OP-740

Android Meter Data Collection Library Implementation Guide

(Version 2.021 and above)

Date:
July 2021

GERMAN METERING
German Metring GmbH
www.german-metering.com

Address:
German Metering GmbH
Reuterweg 65
D-60323 Frankfurt am Main
Germany

Tel.: +49 (0) 69 / 770 622 - 06
Fax: +49 (0) 69 / 770 622 - 05

Index

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com

This manual explains the instruction for using this product. Read this manual before developing your android application. Keep this manual within easy reach, and use it.

1 Introduction

Optical probes are used to communicate and read electricity meters. The OP-740 Bluetooth optical probe of German Metering GmbH, has a library file with it, which has been made for Android developers and use in applications for reading meters. This guide will explain how to implement this library into the target application.

2 How to use the library

After adding the library to the project, there are three general sections that will be explained in order of importance.

2.1 Reading meters

1- First import the class:

```
import com.german-metering.readmeterlib.Classes.Bussiness
```

2- Import the structure:

```
import com.german-metering.readmeterlib.Structures.DigitalCounterInfo
```

3- Define a new intent:

```
intent = new Intent("com.german-metering.readmeterlib.Activities.ReadMeter")
```

4- Add the Bundle as described table below:

Key	Value	Description
Prob_Type	OP_Bluetooth OP_USB OP_Serial	Type of probe used
BT_Serial	It's a string with 10 characters.	You must have previously connected and paired with the device via the device's Bluetooth menu.
Special_Meter_Name	TRUE or FALSE	Indicates whether the meter is in the list of specific meters or not.
Time_Diff	Min = 10 Max = 120	The value defined for the difference between the android device clock and the meter clock in minutes, which sets the date and time of the meter if it's available.
Meter_Pass_Json	Appendix Table 2	Json string contains passwords for various meter models

Note: The list of specific meters is given in Appendix Table 1.

5- Call Intent with the following method:

```
startActivityForResult(intent, 100)
```

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com

2.1.1 Get Collection Results

After reading the meter, the results in the method are received as a Json string from the library:

```
onActivityResult
```

If the meter reading is successful (resultCode == RESULT_OK) we go to the data processing section and otherwise we process the operation error.

2.1.2 Processing Collection Results

In the Intent method there are three arguments:

2.1.2.1 Read_Result

It is parsed in the class below:

```
public class DigitalCounterInfo {

   private String companyName;
   private String meterModel;
   private String meterString;

   private Boolean setDateTime;
   private Integer readMode;
   private Integer sendPassMode;
   private Integer floatPointNum;

   private String demand1;

   private String active1;
   private String active2;
   private String active3;

   private String reactive1;
   private String reactive2;
   private String reactive3;

   private String weekend1;
   private String weekend2;
   private String weekend3;

   private String reserve1;
   private String reserve2;
   private String reserve3;

   private String serialNum;
   private String serialNum2;

   private String curDate;
   private String curTime;

   private String activeSum;
   private String reversEnergy;
   private String reversEnergy1;
   private String reversEnergy2;
   private String reversEnergy3;
   private String reversEnergy4;

   private String funcErr1;
   private String funcErr2;
   private String funcErr3;
   private String funcErr4;
   private String funcErr5;

   private String curVolt1;
```

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com

```
    private String curVolt2;
    private String curVolt3;

    private String curCurrent1;
    private String curCurrent2;
    private String curCurrent3;

    private String pass1;
    private String pass2;
    private String pass3;

    private String valueFormat;
}
```

2.1.2.2 Meter_Name

It's the Name of meter.

2.1.2.3 Result_Message

Error received:

<string name="read_notRegisteredMeter">Meter is not defined</string>
<string name="toast_error_invalid_data_from_meter">Invalid data from meter</string>
<string name="toast_password_error">Wrong password</string>
<string name="toast_error_no_response_from_meter">No response from meter</string>
<string name="toast_timeout_error">Timeout</string>
<string name="toast_cancel_by_user">Canceled by user</string>
<string name="toast_error_read_obisfile">Error while reading obis code file</string>
<string name="toast_error_connecting_bluetooth_device">Bluetooth device connection error</string>

2.2 Update the passwords for various meters

The library has default passwords for all meters. If you want to send other passwords to the meter, a string in Json format must be sent that contains the desired meter model and a maximum of three passwords for each meter model under the Meter_Pass_Json argument. To ease this operation, a small software to define the password is provided with the library. The Json string is eventually produced by the same software. It should be noted that passwords are encrypted in the application.

2.2.1 Used algorithm for changing the password

```
public static String EncryptDecrypt(String szPlainText, int
szEncryptionKey)
{
    StringBuilder szInputStringBuild = new StringBuilder(szPlainText);
    StringBuilder szOutStringBuild = new
StringBuilder(szPlainText.length());
    char Textch;
    for (int iCount = 0; iCount < szPlainText.length(); iCount++)
    {
        Textch = szInputStringBuild.charAt(iCount);
        Textch = (char)(Textch ^ szEncryptionKey);
        szOutStringBuild.append(Textch);
    }
```

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com

```
    return szOutStringBuild.toString();
}
```

Note: szEncryptionKey = 146

Example of a Json string:

```
{  "Pass": [
    {
      "meterModel": "ABB_Family",
      "meterString": "/ABB",
      "pass1": "¢¢¢¢¢¢¢",
      "pass2": "",
      "pass3": ""
    },
  ]
}
```

2.3 Get the reading library version

By calling:

```
com.german-metering.readmeterlib.BuildConfig.VERSION_NAME;
```

2.4 Supported Standards

- Communication with meters with IEC62056-21 Mode C (READOUT, PROGRAMMING) communication protocol.
- Communication with meters with IEC62056-21 Mode E communication protocol.
- Ability to read meters that have DLMS communication protocol. In this case, only the parameters are read at the LLS access level.
- Ability to read meters whose communication protocol is IEC62056-31 but the optical port of meter physically supports IEC62056-21 standard.

3 Appendix Table 1

Table of special meters:

#	Company	Models
1	EDMI	MK10 – MK6
2	SAGEM	CX2000

4 Appendix Table 2

Table of meters that allow application to adjust the date and time:

#	Company	Models	Description
1	AMPY	Grey_1_Phase	-
2	AMPY	White_3_Phase	-
3	AMPY	Grey_3_Phase	Applied after turning off and on

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com

4	INTECH	HEX_12_34	-
5	ACTARIS	ACE_2000	-
6	ELESTER	A1350i	-
7	ELESTER	A120	-
8	ABB	ABB_Family	The bottom frame of the meter must be open when configuring A220 model.

5 Appendix Table 3

Table of supported meter models:

#	Company	Model
1	ABB	A1350
2		A1440
3		A1500
4		A220
5		A220R
6	ELESTER	A120
7		A1350i
8		A1700
9	ACTARIS	ACE5000
10		ACE6000
11		ACE2000
12	AMPY	5194E
13		5219J
14		5192F
15		5192J
16	INTECH	HXE12
17		HXE34
18	ISKRA	MT32H
19		MT830
20		ME420
21	LANDIS	ZMG
22		ZMD
23	EDMI	MK10
24		MK6
25	ELGAMA	GAMA300

If the intended meter model was not supported, in the collection process a try will initiate by the name DEFAULT_IEC_READOUT in Readout mode. If the meter responds to the request, received data will be parsed using standard OBIS codes and if meter does not respond the connection will be terminated and appropriate message will be shown.

In all cases the communication log is returned as an argument which by using the library saving this log is optional to the programmer.

Reuterweg 65, 60323 Frankfurt am Main, Germany
Telephone: +49 (0)69 / 77062206
Fax: +49 (0)69 / 77062226
www.german-metering.com