



## **TagBlower App**

Copyright © 2019 KATHREIN Solutions GmbH

All rights reserved. No part of this document may be reproduced, distributed, stored in a retrieval system, translated into any language or computer language or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of KATHREIN Solutions GmbH.

KATHREIN Solutions GmbH accepts no liability for omissions or inaccuracies in this document or in relation to the provision or use of the information contained in this document. KATHREIN Solutions GmbH reserves the right to change the products described in this document at any time without notice and does not accept any liability in relation to the application or usage of the products described in this manual. The latest version of this manual is available at our website [www.kathrein-solutions.com](http://www.kathrein-solutions.com).

Information provided in this manual is intended to be accurate and reliable. However, KATHREIN Solutions GmbH assumes no responsibility for its use; nor for any infringements of rights of third parties which may result from its use.

This document and the information contained in it are proprietary information of KATHREIN Solutions GmbH and must be treated as confidential. KATHREIN Solutions GmbH provides this document to its customers in connection with contacts of sale for the products described therein. If the person in possession of this document, being a legal or natural person, is not a contractual sales partner of KATHREIN Solutions GmbH, or KATHREIN Solutions GmbH has not intended him by other means as the recipient of the document and the information contained therein, the person in possession is hereby informed that the use of this document is unlawful and a violation of the rights of KATHREIN Solutions GmbH.

# Contents

|            |                            |          |
|------------|----------------------------|----------|
| 1          | About This Guide           | 4        |
| 2          | Explanation of Symbols     | 4        |
| 3          | Function                   | 4        |
| 4          | Configuration and Listener | 4        |
| 5          | Magic Words                | 6        |
| <b>5.1</b> | <b>Coding</b>              | <b>6</b> |
| <b>5.2</b> | <b>MemoryBank</b>          | <b>7</b> |
| 6          | Examples                   | 7        |

# 1 About This Guide

This document describes how to configure and operate the TagBlower app.

# 2 Explanation of Symbols

| Symbol                 | Meaning  |
|------------------------|--|
| ASyncHeartbeatInterval | frames, block structures, data blocks, commands etc. |
| TagBlower              | ReaderStart V3 GUI                                   |

# 3 Function

The TagBlower app can read tags asynchronously and generate messages when a tag is coming or going. The app provides a server on a configurable port to which TCP clients can connect to receive those messages. The user can specify the message format by setting a coming and/or going datagram. A datagram consists of a normal text and magic words.

In addition, it is possible to generate "heartbeat" messages. They are an easy way to check whether the connection between the TCP client and the app is still active. Messages can be generated when the app starts or stops reading tags or is in a cyclic interval. It is possible to configure the cyclic interval in the reader settings via the ASyncHeartbeatInterval parameter.

# 4 Configuration and Listener

In order to operate the TagBlower app, it is necessary to have a valid license key. Licence keys (either a full license or a time-limited demo license) are bound to the specific reader the app is running on. To obtain a license key, the hardware key provided by the app is needed:

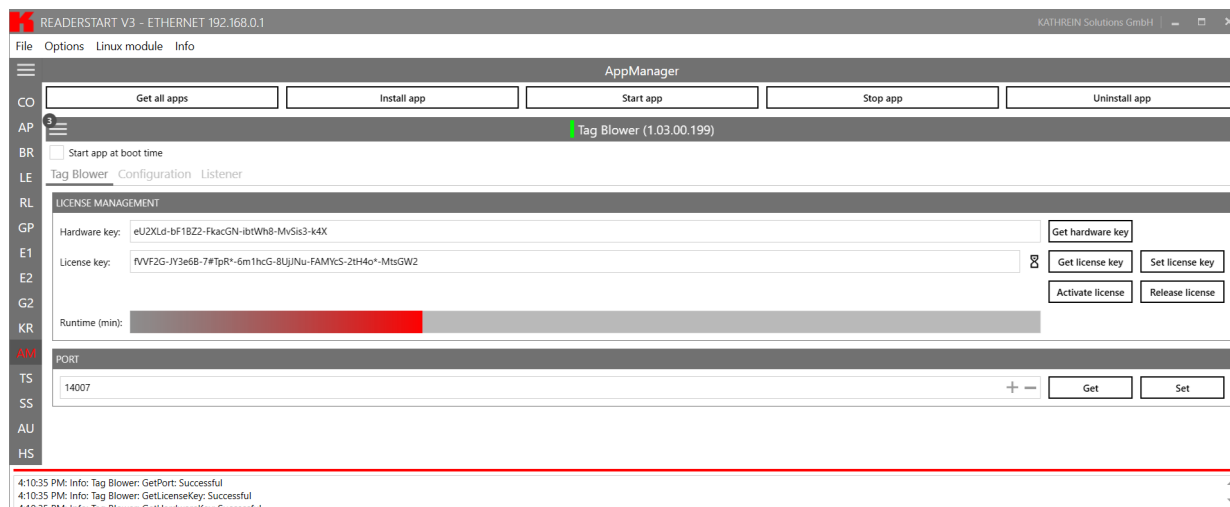


Fig. 1: License Management

The TagBlower is configured in the ReaderStart software:

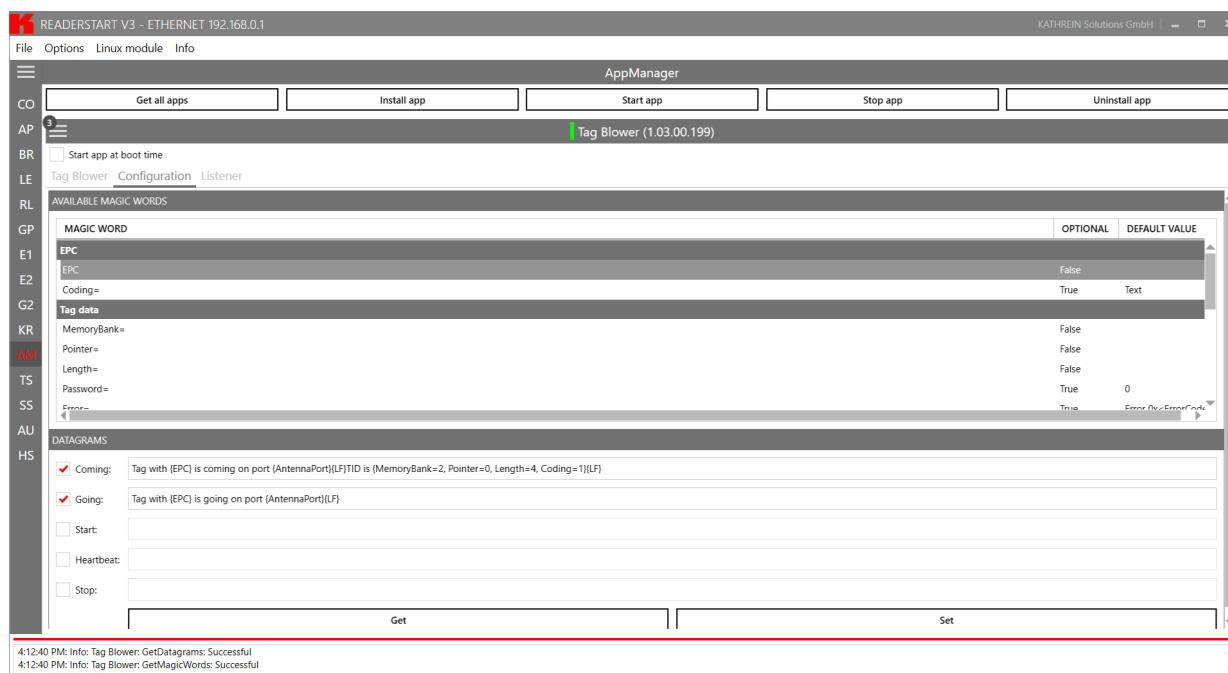


Fig. 2: Configuration tab

It is possible to edit a coming and a going datagram and transfer it to the app. Using the checkboxes, the user can select which event will generate a datagram.

To quickly check the configuration, ReaderStart V3 also offers a TCP Listener which connects to the app:



Fig. 3: Listener tab

Messages generated by the app will be shown here, but it is possible to use any other TCP client. It's possible to connect multiple clients to the app simultaneously; all of them will receive the messages sent by the app.

## 5 Magic Words

Magic words have to be surrounded by '{' and '}'. When a tag is coming or going, the magic word is replaced with the actual data of the tag. Some magic words have to be combined with each other in a comma separated list to be used. If a magic word ends with '=', it usually needs a parameter. If the app can't parse a magic word when the datagram is set, it will not be replaced in the messages.

The following table shows the valid magic word combinations. If a magic word is given in *italic*, it is optional and can be omitted, the app will use its default value.

| <b>Magic Word</b>   | <b>Replaced with</b>  |
|---|---|
| EPC ,<br><i>Coding=</i>   | EPC of a tag  |
| MemoryBank= ,<br>Pointer= ,<br>Length= ,<br><i>Password=</i> ,<br><i>Error=</i> ,<br><i>Coding=</i> | specified memory bank of a tag<br><br>If an error occurs while reading the bank, the tag error code returned by the reader is sent. "Pointer=" is the starting byte address, "Length=" is the number of bytes to be read. If 0xFFFFFFFF is specified, all remaining bytes are read. |
| AntennaPort   | antenna port at which the tag is read   |
| AntennaProperty   | antenna property at which the tag is read   |
| RSSI  | RSSI returned by the reader   |
| Timestamp   | time at which the tag event was registered in the format " <i>yyyy-MM-dd hh:mm:ss</i> "   |
| LF  | LF character 0x0A   |
| CF  | CR character 0x0D   |
| TAB   | TAB character 0x09  |
| NUL   | NUL character 0x00  |
| STX   | STX character 0x02  |
| Hex=  | the character representation of the parameter, e.g. "{Hex=0x31324142}" is replaced with "12AB"  |

The following chapters list the valid parameter values for some magic words.

### 5.1 Coding

| <b>Value</b> | <b>Comment</b>  |
|--------------|---|
| 0            | sends data as a ASCII representation, e.g. 0x31324142 is sent as "12AB"             |
| 1            | sends data as a hex representation, e.g. 0x31324142 is sent as "31324142" (default) |
| 2            | sends data as 6-bit coded data, e.g. 0xC7204286 is sent as "12AB"                   |
| other        | invalid   |

## 5.2 MemoryBank

| Value | Comment |
|-------|---------|
| 0     | RFU     |
| 1     | EPC     |
| 2     | TID     |
| 3     | User    |
| other | invalid |

Note that it's only possible to read one memory bank. If you want to use the "MemoryBank" magic word twice, it has to be configured the same way.

## 6 Examples

Let's assume there is a coming tag with the EPC 0x3152464944323334 ("1RFID234").

| Datagram  | Message                                | Comment   |
|---|--|---|
| EPC is {EPC}.   | EPC is 3152464944323334.               | reads the whole EPC, sends it as a hex representation   |
| EPC is {EPC, Coding=0}.   | EPC is 1RFID234.                       | reads the whole EPC, sends it as a raw binary   |
| EPC is<br>{MemoryBank=1,<br>Coding=0, Pointer=4,<br>Length=8}.      | EPC is 1RFID234.                       | reads the first 8 bytes of the EPC<br><br>Note that the reading begins at byte 4 to skip the CRC and PC of the EPC memory bank. |
| EPC part is<br>{MemoryBank=1,<br>Coding=0, Pointer=5,<br>Length=4}. | EPC part is RFID.                      | reads 4 bytes of the EPC, skipping the first byte of the EPC  |
| EPC part is<br>{MemoryBank=1,<br>Coding=0, Pointer=9,<br>Length=3}. | EPC part is 234.                       | reads 3 bytes of the EPC, skipping the first 5 bytes of the EPC   |
| EPC is {EPC} or "{EPC,<br>Coding=0}".                               | EPC is 3152464944323334 or "1RFID234". | reads the whole EPC and sends it as a hex representation and a raw binary   |
| EPC is<br>{MemoryBank=1,<br>Pointer=4,<br>Length=100}.              | EPC is Error 0x83.                     | trying to read more bytes than available<br>returns error code 0x83 (TEC_MemoryOverrun).  |



Electronic equipment is not domestic waste – in accordance with directive 2002/96/EC OF THE EUROPEAN PARLIAMENT AND THE COUNCIL dated 27th January 2003 concerning used electrical and electronic appliances, it must be disposed of properly. At the end of its service life, take this unit for disposal at a designated public collection point.