SERMA
PRODUCTIVITY ENGINEERING

Documentation

EVA3001U User Guide

Table of Content

1 Revision History

Version	Date	Changes	Page
Initial Version V1.0	04/2010		
V1.1	10/2010	Add Reader DESKID UHF	4, 6
V1.2	11/2010	New Layout	25
V1.3	01/2011	New Layout	3,25

2 Overview

The EVA3001U is especially designed to evaluate the PE3001 UHF RFID Integrated Circuit of Productivity Engineering GmbH featuring an integrated temperature sensor and real time clock as well as an 8kBit EEPROM for data monitoring. External sensors can also be incorporated with the help of a microcontroller through the SPI interface of the chip.

The PE3001 can also be used as a simple data storage system for ASCII data in the integrated EEPROM. This document is intended to describe the use of the EVA3001 to store ASCII data on the RFID chip. This application can be run without battery. A UHF reader is required.

The EVA3001 serves as a demonstrator and evaluation kit with these features:

- PCB with IC, LED(opt.), resistors(opt.) and battery
- passive RFID UHF transponder with „EPC Class 1 Generation 2” protocol interface
- recording of saving date and data in ASCII-Code on tag memory
- readout of all stored data via UHF-RFID interface (EPC Class1 Gen2)
- detailed representation of data in text file for detailed view in Windows OS

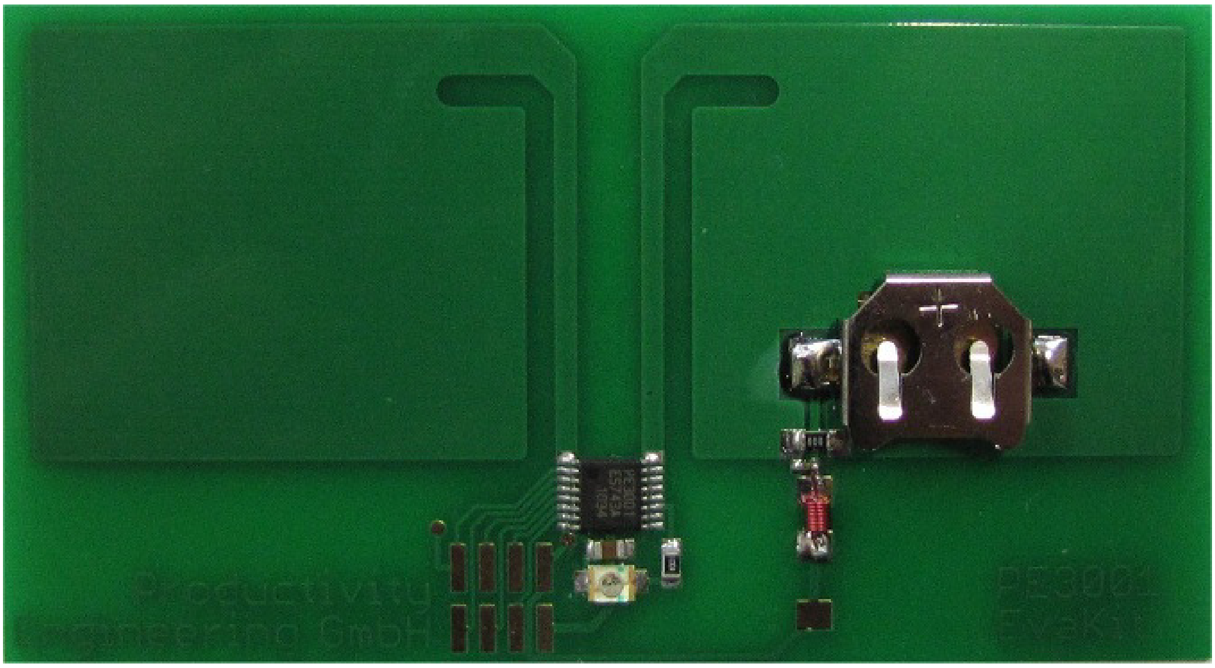


Figure 1 - Evaluation Kit PE3001

3 Controlling software „EVA3001U“

3.1 Installation

After downloading the software to manipulate the Data Monitor with a FEIG, SCEMTEC, Harting/Deister or CAEN reader it needs to be installed by executing the file **“Install_EVA3001U_1.0.exe”**.

The EVA3001U comes with a graphical user interface for WindowsXP ™ platforms. It is recommended to start with this simple software interface to learn about the basic functionality of the chip on the board.

All necessary files will be copied into the target directory and a group will be created in the “Start” Menu. Readers from other vendors might now work properly. Typically each reader requires its own USB driver interface. Driver installation can be done through standard software installation or through the “Start” Menu.

FEIG

To communicate with the FEIG reader it is necessary to install the USB Driver of the provider. The FEIG driver is in the package. For the operation of the software FEIG reader DLLs are required, which are automatically extracted to the destination directory. With these DLLs a FEIG reader MRU200 or LRU1000 can communicate through the USB or COM Port.

SCEMTEC

The driver interface for USB must be installed with the reader software from reader vendor.For the operation with SCEMTEC a DLL and special internal C++ functions are designed. With these functions a SCEMTEC reader SIH900 can communicate through the USB to COM conversion. The software automatically checks the devices connected to the ports and connects with the reader.

Deister/Harting

The driver interface for USB must be installed with the reader software from reader vendor. For the operation with Deister/Harting special internal C++ functions are designed. With these functions a Harting reader with Deister debus (UDL500) protocol can communicate through the USB to COM conversion. The software automatically checks the devices connected to the ports and connects with the reader.

CAEN

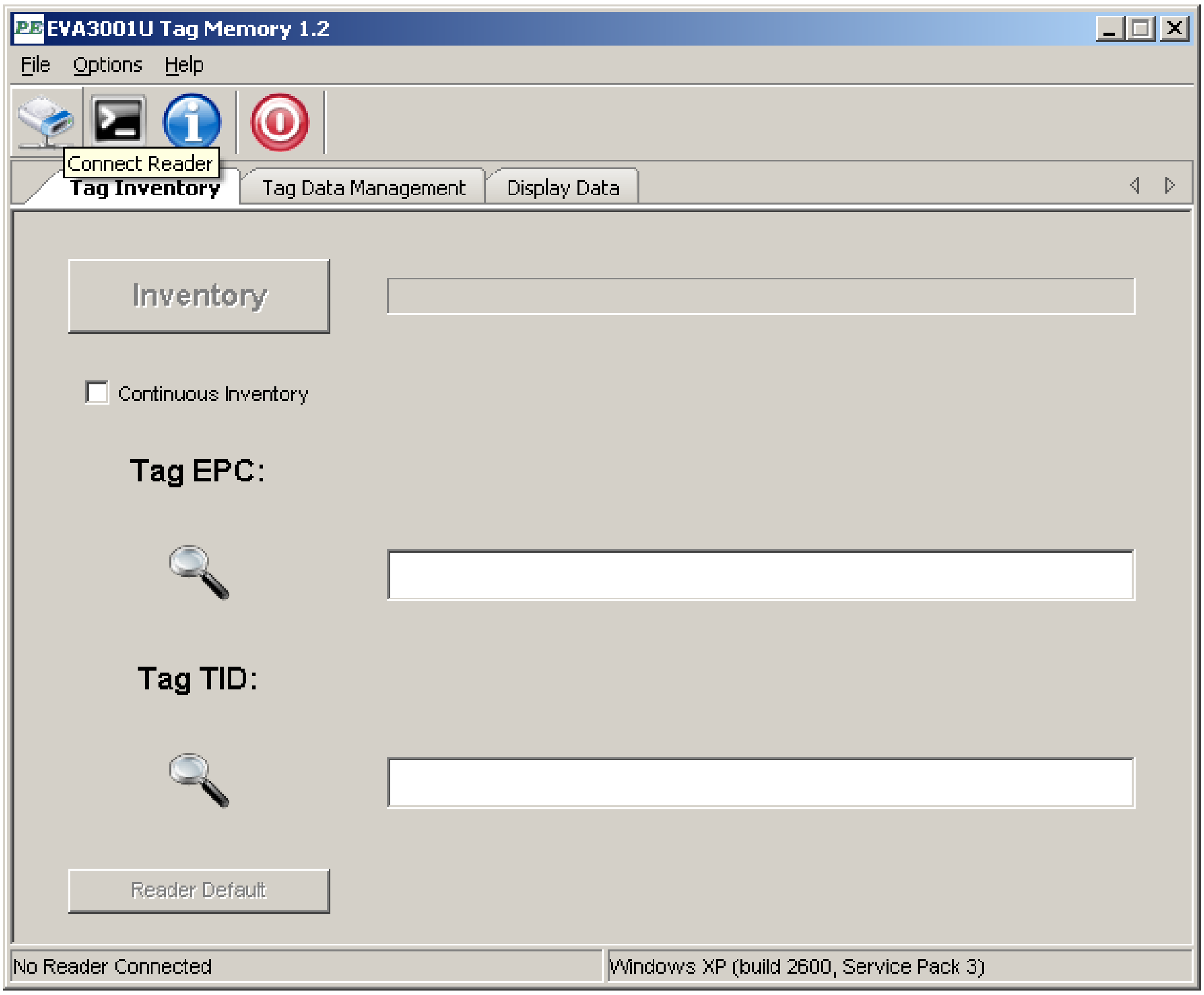
The driver interface for USB must be installed with the reader software from reader vendor. The CAEN reader (A828EU and A829EU) is implemented via DLL to an USB-Port for the C1G2 protocol. The software automatically checks the devices connected to the ports and connects with the reader.

Metratec

The driver interface for USB must be installed with the reader software from reader vendor. For the operation with Metratec special internal C++ functions are designed. With these functions the Metratec reader DESKID UHF can communicate through the USB to COM conversion. The software automatically checks the devices connected to the ports and connects with the reader.

3.2 Software description – Connect Reader

After starting the software (over “Start” Menu) and the access of an UHF-Reader the connection can be established. For this action the menu **"File"** and the point **"Connect Reader"** or the icon should be used.

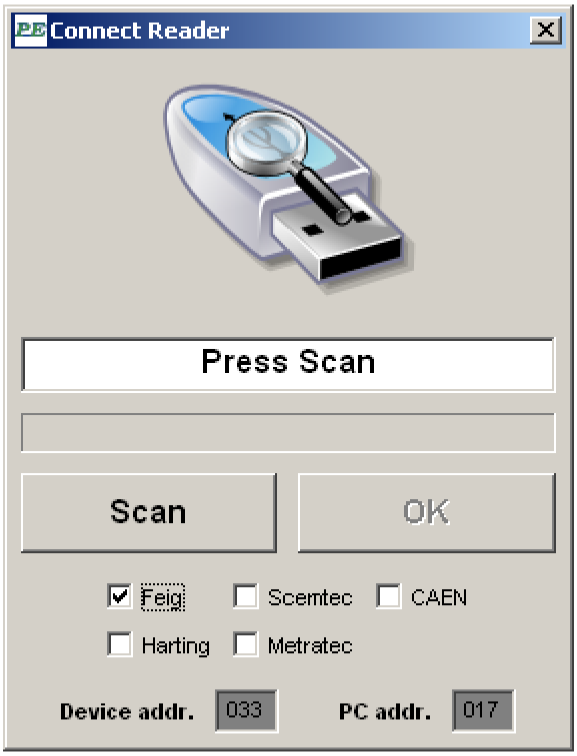


☐ Continuous Inventory

No Reader Connected

Windows XP (build 2600, Service Pack 3)

Please preselect the reader Firm and after pressing Scan it will now scan the ports for a connected reader. If it finds a valid reader the connection can be established by confirming with **"OK"**.



Device addr. 033

PC addr. 017

The “Device addr.” and “PC addr.” are decimal coded addresses. These text fields are necessary for the deBus protocol at Harting/Deister Reader. It is necessary to put before scanning the address to debus Device Address of Reader.

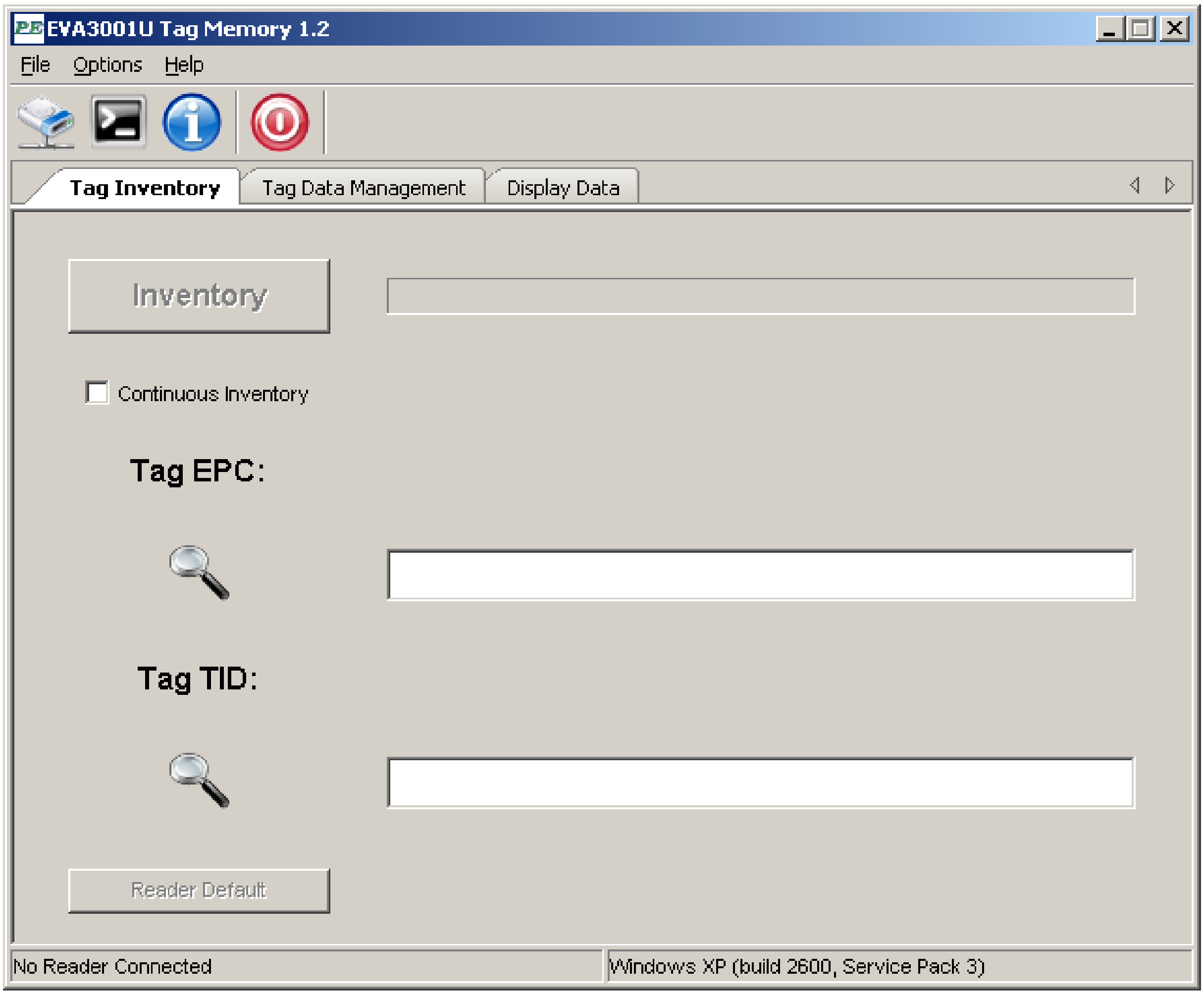
NOTICE for deBus protocol on Harting/Deister Reader:

1. Set Reader in Trigger Mode (use tool “deBus WebConfig” from Deister)
   - Tab Basic Setup: Operating mode → trigger mode
   - Apply Changes
   - Tab Trigger: select stop trigger after tag read
   - Apply Changes

2. Identify the used Reader address (use tool “RDemo” of Deister)
   - use “Quick device detection”
   - set device address with hexadecimal to decimal conversion in text field Device addr.

3. Set baud rate to 115200 bps (use tool “RDemo” of Deister)
   - Port→Setting→baud rate 115200

If no reader is found the Inventory button is not usable and the software is working without any tag information and activities.



□ Continuous Inventory

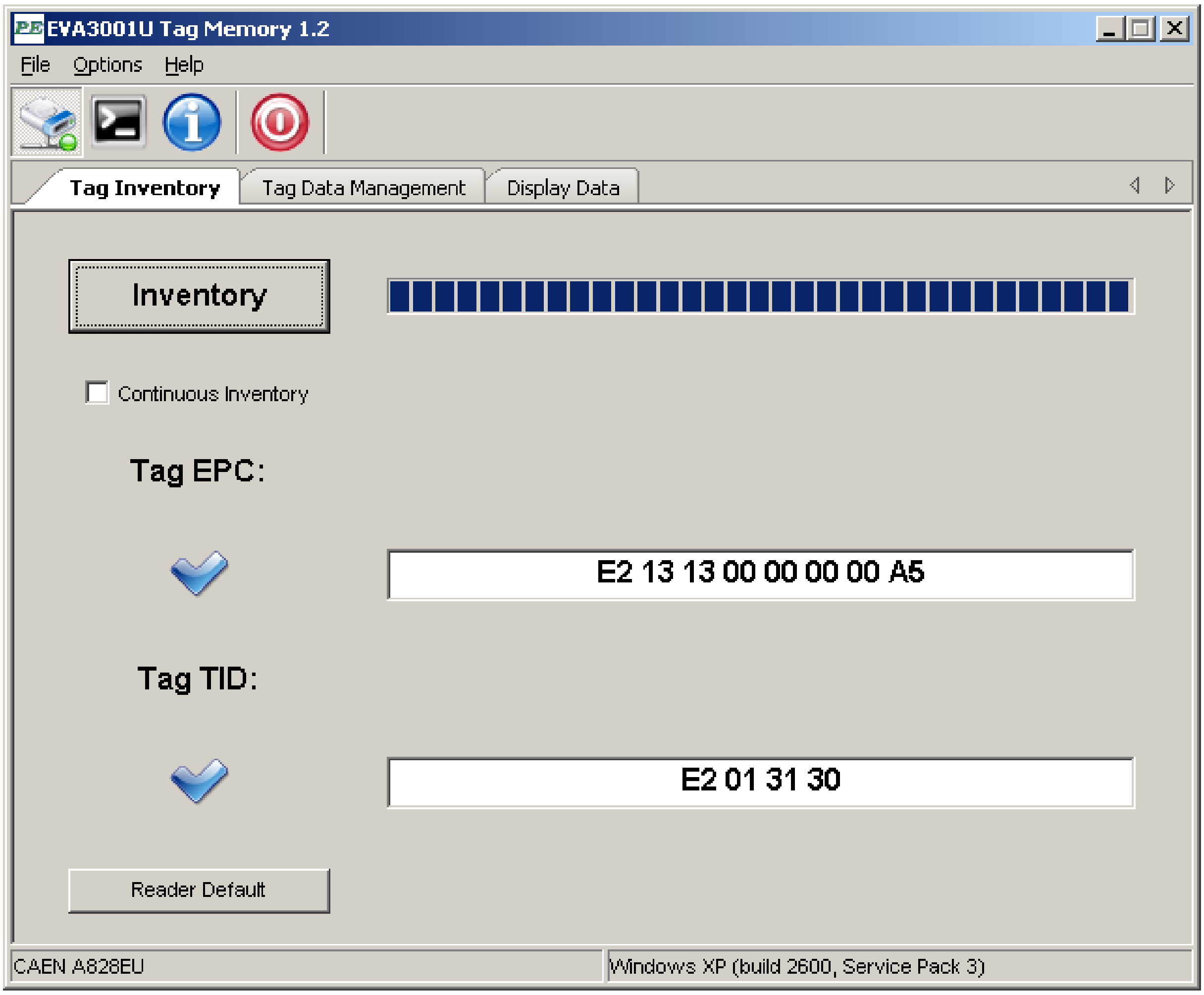
Tag EPC:

Tag TID:

No Reader Connected

Windows XP (build 2600, Service Pack 3)

After a successful reader connection the software can **"Inventory"** a tag for EPC and TID data.



EVA3001U Tag Memory 1.2

File Options Help

Tag Inventory | Tag Data Management | Display Data

Inventory

□ Continuous Inventory

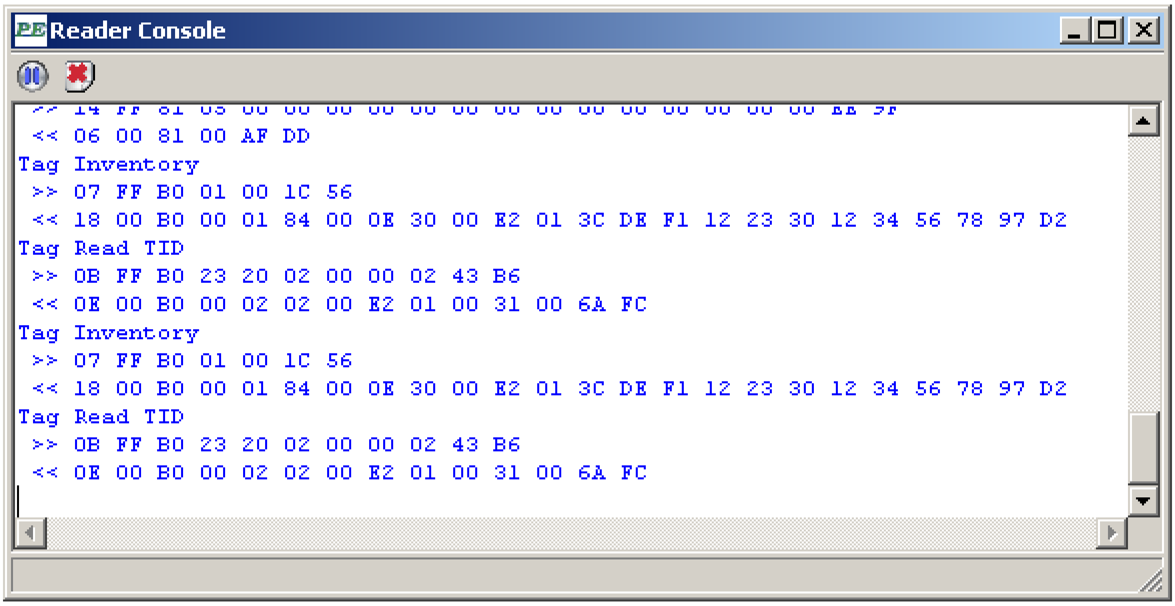
Tag EPC: ✓ E2 13 13 00 00 00 00 A5

Tag TID: ✓ E2 01 31 30

Reader Default

CAEN A828EU | Windows XP (build 2600, Service Pack 3)

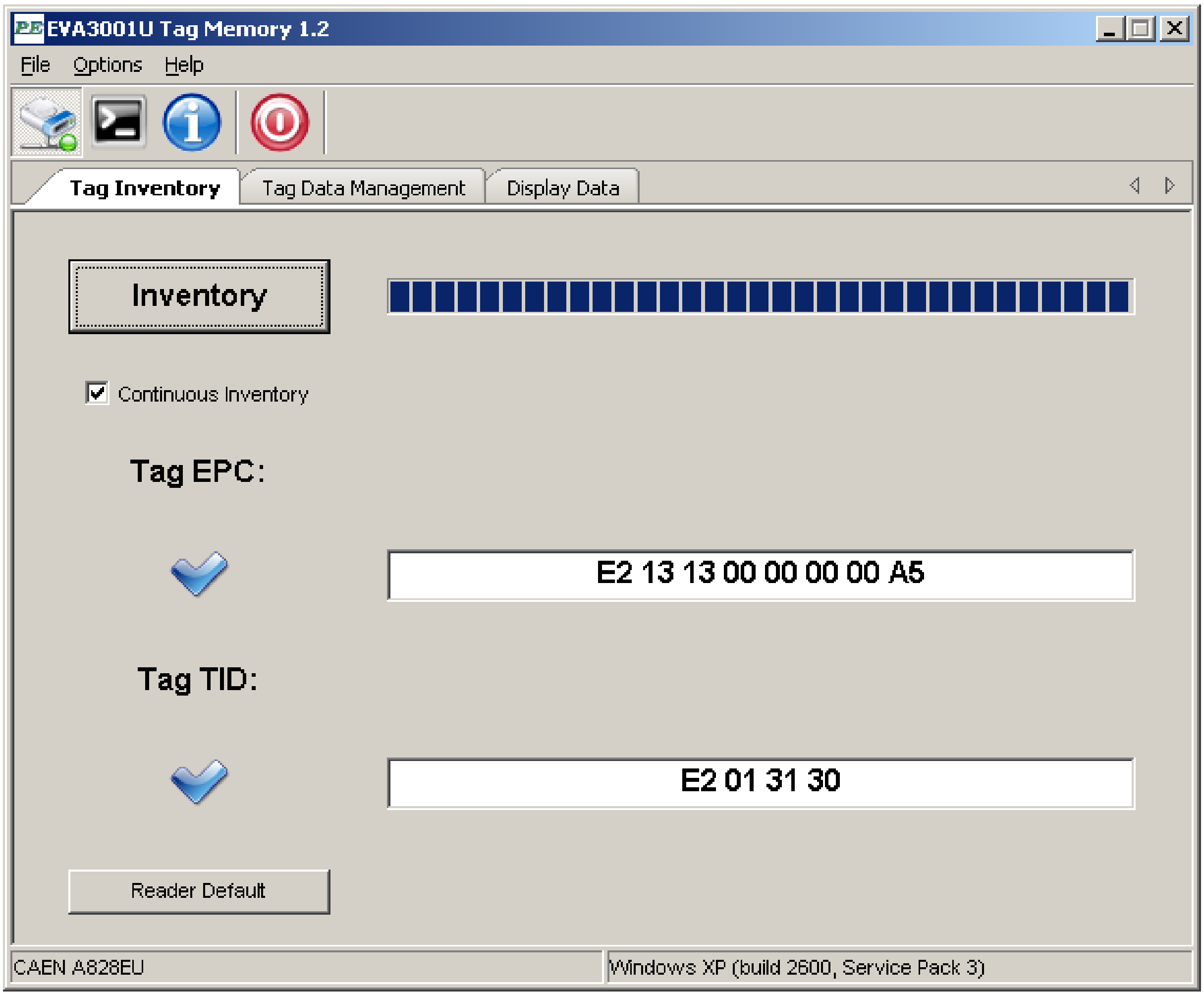
It is possible to use the Reader Console (menu **"OPTIONS"** -> **"Reader Console"** or the icon) to see the internal data transfer between software reader and tag.



Reader Console

```
 >> 14 FF 81 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 EE 9F
 << 06 00 81 00 AF DD
Tag Inventory
 >> 07 FF B0 01 00 1C 56
 << 18 00 B0 00 01 84 00 0E 30 00 E2 01 3C DE F1 12 23 30 12 34 56 78 97 D2
Tag Read TID
 >> 0B FF B0 23 20 02 00 00 02 43 B6
 << 0E 00 B0 00 02 02 00 E2 01 00 31 00 6A FC
Tag Inventory
 >> 07 FF B0 01 00 1C 56
 << 18 00 B0 00 01 84 00 0E 30 00 E2 01 3C DE F1 12 23 30 12 34 56 78 97 D2
Tag Read TID
 >> 0B FF B0 23 20 02 00 00 02 43 B6
 << 0E 00 B0 00 02 02 00 E2 01 00 31 00 6A FC
```

The checkbox **"Continuous Inventory"** select a permanently read with the button **"Inventory"**. In this way it is possible to find an optimal position between tag and reader. A stop of thes function is possible with deselecting checkbox **"Continuous Inventory"**.



☑ Continuous Inventory

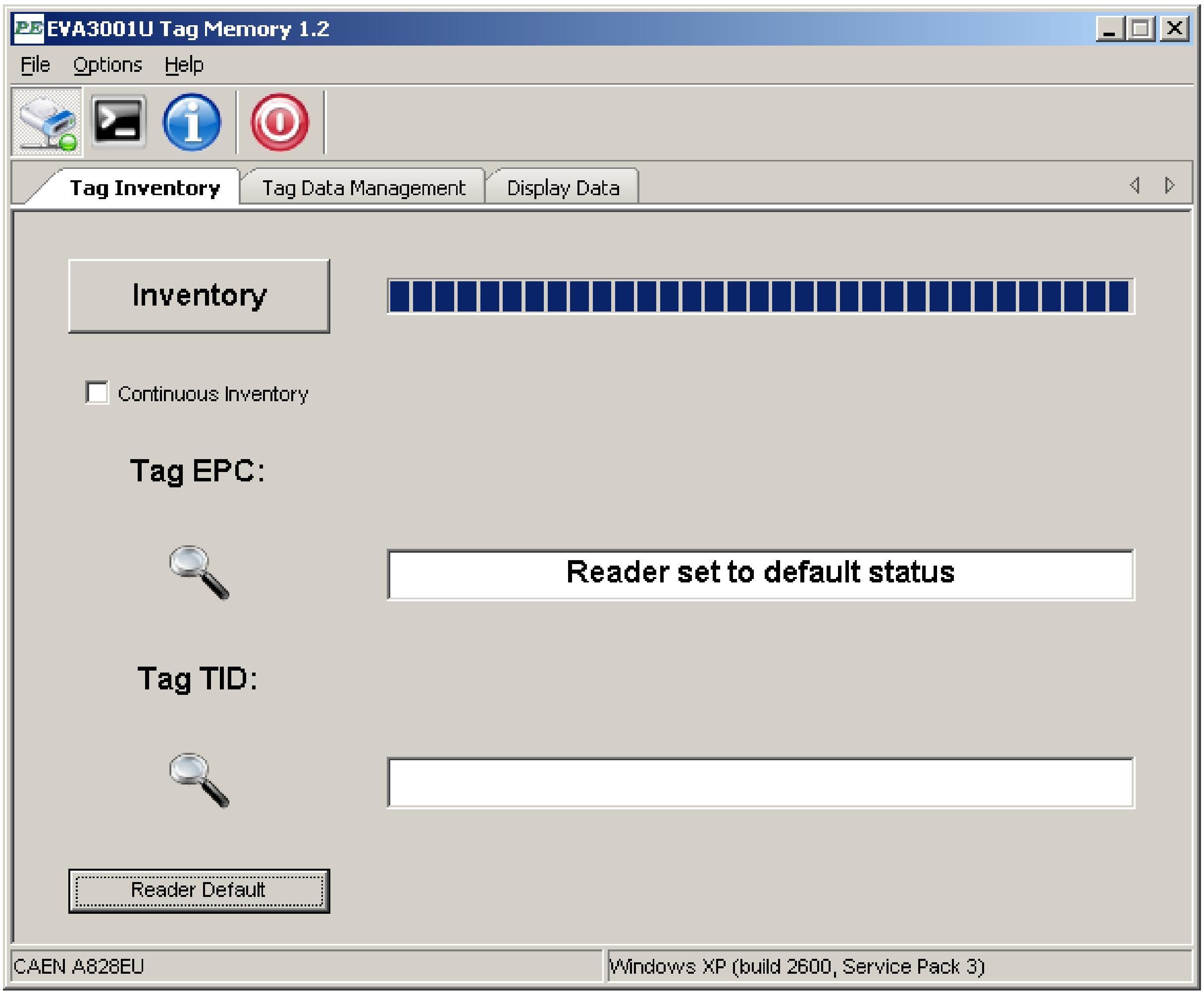
Tag EPC: E2 13 13 00 00 00 00 A5

Tag TID: E2 01 31 30

CAEN A828EU

Windows XP (build 2600, Service Pack 3)

The button **"Reader Default"** sets all reader specific flags to default configuration. This is necessary if the reader is not set to EPC Class1 Gen2 conformity.



□ Continuous Inventory

Tag EPC:

Reader set to default status

Tag TID:

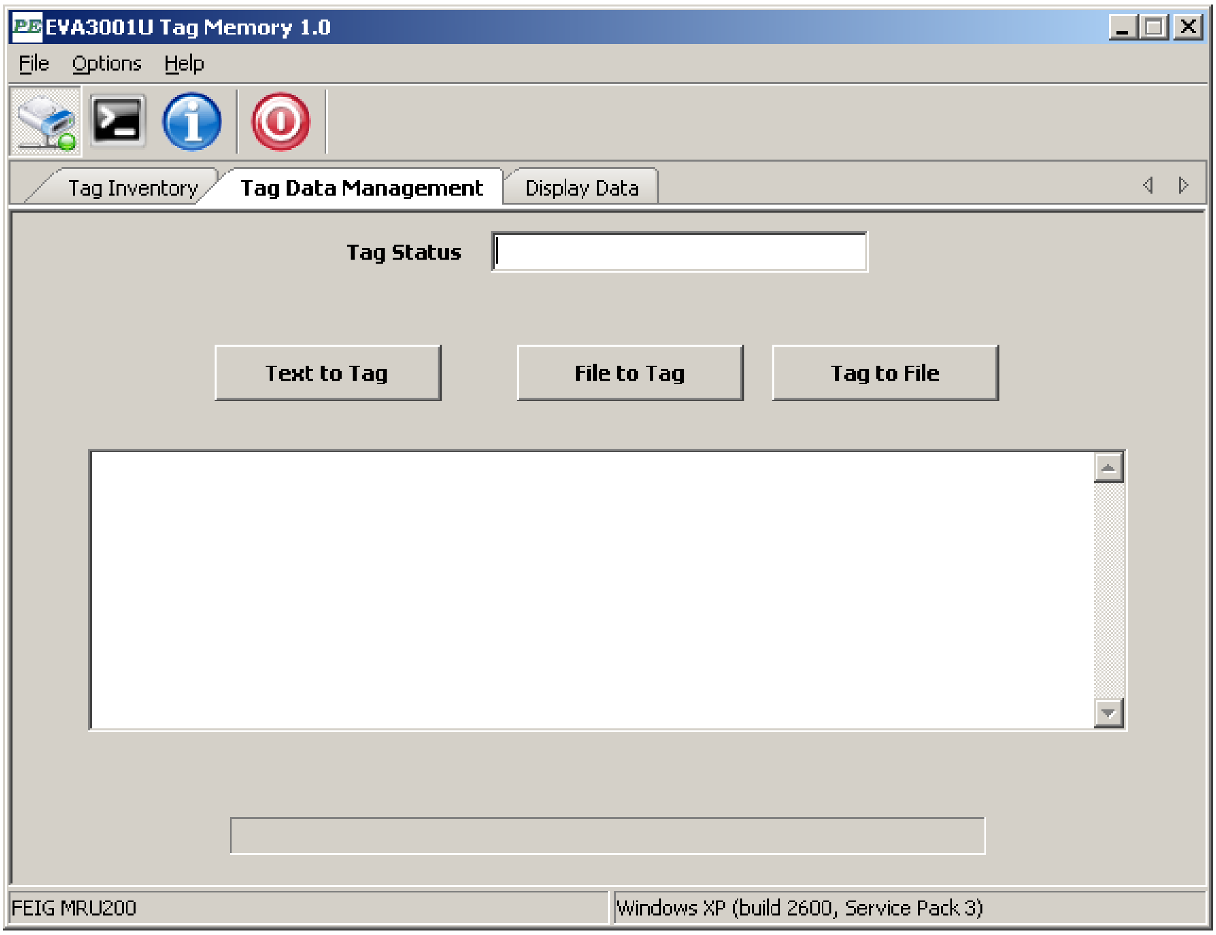
CAEN A828EU

Windows XP (build 2600, Service Pack 3)

This way all reader and tag pre-definitions are completed.

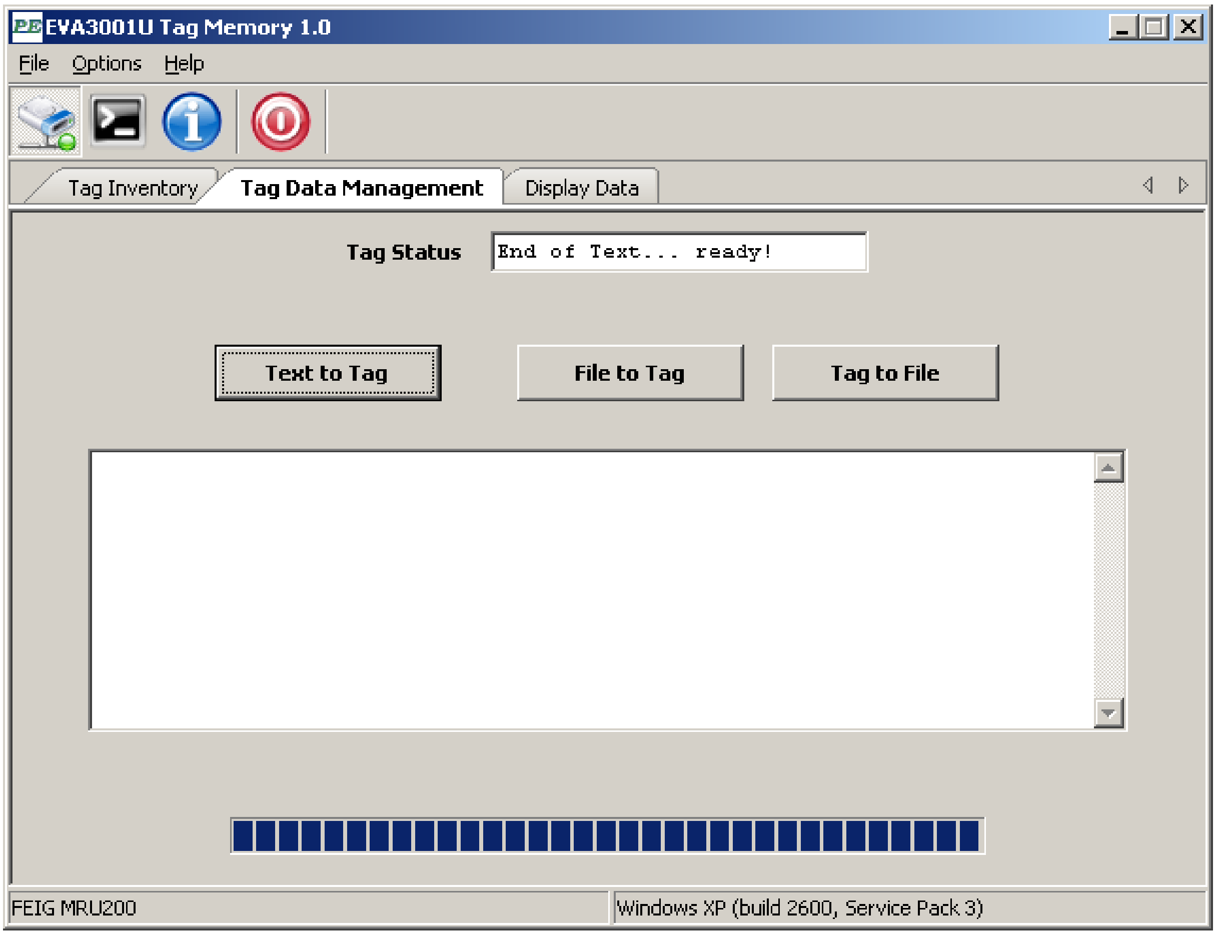
3.3 Software description – Tag Data Management

To read and manipulate the Data memory of the PE3001 tag the tab **"Tag Data Management"** can be used.



Tag Status

At first it is possible to see the field **"Tag Status".** This shows the current status of the tag in text field during any action on this tab.



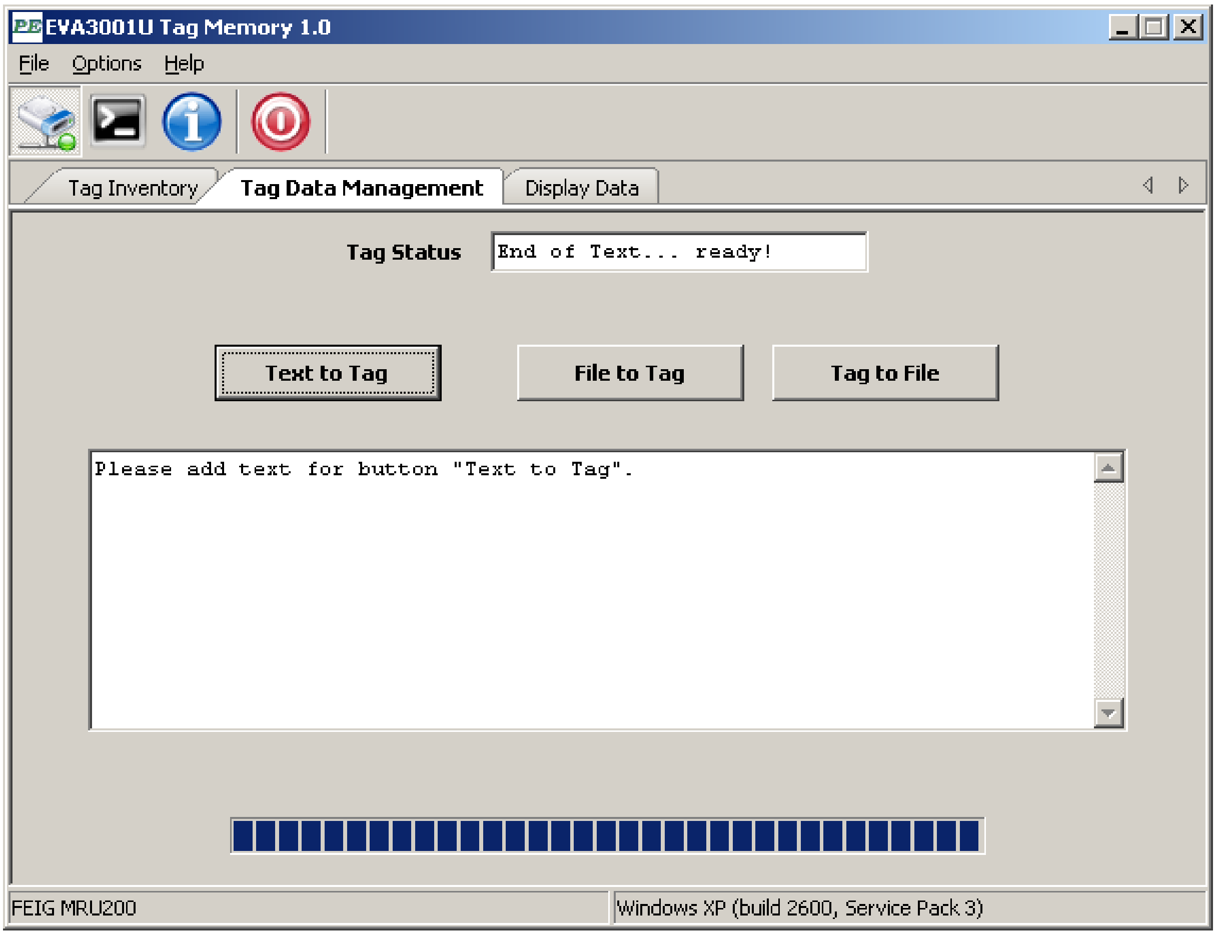
Tag Status: End of Text... ready!

FEIG MRU200

Windows XP (build 2600, Service Pack 3)

Text to Tag

The button **"Text to Tag"** saves the text from the text field below in the tag. After this event the PE3001 tag contains the saving date, the length of data and the data itself as ASCII-Code in the USER-Bank of the tag.



Tag Status: End of Text... ready!

Please add text for button "Text to Tag".

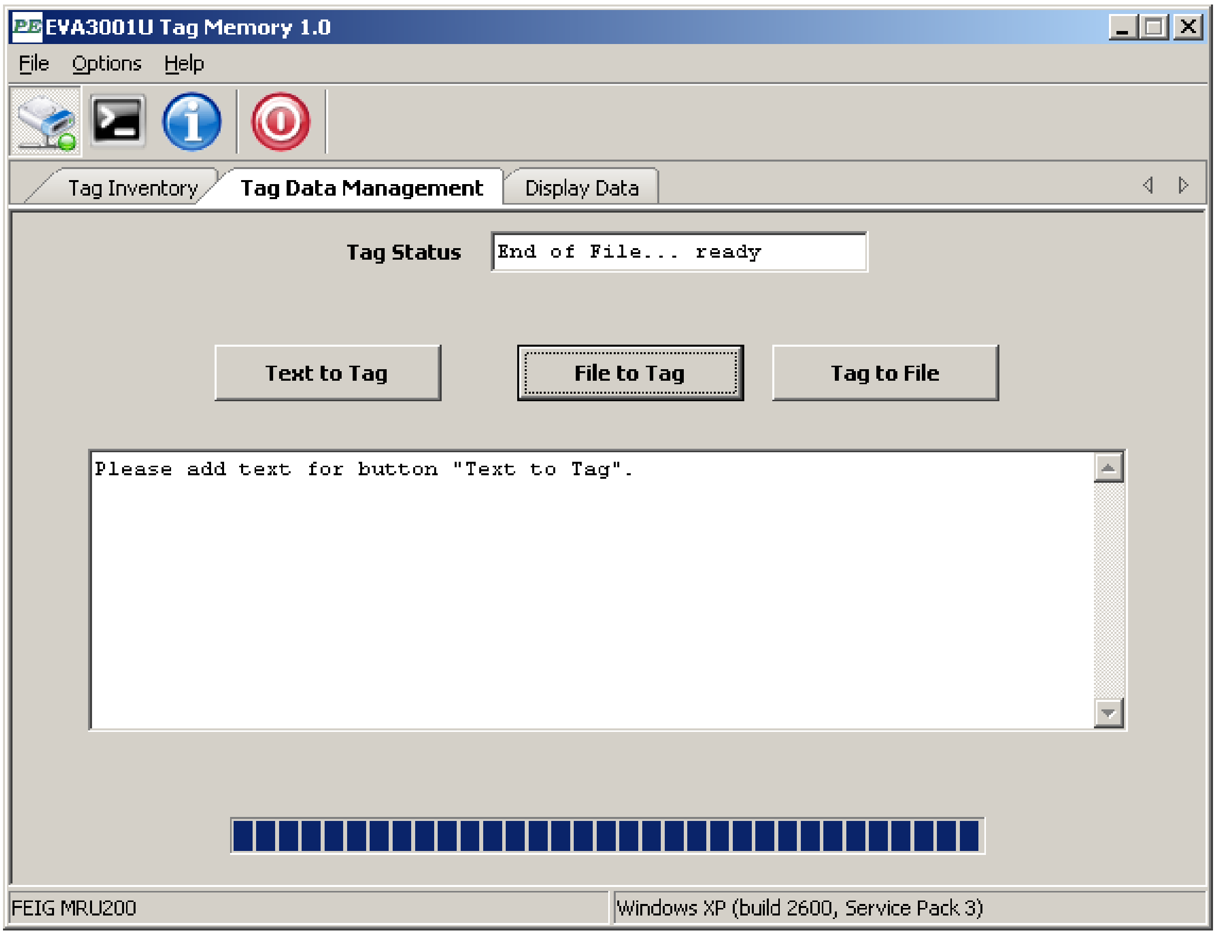
FEIG MRU200

Windows XP (build 2600, Service Pack 3)

Note: If the text is larger than the memory space the text will be truncated.

File to Tag

Push the button **"File to Tag"** to save data from text file in the tag memory.



Tag Status: End of File... ready

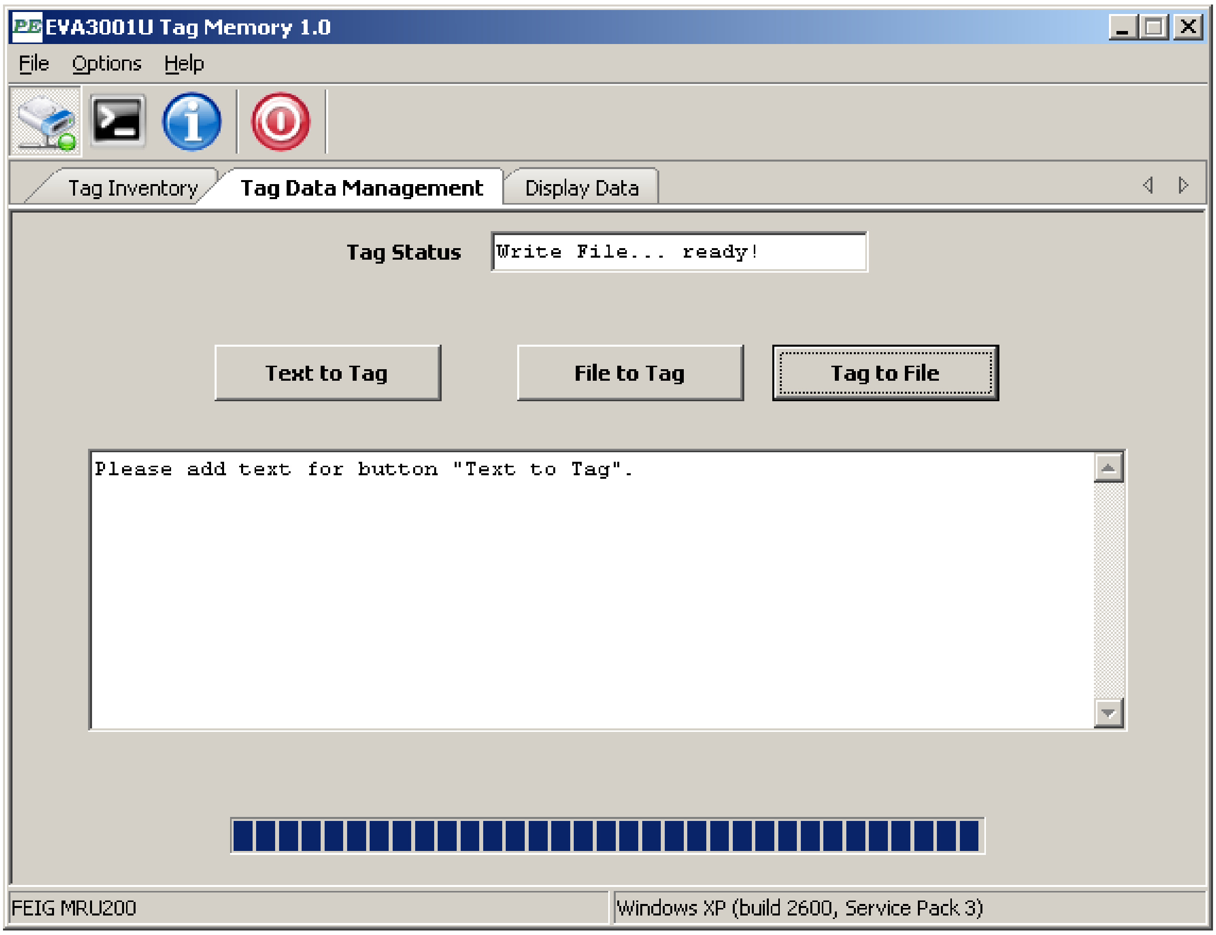
```
Please add text for button "Text to Tag".
```

FEIG MRU200 | Windows XP (build 2600, Service Pack 3)

Note: If the text is larger than the memory space the text will be truncated.

Tag to File

After successful saving of data it is possible to write detailed information to a text file in “UserData_YYYYMMDD_hhmmss.txt” format. Writing data to and reading back data from the PE3001 can also be done without any battery support.



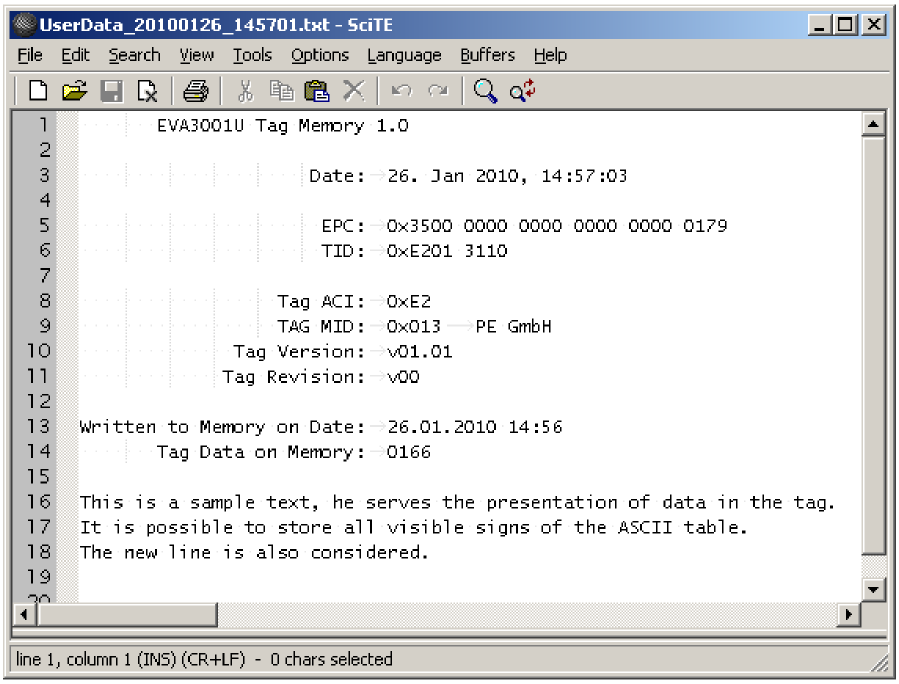
Tag Status: Write File... ready!

Please add text for button "Text to Tag".

FEIG MRU200

Windows XP (build 2600, Service Pack 3)

The current status of User Data is displayed below. The fields show the read values, the saving time and counted length of data. At the end of the file the saved ASCII data will be provided.


UserData_20100126_145701.txt - SciTE
File Edit Search View Tools Options Language Buffers Help

```
       EVA3001U Tag Memory 1.0

                     Date:  26. Jan 2010, 14:57:03

                      EPC:  0x3500 0000 0000 0000 0000 0179
                      TID:  0xE201 3110

                  Tag ACI:  0xE2
                  TAG MID:  0x013   PE GmbH
              Tag Version:  v01.01
             Tag Revision:  v00

Written to Memory on Date:  26.01.2010 14:56
       Tag Data on Memory:  0166

This is a sample text, he serves the presentation of data in the tag.
It is possible to store all visible signs of the ASCII table.
The new line is also considered.
```

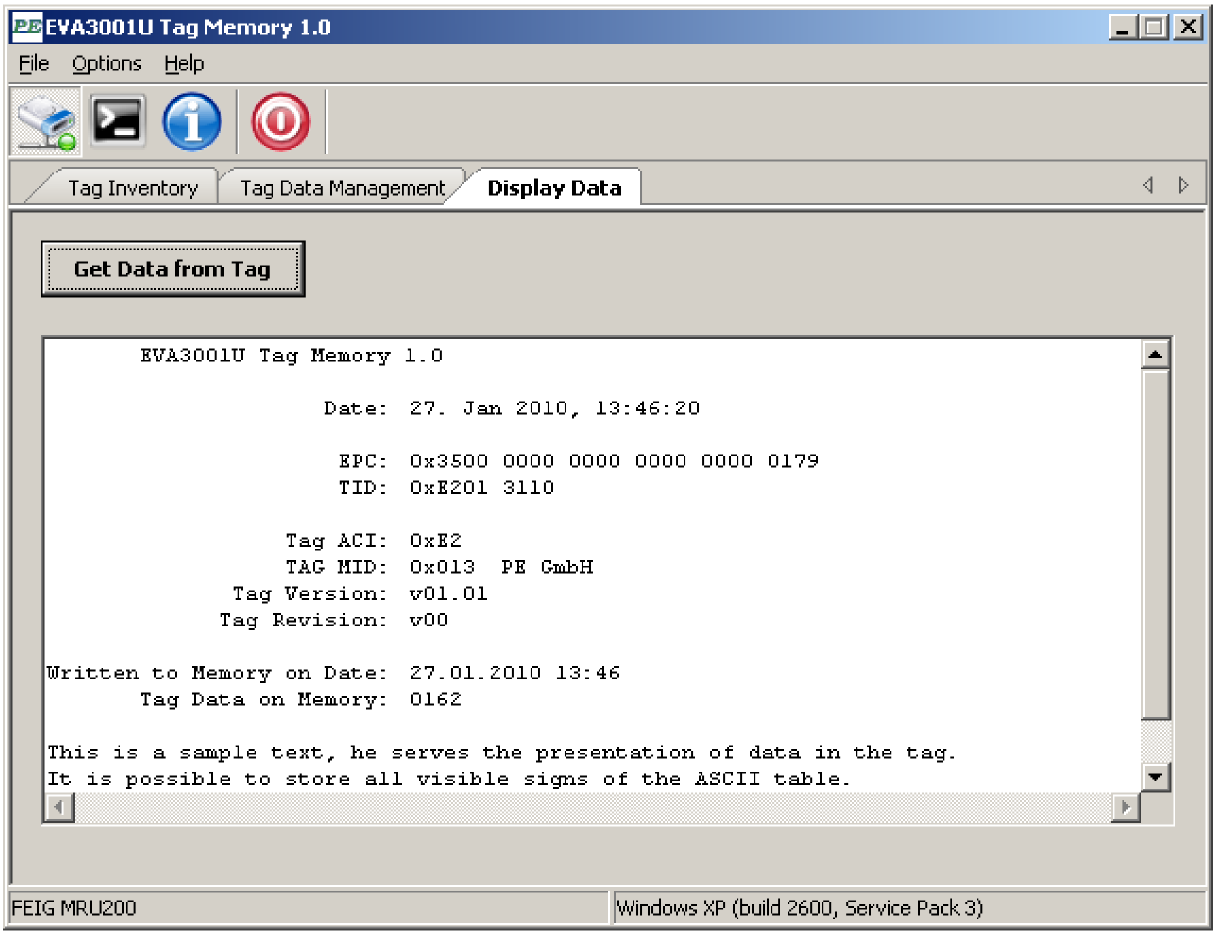
line 1, column 1 (INS) (CR+LF) - 0 chars selected

3.4 Software description – Display Data

This tab has been programmed to display the contents of the tag memory (EPC data, saving date, ASCII length) and the stored ASCII data.

Get Data from Tag

It is possible to use the button **"Get Data from Tag",** this clears the old data and shows the current content of the RFID tag.



```
       EVA3001U Tag Memory 1.0

                         Date:  27. Jan 2010, 13:46:20

                          EPC:  0x3500 0000 0000 0000 0000 0179
                          TID:  0xE201 3110

                      Tag ACI:  0xE2
                      TAG MID:  0x013  PE GmbH
                  Tag Version:  v01.01
                 Tag Revision:  v00

Written to Memory on Date:  27.01.2010 13:46
       Tag Data on Memory:  0162

This is a sample text, he serves the presentation of data in the tag.
It is possible to store all visible signs of the ASCII table.
```

4 Memory definition

The memory for the Data Monitor is in the RFID IC and is organized as specified in the EPC Gen2 standard. The monitor configuration and the trimming values are stored in the TID bank.

Table 1 -

EPC Bank	EPC Addr	EEPROM Block	Data
EPC Bank 01b	000h	000h	CRC16
	001h	001h	PC[15:0]
	002h	002h	EPC[95:80]
	003h	003h	EPC[79:64]
	004h	004h	EPC[63:48]
	005h	005h	EPC[47:32]
	006h	006h	EPC[31:16]
	007h	007h	EPC[15:0]
TID Bank 10b	000h	008h	TID[31:16]
	001h	009h	TID[15:0]
	002h	00Ah	Status / Log Counter
	003h	00Bh	Temperature Scaling
	004h	00Ch	Calibration Temperature
	005h	00Dh	TAG Trim
	006h	00Eh	RTC Trim
	007h	00Fh	TMS Offs
	008h	010h	TMS Trim
	009h	011h	Alarm Log Count
	00Ah	012h	Mean Value / Log Size (1AEh)
	00Bh	013h	Startup Time, Wait-Till-Log-Start
	00Ch	014h	Intervall Time
	00Dh	015h	Temp High Limit / Temp Low Limit
	00Eh	016h	CMD, Command data
	00Fh	017h	RSP, Response data
RESERVED Bank 00b Read Only!	000h	018h	KillPwd[31:16]
	001h	019h	KillPwd[15:0]
	002h	01Ah	AccessPwd[31:16]
	003h	01Bh	AccessPwd[15:0]
	004h	01Ch	Chip UID
	005h	01Dh	Chip UID
	006h	01Eh	Chip UID
	007h	01Fh	Chip UID
	008h	020h	Operation Cycle Counter
	009h	021h	Log Data Temp
USER Bank 11b	1DCh	022h	User Data
	…	…	…
	002h	1FDh	Length of Data
	001h	1FEh	Date at saving Data
	000h	1FFh	Date at saving Data

Generally all measured data of temperature sensor will be stored in RESERVED-Bank. In this application this function is not used. For this application the USER-Bank will be used. The structure is stored with saving date, text length and ASCII data. The edge between USER and RESERVED-bank can be changed by configuration byte LogSize. In this application the LogSize is defined as one.

473 addresses are reserved for recording text data in the memory. The total number of values to be stored is 473*2 = 946.

Note: PE GmbH delivers samples and production volume ICs in calibrated condition (RTC, TMS). The nature of the chip allows everybody to access AND change these values at any time as long as the TID bank is not locked. If the TID bank will be locked the data monitoring setup can not be changed any more. The values are trimmed in the test process during manufacturing and are guaranteed to be within specified limits.

4.1 Header for PE3001 Tag definition in C++

The complete source code is available from Productivity Engineering GmbH.

```
/***************************************************************
 * Name:    tagdefs.h
 * Purpose: tag defines
 **************************************************************/

#ifndef __tagdefs__
#define __tagdefs__

// memory bank organisation
#define BANK_EPC      0x01
#define BANK_TID      0x02
#define BANK_USER     0x03
#define BANK_RESERVED 0x00

// memory size
//#define MAX_MEMSIZE 0x200
#define MIN_LOGSIZE 0x000
#define DEF_LOGSIZE 0x001
#define DEF_LOGSIZESCEM 0x001

// memory bank size
#define MAX_EPC      0x008
#define MAX_TID      0x010
#define MAX_USER     0x1DD // user bank size (USER_SIZE = MAX_USER - LOGSIZE)
#define MIN_RESERVED 0x00A // reserved bank size (RESERVED_SIZE = MIN_RESERVED + LOGSIZE)
#define MAX_RESERVED 0x1E7

// memory addresses epc bank
#define EPC_CRC16 0x000
#define EPC_PC    0x001
#define EPC_EPC5  0x002
#define EPC_EPC4  0x003
#define EPC_EPC3  0x004
#define EPC_EPC2  0x005
#define EPC_EPC1  0x006
#define EPC_EPC0  0x007

// memory addresses tid bank
#define TID_TID1       0x000
#define TID_TID0       0x001
#define TID_LOGSTATUS  0x002
#define TID_TMPSCALING 0x003
#define TID_TAGTRIM    0x005
#define TID_TMSOFFSET  0x007
#define TID_TMSTRIM    0x008
#define TID_ACCELTHRES 0x009
#define TID_ALARM      0x009
#define TID_LOGMODE    0x00A
#define TID_STARTTIME  0x00B
#define TID_INTTIME    0x00C
#define TID_TEMPLIMIT  0x00D
#define TID_CMD        0x00E
#define TID_RSP        0x00F

// memory addresses user bank
#define USER_USERDATA 0x000
#define USER_STARTDATE0 0x000
#define USER_STARTDATE1 0x001
#define USER_DATALENGTH 0x002
#define USER_DATA       0x003

// memory addresses reserved bank
#define RESERVED_KILLPWD1   0x000
#define RESERVED_KILLPWD0   0x001
#define RESERVED_ACCESSPWD1 0x002
#define RESERVED_ACCESSPWD0 0x003
#define RESERVED_LOGDATA    0x008
```

```
// epc block PC cnt
#define GET_PCEPC(reg)      ((reg & 0x3800) >> 11)

// tid block version register
#define GET_TAGACI(reg)      ((reg & 0xFF00) >> 8)
#define GET_FIRMAMAJOR(reg)  ((reg & 0x00FF) >> 0)
#define GET_FIRMAMINOR(reg)  ((reg & 0xF000) >> 12)
#define GET_VERSIONMAJOR(reg)((reg & 0x0F00) >> 8)
#define GET_VERSIONMINOR(reg)((reg & 0x00F0) >> 4)
#define GET_REVISION(reg)    ((reg & 0x000F) >> 0)

// tid block status register
#define GET_LOGSTATUS(reg)   ((reg & 0xFE00) >> 9)
#define GET_LOGCOUNT(reg)    ((reg & 0x01FF) >> 0)
#define GET_LOGMODE(reg)     ((reg & 0xF000) >> 12)
#define GET_LOGSIZE(reg)     ((reg & 0x0FFF) >> 0)
#define GET_ALARMMONCNT(reg) ((reg & 0x0FFF) >> 0)
#define GET_MWSIZE(reg)      ((reg & 0xF000) >> 0)
#define GET_TEMPLIMITHI(reg) ((reg & 0xFF00) >> 8)
#define GET_TEMPLIMITLO(reg) ((reg & 0x00FF) >> 0)
#define GET_TEMPOFFSET(reg)  ((reg & 0xFF00) >> 8)
#define GET_TEMPSLOPE(reg)   ((reg & 0x00FF) >> 0)
#define GET_LOGWORD(reg)     ((reg & 0x3FFF) >> 0)

#define GET_ACTDATEH(reg)   ((reg & 0xFFFF0000) >> 16)
#define GET_ACTDATEL(reg)   ((reg & 0x0000FFFF) >> 0)
#define SET_ACTDATEH(reg)   (reg << 16)
#define SET_ACTDATEL(reg)   ((reg & 0x0000FFFF) >> 0)

// user block data (logsize = 0x1AE)
#define USER_START 0x000 // 48 blocks
#define USER_STOP  0x02F

// reserved block data (logsize = 0x1AE)
#define TEMP_START  0x009 // 384 blocks
#define TEMP_STOP   0x187

// status values
#define LOGSTATUS_STOPLOG     0x00
#define LOGSTATUS_STARTLOG    0x40
#define LOGSTATUS_ACTIVE      0x20
#define LOGSTATUS_MEMOVERFLOW 0x10
#define LOGSTATUS_ALARM       0x02
#define LOGSTATUS_OUTBAND     0x01
#define LOGSTATUS_BATT        0x04

// calibration values
#define CAL_TEMPSLOPEDIV  100.0 //
#define CAL_TEMPOFFSETDIV  1.0 //

#endif // __tagdefs__
```

5 Introduction of EVA3001 Hardware

Features PE3001:

- Passive RFID UHF transponder chip with integrated data monitor
- Compliance with EPC Class 1 Generation 2 (UHF RFID version 1.0.9)
- 8kBit EEPROM read-/writeable access about RF field and/or data monitor
- EEPROM memory for capture measurement data with time stamp or simple data saver
- Power supply about RF field (for communication) or with battery (data monitoring)
- Intelligent power management for different power domains
- Continuous battery control and automatic shutdown
- Password protection for monitor data according EPC Gen2 protocol
- Extra signal output for “out-of-limits” temperature detection
- Additional function with SPI interface (for using external devices)

5.1 PE3001 General Description

The PE3001 is an integrated circuit for tracking and controlling logistics. It monitors temperature and related time data of goods during transport or storage. While not in an RFID reader field and so not being supplied through the reader the system draws the required energy from the battery for a temperature measuring. While in a UHF reader field the system is supplied by the reader’s field energy and communicates to the reader based on the standard protocol. Besides standard EPC communication additional EPC functionality to read out temperature or other data is implemented. The integrated SPI interface allows for communication with other external devices like a microcontroller that can provide additional sensor functionality like for MEMS sensors.

Memory access is granted through the UHF reader controller as well as through the data monitoring controller. Both blocks have the same priority. No started memory access will be interrupted by a request for another access. The started access will be finished first before the new access request will be acknowledged. Detailed information can be found in the PE3001 datasheet.

6 How to design an application

6.1 Given PCB specification

The PCB (printed circuit board) setup in the original state has the following specification parameters. It consists of a network for antenna matching, a battery with corresponding connector to the PE3001 and an interface for SPI communication.

Supply voltage (battery)	1,5V / 3V

It is possible to assemble more devices for additional tasks. This implies an optical indication for an active alarm port and a possibility to boost the wireless communication range of the PE3001.

6.2 SPI Interface

When using a microcontroller with SPI it is possible use also external sensors. The measurement data of these sensors can be written to the EEPROM via the SPI and can be read out through the RFID interface.

Note: Refer to the datasheet of PE3001 for read/write commands to EEPROM via SPI.

6.3 Alarm Port

The alarm port becomes active, when user defined temperature limits are violated and a measurement cycle is running. Refer to the datasheet of PE3001 for an active alarm. For getting an optical indication when the alarm port is activated, it is necessary to assemble resistor R3 and the LED. The recommend data for R3 and LED can be found in the BOM.

When the alarm port gets activated, the signal is switched to ground and the LED illuminates.

Calculation for series resistor:

$$R_3 = \frac{V_{BAT} - V_F}{I_F}$$

For a red LED with low current the following parameters have to be accounted for:

Forward voltage V_F=1,8V , forward current I_F=2mA, battery voltage 3V:

$$R_3 = \frac{3V - 1{,}8V}{2mA} = 600Ohm$$

R_3=560Ohm shall be chosen here.

Note: The LED can only be applied, when using a 3V battery.

6.4 Boost Range

For getting a higher communication range with antenna, it is necessary to assemble and connect the connector JP2. In this case the passive RFID front end is supplied with battery voltage and draws less or no energy from the field.

7 Description

The Evaluation Kit was designed to help understand and evaluate the features of the PE3001 UHF RFID IC. Used external devices are standard components and do not represent a completely fine-tuned OEM application. The BOM (bill of material) for a final application may look different.

The PE3001 is a passive UHF RFID transponder circuit with an integrated data monitor.

7.1 Evaluation Board Specifications

Table 2 -

Parameter	Symbol	Min	Typ	Max	Unit	Notice
Operating temperature	T_{amb}	-40	27	80	°C	
Frequency RF field	f_{rf}	860		960	MHz	
Battery Voltage	V_{BAT}	1,1	1,4	1,7	V	Not possible if LED is used to indicate alarm port
				3,6	V	
Current Alarm Port (PE3001)	I_{Alarm}		5		mA	Open drain output

7.2 Wire bridges

The table presents an overview of required and optional jumper headers and their function. Status “close” mean that the pins of the jumper are connected through the header.

Table 3 -

PE3001	Notice
JP1	Closed: data monitor active (battery supply voltage)
JP2 (optional)	Closed: RFID front end supplied with battery voltage

7.3 Bill of Material (BOM)

Table 4 - Required devices

Board position	Amount	Value	Description Package	Notice
R1, R2	2	0Ohm	0805	Could be changed for a capacitor (antenna adjustment)
L1	1	0Ohm	0805	Resistor could be changed for an inductor (antenna adjustment)
PE3001	1	PE3001	SOIC16	
JP1	1	Header	2mm	Connecting battery voltage with PE3001
	1	Jumper	2mm	
BAT	1			SMD battery retainer

Table 5 - Optional devices (not assembled on Evaluation Kit)

Board position	Amount	Value	Description / Package	Notice
JP2	1	Header	2mm	Connecting battery voltage with RFID front end PE3001 (boost range)
	1	Jumper	2mm	
LED1	1		0805	V_F=1,8V; I_F=2mA; red
R3	1	560Ohm	0805	Refer to chapter 6.3

7.4 Schematic

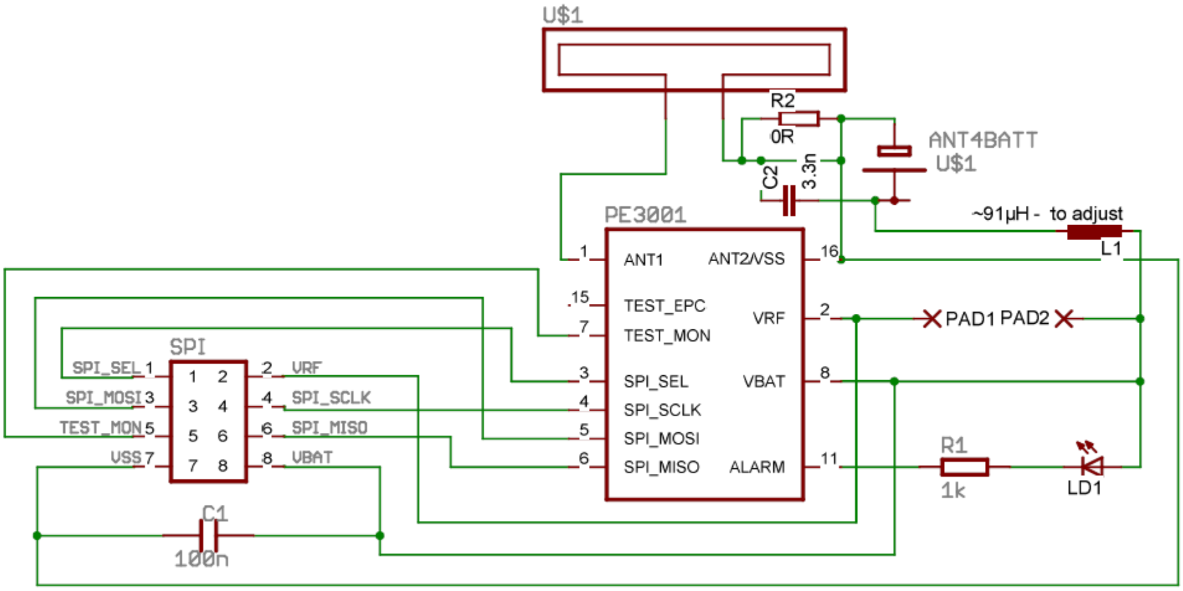


Figure 2 - Schematic of the Evaluation Kit

7.5 Layout

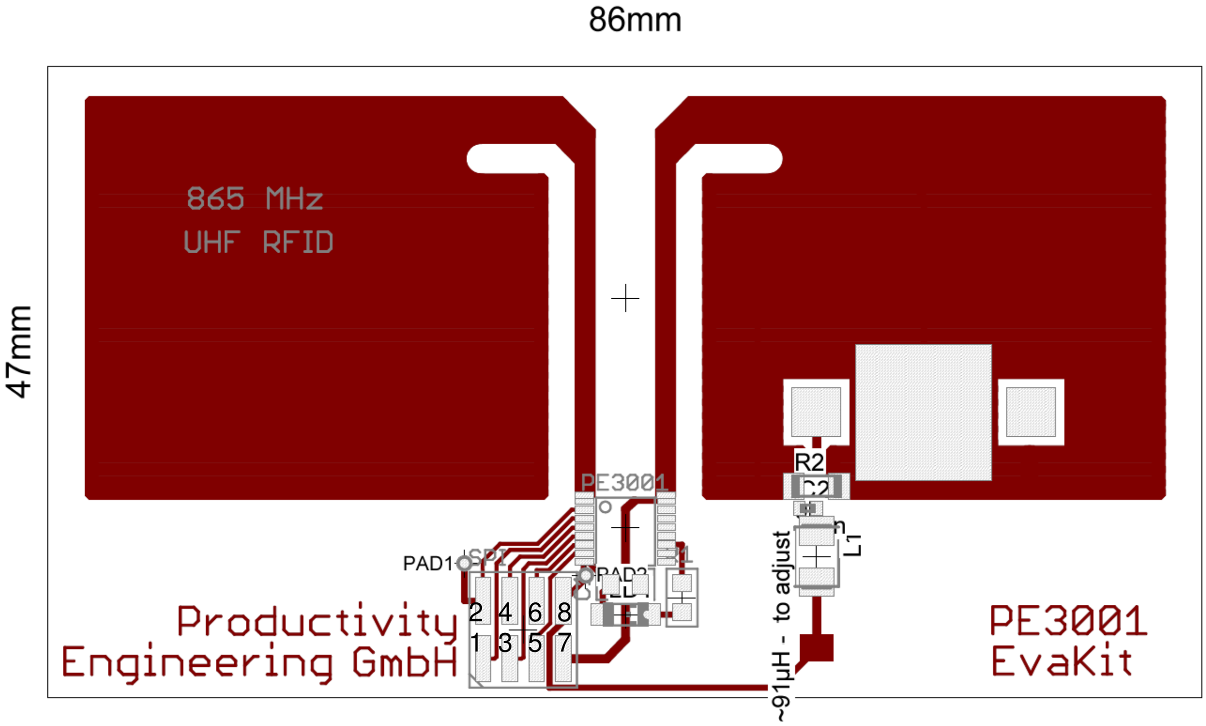


Figure 3 - Layout of the Evaluation Kit

7.6 Using only UHF interface

The UHF interface is accessible at any time on the EVA3001. The battery has not to be populated for this operational mode. JP1 can be closed or open. The PE3001 can be controlled with a UHF Reader. The UHF transmission distance can be extended in certain environments by attaching JP1 and JP2 so that the UHF functional electrical blocks have battery backup for operation. The antenna network is optimized for good electro-magnetic performance of the antenna with the chip. It is recommended to use the GUI3001 software to configure the PE3001 registers. The chip shall respond on any standard EPC Gen2 Reader by sending the UID.

The configuration and measurement commands are described in the PE3001 datasheet.

7.7 Using only data monitor

For using only the data monitor with temperature measurement the battery has to be mounted (default) and JP1 has to be closed.

The PE3001 can be controlled via the SPI interface or the RF field when the GUI3001 software will be used.

The configuration and measurement commands are described in the PE3001 datasheet.

7.8 Using UHF interface and data monitor

For using both UHF interface and data monitor the same requirements are necessary as described in chapter 7.7

This mode is recommended for temperature measurement tasks.

Accessing the EEPROM is possible in two ways (RFID interface and data monitor). Both have the same priority. If an access to the EEPROM occurs, then memory is locked until the access finished.

The configuration and measurement commands are described in the PE3001 datasheet.

8 Notes

9 Contact

Germany	Stuttgart	Dresden
	Productivity Engineering Process Integration GmbH Behringstrasse 7 D-71083 Herrenberg Germany Phone.: +49 (0) 70322798 0 Fax: +49 (0) 70322798 29 Email: info@pe-gmbh.com Web: www.pe-gmbh.com	Productivity Engineering GmbH Branch Sachsenallee 9 D-01723 Kesselsdorf Germany Phone.: +49 (0) 35204777 00 Fax: +49 (0) 35204777 000 Email: info@pe-gmbh.com

Important Notice

Productivity Engineering GmbH (PE) reserves the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to PE's terms and conditions of sale supplied at the time of order acknowledgment. PE warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with PE's standard warranty. Testing and other quality control techniques are used to the extent PE deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed. PE assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using PE components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards. PE does not warrant or represent that any license, either express or implied, is granted under any PE patent right, copyright, mask work right, or other PE intellectual property right relating to any combination, machine, or process in which PE products or services are used. Information published by PE regarding third–party products or services does not constitute a license from PE to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from PE under the patents or other intellectual property of PE. Resale of PE products or services with statements different from or beyond the parameters stated by PE for that product or service voids all express and any implied warranties for the associated PE product or service and is an unfair and deceptive business practice. PE is not responsible or liable for any such statements.