



iMeter
API Specification
Version 1.5

© 2008 OSS Ltd
<http://www.imeterbox.com>
<mailto:sales@outrak.com>

Ireland Tel: +353-1-2570400, Fax: +353-1-4588573
UK Tel: +44 161 210 2129

File Reference

Meter_API_V1_0.doc

Version History

Version	Author	Date	Description
1.0	Sean Finnerty	16/01/2008	First draft version.
1.1	Paul Foley	19/02/2008	i-Meter Marketing Update.
1.2	Thomas Slattery	30/07/2008	SJV rev.006 changes.
1.3	Thomas Slattery	09/01/2009	Updated Switch settings and tables for Mechtronic meter.
1.4	Thomas Slattery	12/02/2009	Updated Switch settings and tables for New Alpeco Meter TE550
1.5	David Bagglely	08/06/2009	Power connections and Bluetooth serial port settings for TE550 and Mechtronic.

Approvals

Company	Name	Signature	Date
OutTrak Software Services Ltd.	Sean Finnerty		
OutTrak Software Services Ltd.	Shane Boyle		

Table of Contents

1.	ABREVIATIONS	4
2.	INTRODUCTION	4
3.	CONTACT	4
4.	SYSTEM OVERVIEW	5
5.	HARDWARE	5
5.1	Meter Type Selector	6
5.2	Bluetooth CSA Adapter	7
6.	SOFTWARE	8
6.1	Meter Box Commands	8
6.1.1	AT+PORT=?	9
6.1.2	AT+PORT=A	9
6.1.3	AT+PORT=B	9
6.1.4	AT+PORT=C	9
6.2	iMeterBox SDK	10
6.2.1	stMETER Structure	11
6.2.2	LCC_API int WINAPI MeterInit(stMETER *pstMeter)	11
6.2.3	LCC_API int WINAPI MeterPrintData(char *pszData, int nSize)	12
6.2.4	LCC_API int WINAPI MeterGetTicket(stMETER *pstMeter)	12
6.2.5	LCC_API void WINAPI MeterEndPrint(void)	13
6.2.6	LCC_API int WINAPI MeterVersion(char *pszVersion)	13
6.2.7	LCC_API void WINAPI MeterClose(void)	13
7.	END-USER LICENSE AGREEMENT FOR OUTTRAK	13
7.1	DISTRIBUTION	13
7.2	DISCLAIMER OF WARRANTY	13
7.3	TERMINATION	14
8.	DISCLAIMER	14

1. ABBREVIATIONS

Abreviation	Meaning
API	Application Programming Interface
BT	Bluetooth
CSA	Cordless Serial Adapter

2. INTRODUCTION

The following document describes the meter interface box (iMeterBox) hardware and software. This box was developed to allow easy integration of existing mobile software applications to the existing printer, connected to the electronic meter on fuel delivery vehicles. In particular this box supports both the LectroCount and Drum meter solutions. It is planned to support more as they become available.

The same hardware is used for both solutions, the box is however configured different for both solutions via external switch settings described in the hardware section. This solution also supports a Bluetooth connection to the box via the Socket CSA Adapter. This BT connection can be used to print the delivery ticket by the software vendor.

3. CONTACT

The iMeterBox is a product from OSS Ltd.

Visit our website <http://www.imeterbox.com/> to:

- Get latest information about this and other products developed by OutTrak Software Services Ltd.
- Get updates of iMeterBox documentation and SDK.

Contact details:

Email: support@outtrak.com

Telephone: +353-1-2570400 Fax: +353-1-4588573

4. SYSTEM OVERVIEW

Today most electronic meter manufacturers supply their system with all the hardware including a printer which is required to print the metered ticket as part of the delivery process. This ticket is mandatory under weights and measures guidelines. On board computing such as the OutTrak solution may also require a printer to print the invoice details for the customer. This would then require two printers in the vehicle and potentially the same two. This is not a practical solution.

The iMeterBox offers you a solution where you only require one printer that is linked to both the electronic meter and the on board computer. It also offers you a wireless link from the computer to the printer via the Bluetooth connection which eliminates extra cables from the vehicle.

5. HARDWARE

The hardware used for this solution comes in the form of a black box with a dual power port plus three serial ports, see (Picture 1) below.

The power port on the i-meter box is only needed if you are connecting the i-meter box to a LectroCount LC3, TE550 or Mechtronic meter. For the Drum meter the power comes through the serial printer cable attached to port 0 on the i-meter box.

For the LectroCount LC3, TE550 and Mechtronic meters there are two power connections on the black box and they are bi-directional, they can be either input or output, so it does not matter which one you plug the power into, plug the supplied DC21 cable into the other power connection and that loops power back to the printer.

There are three serial ports on the iMeterBox:

- Port 0 is the input port from the meter, this requires the purple cable.
- Port 1 is the output port from the box to the printer, use the white cable for this.
- Port 2 is the control port, this can either be connected directly from the on-board computer or you can connect the Socket CSA adapter to this port.

Picture 1: iMeterBox

5.1 Meter Type Selector

The meter box supports at present the four most popular meters on the market today and will support more as and when OutTrak Software Services test them.

To configure the box for the different meters you must adjust the switch (See Picture 2) as follows, see Table 1.

Picture 2: Switch**Table 1: Meter Type Switch Settings**

Meter	Switch1	Switch2
LectroCount LC3	On	Off
Mectronic	Off	On
LectroCount TE550	On	On
Drum	Off	Off

Switch down is in the **ON** position. Up is **Off**

****Note: Please note that before adjusting any switch settings the box must be powered off. Do not adjust switch when power is applied to box.**

Table 2: Print I-Meter Software Version Number

Meter	Switch3
Print Version Number	On
Don't Print Version Number	Off

Switch 3 comes set to **Off** as default. If you turn it **On** it will override all other settings and will print out the I-Meter Software version number on any paper loaded into the printer.

Table 3: Bluetooth Socket Power On/Off

Meter	Switch4
Turn on Power for attached Bluetooth Socket	On
Turn off Power for attached Bluetooth Socket	Off

Switch 4 is only used if you are attaching a bluetooth socket to the I-Meter Box and you wish to power it on from the I-Meter Box.

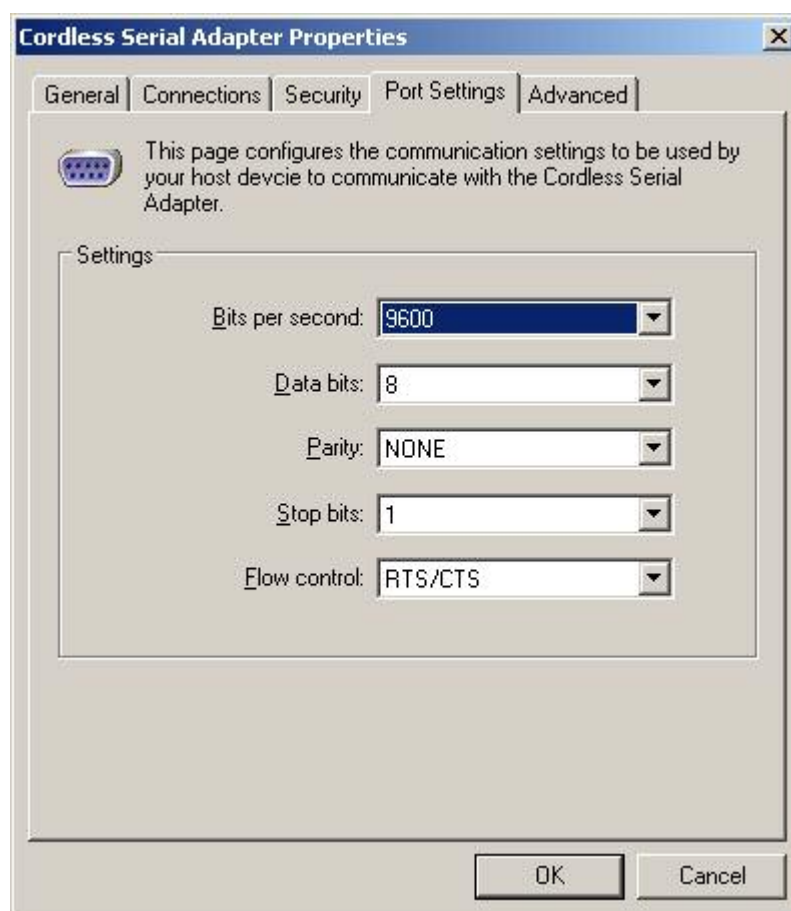
5.2 Bluetooth CSA Adapter

When connecting the BT adapter you must ensure that the bluetooth socket attached to port 2 is configured correctly for the type of meter selected in Table 1. See Table 4. for settings when BT adapter is used.

Table 4: Bluetooth Adapter Serial Port Settings

Meter	Baud Rate	Data Bits	Parity	Stop Bits
LC3, TE550, Mechtronic	9600	8	None	1
Drum	1200	8	Even	1

The BT adapter settings can be configured using the Cordless Adapter Configuration Utility that come with it. Once the switch is changed it changes the serial port parameters of the box and we therefore need to configure the BT adapter with the same serial port settings, see Table 4 and screen shots (Figure 1) of CSA Utility.

Figure 1: CSA Adapter Settings

6. SOFTWARE

The meter box (iMeterBox) is controlled by access to Port 2 on the device. As previously explained the port takes a different port setting depending on its configuration. See Table 2 for settings. We have provided a Software Development Kit (SDK) with the box so that software suppliers can easily get up and running with the solution quickly.

NOTE: All iMeterBox SDKs are backward compatible with iMeterBox hardware.

If you choose not to use the SDK you can use the (AT) commands provided via Port 2 to interface with the meter box.

6.1 Meter Box Commands

The following raw commands provide you with the following functionality if you wish to write your own interface to the meter box. For all the following commands you do NOT require a carriage return linefeed command i.e. (CR LF). You can connect the meter box

to your PC via a serial port connection and then using HyperTerminal you can issue the following commands to simulate control via Port 2.

6.1.1 AT+PORT=?

This command returns the meter box version along with the current mode setting of the device. When the box is first powered up the mode is set to 'A' as the example below shows.

Example of data returned from box:

```
OutTrak Software Services Ltd.
RS232 Multiplexer 3-Port
Electronic Meter Interface Module
Build Version 1.1.5 Test 12/01/08
Operating Mode: A
```

6.1.2 AT+PORT=A

This sets the meter box in operating mode 'A', this is normal operating mode. We use this command when we do not wish to interface to the meter. When we have completed delivery we would normally set the meter box in this mode.

6.1.3 AT+PORT=B

This mode sets the meter box in delivery mode ready to accept the data sent to the printer by the electronic meter. When data is sent to the printer from the meter we capture this data and send it to Port 2. The SDK handles parsing the data for us into the relevant data fields.

6.1.4 AT+PORT=C

The mode sets the meter box in print mode, it allows us to print our data to the printer safely knowing that the meter box is taking care of responses from the electronic meter. If we did not have this mode the printer could power down during printing which is controlled by the meter. This command eliminates this error from occurring. You issue this command before you want to print any data to the printer and you then issue command 'B' or 'A' depending on where you are in the delivery process.

A typical delivery process would be as follows:

Table 5: iMeterBox Delivery Mode

Command	Action
AT+PORT=C	Send ticket header to printer, normally the customer

	name and address details. Static data.
AT+PORT=B	At this point the meter is used to deliver the product to the customer.
Application Waits for Data.....	User prints metered ticket. We capture data from ticket.
AT+PORT=C	Using data captured from the delivery we send the ticket footer to the printer including any data that is calculated from the captured data. This would normally be the product and quantity.
AT+PORT=A	Once the delivery is completed we then change the meter box into idle mode.

6.2 iMeterBox SDK

The following outlines the different API's that are available in the SDK module for the different platforms. Because this is an on board vehicle computing environment we have assumed that the main platform for development is the Pocket PC or Windows Mobile environment.

We have support for both the Native 'C/C++' environment or the newer '.NET' platform.

The SDK available in a zip package (imeter_sdk_v???.zip, where ??? is the version number) consists of the following files:

Table 6: SDK Files

Filename	Description
LCCAPI.H	Native C/C++ API header include file.
sdk\Windows Mobile 5.0 Pocket PC SDK (ARMV4I)\Release\meter.lib	Native C/C++ API link lib file.
sdk\Windows Mobile 5.0 Pocket PC SDK (ARMV4I)\Release\meter.dll	Meter interface release module for Pocket PC
MeterInterface.cs	.NET C# Meter Interface Module
MeterTestForm.cs	.NET C# Meter Interface Sample Application

We will outline the Native 'C/C++' interface file as the .NET application example give a clear indication of how this API functions.

6.2.1 stMETER Structure

This following structure defines the stMETER interface between the iMeterBox and the SDK. Not all fields are used at present. You will know from the sample source applications what API parameters are necessary, especially for setting up the delivery.

```
typedef struct
{
    int    nComPort;           //Com Port to use
    int    (*lpStatusFunction)(char*, void*); //Callback function
    FILE   *fOutput;          //Output file handle
    char   initdll[20];       //Initialisation DLL module
    BOOL   bUseDCB;           //Use current DCB structure
    DCB    dcb;               //DCB structure
    char   szVersion[40];     //Module version string
    BOOL   bDebug;           //Debug flag
    FILE   *fDebug;          //Debug file handle
    RX_STATE rxState;        //Receive state machine status
    METER_STATE mState;      //Meter state machine status
    BYTE    rxBuf[1024];     //Receive buffer
    int     rxBufCount;      //Receive buffer count
    BOOL    bDelStart;       //Delivery start flag
    BOOL    bDelEnd;         //Delivery end flag
    int     nErrorCount;     //Error count
    BOOL    bExit;           //Exit flag
    Double  dLitres;         //Number of litres delivered
    int     nProduct;        //Product delivered
    BOOL    bData;           //Data received
    BOOL    bScrUpdate;     //Update status screen
    int     nProductCode;    //Product ordered code
    double  dQtyOrdered;     //Quantity ordered
    int     nMeterType;      //Meter type
    int     nVersion;        //Meter Version
    HWND    hStatusWindow;  //Status window handle
} stMETER;
```

6.2.2 LCC_API int WINAPI MeterInit(stMETER *pstMeter)

This function initialises the meter interface module. The pstMeter structure needs to be initialized with certain parameters before it can be used, see example as follows:

```
/*
 * Setup meter structure
 */
memset(&stMeter, 0, sizeof(stMeter));
stMeter.nComPort = COM1;
stMeter.lpStatusFunction = (*nStatusFunction);
stMeter.mState = STATE_METER_INIT;
stMeter.fOutput = NULL;
stMeter.bDebug = FALSE;
stMeter.nProductCode = 1;
stMeter.dQtyOrdered = Quantity;
stMeter.nMeterType = LCC_METER | DRUM_METER;
stMeter.bUseDCB = FALSE;
```

This function returns zero if successful and non-zero if there is an error.

6.2.3 LCC_API int WINAPI MeterPrintData(char *pszData, int nSize)

This function is used to print data to the printer connected to the meter box. pszData is a pointer to the data to be printed and nSize is the length of the string to print. Example of function as follows:

```
/*
 * Print Data to Printer
 */
char szTestData[] = "This is a test for the printer\r\n";
MeterPrintData(szTestData, strlen(szTestData));
```

This function returns zero if successful and non-zero if there is an error.

6.2.4 LCC_API int WINAPI MeterGetTicket(stMETER *pstMeter)

This function is called when the meter starts to deliver the product. This function must be called before the meter prints the ticket.

This function captures the printer data in the pstMeter structure. It will also parse certain parameters (Table 7) from the data and allow you to use those values for your own data processing, you will use that data to print the ticket footer.

For example calculate a total invoice amount based on the quantity captured and product. On returning from this function the following members of the stMETER structure will be filled:

Table 7: Returned Values from MeterGetTicket function

Type	Name	Description
BYTE pointer	rxBuf	This hold the raw print data that the meter sends to the printer including escape sequences and printer control codes.
int	rxBufCount	The holds the number of characters received in rxBuf.
double	dLitres	This holds the number of litres that the ticket printed.
int	nProduct	This holds the product code delivered on the ticket.

6.2.5 LCC_API void WINAPI MeterEndPrint(void)

This function is used to indicate that printing has completed. This function must be used when using the Bluetooth option as this closes the link on the meter box. Once the user has confirmed (visually) that the printed docket is complete only then can you close the link to the meter box.

6.2.6 LCC_API int WINAPI MeterVersion(char *pszVersion)

If the user wants to check the version of SDK in use then this function can be used. It returns the version string in the char pointer pszVersion.

6.2.7 LCC_API void WINAPI MeterClose(void)

When printing has completed and the user has completed using the meter printer then this function must be called which closes off the serial port connection and releases any memory resources used during the printout.

7. END-USER LICENSE AGREEMENT FOR OUTTRAK

This End-User License Agreement ("AGREEMENT") is a legal agreement between OSS Ltd. and an individual or a single entity who agrees and accepts the following terms and conditions by installing, copying, or otherwise using the software product identified above ("SOFTWARE PRODUCT").

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties. The SOFTWARE PRODUCT is licensed, not sold. This AGREEMENT does not give you any license to the copyrights or other intellectual property rights. All rights not expressly granted by this AGREEMENT are reserved to OSS Ltd.

7.1 DISTRIBUTION

You may distribute the SOFTWARE PRODUCT in unmodified form via electronic means or on CD compilations. You may not distribute the SOFTWARE PRODUCT as a part of other software without prior written permission of OutTrak Software Services Ltd. You may not sell the SOFTWARE PRODUCT unless you also sell the iMeterBox hardware.

7.2 DISCLAIMER OF WARRANTY

This SOFTWARE PRODUCT is provided as is without warranty of any kind, either express or implied, including, but not limited to, warranties of merchantability or fitness for a particular purpose. OSS Ltd. assumes no liability for damages, either direct or consequential, which may result from the use of the SOFTWARE PRODUCT.

7.3 TERMINATION

Without prejudice to any other rights, OSS Ltd. may terminate this AGREEMENT if you fail to comply with the terms and conditions of this AGREEMENT. In such event, you must destroy all copies of the SOFTWARE PRODUCT.

8. DISCLAIMER

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted.

Other names of actual companies and products mentioned herein may be the trademarks of their respective owners.

END OF DOCUMENT