SERMA
PRODUCTIVITY ENGINEERING

Documentation

EVA3001 TEMP User Guide

Table of Content

1 Revision History

Version	Date	Changes	Page
Initial Version V1.0	10/2011		

2 Overview

The Ref.Des. PE3001 has been designed to evaluate the PE3001 UHF RFID Integrated Circuit of Productivity Engineering GmbH as a wireless battery-less temperature sensor system.

This document is intended to describe the use of the EVA3001 to store temperature data without a battery. A UHF reader is required.

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

- PCB with IC, capacitors and diode
- passive RFID UHF transponder with „EPC Class 1 Generation 2" protocol interface
- measurement and storage of temperature data in tag memory without any battery support
- readout of all stored data via UHF-RFID interface (EPC Class1 Gen2)

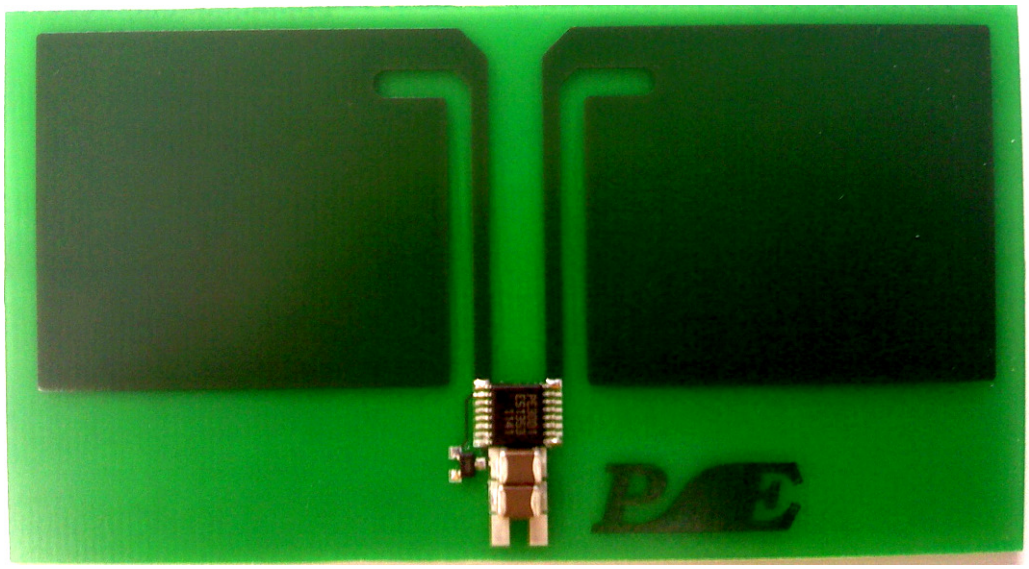


Figure 1 – Wireless battery-less reference design for PE3001

3 Graphical user interface for the „Ref.Des. PE3001“

3.1 Installation

Download the software from the web site www.pe-gmbh.com. It allows to manipulate the Data Monitor with a FEIG, SCEMTEC, Harting/Deister or CAEN reader.
Execute the file **“Install_EVA3001_TEMP_1.0.exe”**.

The Ref.Des. PE3001 requires a graphical user interface for WindowsXP ™ plattforms. It is recommended to start with this simple software interface to learn about the basic functionality of the chip on the board for this specific application.

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

FEIG

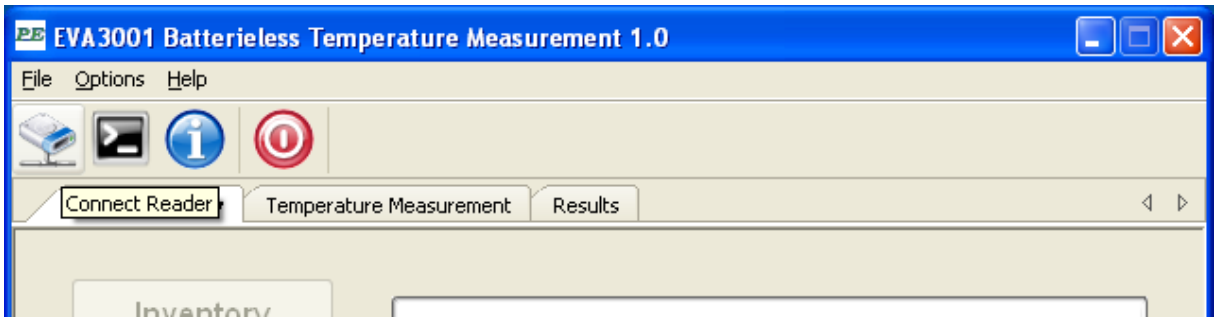
To communicate with the FEIG reader it is necessary to install the USB Driver of the provider. The FEIG reader driver is in the package. For operation of the software FEIG reader DLLs are required which are automatically extracted to the destination directory. With these DLLs a FEIG reader MRU200 or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

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.

3.2 Software description – Connect Reader

After starting the software (in “Start” Menu) and assumed physical connection of a UHF-Reader the connection can be established. For this action the menu **"File"** and the sub menu **"Connect Reader"** or the icon has to be used.



Please preselect the reader company name and after pressing Scan it will now scan the ports for a connected reader. When a valid reader has been detected the connection can be established by confirming with **"OK"**.

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

NOTICE for deBus protocol on Harting/Deister Reader:

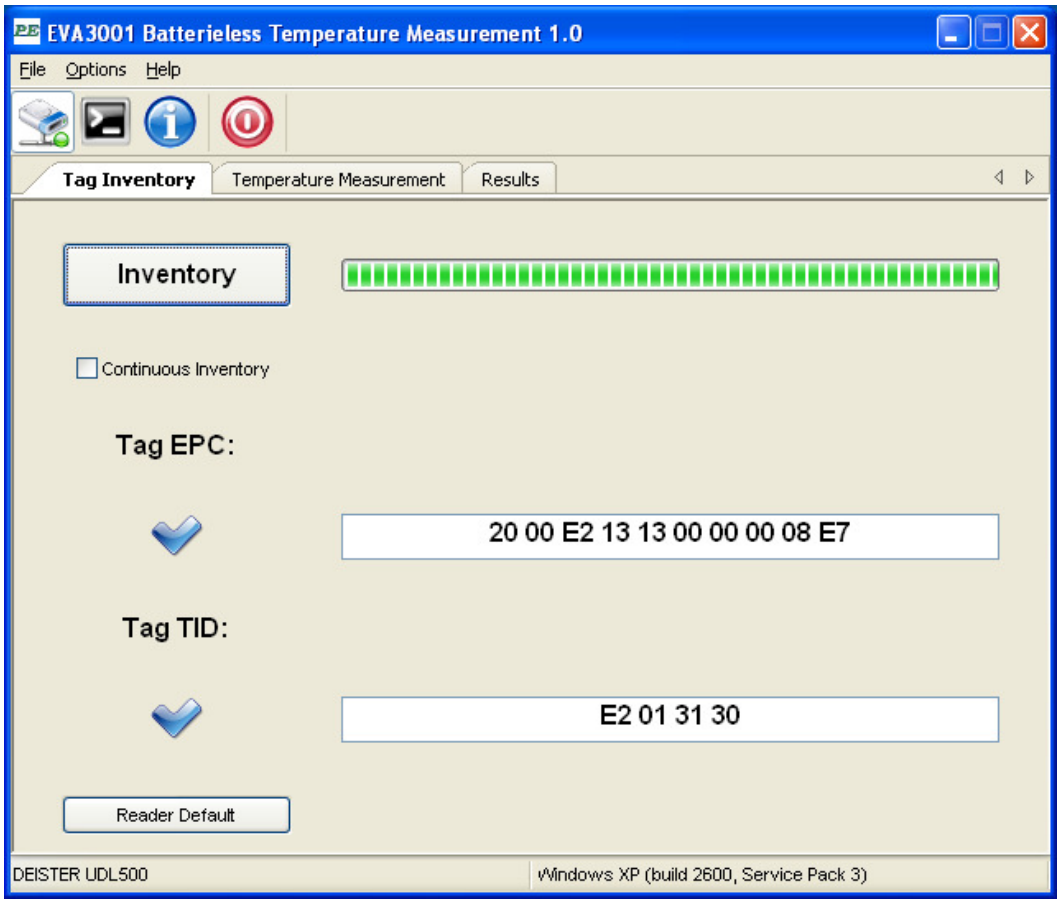
1. Set Reader in Trigger Mode (use tool “deBus WebConfig” from Deister)
   - Tab Basic Setup: Operating mode → trigger mode
   - Apply Changes
   - Tab Trigger: select stop trigger after tag read
   - Apply Changes

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.

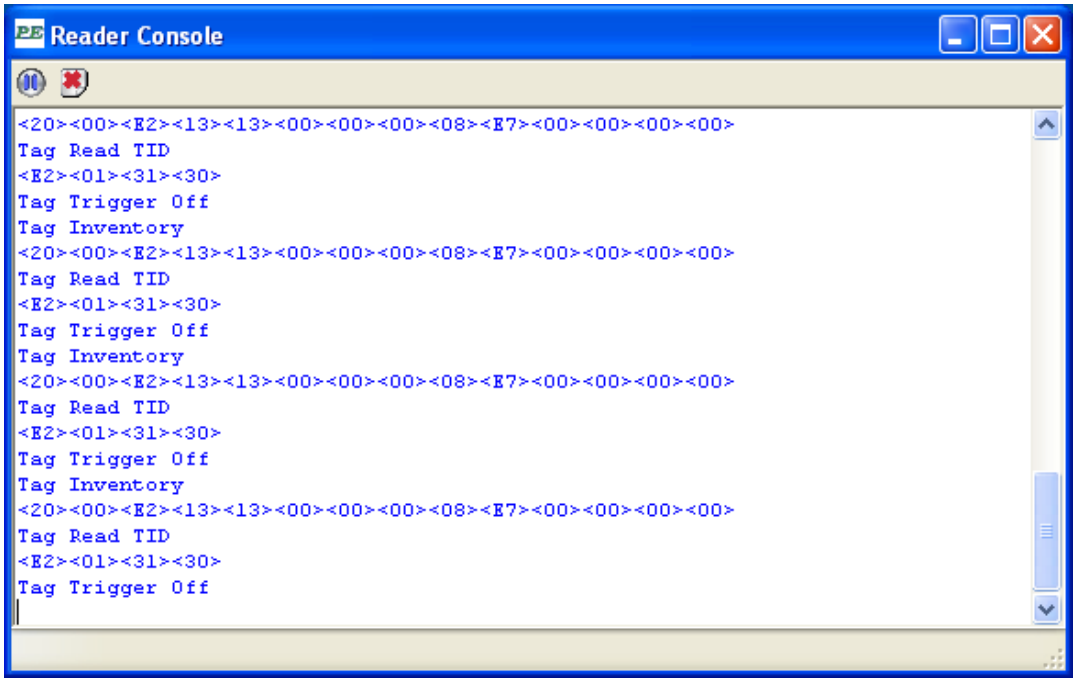
3. Set baud rate to 115200 bps (use tool “RDemo” of Deister)
   - Port→Setting→baud rate 115200

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

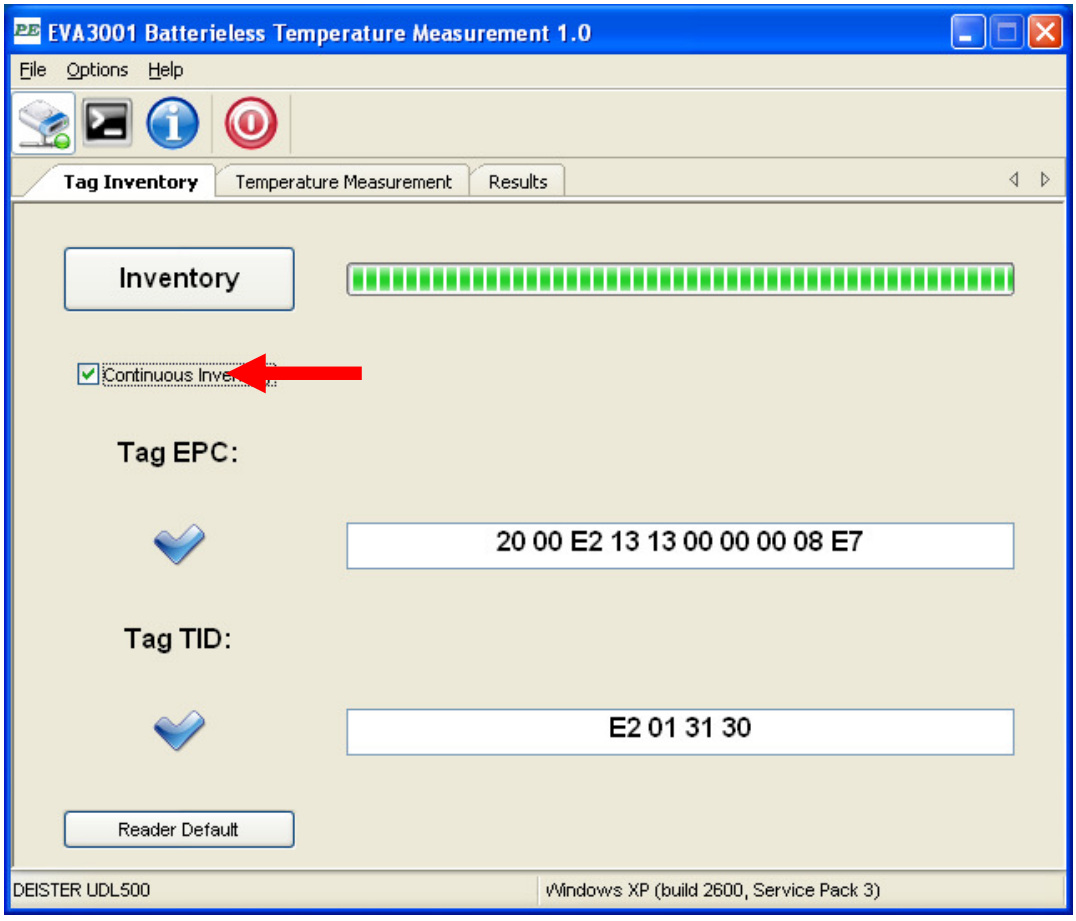
After a successful reader connection the software can **"Inventory"** a tag for EPC and TID data.



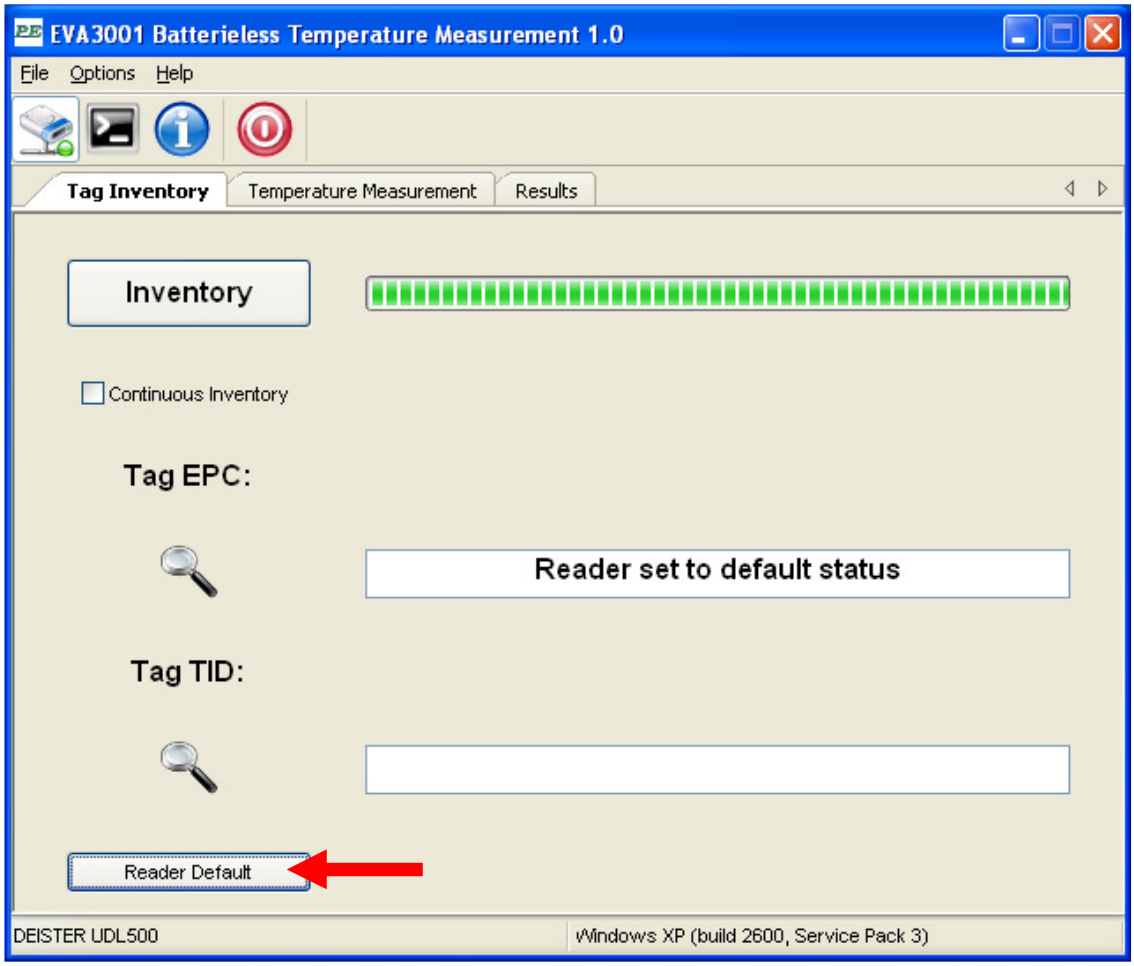
It is possible to use the Reader Console (menu **"OPTIONS"** -> **"Reader Console"** or the icon) to see the internal data transfer between software reader and tag.



The checkbox **"Continuous Inventory"** selects a permanent read with the button **"Inventory"**. This way it is possible to find the optimal position between the tag and the reader. Deselecting the checkbox **"Continuous Inventory"** stops this function.



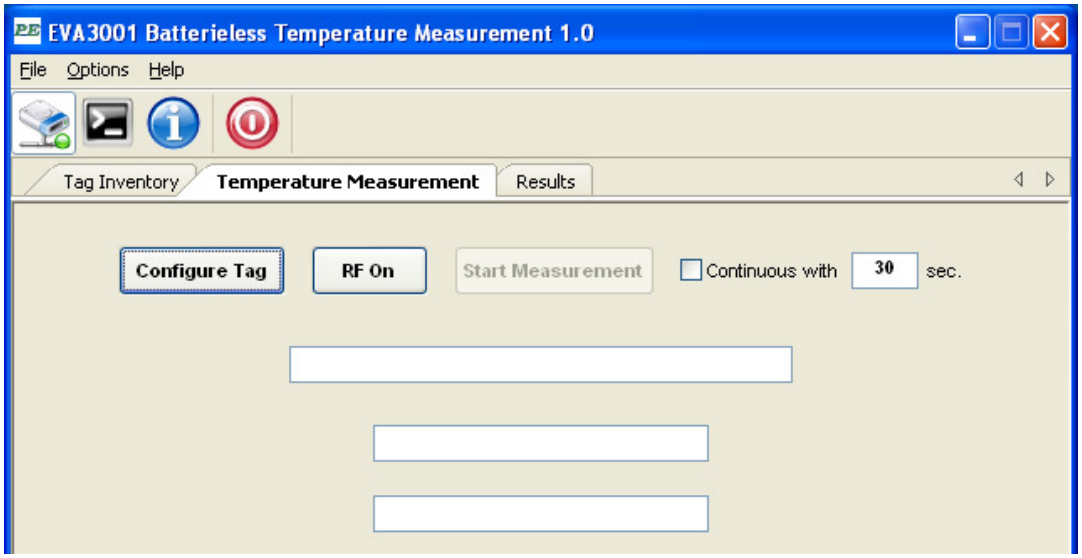
The button **"Reader Default"** resets all reader specific flags to the default configuration. This is necessary if the reader is not set to EPC Class1 Gen2 conformity.



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

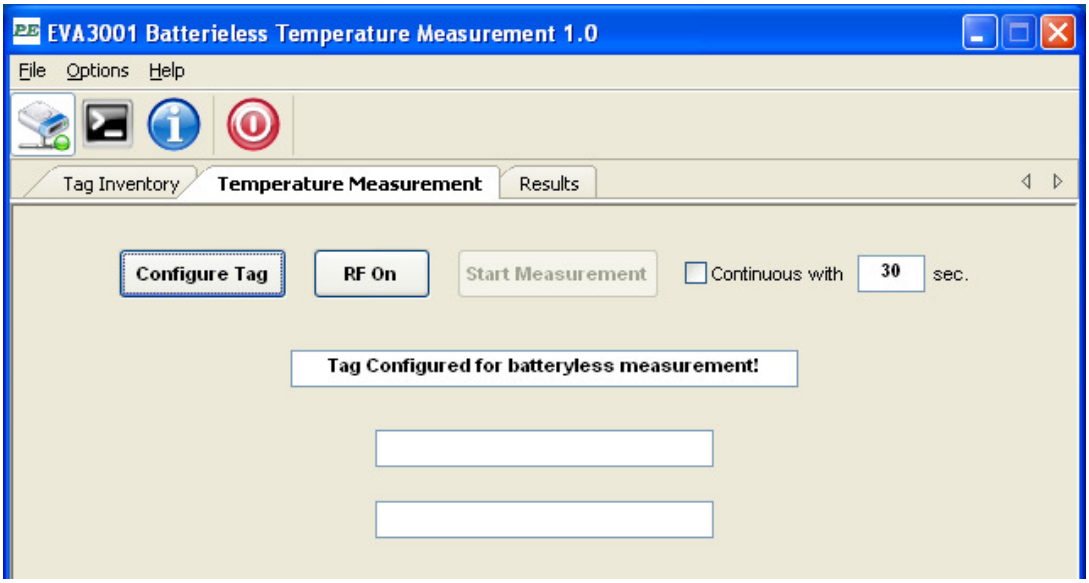
3.3 Software description – Temperature Measurement

To automatically configure the tag with the PE3001, set the RF-Field for measurement. To start the measurement the tab **"Temperature Measurement"** has to be used.



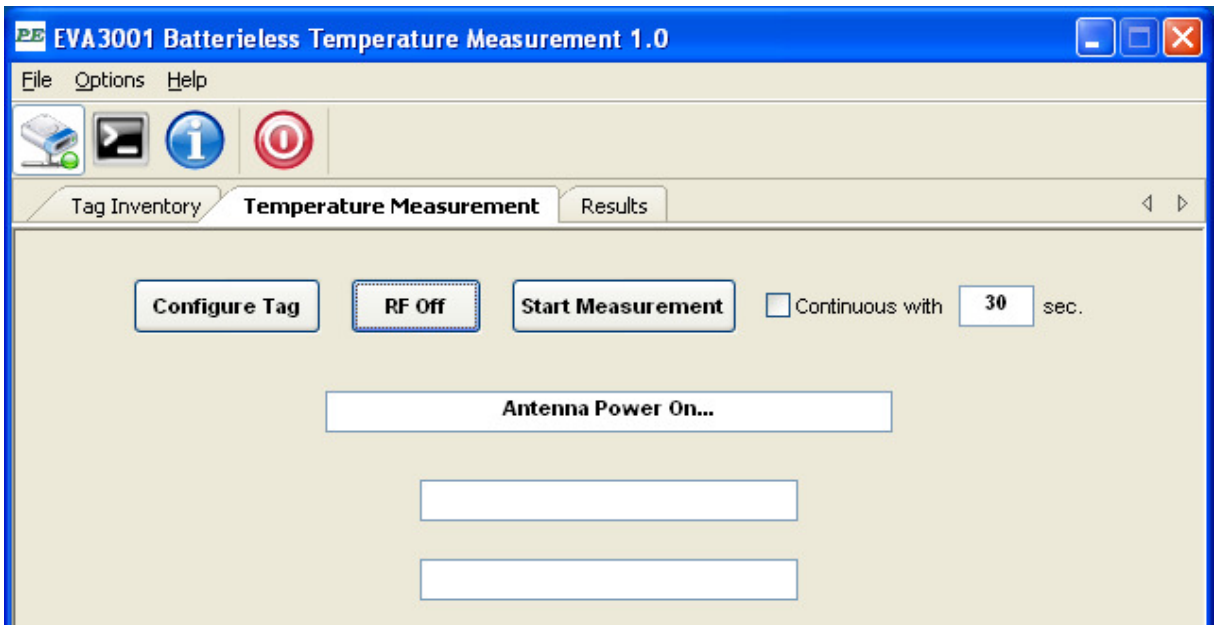
Configure Tag

The button **"Configure Tag"** automatically saves all necessary setup data (registers of the PE3001, for reference see the PE3001 data sheet and compare with source code of this graphical user interface) to store battery-less temperature data in EEPROM of PE3001. The startup time and the logintervall time will be set to “0”. The logsize is set to 10 for a maximum of 20 temperature data.



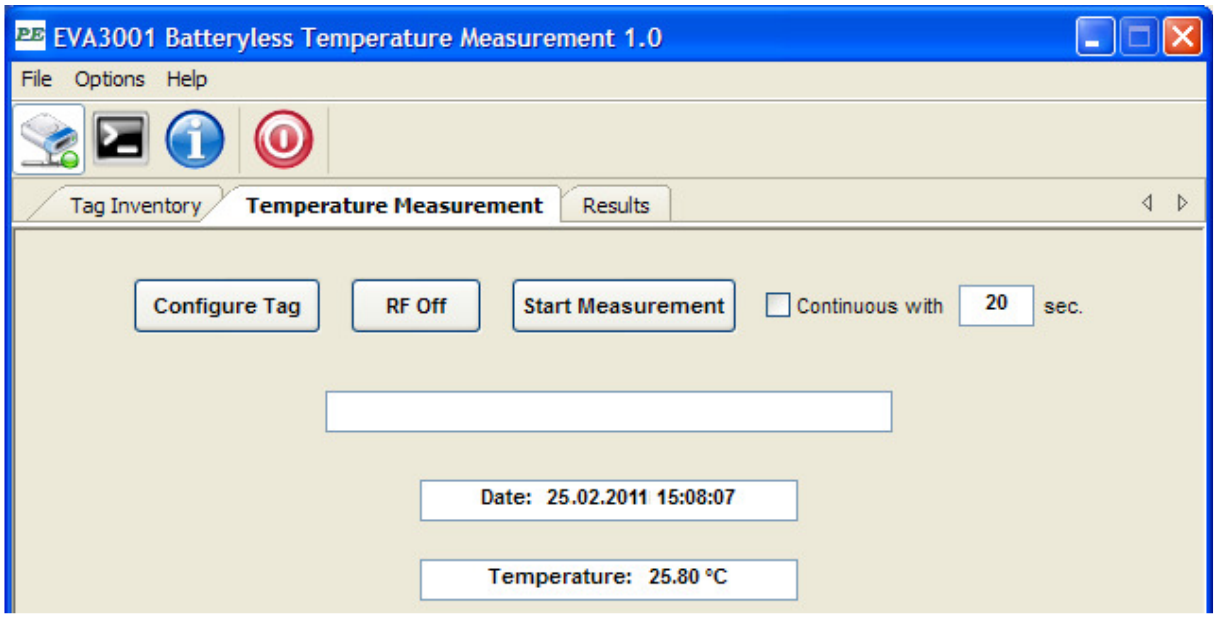
RF On / RF Off

The button **"RF On"** / **"RF Off"** sets the reader to continuous RF-transmission mode. This is necessary for a battery free measurement and loads the capacitors on the EVA3001_Temp board. After the click on this button the button function changes from **"RF On"** to **"RF Off"** and the button **"Start Measurement"** can be used.

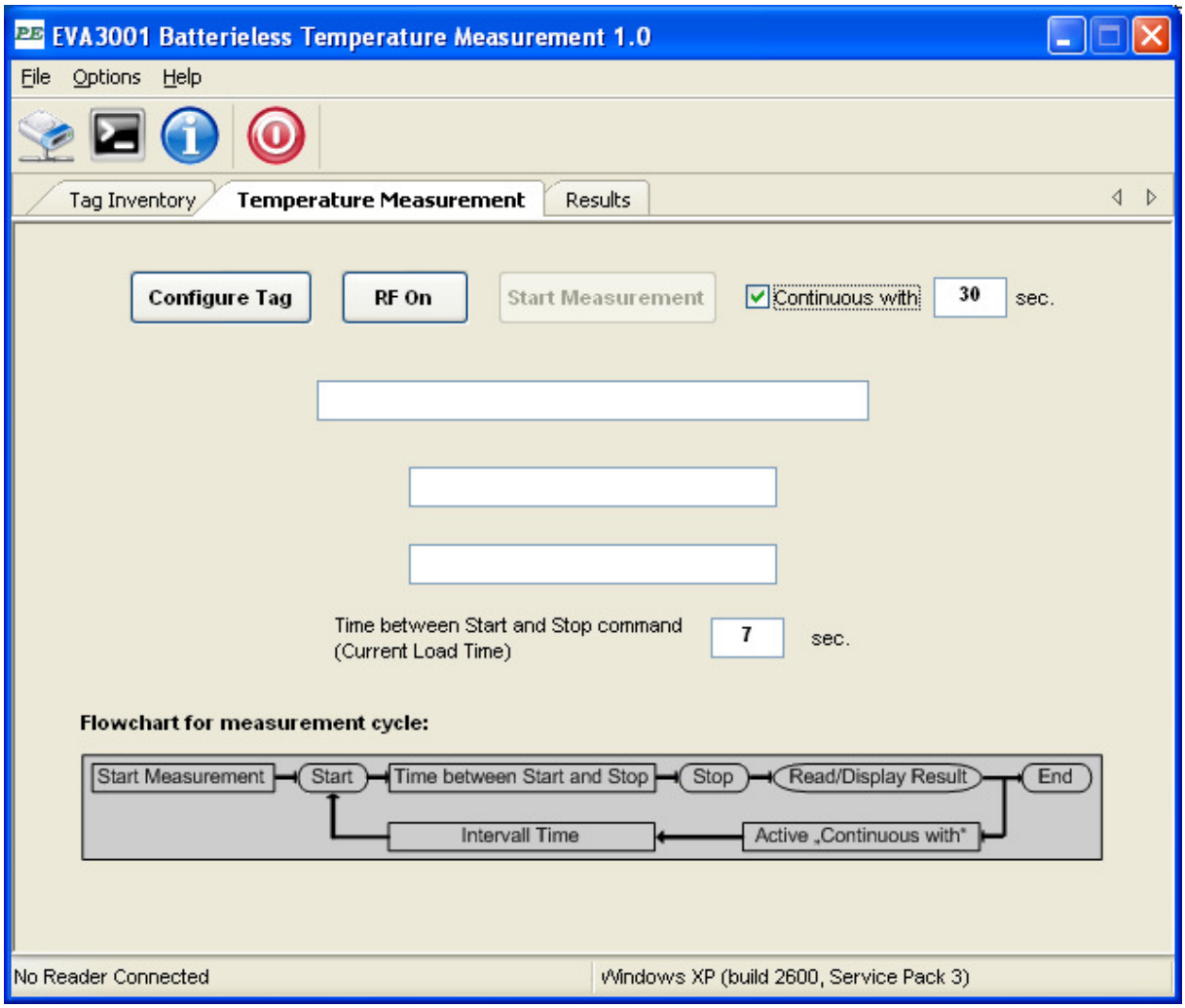


Start Measurement

This button is only usable after the button "**Configure Tag**" and "**RF On**" has been pushed. Push the button **"Start Measurement"** to measure the temperature. The measurement takes about 6 seconds.



It is possible to measure the temperature continuously by setting the continuous mode in seconds and activation of the checkbox "**Continuous with**" and pushing the button "**Start Measurement**". During this mode this button will be deactivated. The mode can be stopped by deactivating the checkbox "**Continuous with**", after the next measurement the mode will be stopped.

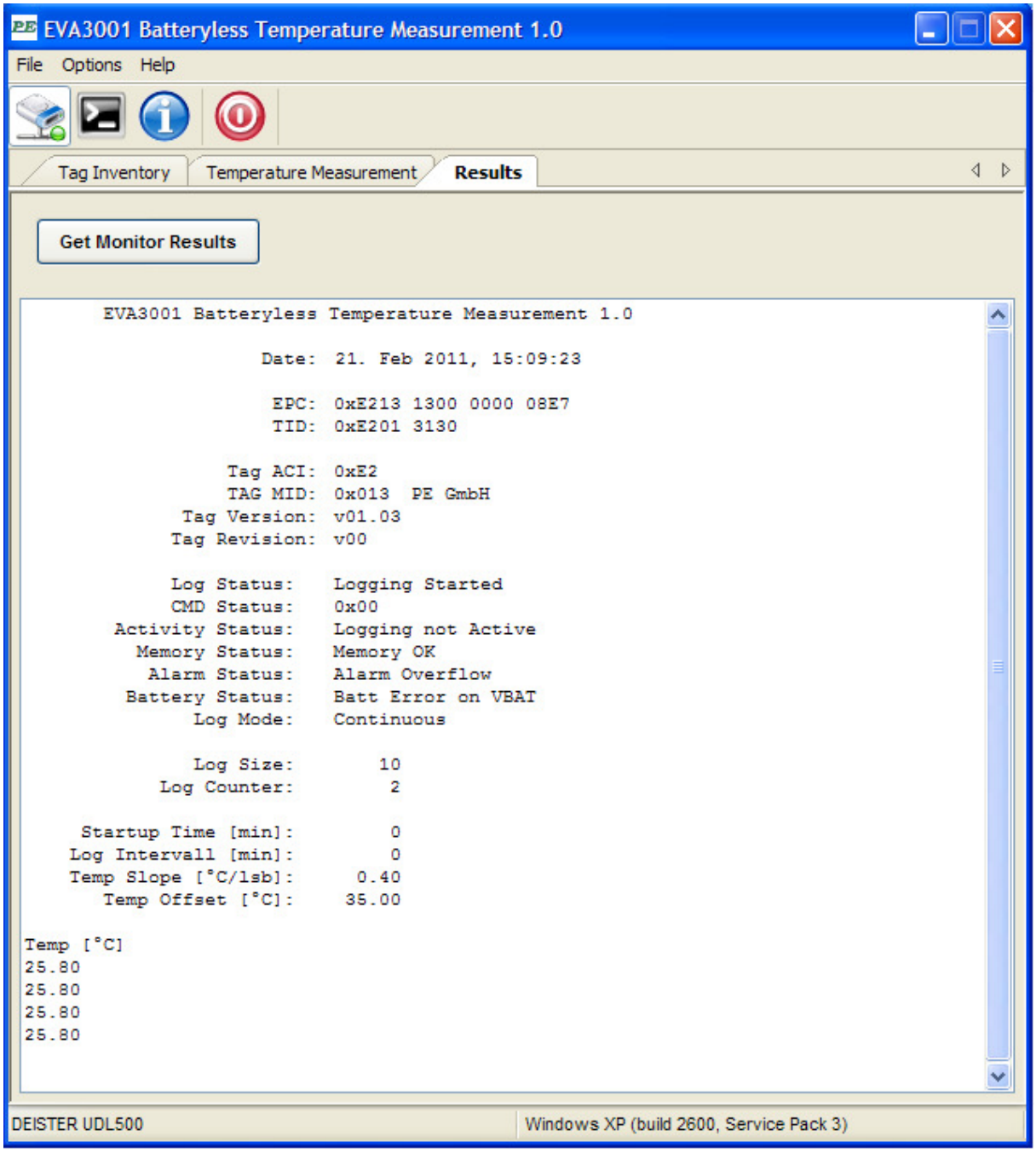


3.4 Software description – Results

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

Get Monitor Results

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



4 Source Code

The source code of the graphical user interface will be copied into the installation directory and can be used for further product development.

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 0x1AE
#define DEF_LOGSIZE 0x00A // Batterielose Temperaturmessung

// memory bank size
#define MAX_EPC      0x008
#define MAX_TID      0x010
#define MAX_USER     0x1DD // user bank size (USER_SIZE = MAX_USER - LOGSIZE)
//#define MAX_USER     0x0FD // user bank size (USER_SIZE = MAX_USER - LOGSIZE) mit Reader
Bereichseinschränkung
#define MIN_RESERVED 0x00A // reserved bank size (RESERVED_SIZE = MIN_RESERVED + LOGSIZE)
#define MAX_RESERVED 0x1E7
//#define MAX_RESERVED 0x074 // Reader Bereichseinschränkung

// 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
```

```
// 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_RSPCMD(reg)       ((reg & 0xF000) >> 12)

#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 Ref.Des. PE3001 Hardware

Features PE3001:

- Passive RFID UHF transponder chip with integrated temperature data monitor
- Compliance with EPC Class 1 Generation 2 (UHF RFID version 1.0.9)
- 8kBit EEPROM read-/writeable access via RF field and/or data monitor
- EEPROM memory to capture measurement data with time stamp or simple ASCII data
- Power supply via RF field (for communication) or from capacitors charged by RF field

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 (or from capacitors in this application) for temperature measurement. 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. Detailed information can be found in the PE3001 datasheet.

5.2 Board 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 PCB (printed circuit board) consists of a network for antenna matching, a capacitor with corresponding connector to the PE3001 and a diode from Vref to the capacitor.

5.3 Evaluation Board Specifications

Table 1 – Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Notice
Operating temperature	T_{amb}	-40	27	80	°C	
Frequency RF field	f_{rf}	860		960	MHz	
Capacitance Voltage	V_{BAT}	1,1	1,4	3,6	V	

5.4 Bill of Material (BOM)

Table 2 - Required devices

Board position	Amount	Value	Description Package	Notice
C1, C2, C3	3	100µF /6,3V	0805	Charge supply for measurement
D1	1	HSMC-285C	SOT-323	Schottky Diode
PE3001	1	PE3001	TSSOP16	

5.5 Schematic

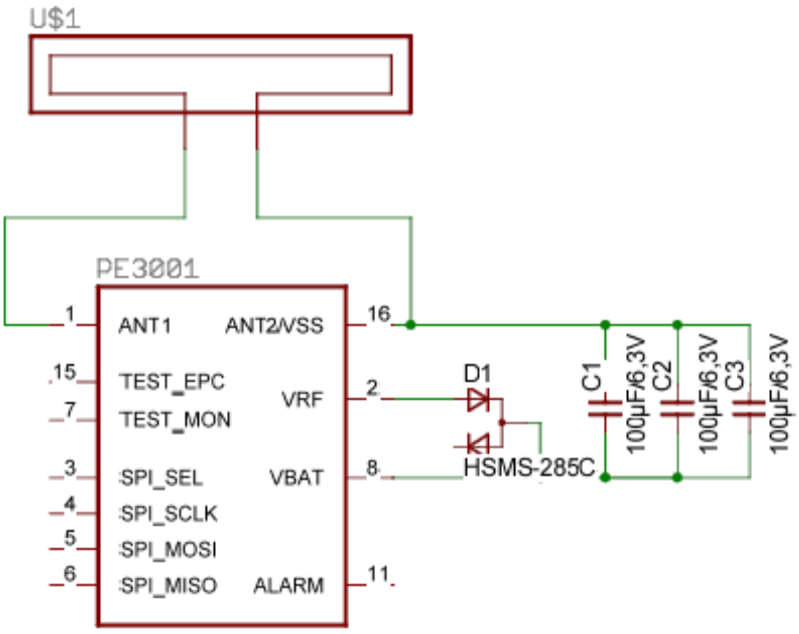


Figure 2 - Schematic of the Evaluation Kit

5.6 Layout

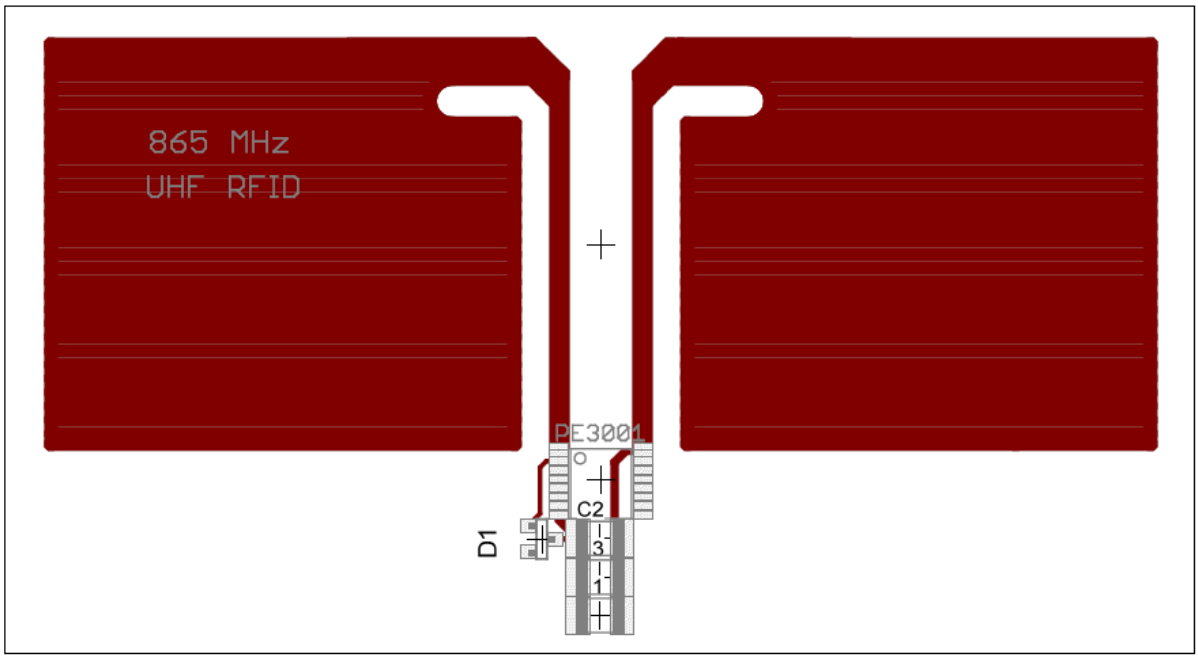


Figure 3 - Layout of the Evaluation Kit

5.7 Operation using UHF interface only

The UHF interface is accessible at any time on the EVA3001. The PE3001 can be controlled with a UHF Reader. 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 responds to any standard EPC Gen2 Reader by sending the UID.

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

5.8 Operation using UHF interface and data monitor

This mode has to be used 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 will be finished.

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

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.

6 Notes

7 Contact

Germany

Stuttgart

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

Dresden

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