write Data Block command takes as arguments block number of the first block to write (Block Address), the number of blocks to write (Write Length), the byte offset in the Data Buffer (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. The desired data byte array which the user wants to store in the tag memory must be in the first place stored in the Data Buffer starting from the Data Buffer Offset (byte index). The difference between the Write Data Block command and the Write Block command is that the first one omits blocks containing authentication keys and lock bits. It only writes to the data blocks in tag memory. If subsequent blocks fall into another tag memory sector, this sector is authenticated using the same key.

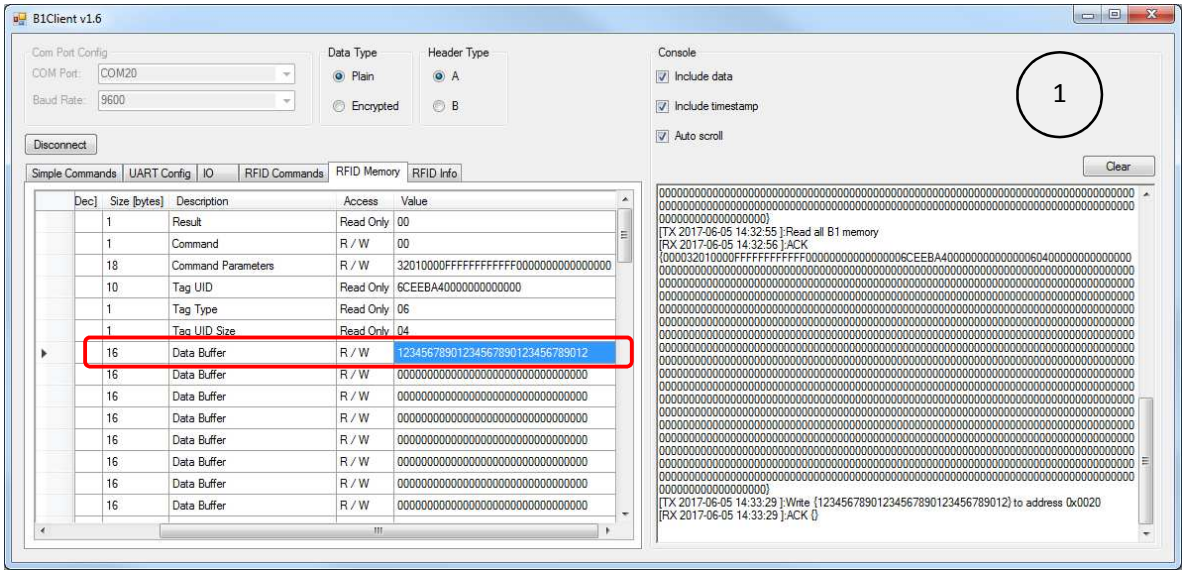


Figure 7. Writing data to the Data Buffer.

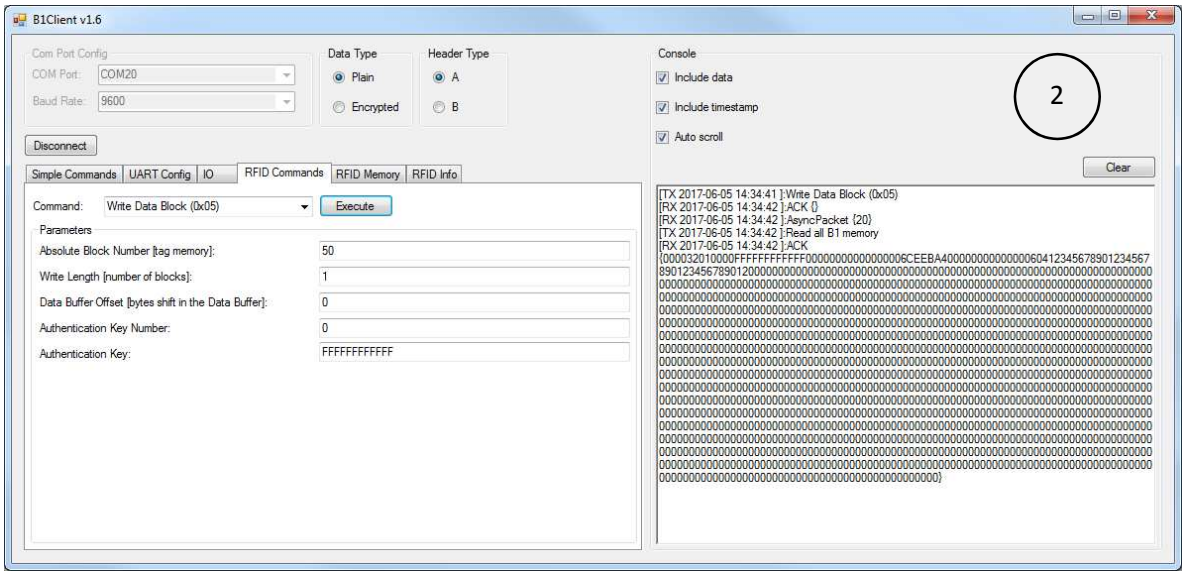
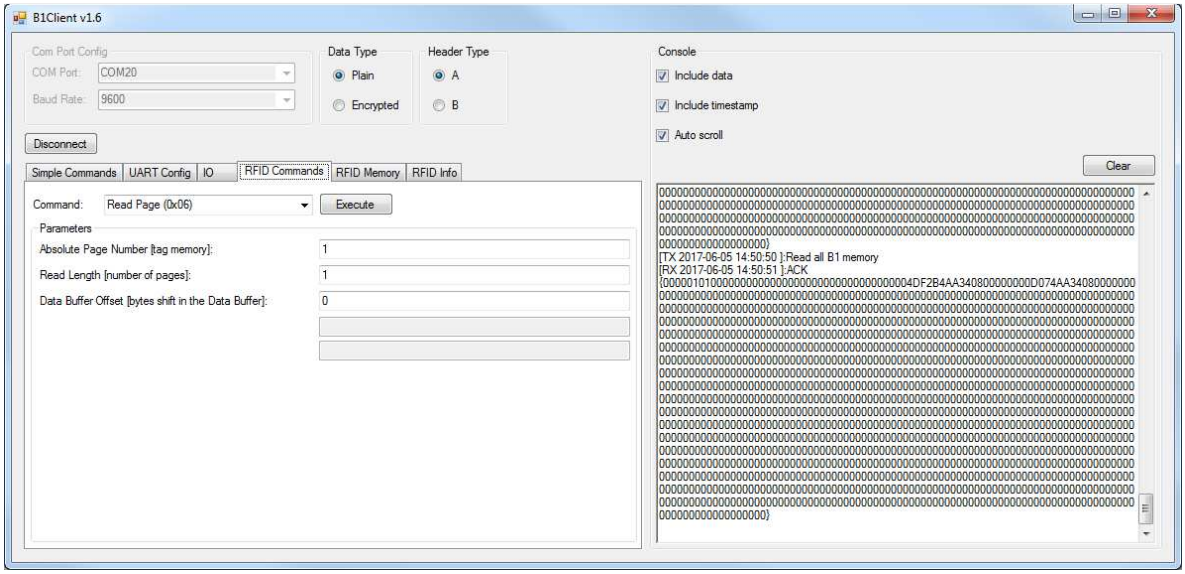
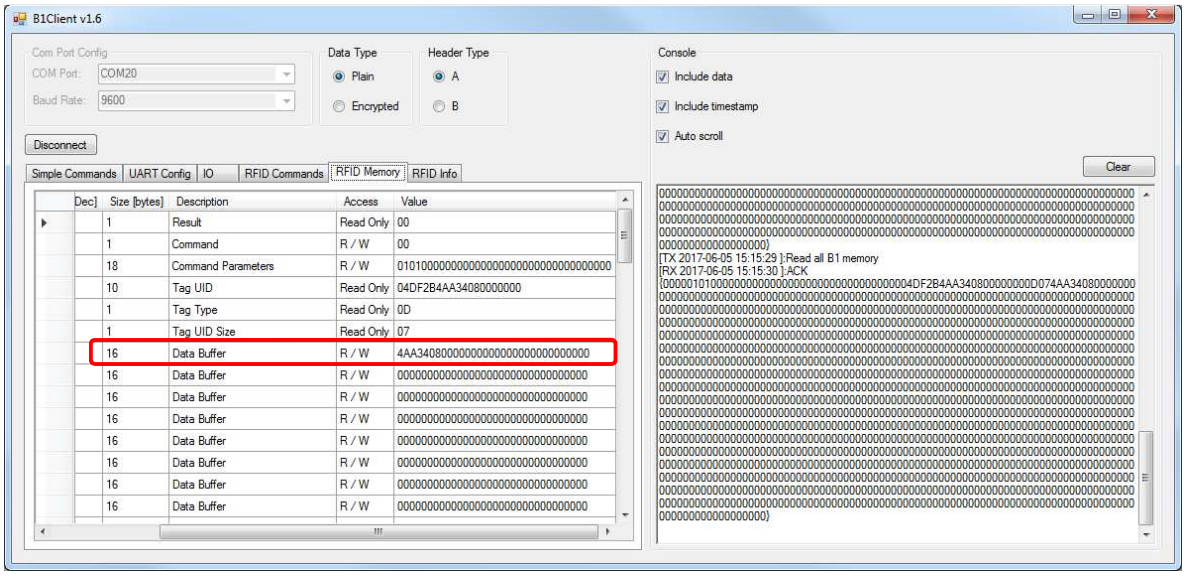


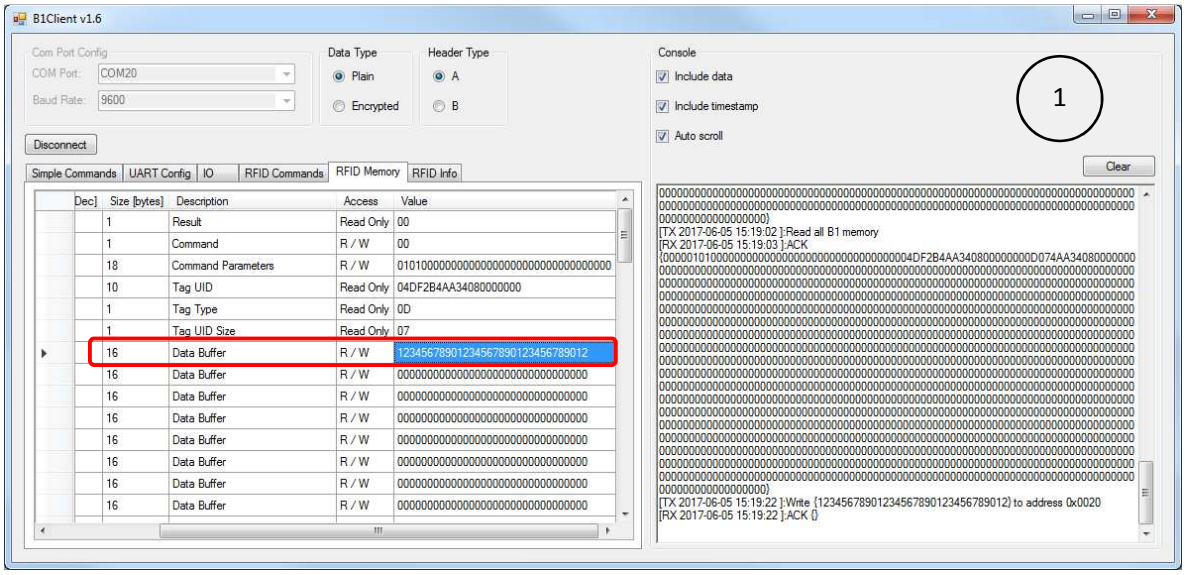
Figure 8. Writing parameters and executing the command.

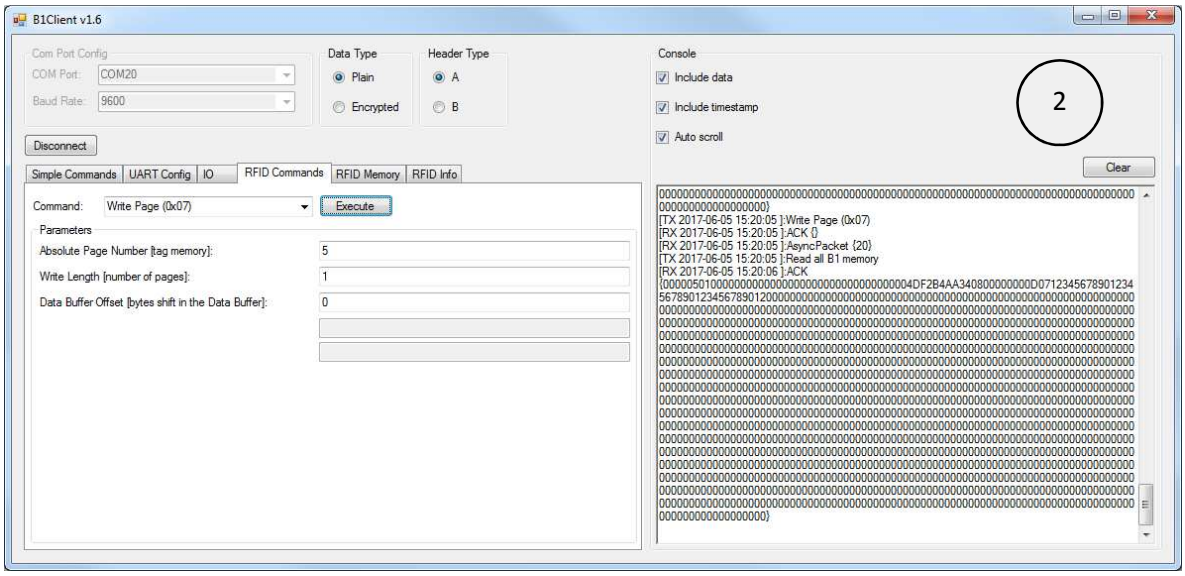
Read Page (0x06) – (only Mifare Ultralight, Mifare Ultralight EV1 and NTAG) The Read Page command takes as arguments page number of the first page to read (Absolute Page Number), the number of pages to read (Read Length) and the byte offset in the Data Buffer (Data Buffer Offset). If any of the pages to be read are password protected a Password Authentication is necessary before doing a read operation.





Write Page (0x07) – (only Mifare Ultralight, Mifare Ultralight EV1 and NTAG) The Write Page command takes as arguments page number of the first page to be written (Absolute Page Number), the number of pages to be written (Write Length), and the byte offset in the Data Buffer (Data Buffer Offset). If any of the pages to be written are password protected, then a Password Authentication is necessary before doing a write operation. The desired data byte array which the user wants to store in the tag memory must be in the first place stored in the Data Buffer starting from the Data Buffer Offset (byte index).





Encrypt Data (0x08) – The Encrypt Data command takes as command arguments the Encryption Key Number (0x00 or 0x01), the Initialization Vector Number (0x00 or 0x01), the Data Buffer Offset (16-byte blocks) and the Data Length (16-byte blocks – this value can be between 1 and 16). This command encrypts the 'Data Length' number of 16-byte blocks using the AES128 CBC encryption. It operates only within the Data Buffer.

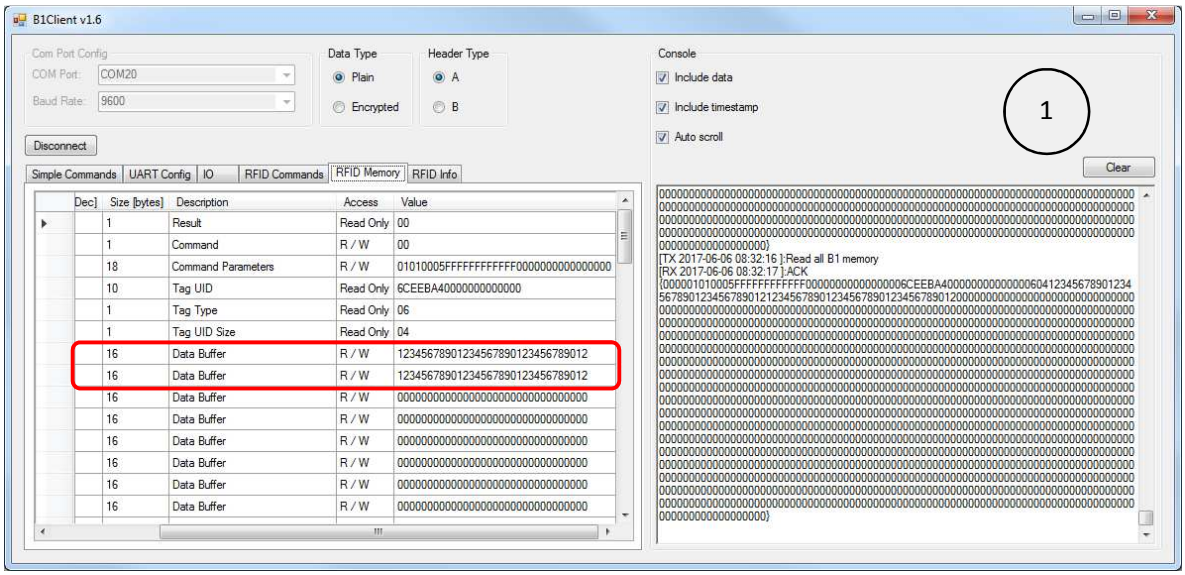


Figure 9. Example data written to the Data Buffer.

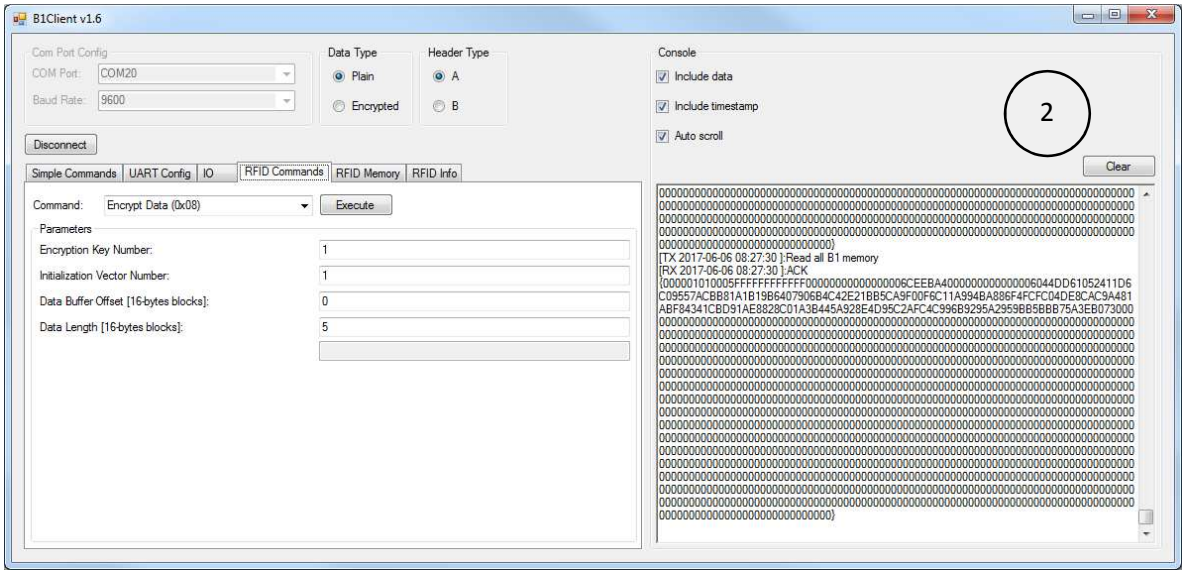


Figure 10. Writing parameters and executing the command.

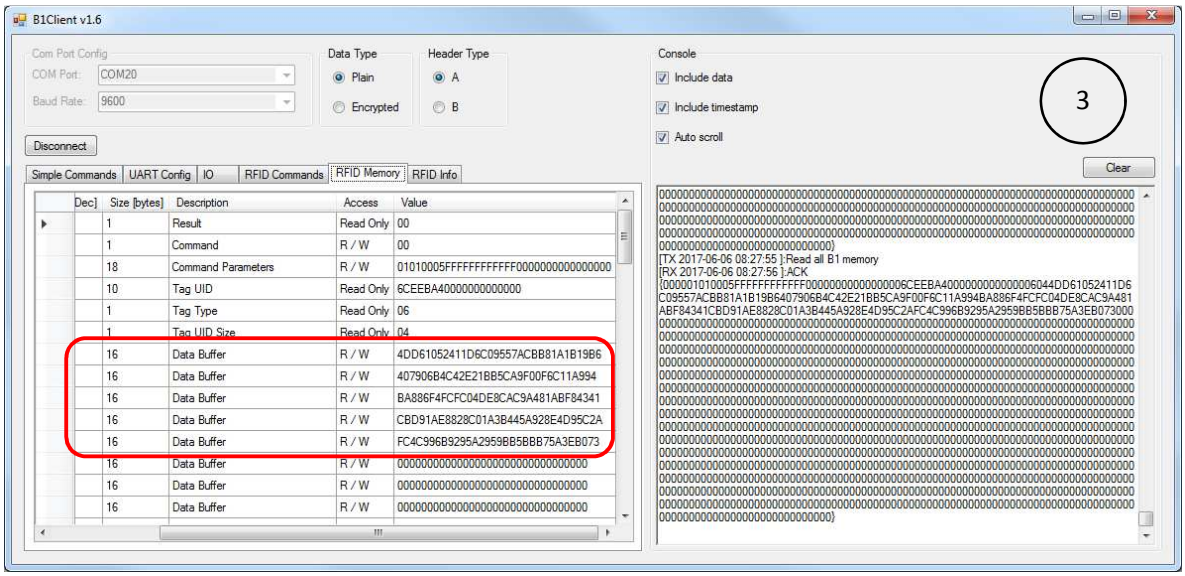


Figure 11. Result of executing the command.

Decrypt Data (0x09) – The Decrypt Data command takes as arguments the Decryption Key Number (0x00 or 0x01), the Initialization Vector Number (0x00 or 0x01), the Data Buffer Offset (16-bytes) and the Data Length (16-blocks – this value can be between 1 and 16). This command decrypts the 'Data Length' number of 16-byte blocks using the AES128 CBC decryption methodology. It operates only within the Data Buffer.

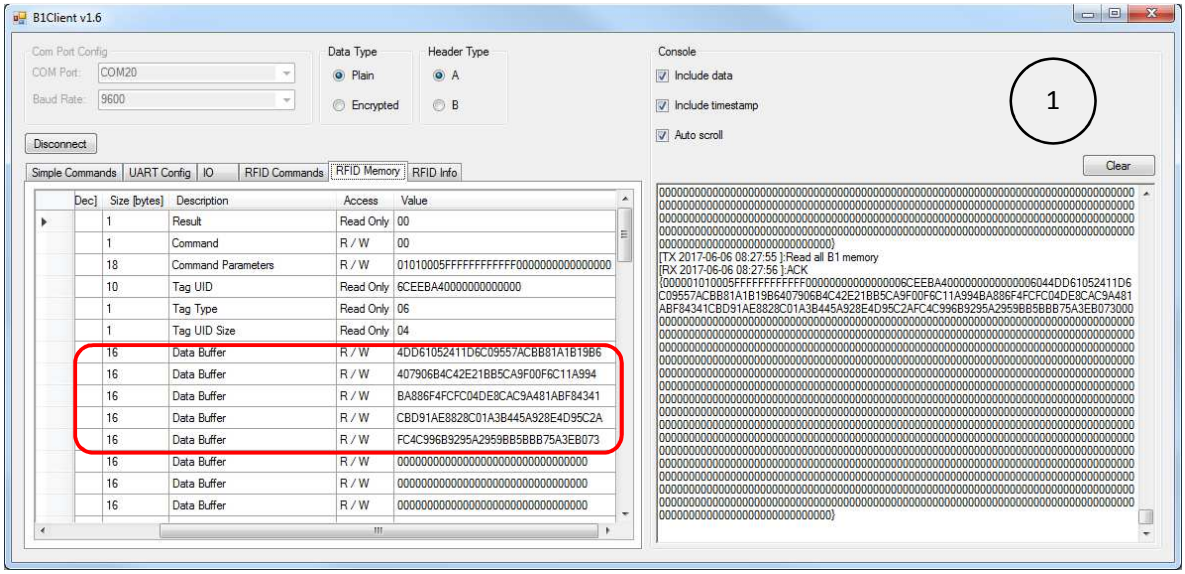


Figure 12. Encrypted data in the Data Buffer.

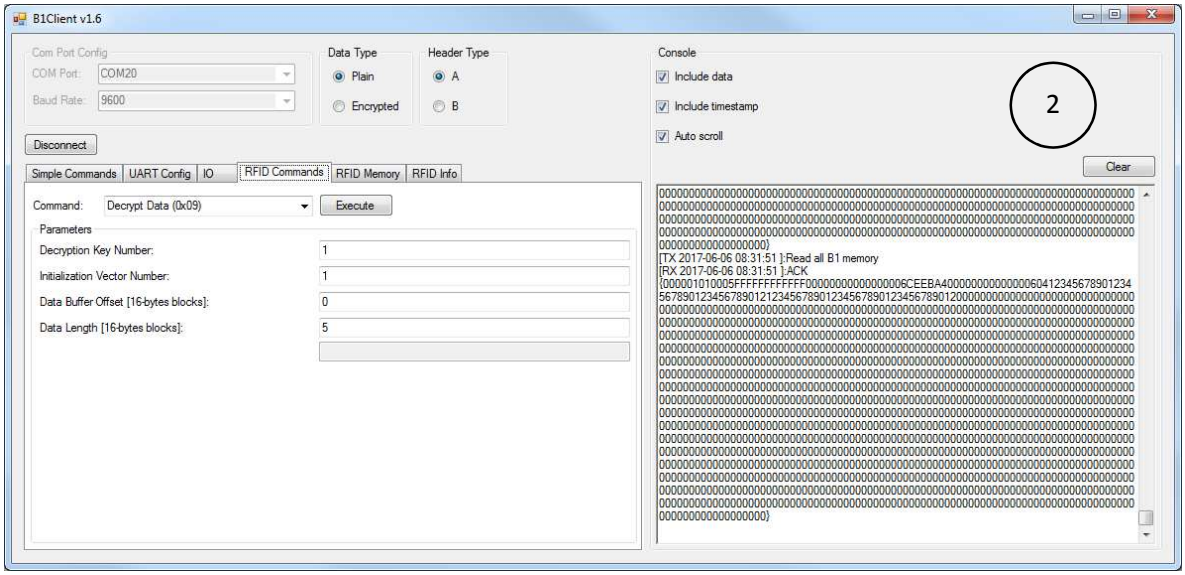


Figure 13. Writing parameters and executing the command.

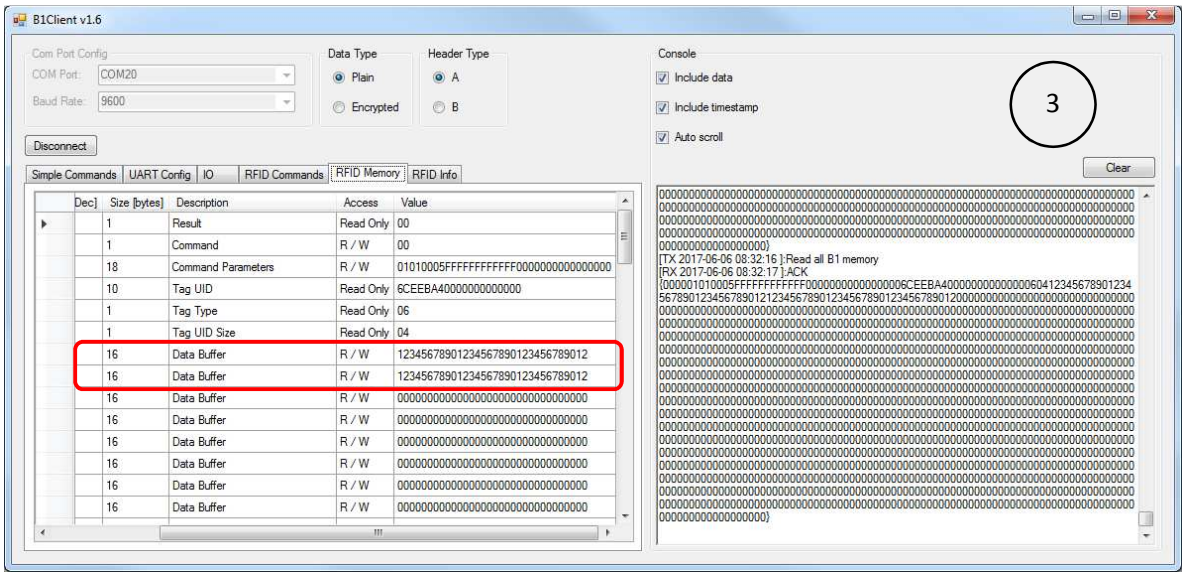
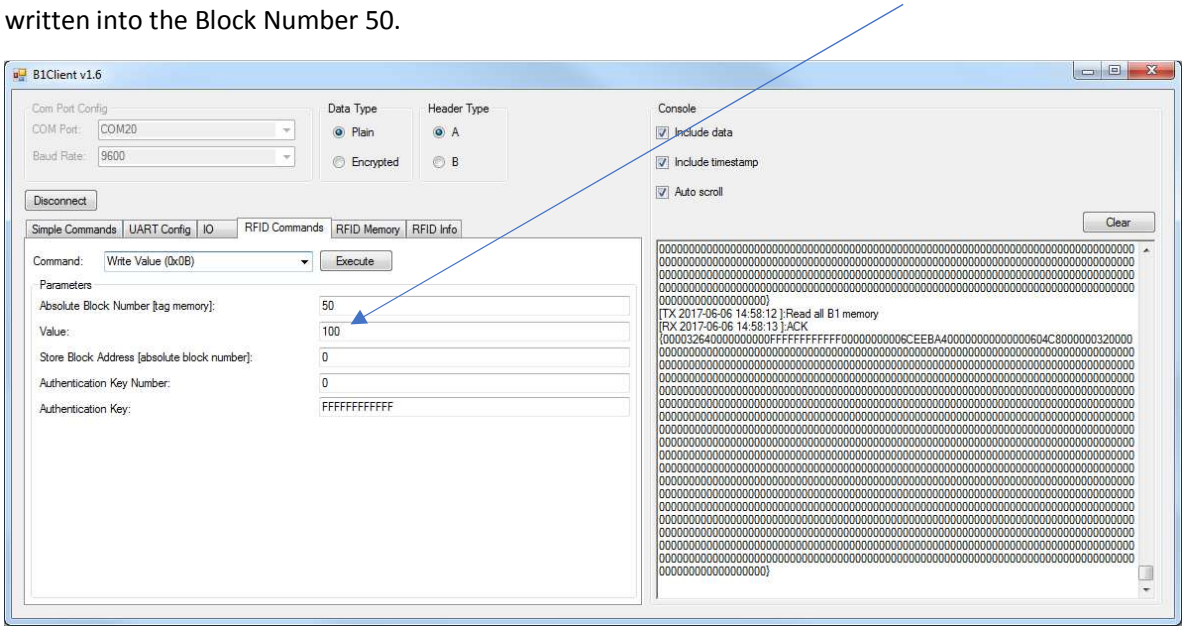
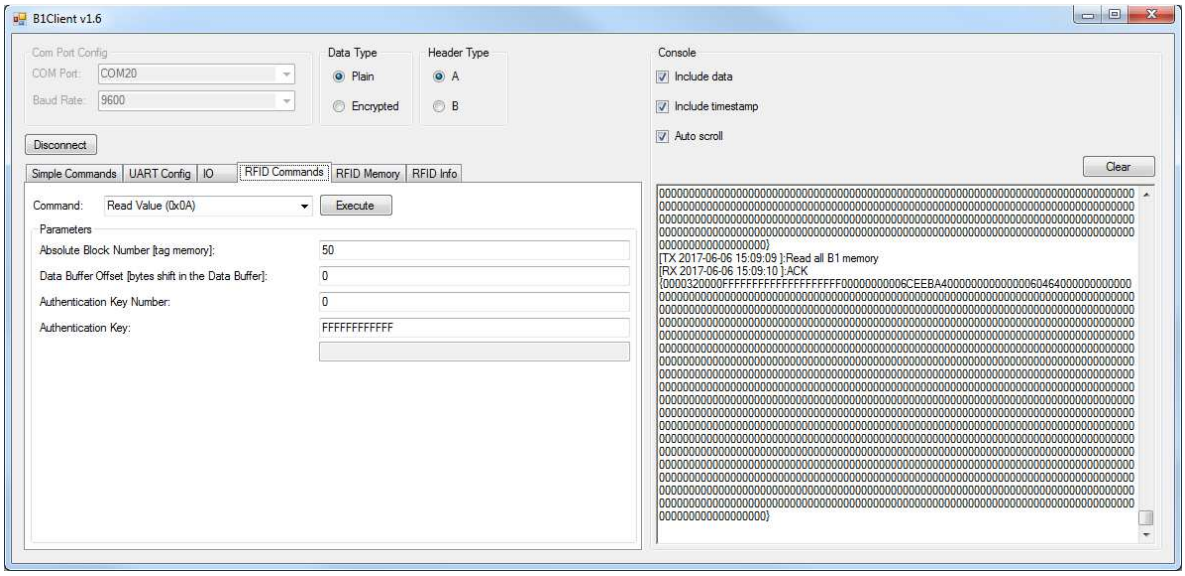


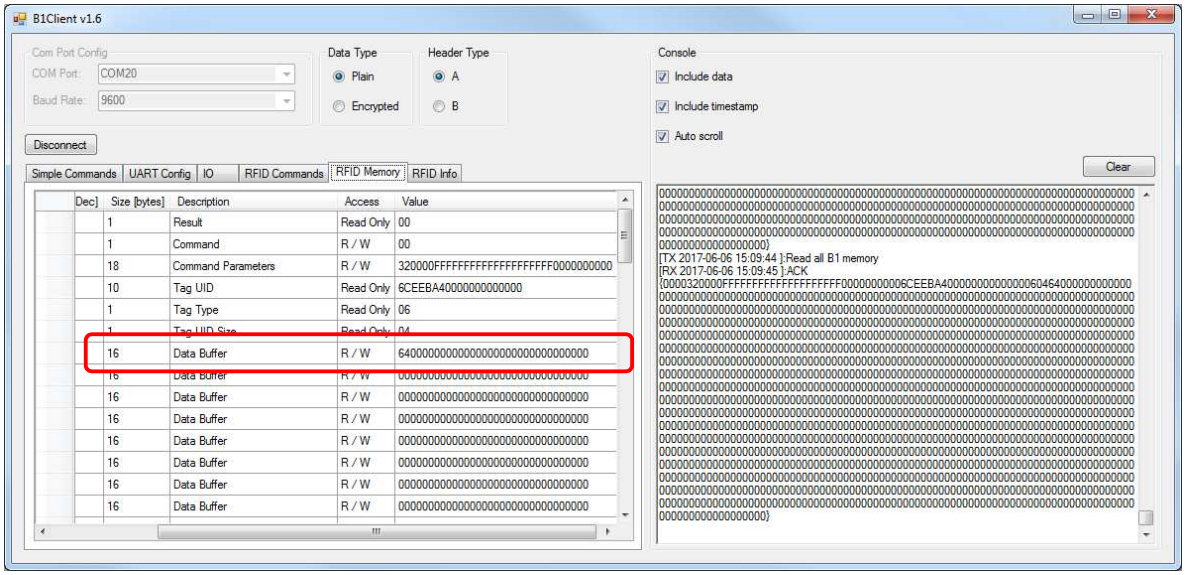
Figure 14. Result of executing the command.

Write Value (0x0B) – (only Mifare Classic) The Write Value command takes as arguments block number of the block to write (Absolute Block Number), the signed 32-bit integer value to be stored (Value), the block address value to store in the memory block (Store Block Address), the Authentication Key Number and (optionally) the Authentication Key. This command has the same functionality as the Write Block command, except that it treats the block as a special value type defined by the Mifare standard. This command has no visible results. The user can only check in the Result Register (the RFID Memory tab) if the Write Value operation was successful. In the following example the value: 100d (0x64h) was written into the Block Number 50.



Read Value (0x0A) – (only Mifare Classic) The Read Value command takes as arguments block number of the first block to read (Absolute Block Number), the offset of the Data Buffer where the value and read address will be stored (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. This command has the same functionality as the Read Block command except it treats the block as a special Value type defined by the Mifare standard. It tries to parse the block content to this 4-byte signed value and to read a byte address stored in the last four bytes of the block. In this example previously written value (100d –> 0x64h) can be read from the Block Number 50.





Increment Value (0x0C) – (only Mifare Classic) The Increment Value command is an operation on a value-type block as defined by the Mifare standard. It takes as arguments block number of the block where the value is stored (Absolute Block Number), the signed 32-bit integer number which will be added to the value (Delta Value – in decimal units), the Authentication Key Number and (optionally) the Authentication Key. The command reads the value from the block to the volatile memory on the tag and increments it by the selected Delta Value. This command has no visible results. The user can only check in the Result Register (the RFID Memory tab) if the Increment Value operation was successful. To store the value in the same or another block (copy to non-volatile memory on the tag), the user must execute the Transfer Value command. In the following example, in the Block Number 50. the initial value equals 0x64h.

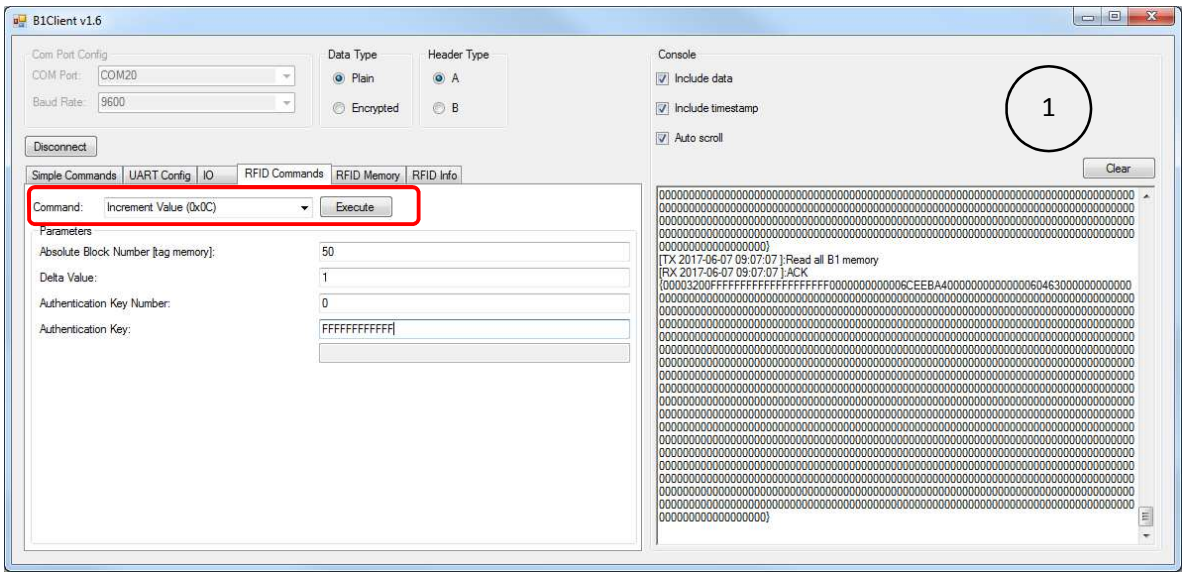


Figure 15. Incrementing the value in the Block Number 50 by 1 and copy it to the volatile memory.

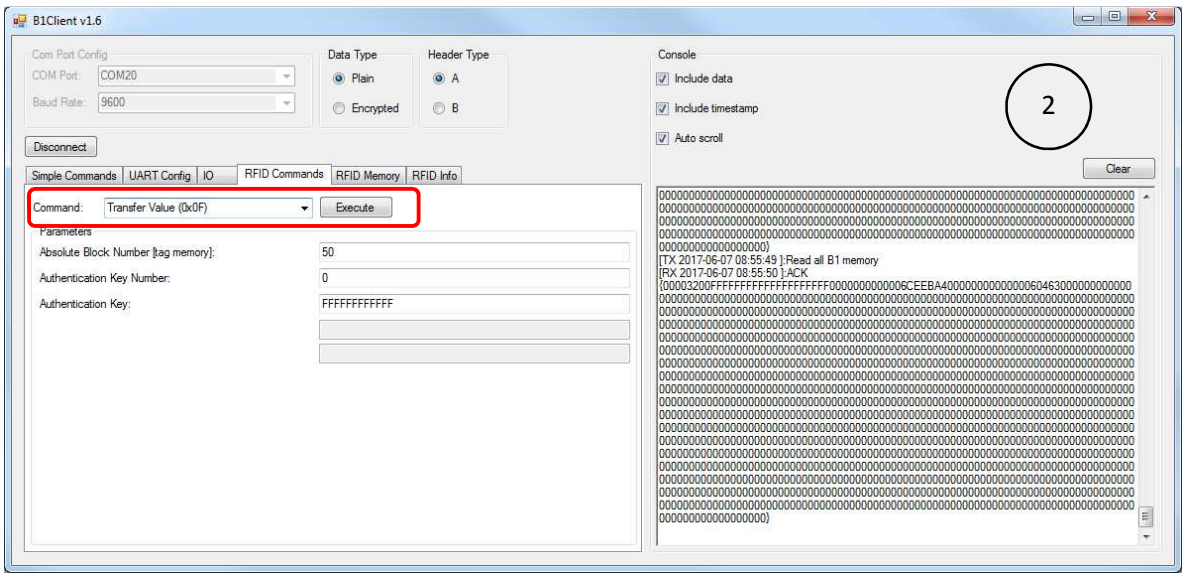


Figure 16. Copy the incremented value to the non-volatile memory and save it to the Block Number 50.

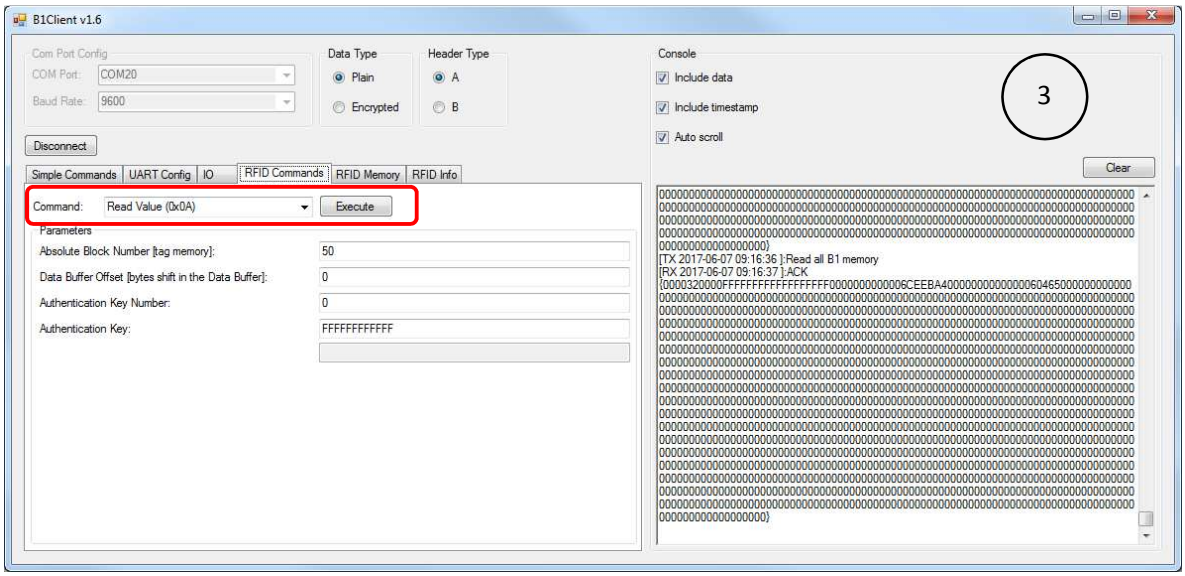


Figure 17. Reading the value in the Block Number 50.

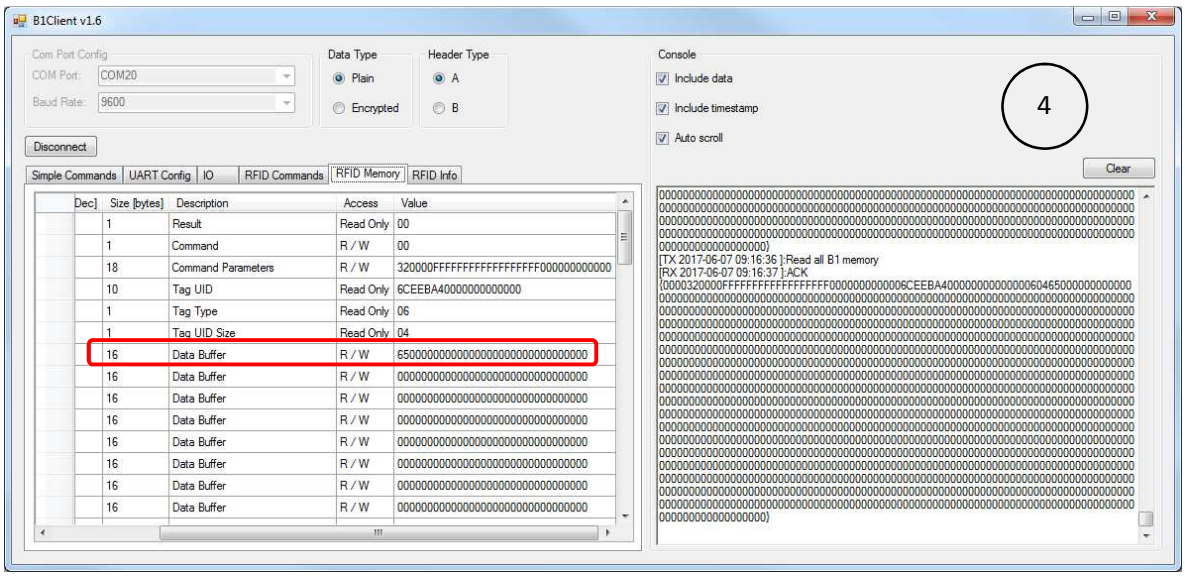


Figure 18. Result of executing all above described commands.

Decrement Value (0x0D) – (only Mifare Classic) The Decrement Value command is an operation on a value-type block, as defined by the Mifare standard. It takes as arguments block number of the block where the value is stored (Absolute Block Number), the signed 32-bit integer number which will be subtracted from the value (Delta Value – in decimal units), the Authentication Key Number and (optionally) the Authentication Key. The command reads the value from the block into the volatile memory on the tag and decrements it by the delta value. This command has no visible results. The user can only check in the Result Register (the RFID Memory tab) if the Decrement Value operation was successful. To store the value in the same or another block (copy to non-volatile memory on the tag), the user must execute the Transfer Value command. In the following example, in the Block Number 50. the initial value equals 0x64h.

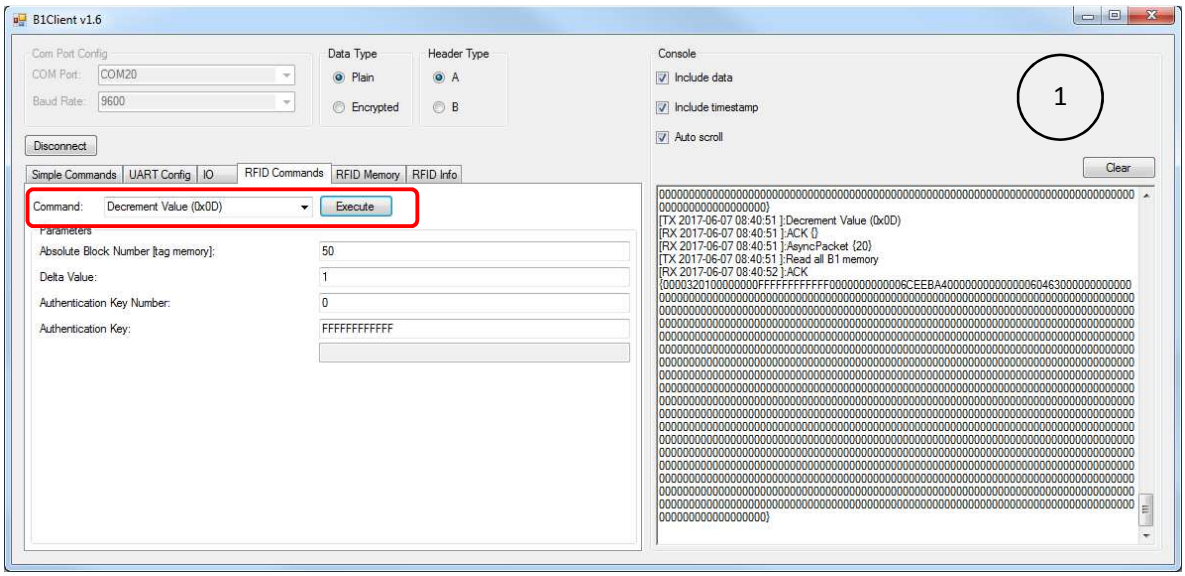


Figure 19. Decrementing the value in the Block Number 50 by 1 and copy it to the volatile memory.

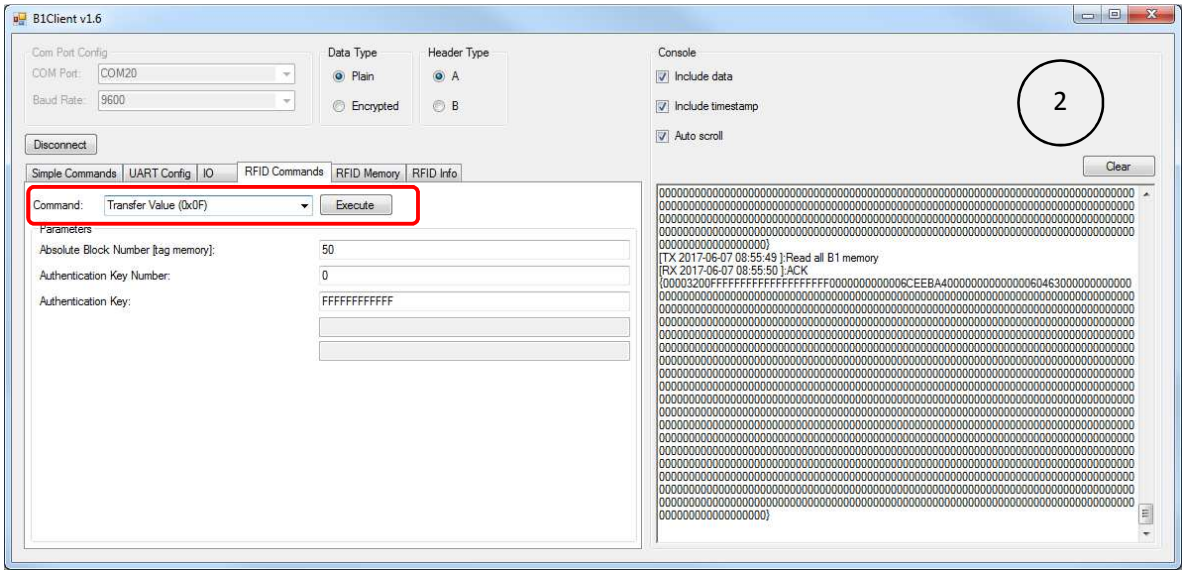


Figure 20. Copy the decremented value to the non-volatile memory and save it to the Block Number 50.

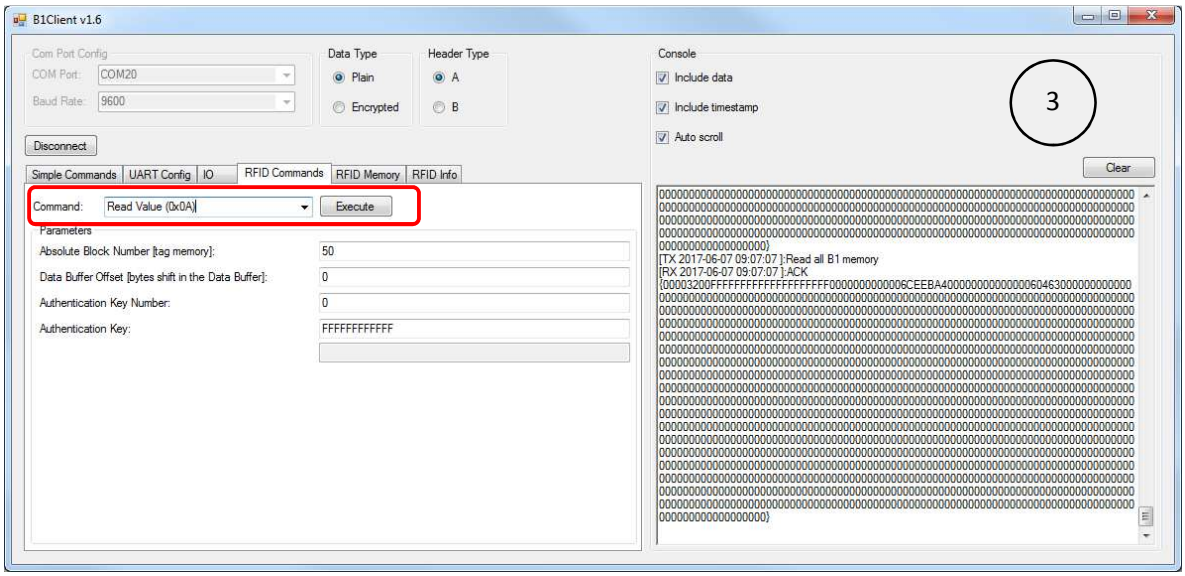


Figure 21. Reading the value in the Block Number 50.

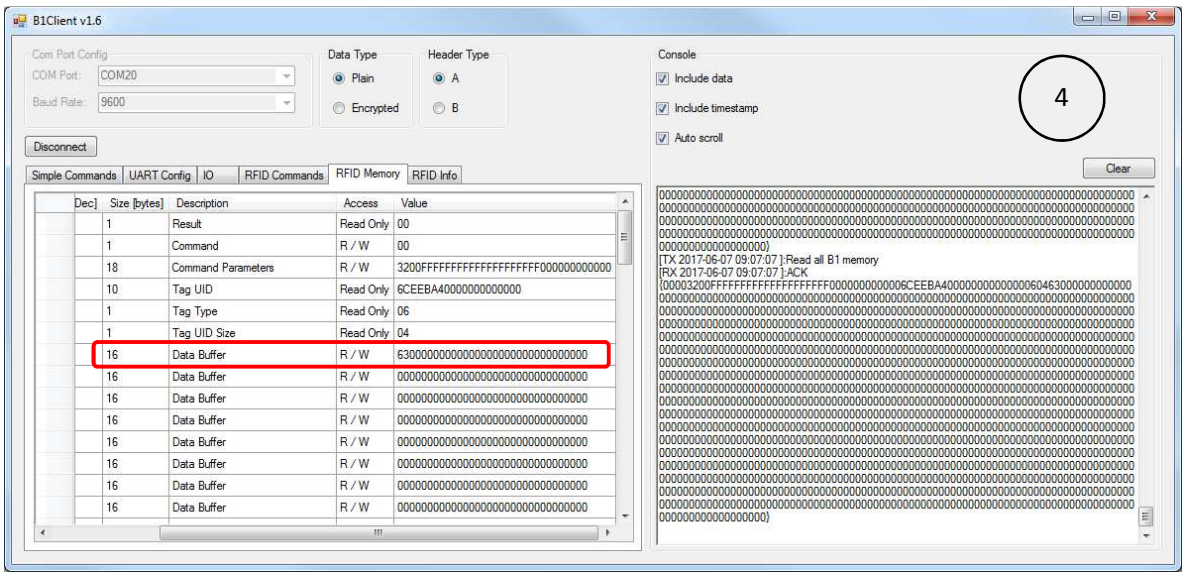
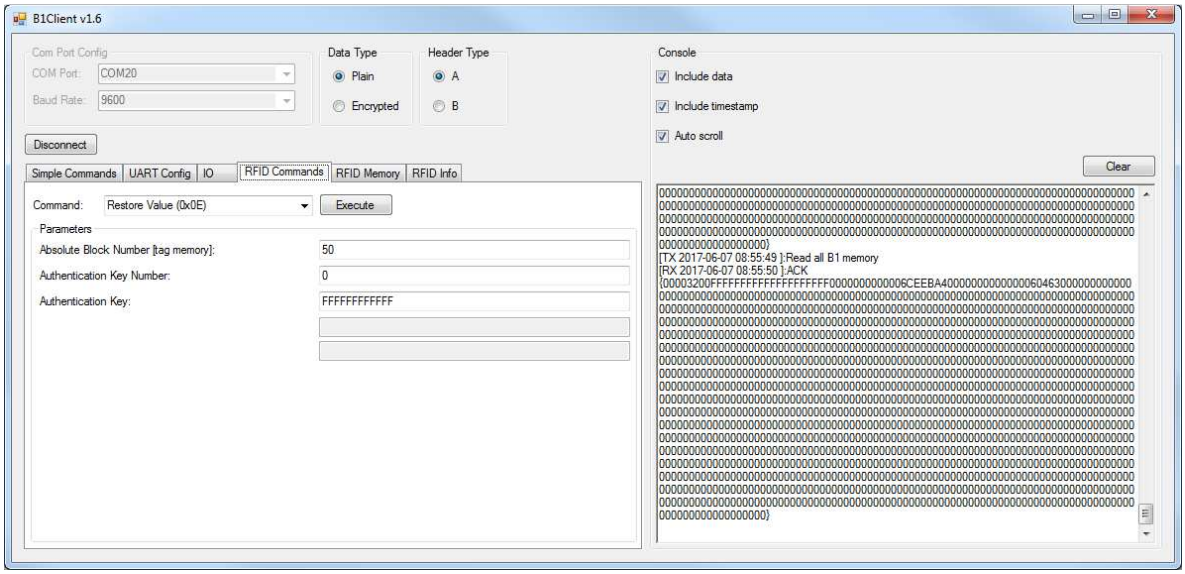
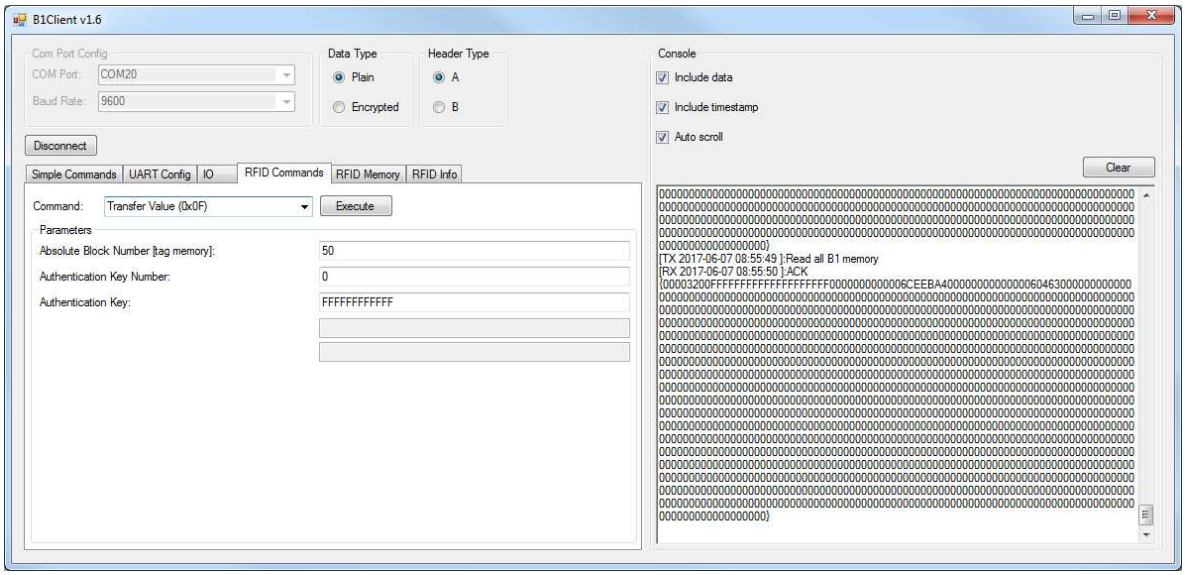


Figure 22. Result of executing all above described commands.

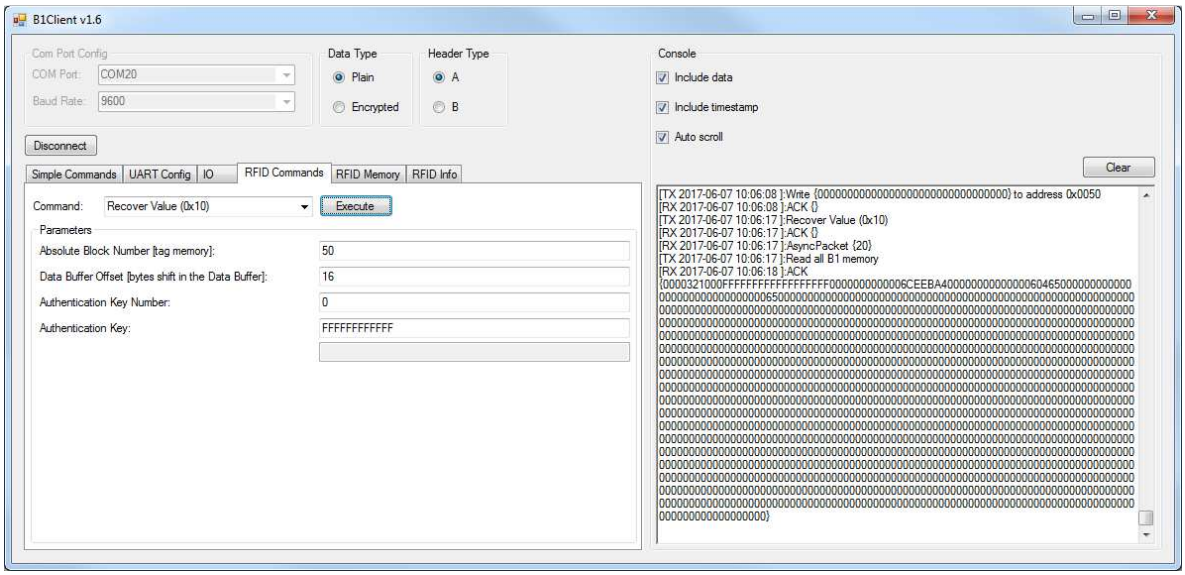
Restore Value (0x0E) – (only Mifare Classic) The Restore Value command is an operation on a value-type block, as defined by the Mifare standard. It takes as arguments block number of the block where the value is stored (Absolute Block Number), the Authentication Key Number and (optionally) the Authentication Key. If the block is properly formatted, then the value from the block is copied to a volatile memory register on the tag. This command has no visible results. The user can only check in the Result Register (the RFID Memory tab) if the Restore Value operation was successful. To store the value in the same or another block (copy to non-volatile memory on the tag), the user must execute the Transfer Value command.

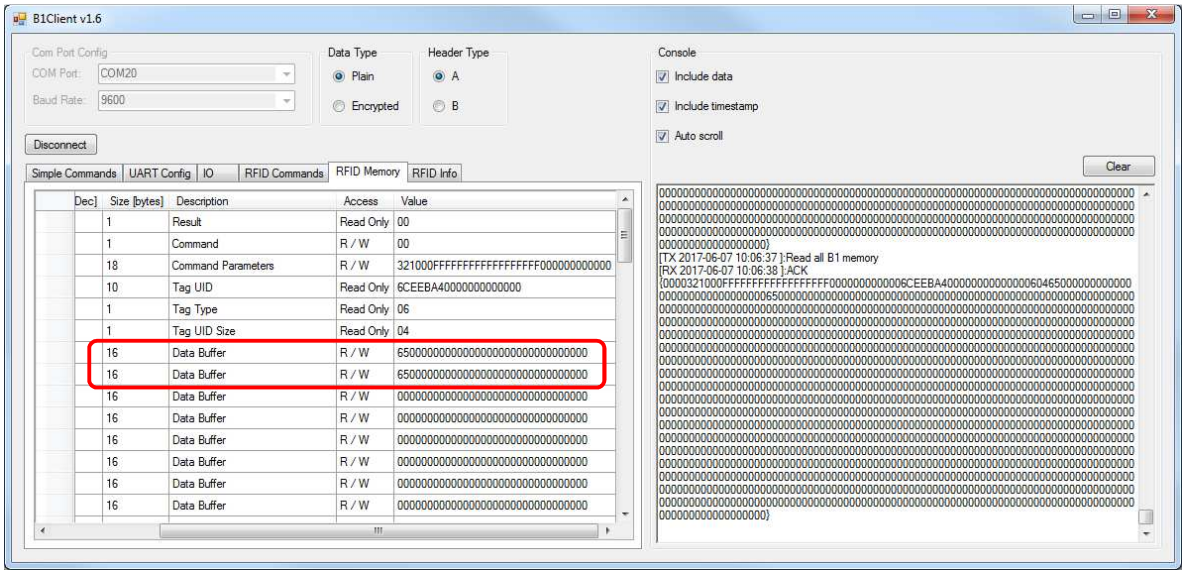


Transfer Value (0x0F) – (only Mifare Classic) The Transfer Value command is an operation on a value-type block as defined by the Mifare standard. It takes as arguments the block number of the block where the value is stored (Absolute Block Number), the Authentication Key Number and (optionally) the Authentication Key. The value from the volatile register on the tag is copied to the pointed block. To verify that this operation was successful the user should execute the Read Value command and check the proper Data Buffer in the RFID Memory tab.

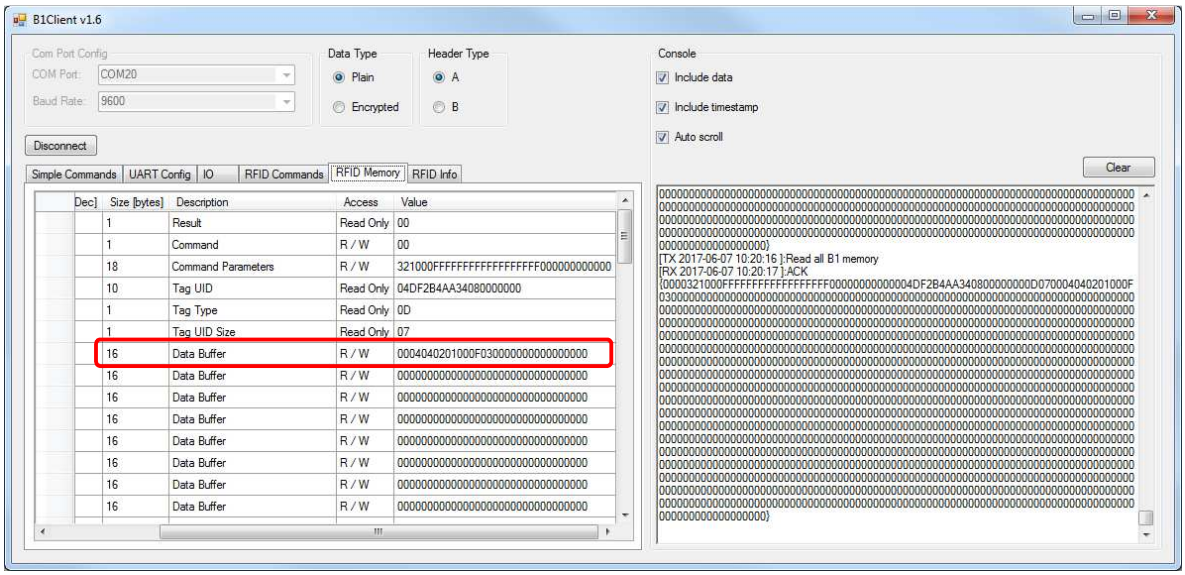


Recover Value (0x10) – (only Mifare Classic) The Recover Value command takes as arguments block number of the block where the value is stored (Absolute Block Number), the offset of the Data Buffer where the value and read address will be stored (Data Buffer Offset), the Authentication Key Number and (optionally) the Authentication Key. This command has the same functionality as the Read Value command, except that it can be used on a block which is corrupted – it tries to recover data from a corrupted block.





Get Version (0x11) – This command doesn't take any arguments. It can be used with Mifare Ultralight EV1, NTAG213, NTAG215 and NTAG216 tags. After successful reading, the first 8 bytes in the Data Buffer are filled with data defined by the NXP standard. Please refer to the RFID B1 or NXP documentation for more information. Example for NTAG are shown in Table 1. More information about tag types can be found in the B1 user manual.

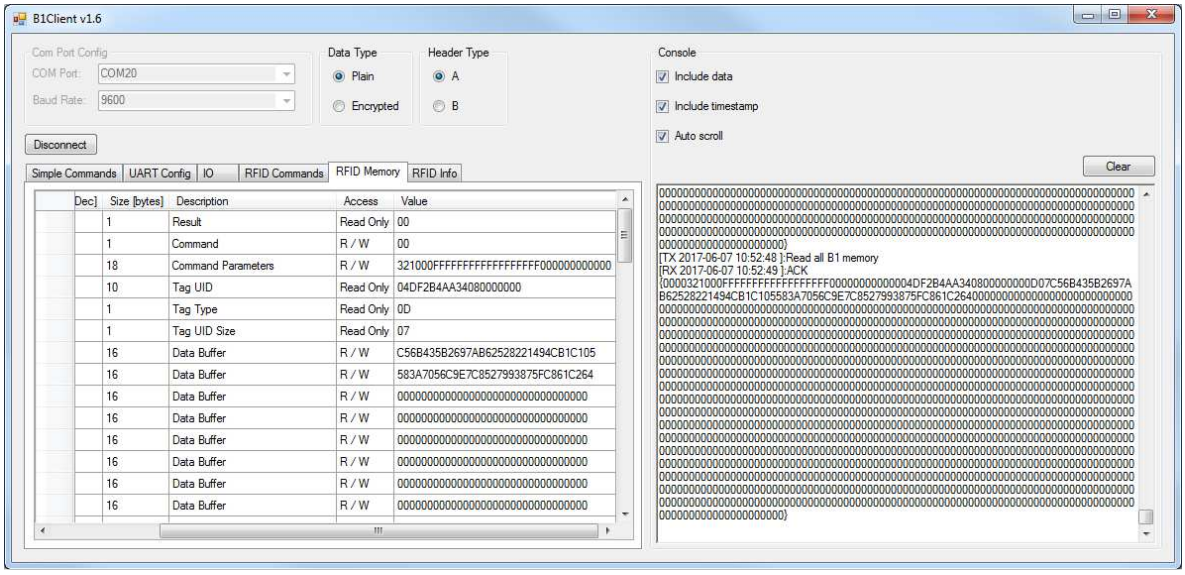


Byte No.	Description	NTAG213	NTAG215	NTAG216	Interpretation
0	Fixed header	0x00	0x00	0x00	
1	Vendor ID	0x04	0x04	0x04	NXP Semiconductors
2	Product type	0x04	0x04	0x04	NTAG
3	Product subtype	0x02	0x02	0x02	50 pF
4	Major product version	0x01	0x01	0x01	1
5	Minor product version	0x00	0x00	0x00	V0
6	Storage size	0x0F	0x11	0x13	
7	Protocol type	0x03	0x03	0x03	ISO/IEC 14443-3 compliant

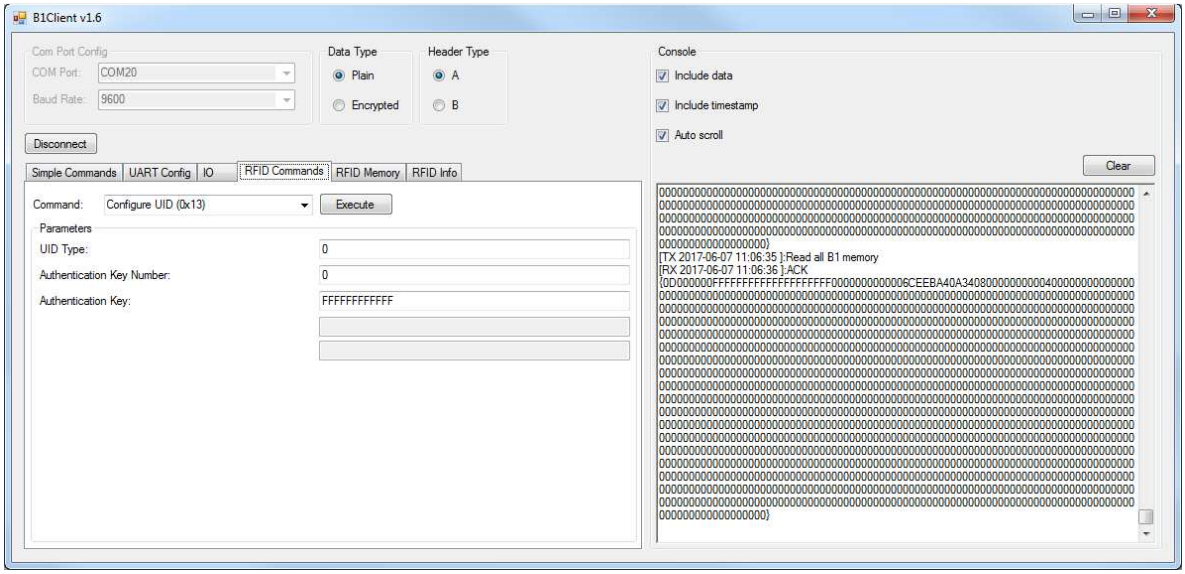
Table 1

In example above there is '0004040201000F030000000000000000' in the Data Buffer and that suggest NTAG213 tag.

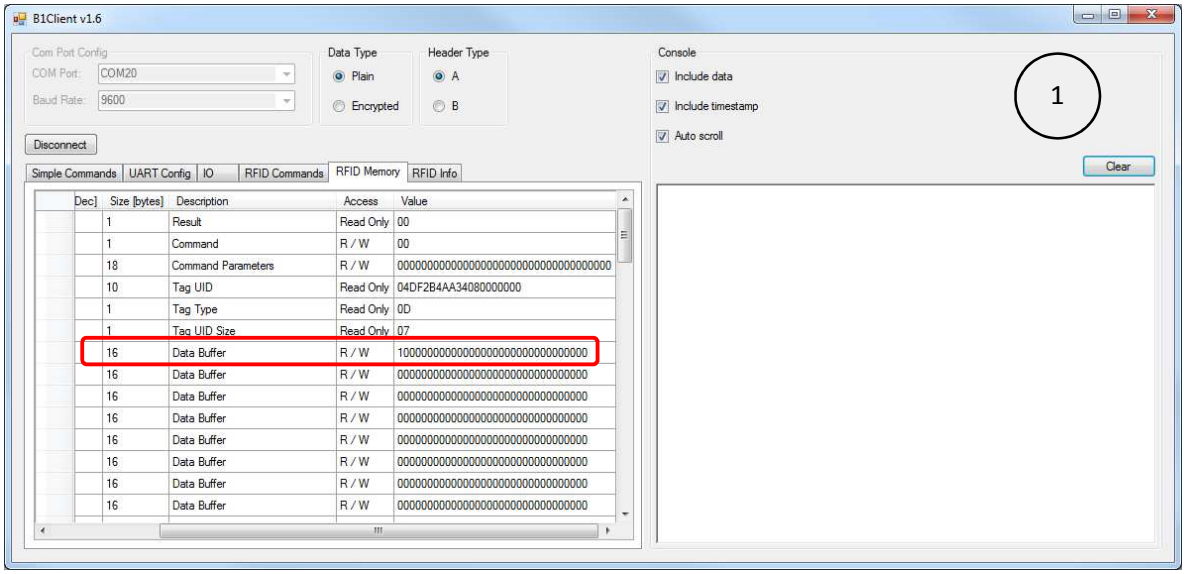
Read Signature (0x12) – The Read Signature command doesn't take any arguments. This command can be used with Mifare Ultralight EV1, NTAG213, NTAG215 and NTAG216 tags. After successful reading, the first 32 bytes in the Data Buffer are filled with the ECC signature defined by the NXP standard. Please refer to the NXP documentation for more information.

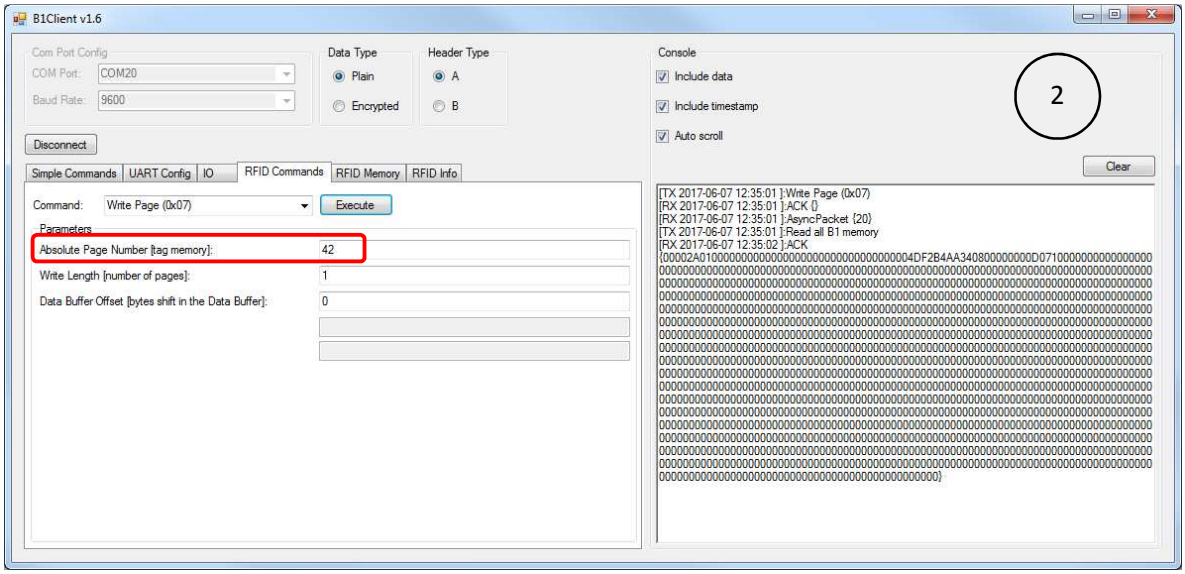


Configure UID (0x13) – The Configure UID command takes as argument the UID configuration byte (UID Type), the Authentication Key Number and (optionally) the Authentication Key. This command changes the configuration of the UID on some Mifare Classic tags. For more information please refer the RFID B1 user manual.



Read Counter (0x14) – The Read Counter command takes as arguments the tag Counter Number and the Data Buffer Offset in bytes. This command reads the counter value of the counter pointed to by the Counter Number and stores it in the Data Buffer at the Data Buffer Offset index as an unsigned 24-bit integer with the least significant byte first. This command works only for Ultralight EV1, NTAG213, NTAG215, NTAG216, NTAG213F, NTAG216F tags. The following example shows how to read the counter in the NTAG213 tag. The counter must be turned on first. This can be done by writing 0x10000000b into page 42 of the tag memory. For further information on how to execute this command with other types of tags please see the appropriate documentation from NXP.





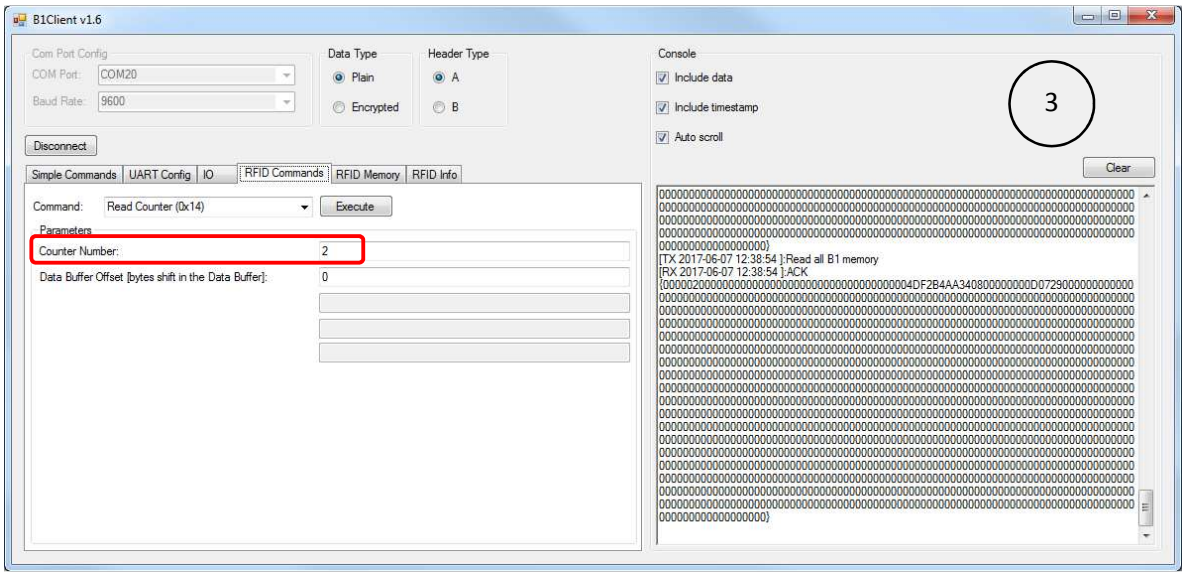


Figure 23. There is the only one counter with number 2.

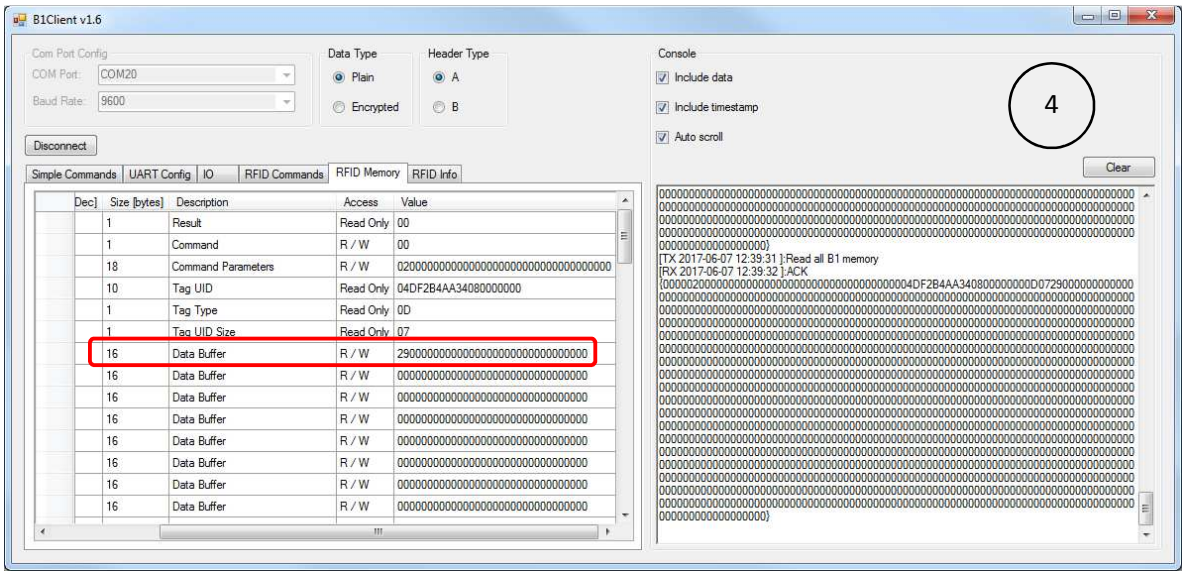


Figure 24. The counter has a value of 0x29h.

Increment Counter (0x15) – The Increment Counter command takes as arguments the tag Counter Number and the Increment Value to be added to the counter value. This command increments the counter value of the counter pointed to by the Counter Number. The Increment Value is a 24-bit unsigned number with the least significant byte first. This command works only with Ultralight EV1 tags. The following example shows how to increment the counter in the Ultralight EV1 tag.

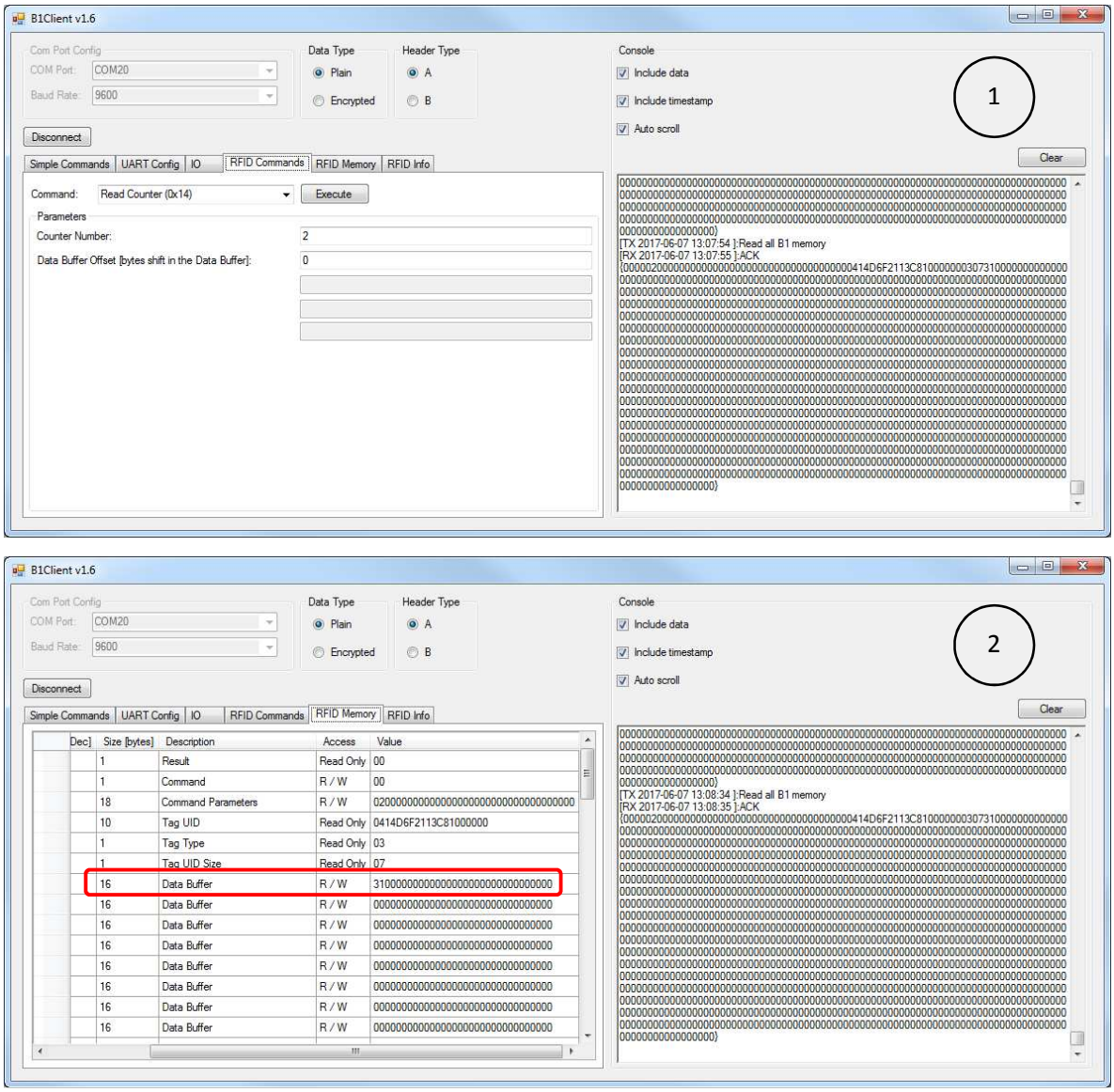


Figure 25. The Data Buffer after the Read Counter command execution.

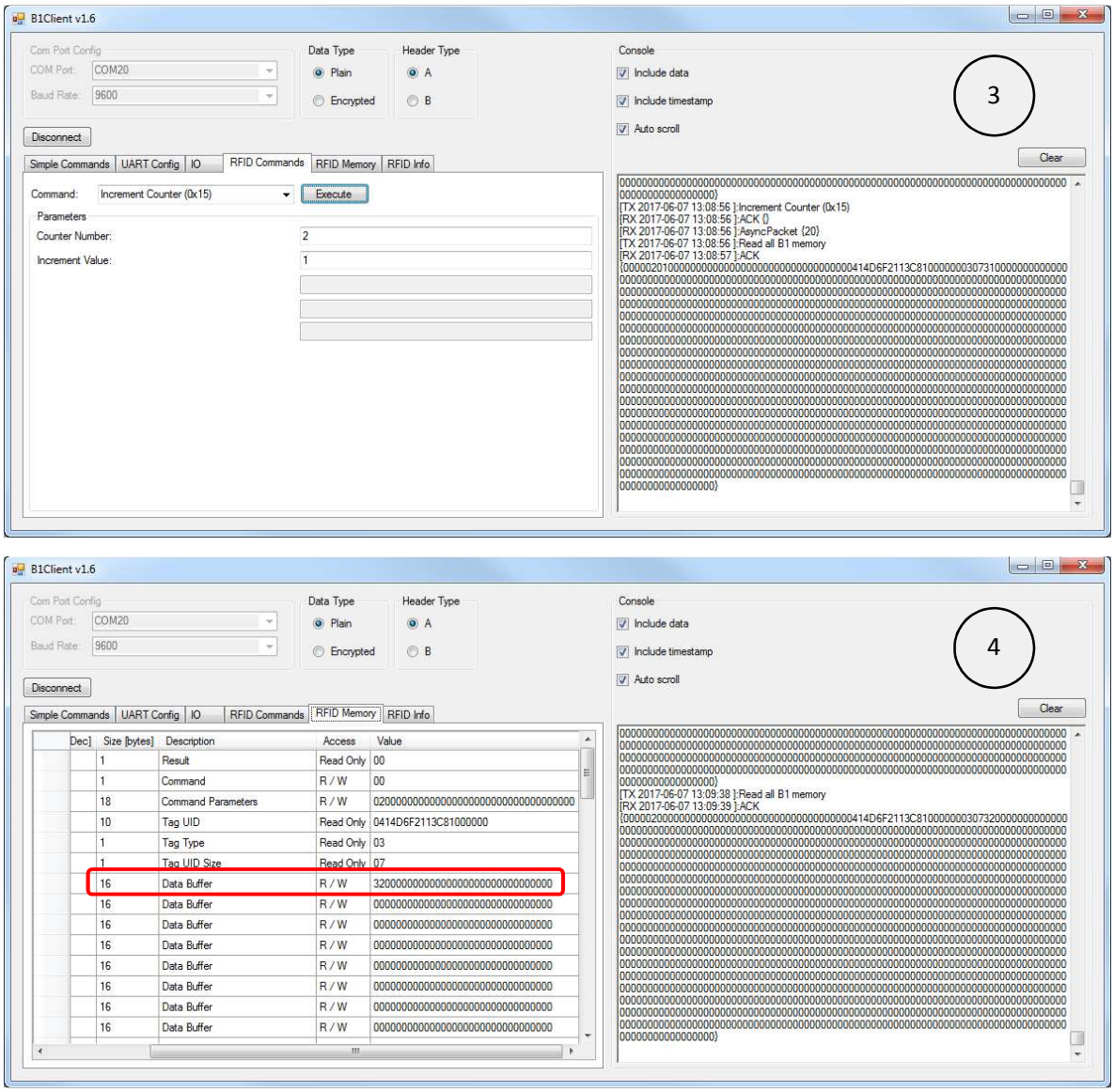
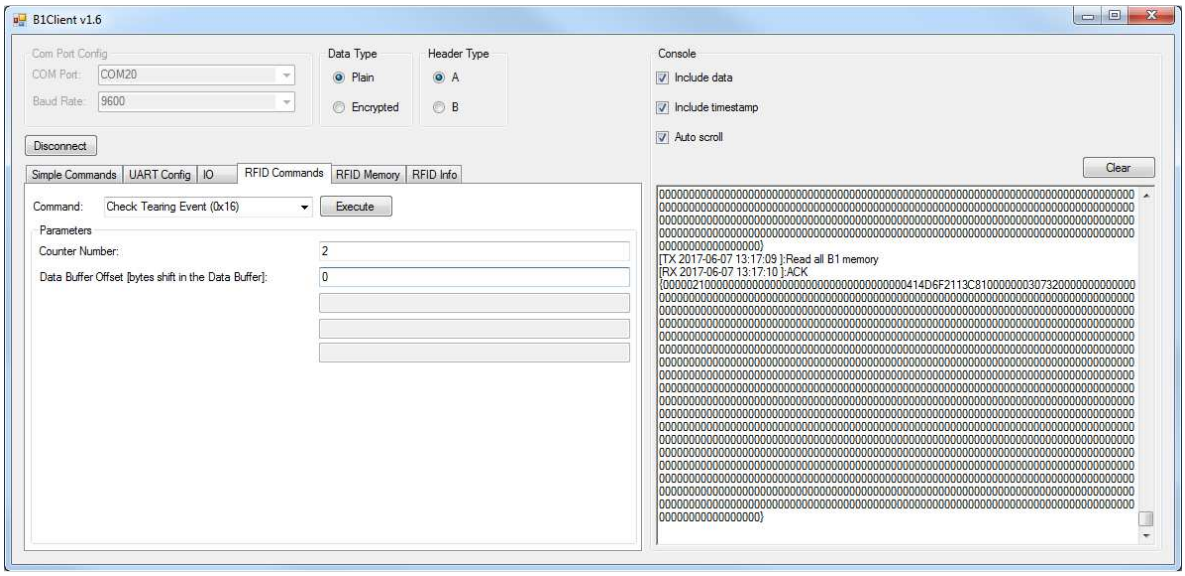


Figure 26. The Data Buffer after the Increment and Read Counter commands execution.

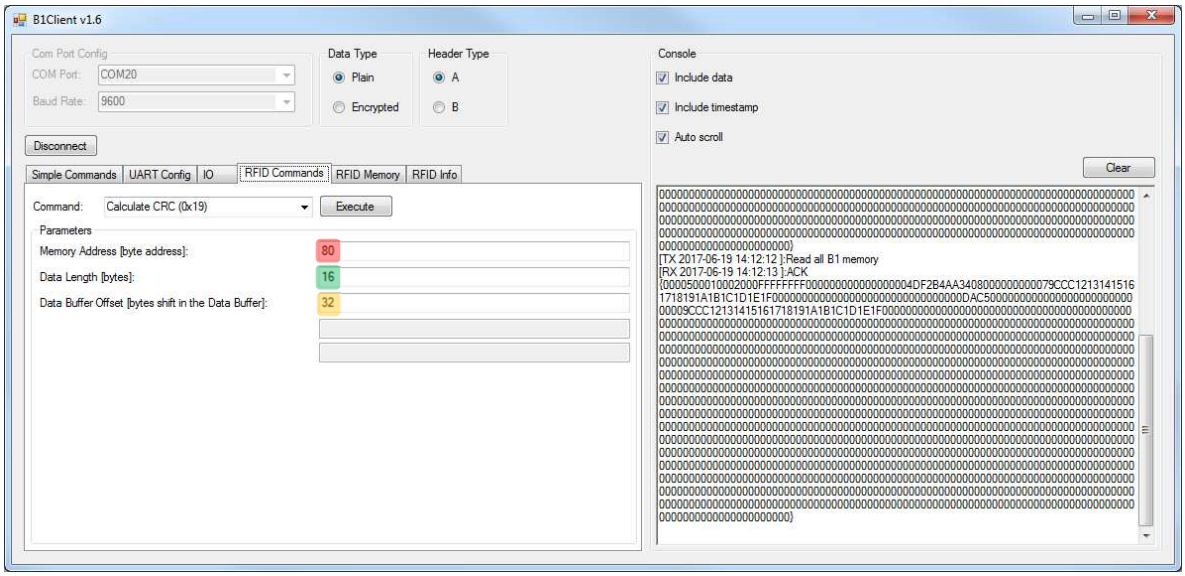
Check Tearing Event (0x16) – (only Ultralight EV1) The Check Tearing Event command takes as arguments the tag Counter Number and the Data Buffer Offset in bytes. This command checks whether there was a tearing event in the counter and stores the flag value in the Data Buffer at the Data Buffer Offset index. The value '0x00' is stored if there has been no tearing event, and '0x01' is stored if a tearing event occurred.

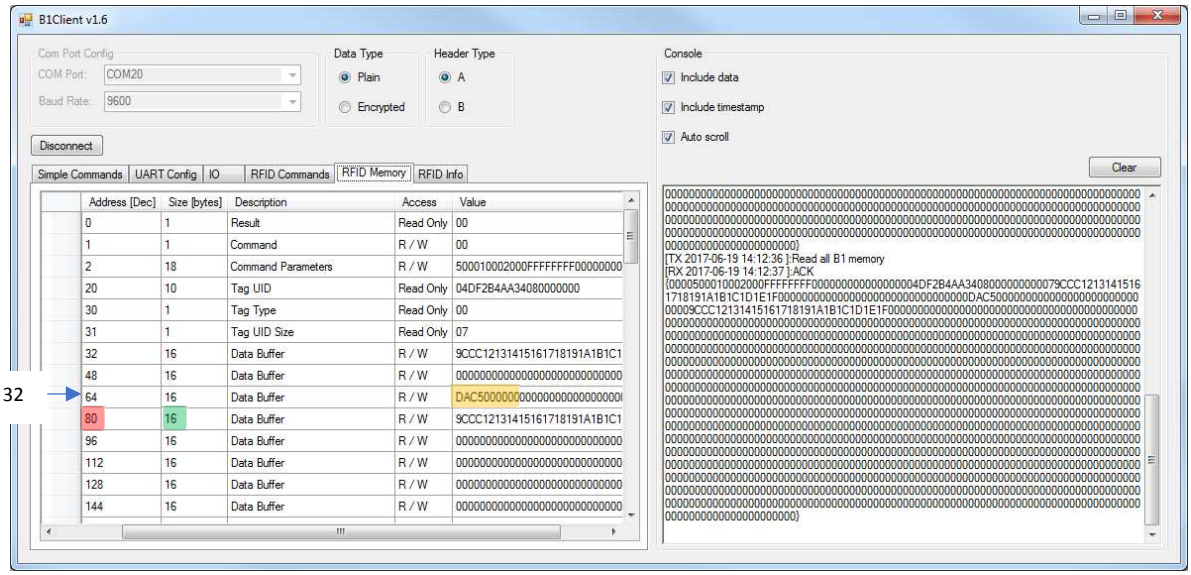


Password Authentication (0x17) – The Password Authentication command takes as arguments the Data Buffer Offset where PACK result will be stored, the Password Number and (optionally) the Password. This command tries to authenticate the tag using the chosen (pointed to) password. The 2-byte PACK result is stored in the Data Buffer under Data Buffer Offset index.

Halt (0x18) – The Halt command takes no arguments. It halts the tag and turns off the RF field. It must be executed at the end of each operation on a tag to disable the antenna and reduce the power consumption.

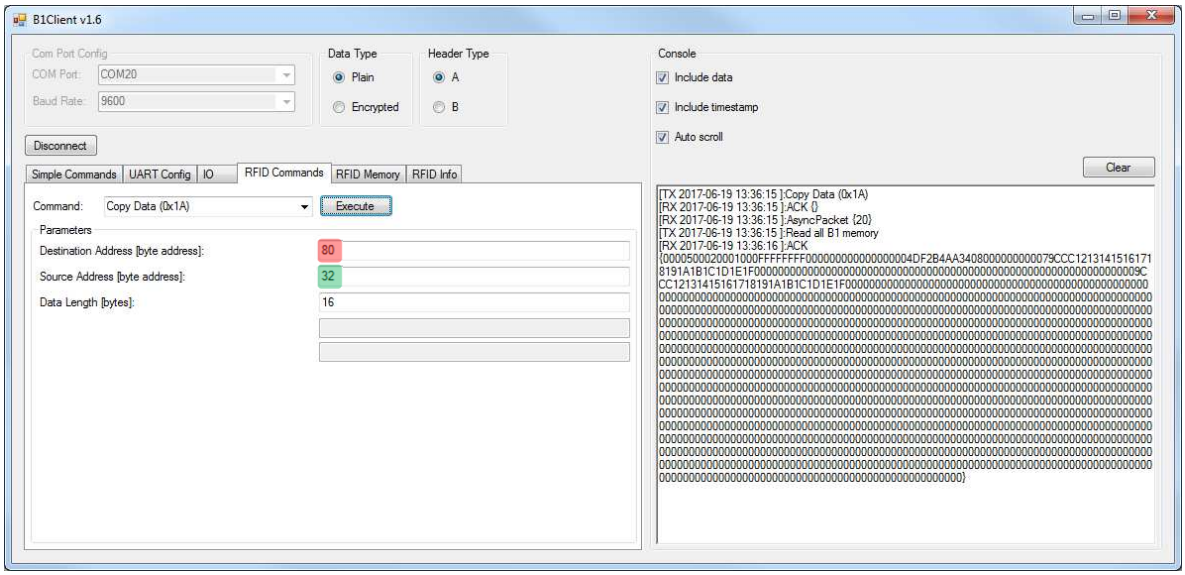
Calculate CRC (0x19) – The Calculate CRC command takes as arguments the Memory Address (B1 based module memory), the Data Length in bytes and the Data Buffer Offset. The CRC calculation starts at the byte pointed at by the memory address and is done on the 'Data length' number of bytes in the volatile memory. The result is stored in the Data Buffer at the Data Buffer Offset index. The result is a 16-bit unsigned value with least significant byte first only when the module's memory is unlocked.

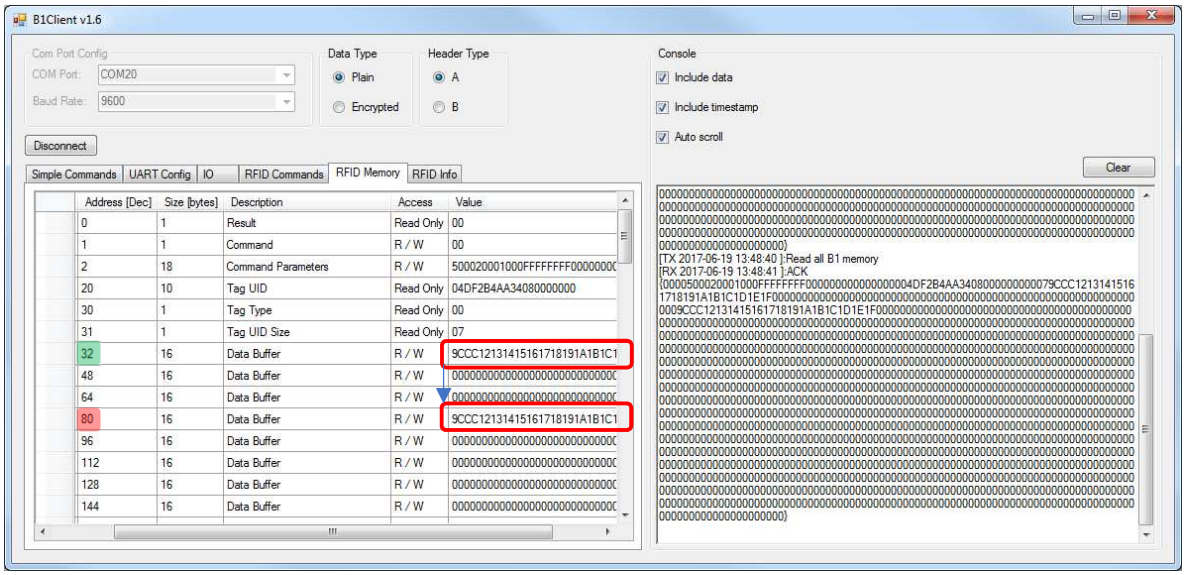




Data Buffer offset - 32

Copy Data (0x1A) – The Copy Data command copies data around inside the RFID Module memory. This command takes as arguments the Destination Address (decimal value), the Source Address (decimal value) and the Data Length in bytes.

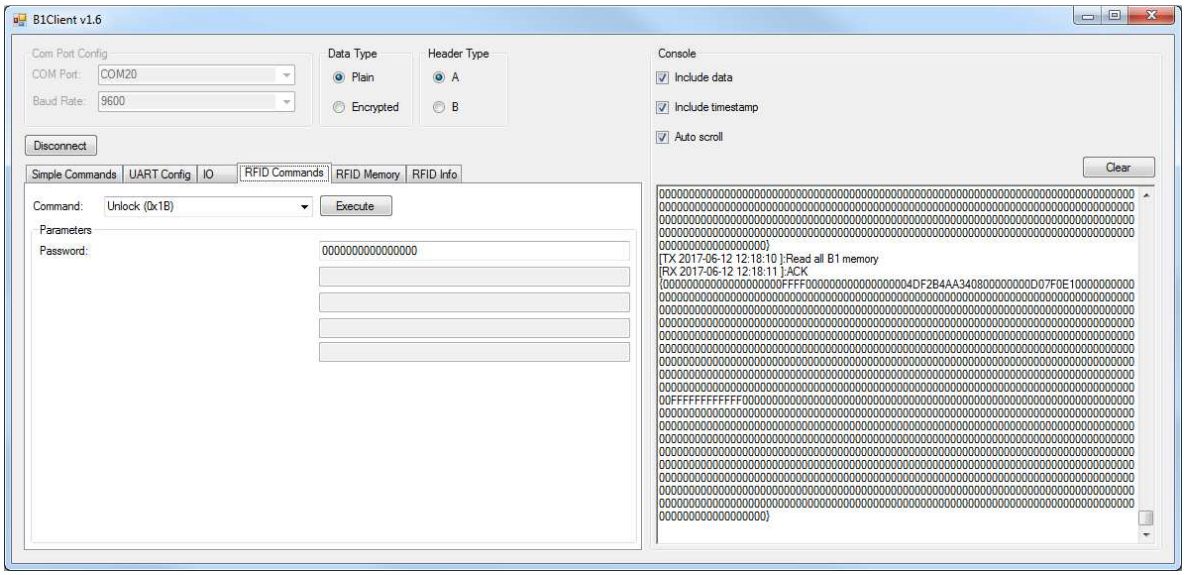


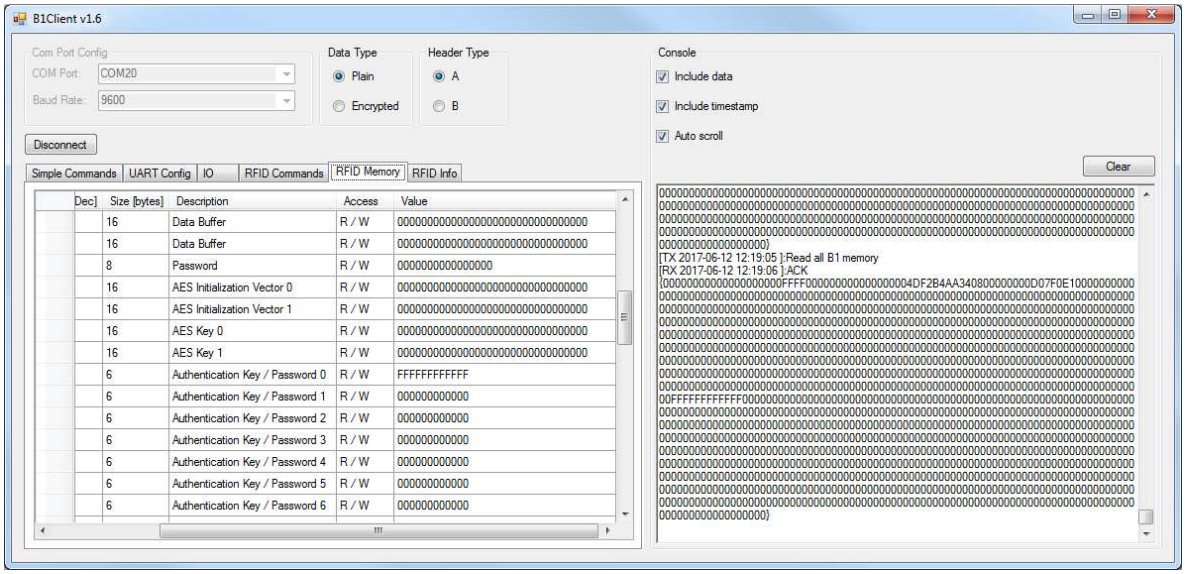


Unlock (0x1B) – The Unlock command takes as an argument an 8-byte long password. If the password matches the actual password in the module the memory which contains:

- AES Initialization Vectors
- AES Encryption Keys
- Authentication Keys and Passwords
- User Memory
- Password

will be unlocked and made accessible for read and write operations.

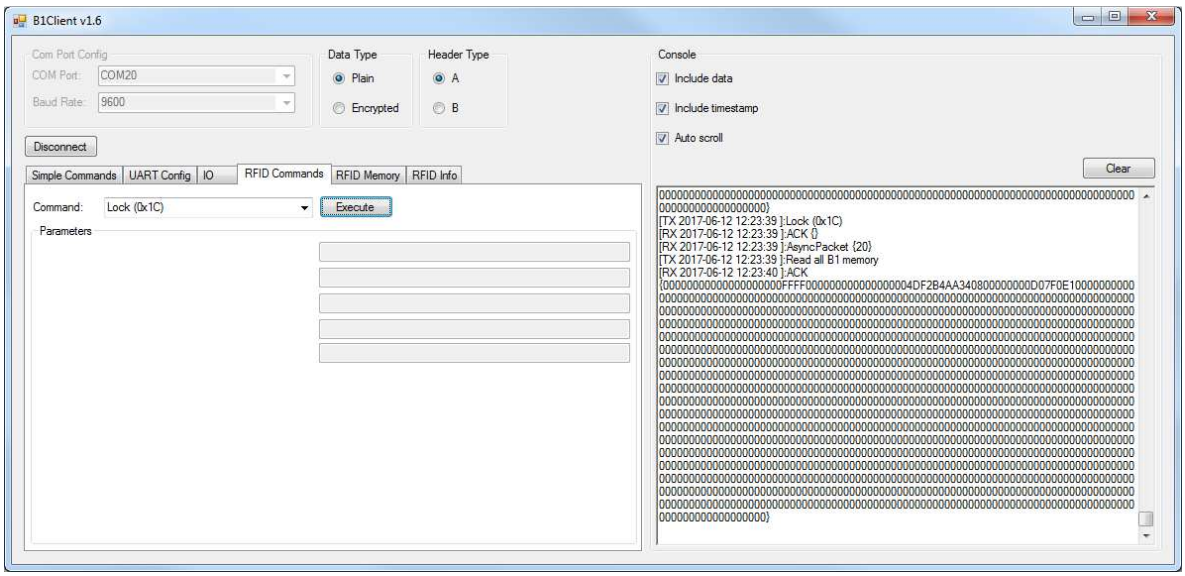


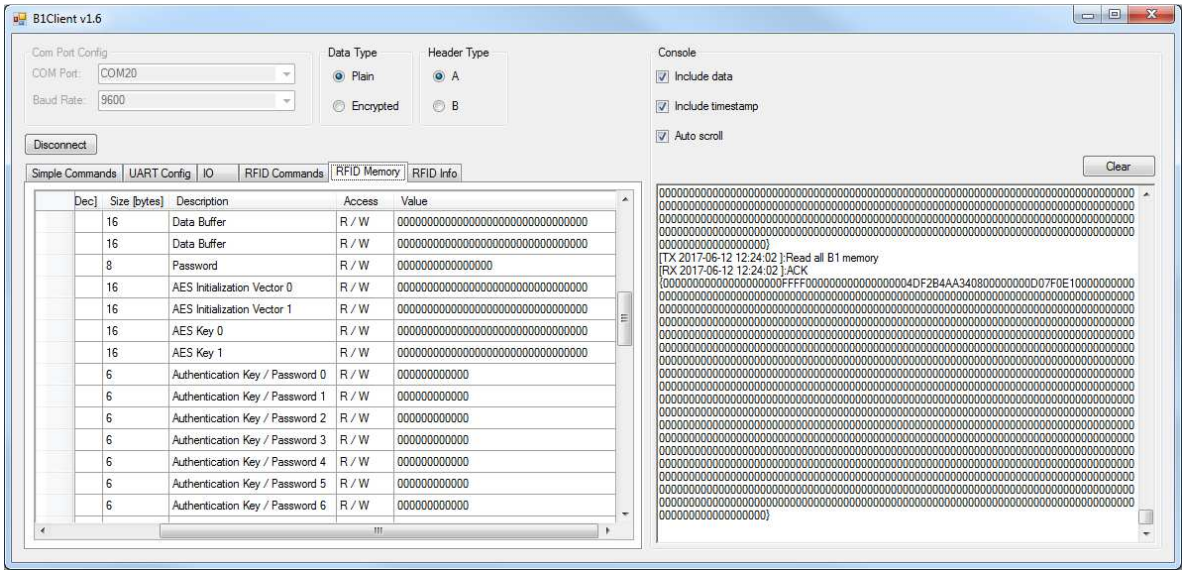


Lock (0x1C) – The Lock command takes no arguments. If the device is unlocked and the user executes the Lock command, protected memory which contains:

- AES Initialization Vectors
- AES Encryption Keys
- Authentication Keys and Passwords
- User Memory
- Password

will be saved to non-volatile memory and locked (direct access will be blocked).





Get Module Version (0x1D) – The Get Module Version command takes no arguments. After execution, the Data Buffer at index 0x00 is filled with a NULL-ended ASCII string which describes the hardware and firmware version of the module.

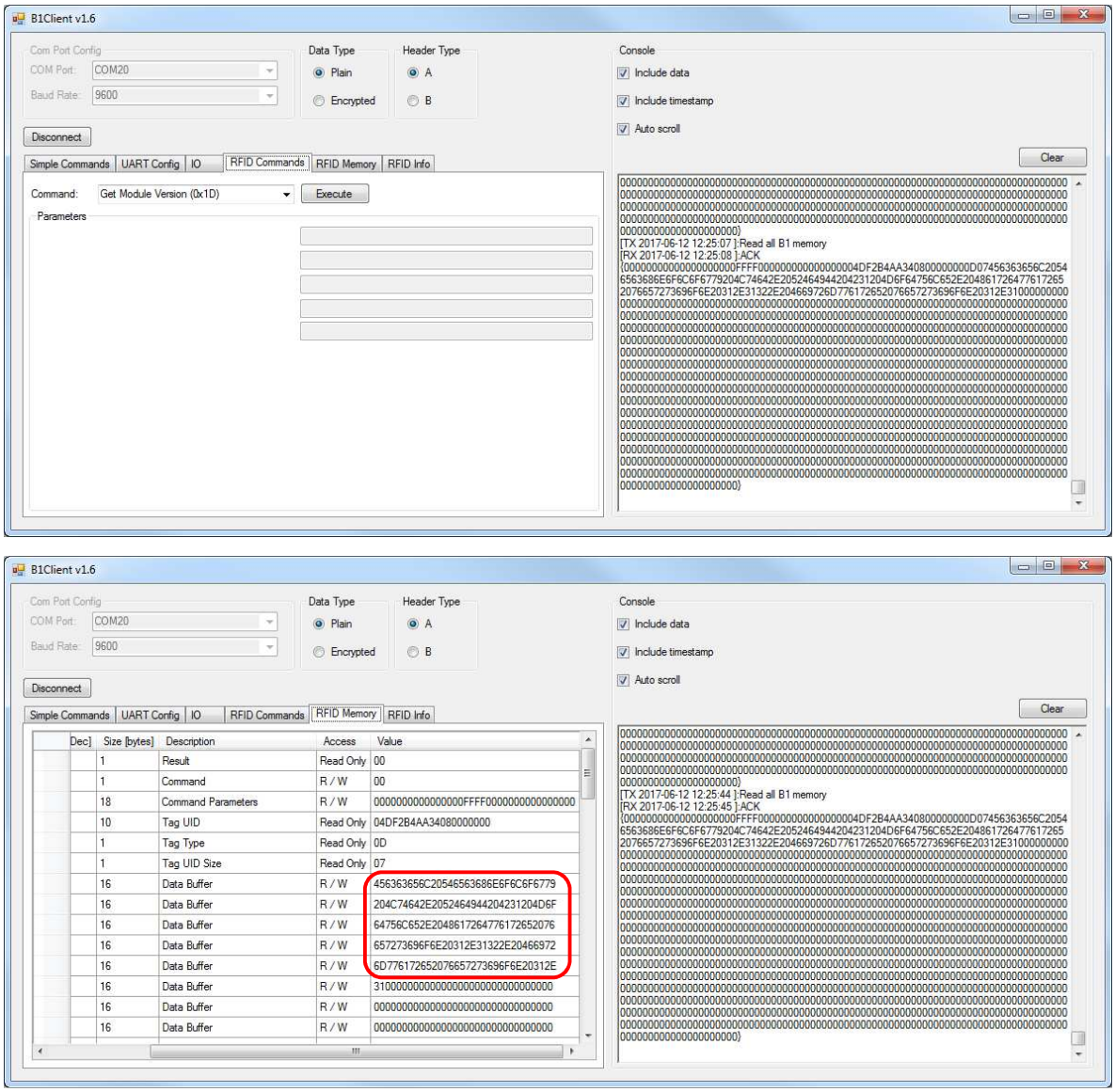
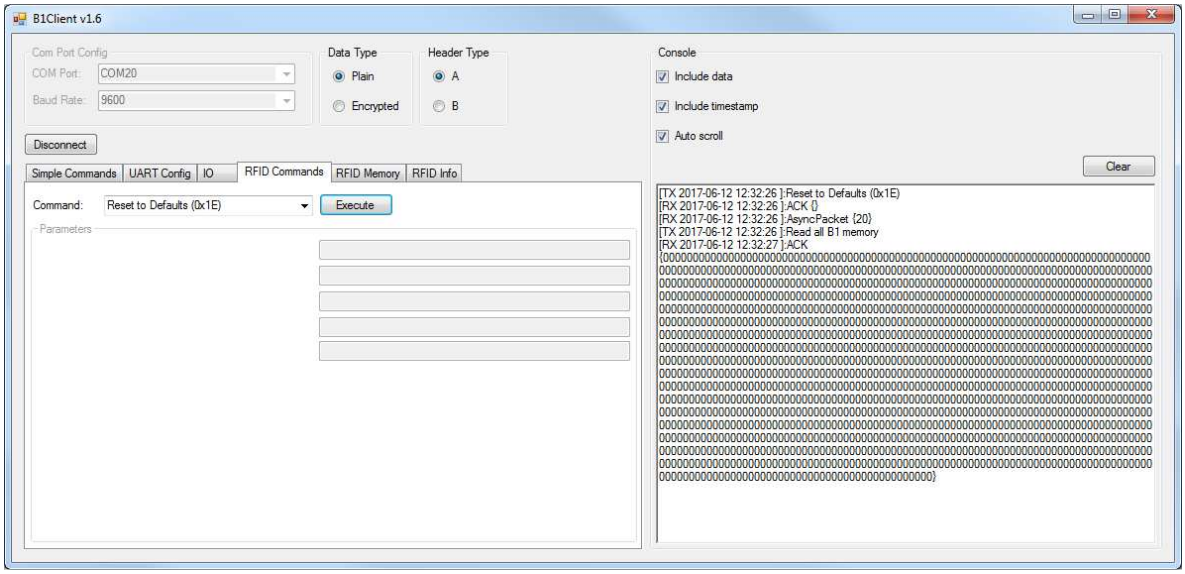


Figure 27. After decoding: Eccel Technology Ltd. RFID B1 Module. Hardware version 1.12. Firmware version 1.1

Reset to Defaults (0x1E) – The Reset to Default command doesn't take any arguments. It resets all memory including protected memory to factory default settings. All locked information is lost.



Enumerate tags UID (0x1F) – This command is only for the B1MT (Multiple Tag) based modules. The Enumerate Tags UID command doesn't take any arguments. This command scans for all tags within the range of the antenna and enumerates their UIDs in the Data Buffer. The first byte in the Data Buffer says how many tags were discovered. If there are no tags in the range, then the Result Register is updated with 0x08 (Tag is not present) value. After enumeration, if there is at least one tag in the field, the Result Register is updated with 0x00 (No Error) value. Information about all enumerated tags is stored in the Data Buffer, starting from the second byte of the buffer. There are as many records in the Data Buffer as there were tags found. Each record contains the Tag UID Size followed by the Tag UID is stored in the rest of the bytes in the record. The record length is variable and equal to the UID size plus one byte.

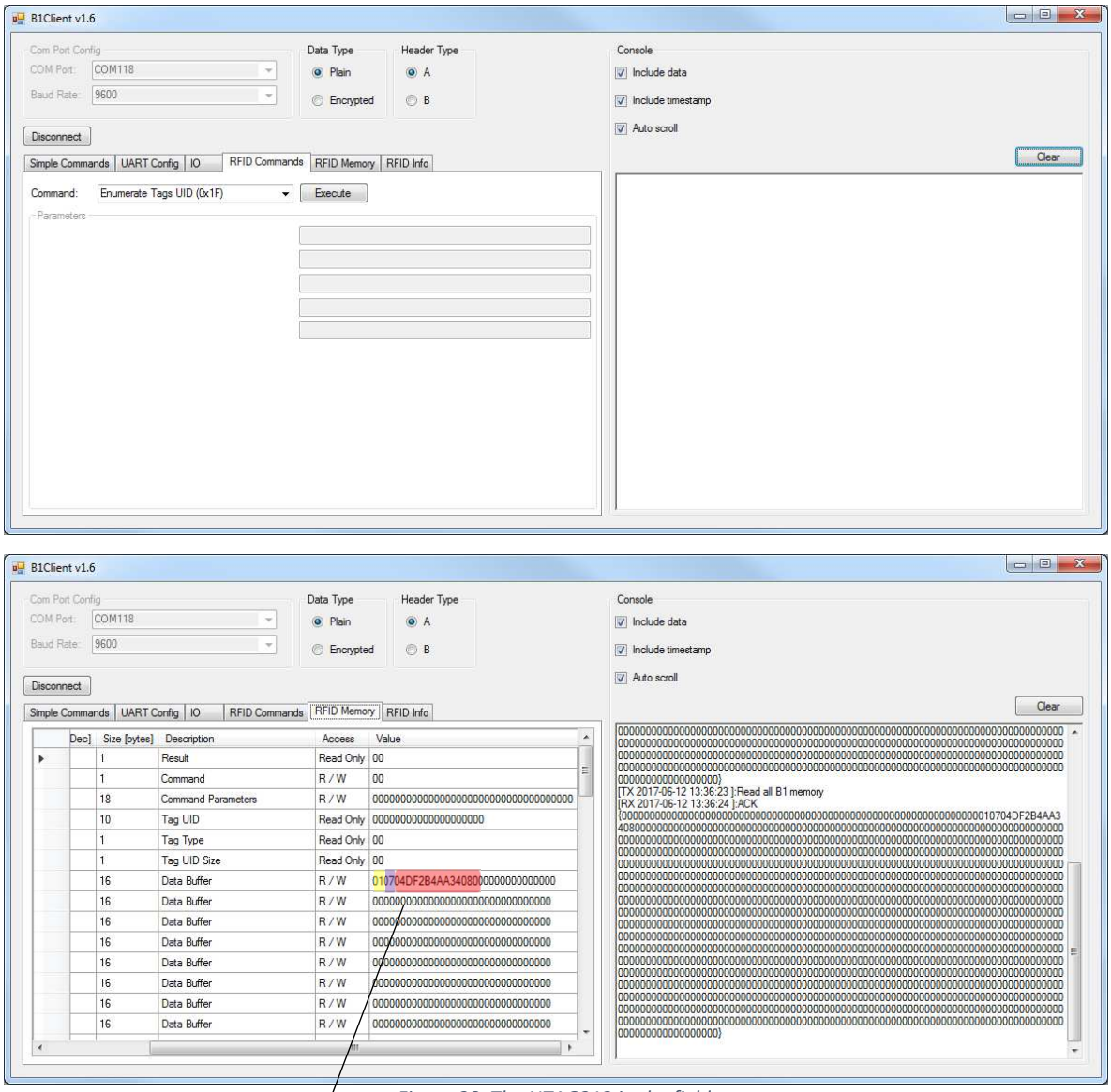


Figure 28. The NTAG213 in the field.

0x01 – Number of tags in the field, 0x07 – Tag UID Size, 0x04DF2B4AA34080 – Tag UID

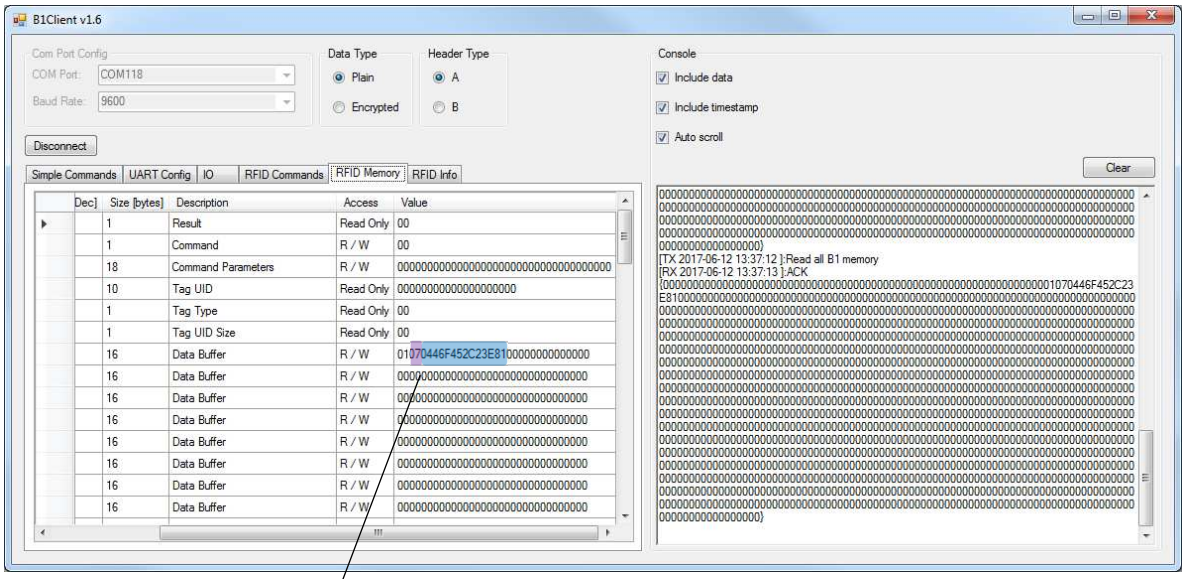


Figure 29. The NTAG215 in the field.

0x01 – Number of tags in the field, 0x07 – Tag UID Size, 0x0446F452C23E81 – Tag UID

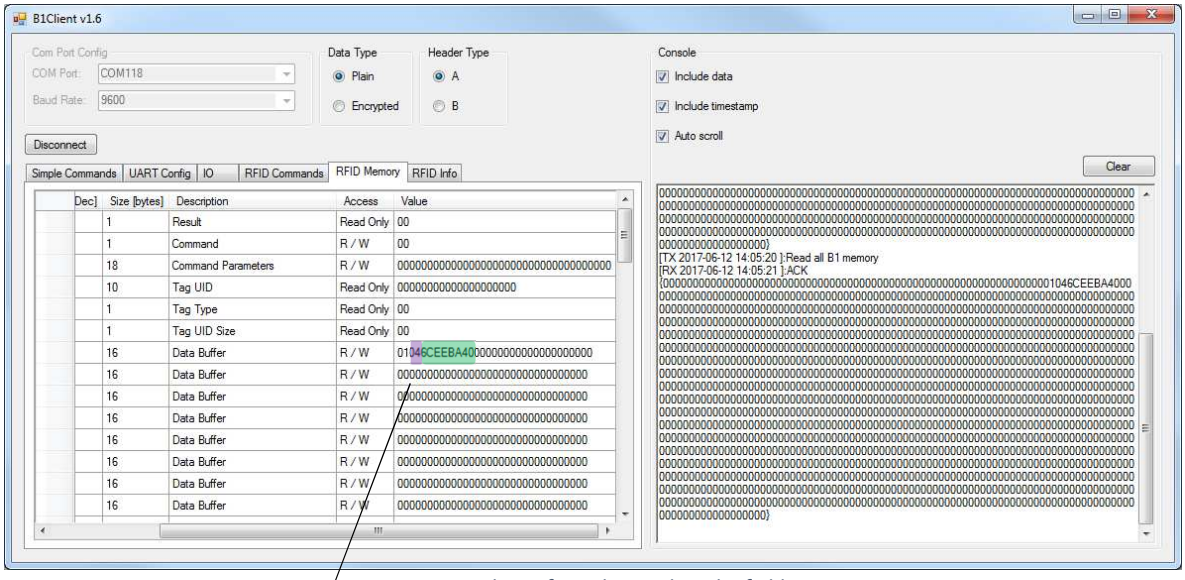
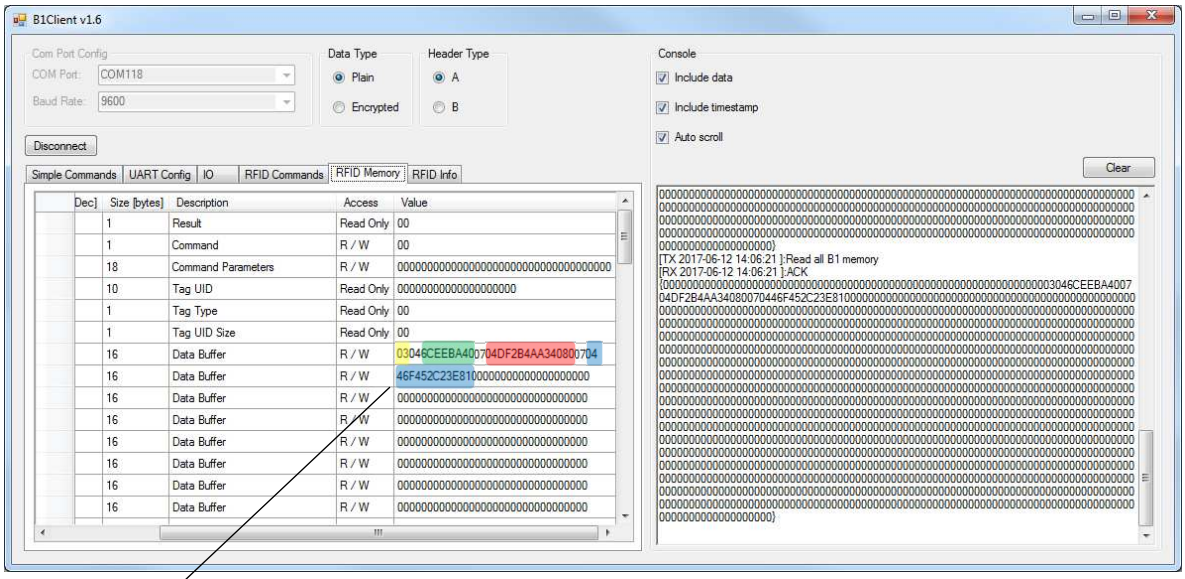
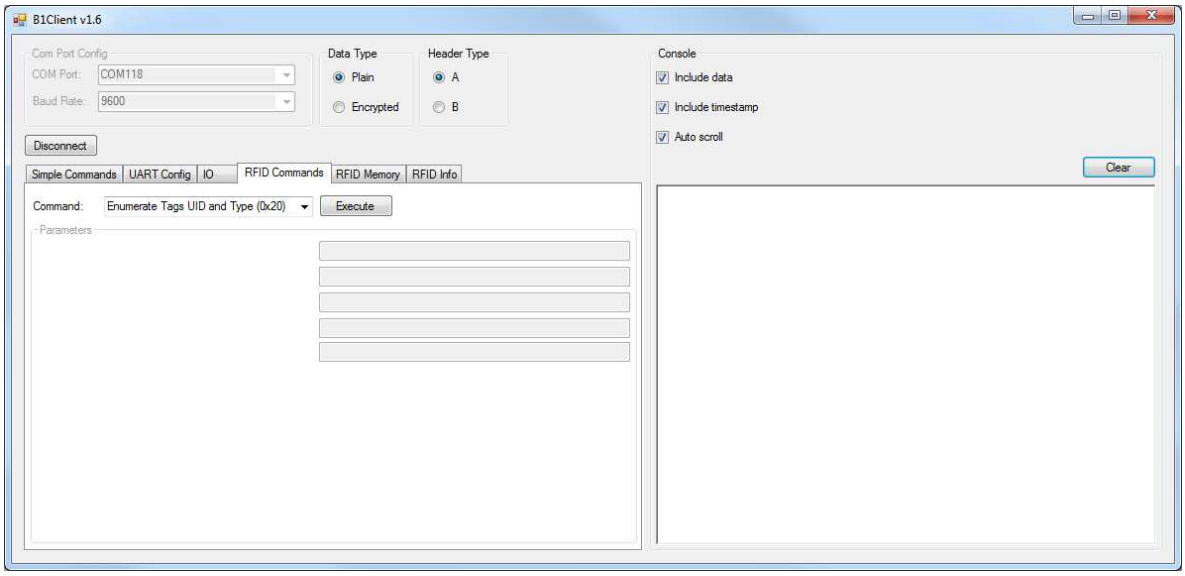


Figure 30. The Mifare Classic 1k in the field.

0x01 – Number of tags in the field, 0x04 – Tag UID Size, 0x6CEEBA40 – Tag UID


B1Client v1.6
Com Port Config
COM Port: COM118
Baud Rate: 9600
Disconnect
Data Type
Plain
Encrypted
Header Type
A
B
Console
Include data
Include timestamp
Auto scroll
Clear
Simple Commands
UART Config
IO
RFID Commands
RFID Memory
RFID Info
Dec
Size [bytes]
Description
Access
Value
1 Result Read Only 00
1 Command R / W 00
18 Command Parameters R / W 000000000000000000000000000000000000
10 Tag UID Read Only 00000000000000000000
1 Tag Type Read Only 00
1 Tag UID Size Read Only 00
16 Data Buffer R / W 03046CEEBA400704DF2B4AA340800704
16 Data Buffer R / W 46F452C23E810000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
16 Data Buffer R / W 00000000000000000000000000000000
[TX 2017-06-12 14:06:21]:Read all B1 memory
[RX 2017-06-12 14:06:21]:ACK
{0000000000000000000000000000000000000000000000000000000000000000000000000003046CEEBA4007
04DF2B4AA340800704446F452C23E810000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000

Enumerate tag UID and Types (0x20) – This command is only for the B1MT (Multiple Tag) based modules. The Enumerate Tags UID and Type command doesn't take any arguments. This command scans for all tags within the range of the antenna and enumerates their UID and type in the Data Buffer. The first byte in the Data Buffer says how many tags were discovered. If there are no tags in the range, then the Result Register is updated with 0x08 (Tag is not present) value. After enumeration, if there is at least one tag in the field the Result Register is updated with 0x00 (No Error) value. Information about all enumerated tags is stored in the Data Buffer starting from the second byte of the buffer. There are as many records in the Data Buffer as there were tags found. Each record contains the Tag Type as the first byte, the Tag UID Size as the second byte and the Tag UID is stored in the rest of the bytes in the record. The record length is variable and equal to the UID size plus two bytes.



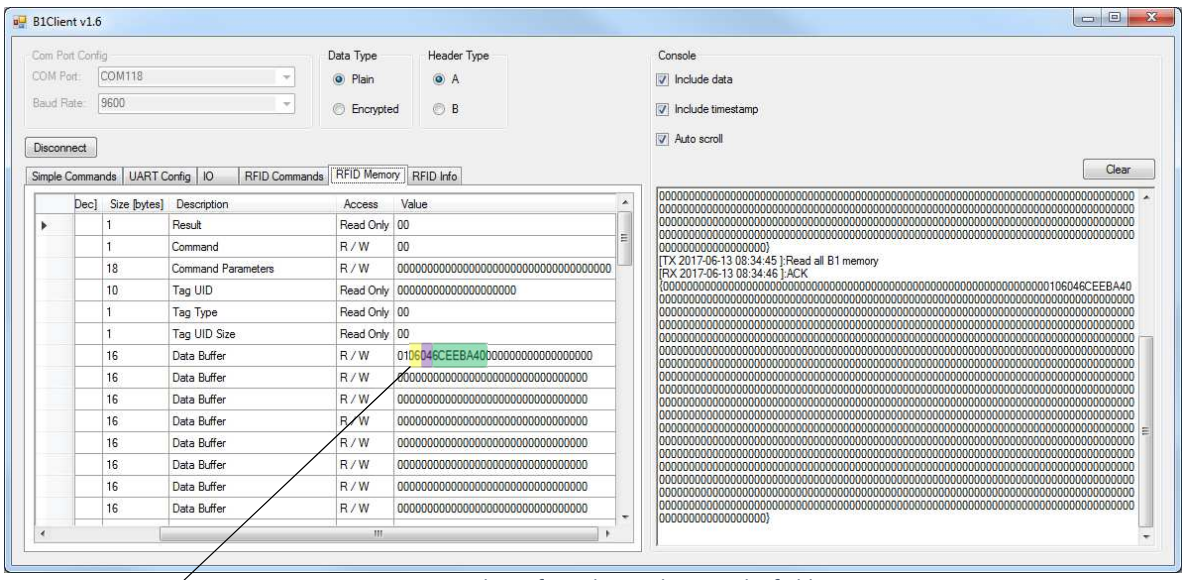


Figure 32. The Mifare Classic 1k tag in the field.

0x01 – Number of tags in the field, 0x06 – Tag Type: Classic 1k, 0x04 – Tag UID Size, 0x6CEEBA40 – Tag UID

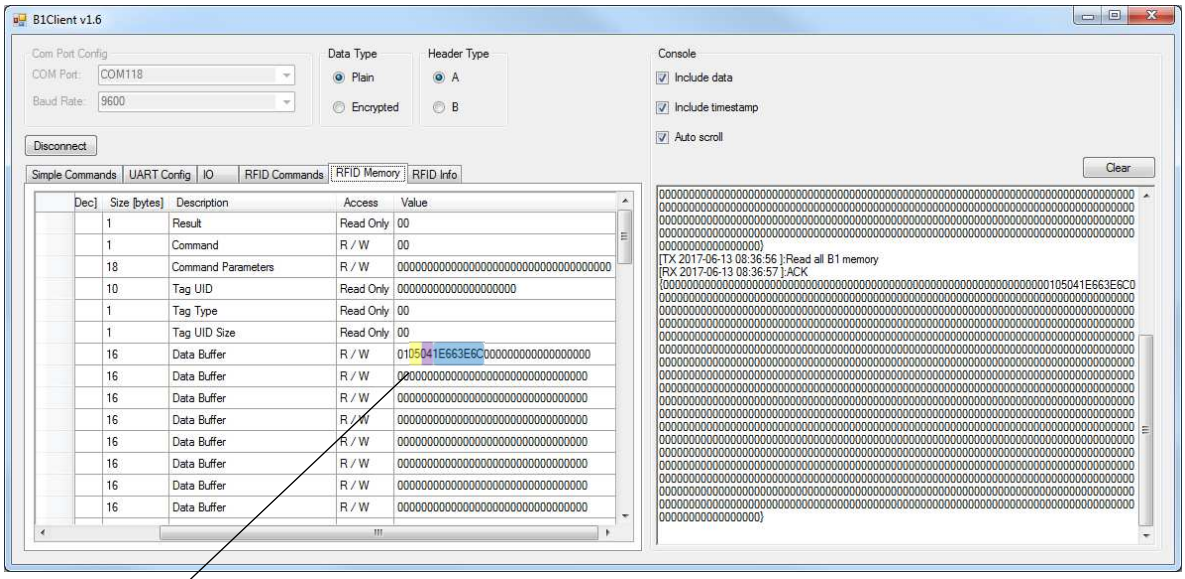


Figure 33. The Classic Mini tag in the field.

0x01 – Number of tags in the field, 0x05 – Tag Type: Classic Mini, 0x04 – Tag UID Size, 0x1E66E6C – Tag UID

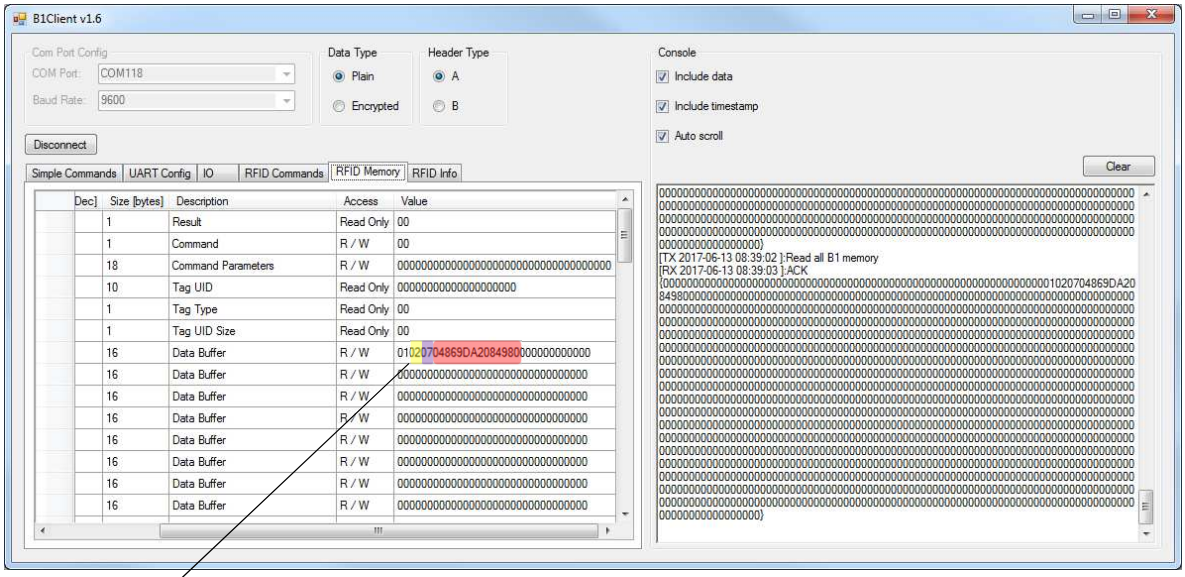


Figure 34. The Ultralight tag in the field.

0x01 – Number of tags in the field, 0x02 – Tag Type: Ultralight, 0x07 – Tag UID Size, 0x04869DA2084980 – Tag UID

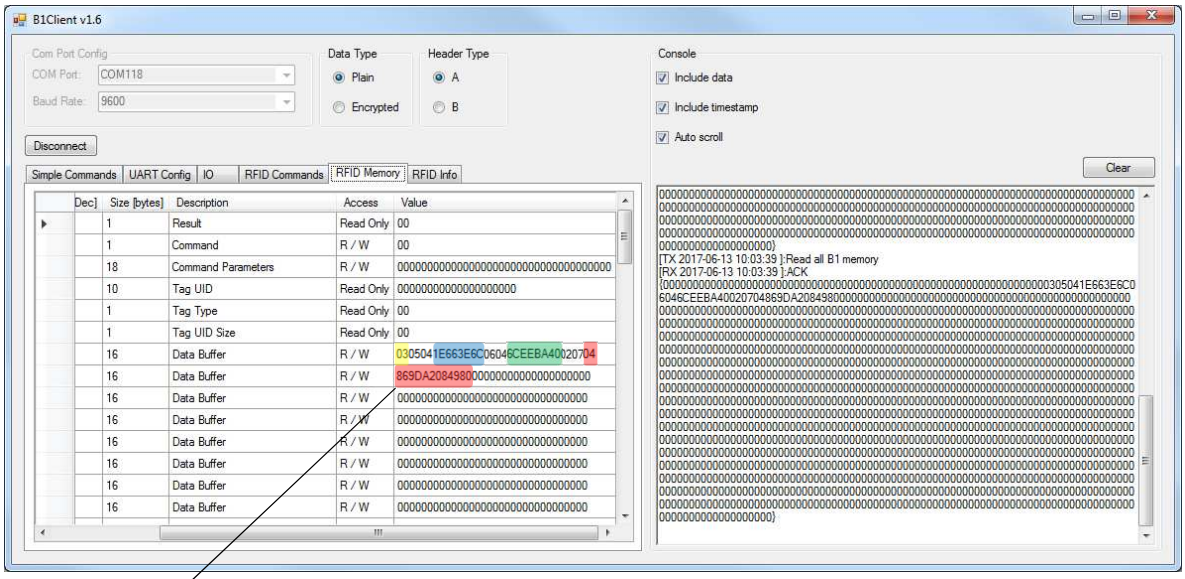


Figure 35. Three tags (Mifare Classic 1k, Classic Mini, Ultralight) in the field.

0x03 – Number of tags in the field,
0x05 – Tag Type: Classic Mini, 0x04 – Tag UID Size, 0x1E66E6C – Tag UID,
0x06 – Tag Type: Mifare Classic 1k, 0x04 – Tag UID Size, 0x6CEEBA40 – Tag UID,
0x02 – Tag Type: Ultralight, 0x07 – Tag UID Size, 0x04869DA2084980 – Tag UID.

Select tag (0x21) – This command is only for the B1MT (Multiple Tag) based modules. The select tag command takes as arguments the UID Size in bytes and the UID bytes themselves. If the command execution is successful, then the addressed tag is selected and the Result Register is updated with 0x00 (No Error) value. In the following example the Mifare Classic 1k tag is chosen.

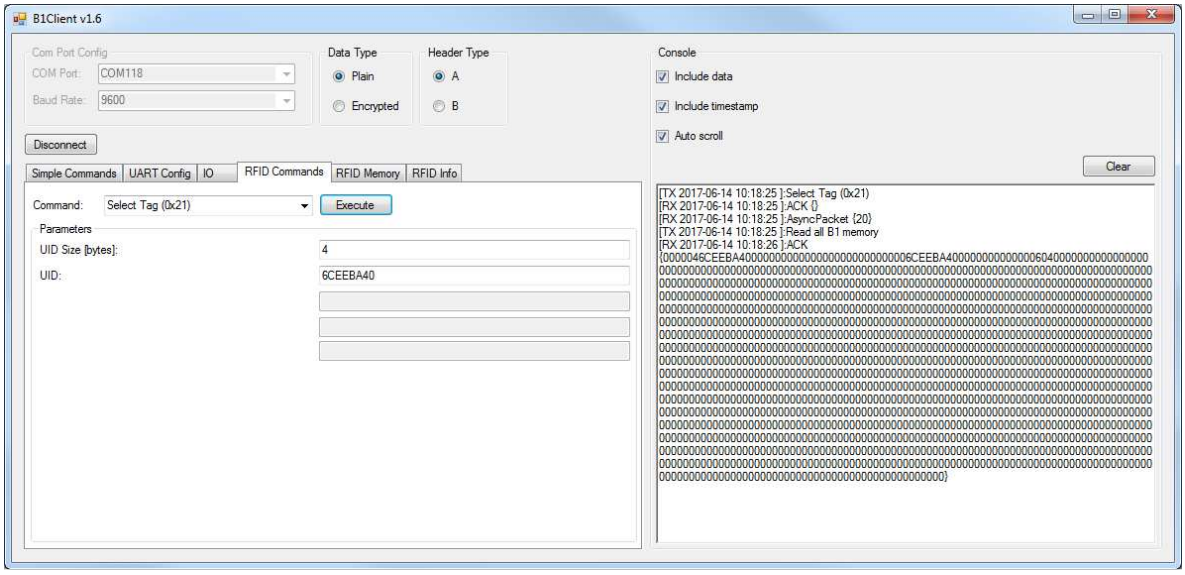
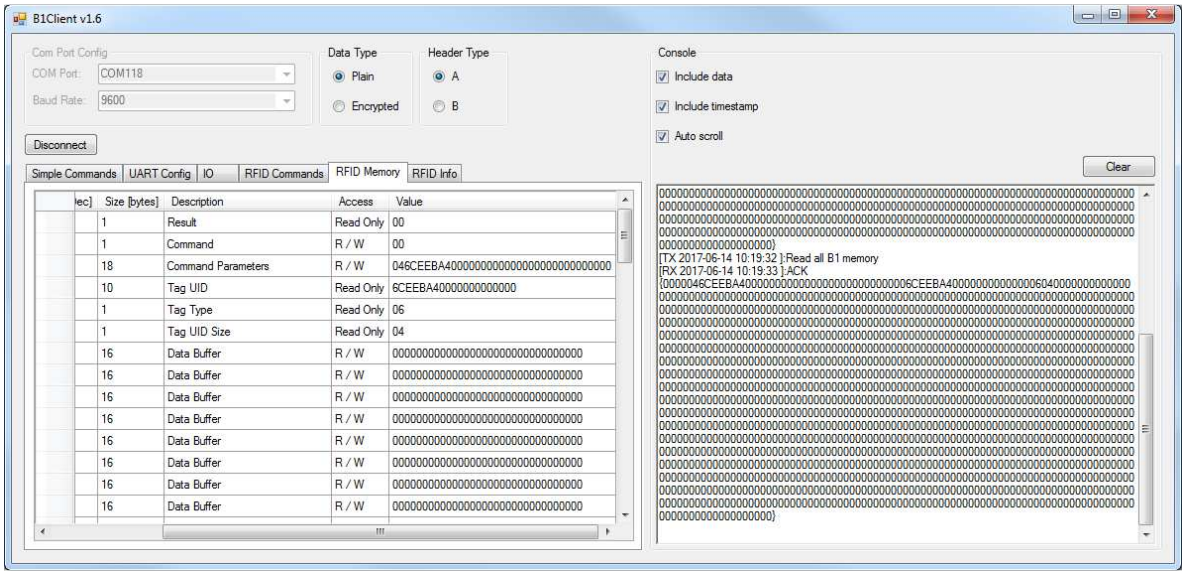
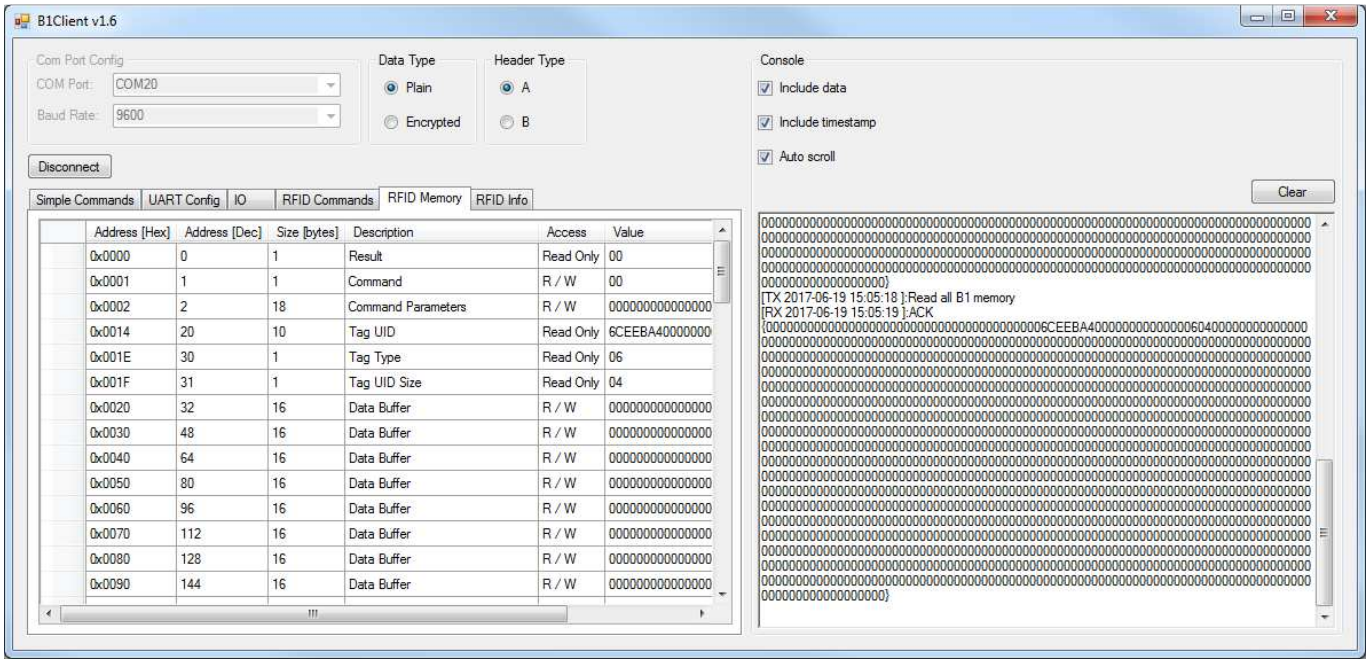


Figure 36. Mifare Classic 1k selected.



3.2.5 RFID Memory

The RFID Memory tab presents the whole memory of the B1 based module chip – RFID section. In the RFID B1 module there are 728 bytes of user accessible memory. Each byte of the memory has a defined factory default value and these values can be recovered by using the 'Reset to Factory Defaults' command. The first 288 bytes are volatile memory which also have a default reset state that is the same as the factory default value. The other memory is buffered non-volatile memory and can be modified and stored using 'Unlock' and 'Lock' commands. The RFID B1 module is automatically returned to factory default state if after power up there is no valid configuration stored in non-volatile memory. The first 288 bytes of this user accessible memory are not stored in non-volatile memory and are always reset to factory defaults after a power-up sequence or after exit from the Power Down Mode.



Address	Size (bytes)	Description	Notes
0x0000	1	Result	The Result Register values.
0x0001	1	Command	Command Register. Writing to this register is recognized by the module as a command execution request.
0x0002	18	Command Parameters	This is the place from where the system parses the arguments necessary to perform the requested operation when a command is executed.
0x0014	10	Tag UID	10-bytes long the Tag UID Register.
0x001E	1	Tag Type	This register contains information about the type of the tag which was last seen in the field. Possible tag types are shown in Table 3.5 in the RFID B1 User
0x001F	1	Tag UID Size	It contains the information of what the UID size in bytes was of the last tag in the field.
0x0020	256	Data Buffer	This buffer is used for data transfers between the tag and the user of the module.
0x0120	8	Password	This register is inaccessible when the module is locked. It contains an 8-byte long password which must be used with the Unlock Command to unlock protected memory.
0x0128	16	AES Initialization Vector 0	They can be used as an initialization vector for the first encrypted data block. These registers are inaccessible when the module is locked.
0x0138	16	AES Initialization Vector 1	
0x0148	16	AES Key 0	Both registers contain an AES encryption key which can be used for encryption of the Data Buffer. These registers are inaccessible when the module is locked.
0x0158	16	AES Key 1	
0x0168	6	Authentication Key / Password 0	When working with Mifare Classic tags these registers contain the password keys used for block authentication in the tag. When working with Ultralight and NTAG transponders, these registers contain 4-byte passwords. These registers are inaccessible when the module is locked.
0x016E	6	Authentication Key / Password 1	
. . .		. . .	
0x0252	6	Authentication Key / Password 39	
0x0258	128	User Memory	There are 128 bytes of memory available for the user as a protected memory space. This memory is inaccessible when the device is locked.

Table 2. Memory map of B1 based module - RFID section.

3.2.6 RFID Info

The most important information about a tag: the Type, the UID and the UID size. The following examples shows the result after execution the *Get UID and Type* command with and without a tag in the RF field.

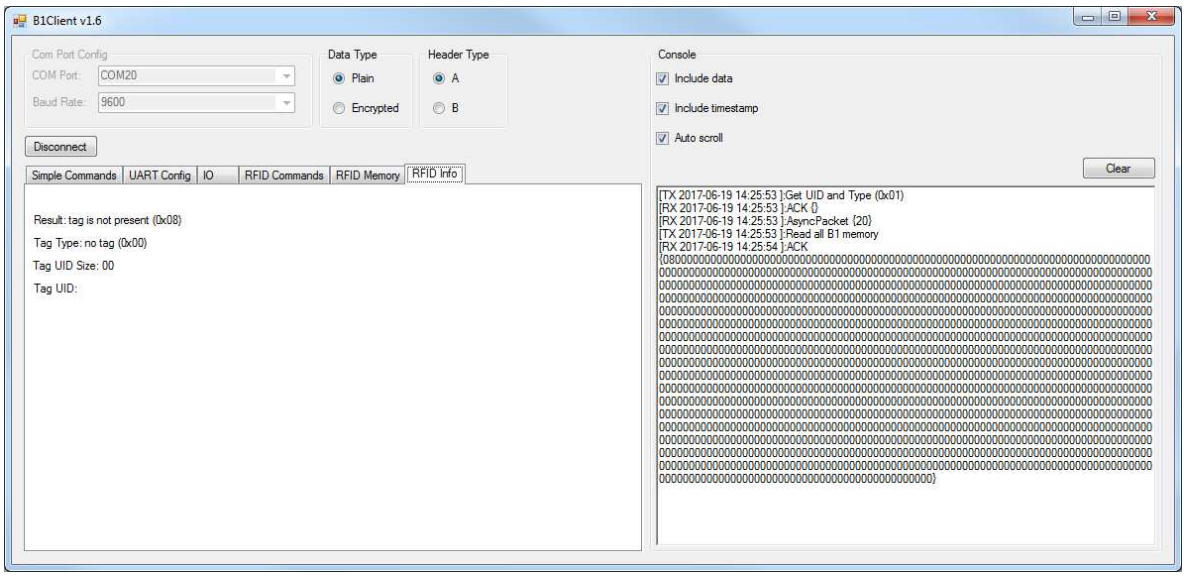


Figure 37. There is no tag in the field.

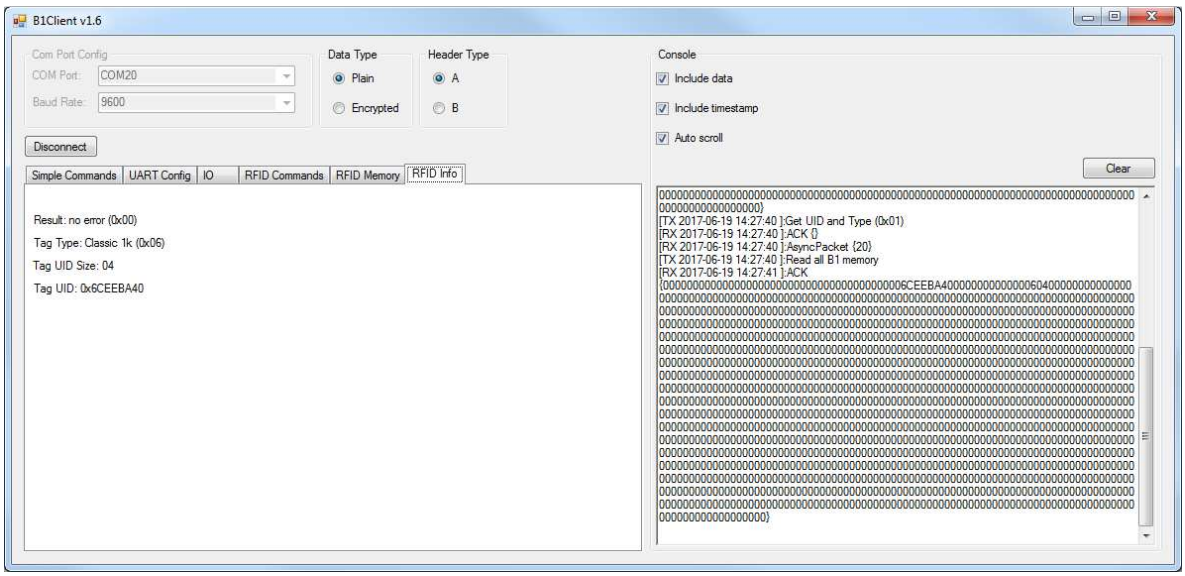


Figure 38. The Mifare Classic 1k tag in the field.

3.3 Output panel

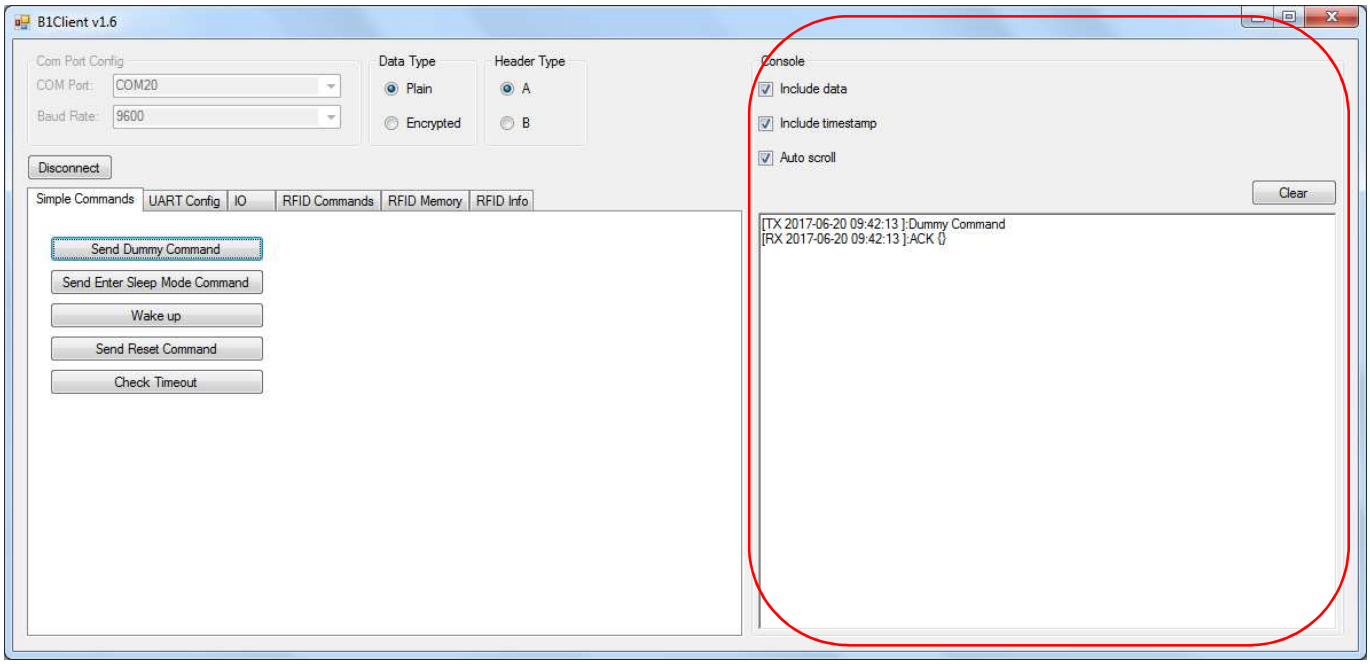
The Output panel is a console where the user can see the result of executed commands, responses from the B1 module for example: ACK or read full memory of a tag. This console is read only.

The B1 module provides an asynchronous UART communication interface. Communication is in a packetized command-response format which means that after every packet is sent from the master there will be a response packet sent back by the module. Apart from this type of user-master generated communication, the module also autonomously sends asynchronous packets when, for example an interrupt is triggered.

There are three checkboxes at the bottom of the Output panel:

- Include data – when checked all the tag memory will be shown on the console after executing any RFID command.
- Include timestamp – when checked the TX and RX commands will have a timestamp (actual data and time).
- Auto scroll – when checked the Console window scroll automatically.

By clicking the Clear button all the console window will be cleared.



4 Error codes

In the RFID Memory tab the user can read the Result Register. It contains the result (error code) of the last executed command. The list of all possible results is shown in table below.

Value	Type	Description
0x00	No Error	Command was executed successfully and results were stored in the registers.
0x01	Invalid Command	Value written to command register is invalid.
0x02	Invalid Command Parameter	One of the parameters taken by the command is invalid.
0x03	Indexes Out Of Range	Indexes passed as command parameters exceed limit.
0x04	Error When Writing To Non Volatile Memory	There was an internal error during writing to the non-volatile memory.
0x05	System Error	Internal system error. Shall be considered as fatal.
0x06	Tag CRC Error	During communication with the tag a CRC was not correct.
0x07	Tag Collision	Reserved for future use.
0x08	Tag is not present	There is no tag within range.
0x09	Tag Authentication Error	Authentication failed due to incorrect Authentication Key or Password.
0x0A	Tag Value Block Corrupted	At least one value block is corrupted in the tag memory.
0x0B	Module Overheated	A overheat was detected.
0x0C	Tag Not Supported	There is a tag in the field which is not supported.
0x0D	Tag Communication Error	There was an error during communication with the tag.
0x0E	Invalid Password	The Password used in the Unlock command string was invalid.
0x0F	Already Locked	You are trying to lock a module that is already locked.
0xFF	Module Busy	Your command was ignored because the module is busy. Retry later.

Table 3. Result Register Values.

No responsibility is taken for the method of integration or final use of the B1 based modules

More information about the B1 module and other products can be found at the Internet site:

http://www.eccel.co.uk

or alternatively contact ECCEL Technology (IB Technology) by e-mail at:

sales@eccel.co.uk