



## j62056 User Guide

Fraunhofer Institute for Solar Energy Systems ISE

[openmuc.org](http://openmuc.org)

# Table of Contents

1. Intro .....	1
2. Distribution .....	1
2.1. Dependencies .....	1
2.2. Reader Application .....	1
3. Using j62056 .....	1
4. Modifying and Compiling j62056 .....	1
5. How IEC 62056-21 Works .....	2
6. Authors .....	2

# 1. Intro

j62056 is a Java implementation of the IEC 62056-21 protocol (modes A, B, C, and D). It can be used to develop individual IEC 62056-21 master applications that read metering data from slaves such as gas, water, heat, or electricity meters. The library includes an application that reads a given meter and prints the received data to stdout. j62056 is licensed under the LGPL version 2.1 or later.

## 2. Distribution

After extracting the distribution tar file the j62056 library can be found in the directory *build/libs-all*.

### 2.1. Dependencies

Besides the j62056 library the folder *build/libs-all/* contains the following external library:

jRXTx

jRXTX for accessing the serial port (UART). jRXTx is licensed under LGPL(v2.1 or later) + linking exception. It consists of a Java part and a native part. Note that you have to install the native part of as explained in our [FAQ](#).

The *dependencies* folder in the distribution contains more detailed license and copyright information about the dependency.

### 2.2. Reader Application

A command line application is part of the library that can be used to read a meter. You can execute it using the *j62056-reader* script found in the folder *run-scripts*. Executing the script without any parameters will print help information to the screen.

Instead of running the application from the terminal you can create Eclipse project files as explained in our [FAQs](#) and run it from within Eclipse.

## 3. Using j62056

See the j62056 [Javadoc](#) for documentation of the API. You may want to take a look at the source code of the reader app which is part of the distribution. You can find the source code of Reader.java in `src/main/java/org/openmuc/j62056/app/`.

## 4. Modifying and Compiling j62056

We use the Gradle build automation tool. The distribution contains a fully functional gradle build file (*build.gradle*). Thus if you changed code and want to rebuild a library you can do it easily with Gradle. Also if you want to import

our software into Eclipse you can easily create Eclipse project files using Gradle. Just follow the instructions on our FAQ site.

## 5. How IEC 62056-21 Works

IEC 62056-21 offers modes A,B,C,D and E. The most commonly used mode is mode C. The mode C message exchange looks like this:

Master -----Request message (300 baud)----- > Slave

Master < ----Identification message (300 baud)----- Slave

Master -----Acknowledgment (300 baud)----- > Slave

Master < -----Data message (new baud rate)----- Slave

The request message may optionally contain a device address. The identification message contains:

- the manufacturer ID consisting of three upper case letters
- the requested baud rate to change to for the data message
- the meter identification field
- the enhanced identification field

## 6. Authors

Developers:

- Stefan Feuerhahn
- Dirk Zimmermann