

# Table of Contents

<b>Hardware.....</b>	<b>1</b>
RFID reader.....	1
Reader antenna.....	1
<b>Software.....</b>	<b>2</b>
How to start up.....	2

# Hardware

This section will give a short overview over the different components necessary to read RFID tags.

## RFID reader

We use a ThingMagic USBPro reader. The reader can be bought at [DigiKey](#) or [RFIDStore](#). RFID reader will be pre configured in Freiburg using the Autonomous Configuration Tool from the manufacturer. The Reader will be configured to start reading as soon as it is connected to USB or on button push. If you want to play with the reader yourself make sure to connect both USB connectors, insufficient power will NOT trigger an error!

## Reader antenna

The plan is to read out tags by bringing the antenna within cm distance from the tag. Any nearfield antenna can serve this purpose. We use a Beta-layout nearfield antenna.

This antenna can be bought [here](#).

# Software

A demo version of the software can be found on GitLab under this [Link](#). Required packages are PyQt5, pyserial and numpy. In addition the iTk production database library is necessary to communicate with the database. To be operational in the subfolder User the file User.txt needs to be filled with the database credentials. This is necessary for communication with the database. In the file Mercury.py the correct COM port for the reader needs to be specified (example for Windows).

## How to start up

Software is started like any python script.

Only tags which are ticked are scanned or uploaded.

Before upload make sure a Barrel Hybrid Flex sheet with Hybrids is pre created in the database and that the sheet number you specified is unique.

-- JanCedricHonig - 2019-06-13

---

This topic: Sandbox > RfidSetup

Topic revision: r2 - 2019-06-13 - JanCedricHonig



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback