
JWaveLib Documentation

Release 1.0.2

Didrik Pinte - Dipole Consulting

May 25, 2011

CONTENTS

1	Introduction	3
1.1	Features	3
1.2	How does it work	4
2	Technical overview	7
2.1	History	7
2.2	Architectural details	7
2.3	Library packages	8
2.4	Test packages	9
2.5	Integrating JWaveLib	9
3	Implementing the JWaveLib	11
3.1	Java 2 Standard Edition	11
3.2	ACTL eWON implementation	11
3.3	Siemens TC65 implementation	11
4	Support	13
4.1	Bug reports	13
4.2	Technical assistance	13
4.3	Frequently Asked Questions	13
5	JWaveLib versions	15
5.1	Version 1.0.2	15
5.2	Version 1.0.0	15
6	Contacts	17

Welcome to the JWaveLib's documentation project.

You will find in this document the latest end-user and technical documentation on the JWaveLib library and its different hardware integration. This document is still a work in progress as the library and its neighbouring application are evolving quickly.

For any missing information, unclear text or errors, please report to support@dipole-consulting.com

For information about our other products (WaveSniffer, WaveConfigurator) and services related to the JWaveLib, please contact info@dipole-consulting.com

Didrik Pinte

INTRODUCTION

JWaveLib offers a simple middleware to interface Coronis¹ sensors using the Wavenis© protocol. It allows easy communication between any Java enabled device and a network of Wavenis© sensor modules using the Coronis Systems products.

JWaveLib focuses on integrator needs by decreasing their development time, assuring code quality and fully tested Wavenis© protocol implementation. JWaveLib respects the ultra-low power philosophy of the Wavenis© modules by limiting to the strict minimum the number of requests between the WavePort and the network of modules.

1.1 Features

1.1.1 Wavenis protocol and Coronis modules

- Full Coronis sensor integration managing alarms, datalogging, extended datalogging, etc. for :
- Wavetherm Dallas (1 sensor)
- Waveflow (1 sensor)
- Wavesense (4-20mA, 0-5V, Wavetank)
- Partial support for WaveTherm Pt100/Pt1000
- Facility functions like getAdvancedDatalog(), startDatalog(), etc.
- Minimal communication between the Waveport and the modules by using the most appropriate communication mode (PTP or MULTIFRAME)

1.1.2 Q&A

- Fully unit tested and complete javadoc available
- Javadoc and class diagram can be downloaded from the Download section of our website.

1.1.3 Integration

- Object-oriented architecture allowing easy integration with custom applications
- Simple customization of the library's behaviour by configuration files and abstract classes.
- Library size is less than 60 Kb

¹ Coronis website

J2ME

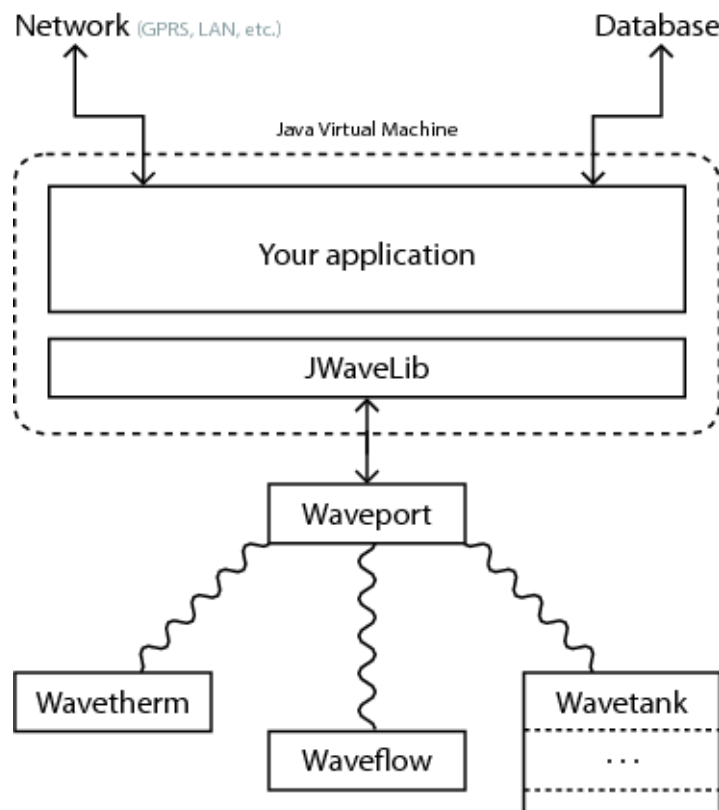
- Java J2ME CLDC 1.1 compliant. The library can be used in most embedded Java platforms but also with the latest Java Runtime Environment.
- Integrated platforms :
 - Siemens TC65,
 - ACTL eWON's 4101 GPRS

J2SE

- Support multiple RS-232 Waveport and unlimited numbers of sensors.
- Officially supported J2SE Virtual Machines :
 - Sun JDK ≥ 1.3 ,
 - OpenJDK ≥ 1.6
 - with Java Communication API 3.0 or RXTX $\geq 2.1.7$

1.2 How does it work

The JWaveLib hides all the Wavenis protocol complexity behind simple objects and data structures.



The following snippet shows how easy it can be

```
import com.coronis.modules.platform.SerialWavePort;
import com.coronis.modules.WaveFlow;
import com.coronis.modules.WavePort;
import com.coronis.modules.WaveTherm;
// initialise Waveport
```



```
WavePort wpt = new SerialWavePort("waveport1", "/dev/ttyUSB0");
// initialise a Wavetherm using no Wavetalk
JWaveLib WaveTherm wth1 = new WaveTherm("031907301989", wpt, null);
int temperature = wth1.getCurrentValue();
// initialise Waveflow using no Wavetalk
WaveFlow wfl1 = new WaveFlow("021607314289", wpt, null);
int index = wfl1.getCurrentValue();
DataSet dst = wfl1.getDailyData();
```


TECHNICAL OVERVIEW

This chapter will give an overview of the JWaveLib architecture and implementation.

2.1 History

The JWaveLib has originally been developed to be integrated into embedded Java hardware. The first running implementation was made for ACTL eWON's in 2007. The application was designed to do some data acquisition at a given frequency. Thus, the main development were focused on datalogging access. The library is now opening to support all the types of communication a user can think about using Coronis devices.

The JWaveLib is now running on different architectures from classic J2SE environments to embedded platforms like the ACTL eWON or the Siemens TC65.

2.2 Architectural details

The JWaveLib hides all of the Wavenis protocol behind easy to use objects (WavePort, WaveTherm, etc.). A rich set of exceptions are used through the library allowing the user to catch and understand what the problem is.

Using the library is limited to declaring a WavePort instance, connecting them to Modules (any Coronis module) and then using those modules. Depending on the module type, a different set of methods are available.

If needed, existing module types can be extended to implement specific functionalities.

2.2.1 WavePort

The **com.coronis.module.WavePort** class is the central point. It is in charge of doing all the job of talking and listening to an RS-232 connected WavePort. It offers the needed methods to do point-to-point or multiframe requests. Some broadcasting queries have already been tested but not officially implemented in the library.

This class is abstract because dependant on the hardware.

2.2.2 Module

The **com.coronis.module.Module** class is the parent of all the Coronis modules (WaveTalk or DataloggingModule). This class contains all the methods shared by the different Coronis modules (get/set firmware, get type, get/set datetime, etc.).

A Module has also references to a list of repeaters if any. This list of repeaters will automatically be used to optimize the type of request between the WavePort and the Module (PTP, Repeated or Multiframe).

2.2.3 DataLoggingModule

The **com.coronis.module.DataLoggingModule** class allow management of modules that have datalogging tables (simple or extended).

The DataLoggingModule class is abstract as the methods reading the data from the modules are module specific (e.g. WaveFlow stores the index information on 4 bytes while the WaveTherm Dallas stores information on 2 bytes). The specific implementation are done inside the children classes :

- **com.coronis.module.WaveTherm**,
- **com.coronis.module.WaveFlow**,
- **com.coronis.module.WaveTank**
- **com.coronis.module.WaveSense**

When retrieving data from the sensors, the methods returns **com.dipole.libs.DataSet** objects that holds the retrieved information. a DataSet is a list of **com.dipole.libs.Measure** objects and some metadata about them (last call date, ...).

This class is the primary input point when the user would probably need to implement specific behaviour dependant on their need.

2.3 Library packages

This section will describe each package of the library and will give a brief overview of their content and use. For a more detailed information, please refer to the javadoc.

The list of packages in the jwavelib.jar library :

2.3.1 com.coronis

General package holding classes shared all over the library

2.3.2 com.coronis.exceptions

Exception package grouping a nice set of exception classes. The base class of all the exceptions is **CoronisException**.

2.3.3 com.coronis.frames

Every incoming and outgoing Coronis frames are managed by this package. A factory frame builder (**CoronisFrameBuilder**) and a state machine frame reader (**CoronisFrameReader**) can read or produce any of the Coronis frames existing in this package.

2.3.4 com.coronis.logging

The JWaveLib uses a logging module a la log4j but very light. Everything is based on the **SimpleLogger** interface. The **Logger** class uses only static methods and is used all over the library. By default, it uses a **StdLogger** that outputs all the logging to the standard output. Depending on the debugging configuration, the different levels of logging will print messages. The **frame()** method allows logging of Coronis frames.

2.3.5 com.coronis.modules

The physical Coronis modules have a class counterpart. The base module is the **WavePort** and the connected module to the WavePort are all children of the **Module** class. The hierarch is the following :

- WavePort
- Module
 - WaveTalk
 - DataloggingModule
 - WaveTherm
 - WaveThermDallas
 - WaveThermPT100
 - WaveFlow
 - WaveSense
 - WaveSense4_20
 - WaveSense5V
 - WaveTank

com.coronis.modules.platform

com.coronis.modules.requests

2.3.6 com.dipole.libs

2.4 Test packages

2.4.1 com.coronis.test

2.5 Integrating JWaveLib

From an integrator point of view, the JWaveLib provides easy extension points by the use of abstract classes and the object orientation of the package :

- The hardware specific classes have all been made abstract to ensure each platform has its implementation of the classes.
- When a user wants to implement specific methods, he can derive from existing classes to implement the new methods or overload existing ones

IMPLEMENTING THE JWAVELIB

This chapter will list the implementation of the JWaveLib into different architectures.

3.1 Java 2 Standard Edition

The JWaveLib is also supported by standard Java runtime environment. Using the Java Communication API or the RXTX library, it can very easily be integrated in any application to offer the full access to the Coronis modules.

3.2 ACTL eWON implementation

This is the original development platform of the JWaveLib and as thus been thoroughly tested. This is the reference implementation of the JWaveLib.

The eWON SDK offered an easy interface to the JWaveLib. The development was mainly focused on interacting with the eWON (using the filesystem, GPRS, leds, etc.) as the JWaveLib integration was limited to implementing the WavePort abstract class for the eWON Java environment.

Further work is planned to allow more interactions with the eWON tags and other configuration.

3.3 Siemens TC65 implementation

The JWaveLib implementation into the TC65 has been tested in our lab but has not been putted under production condition.

SUPPORT

4.1 Bug reports

For any bug reports, do ask a login to support@dipole-consulting.com and report the tickets onto the JWavelib trac instance :

<http://svn.dipole-consulting.com/jwavelib/trac.cgi>

The trac is opened to all of your JWavelib clients. If you require more confidentiality, please report your bug by e-mail to support@dipole-consulting.com

4.2 Technical assistance

For technical assistance, please contact the support by e-mail : support@dipole-consulting.com

4.3 Frequently Asked Questions

This section is a copy of the FAQ content you can find on the JWavelib trac instance.

4.3.1 Supported features

- What kind of Coronis modules have been fully tested using JWavelib ?

The following list of modules have been thoroughly tested and are active on production sites :

1. WaveTherm Dallas - 1 sensor
2. WaveFlow - 1 counter
3. WaveTalk 25 mW and 500 mW
4. Serial WavePort 25/500 mW

- What is the list of all the supported Coronis modules ?

1. WaveTherm Dallas - 1 sensor
2. WaveFlow - 1 counter
3. WaveTank
4. WaveTalk 25 mW and 500 mW
5. Serial WavePort 25/500 mW

- What if the modules I want to use is not already supported by the JWavelib ?

Our politic is to support all the Coronis modules. Thus if you plan to use a module that is not actually supported by the JWaveLib, contact the support and we will give you an estimated time to get full support on this module. Lots of modules just require a very small development to work with the current JWaveLib.

For example : you need to use a WaveFlow? with two counters (A and B). Dipole Consulting will provide you a JWaveLib version supporting this type of module within one month.

- Does JWaveLib support automatic network configuration (SDP) ?

At the moment, JWaveLib does not support the SDP protocol. The feature is in the development process for milestone 2.0 if available in the Coronis modules. The SDP protocol is not already available into the Coronis modules.

4.3.2 Configuration

4.3.3 Coronis products related questions

- How can we know if the module supports extended datalogging ?

This is not possible using the Wavenis protocol. If the module has a strange behaviour when asking for the extended datalogging measures, it does probably not support extended datalogging. Only the Coronis technical support can confirm it. Send an e-mail with the module id to technicalsupport@coronis.com to check that.

Waveport

Waveflow

JWAVELIB VERSIONS

5.1 Version 1.0.2

- Released 11/21/2008
- Important bugfix for extended datalogging parameters check
- See changeset [15]

5.2 Version 1.0.0

- Released 10/01/2008
- First official release. Library used in production using eWon hardware on approximately 30 sites.

CONTACTS

For any information, please contact :

Dipole Consulting SPRL

Rue d'Arnelle, 4

1315 Incourt

Belgium

info@dipole-consulting.com

+32 10 77 90 05

+32 475 665 668

<http://svn.dipole-consulting.com/jwavelib/trac.cgi>

<http://www.dipole-consulting.com>