WebIO

UDP 2 XML
Software Application/Driver
Version 0.1

Manual, draft V1.1

www.keyeleven.com

Introduction

WebIO sends out a UDP network messages when a wireless sensor message is received. These messages contain information on the sensor reading/state.

WebIO Software Applications

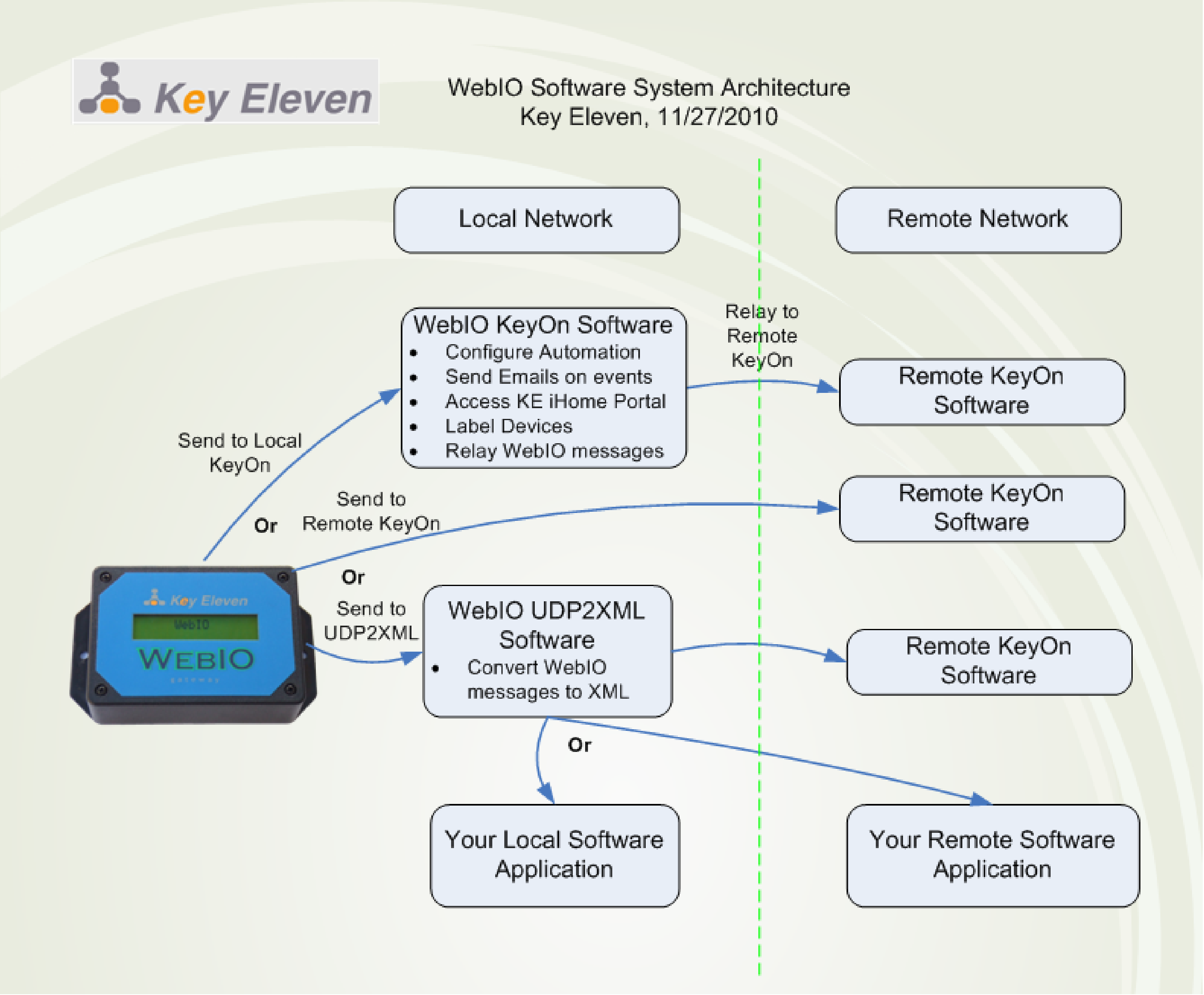
WebIO version 3 (Wireless Temperature)
The WebIO-TM software app is used to receive and process temperature sensor data and Expansion port events sent over a TCP/IP network from WebIO. These messages are sent via UDP datagram.

WebIO version 4 (Wireless Sensors)
The WebIO KeyOn software is used to receive wireless sensor data and expansion port events from WebIO over a TCP/IP network from WebIO. These messages are sent via UDP datagram.

WebIO UDP2XML Driver
The UDP2XML Driver software is used to receive WebIO network message and output the message as XML data for use by your own software application. The XML data is send out using UDP protocol to the IP address and UDP Port of your own application. WebIO UDP2XML runs on a Windows computer with the .NET runtime environment.

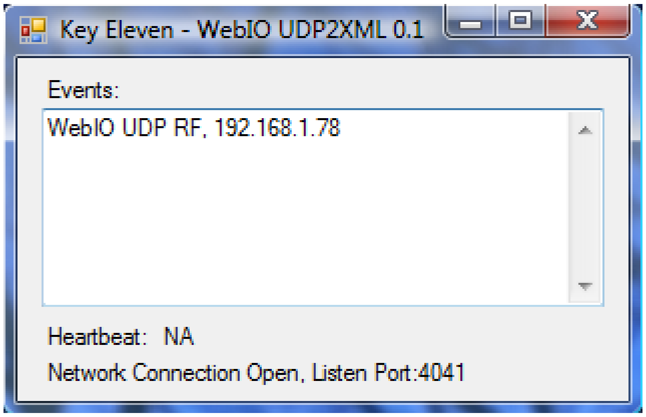
WebIO Software Architecture

The diagram below shows the many possible configurations of handling WebIO network messages using KeyOn and/or WebIO UDP2XML software.
Although WebIO can only send network messages to one IP address/Port, Both KeyOn and UDP2XML have the ability to relay the WebIO messages to a remote installation of KeyOn. UDP2XML also sends XML to your local or remote application.



UDP2XML

WebIO UDP2XML is a Windows app that requires the .Net Runtime environment.



Events:

WebIO UDP RF, 192.168.1.78

Heartbeat: NA

Network Connection Open, Listen Port:4041

To use WebIO_UDP2XML, download WebIO_UDP2XML.zip
This file will contain a directory containing 3 files: App.conf and WebIO_UDP2XML.exe, WebIO_UDP2XML.pdf
To use, configure the app.conf file for you use and have your app start WebIO_UDP2XML.exe
WebIO_UDP2XML can run on the same PC as your app or on another PC on the same Network.

App.conf
This config file is used to configure WebIO_UDP2XML. App.conf must be in the same "working directory" as WebIO_UDP2XML.exe.

App.conf contains the following parameters in key value pairs:

Key=Value	Notes
confversion=1.00	Sets the version of this config file
wip=192.168.1.78	Not currently used
wipverified=False	Not currently used
tracelog=4	Sets the tracelog level. 0-4, where 0 is no logging
udpserverport=4041	Sets the UDP port for UDP2XML to listen on.
relayon=True	Turns on the relay mode as used by KeyOn
relayurl=192.168.1.3	URL or IP address to relay WebIO network message to (send to an installation of KeyOn)
relayport=4042	Port to relay WebIO network messages (Port used by KeyOn)
heartenabled=True	Enable sending of heartbeat message via UDP indication this app is alive
heartinterval=2	Heartbeat interval in minutes
UDPXMLsendtoip=127.0.0.1	The IP address to send the XML data to (IP of PC your app runs on)
UDPXMLsenttoport=4045	The UDP port to send the XML data to (Your app's listen port)

The app.conf key value pairs marked in Red need to be set for your applications IP address and Port number. If your app is on the same PC as UDP2XML then you may use 127.0.0.1 loop back address and an available port (not the same port number used by UDP2XML to receive messages from WebIO).

XML Format

UDP2XML sends out XML representation of the UDP network message received by WebIO.
XML data nodes will vary depending on message type from WebIO.

Message types include:

- WebIO Message – contains: IP address, MAC address, Version
- RF Message – contains: readings from wireless sensors
- Temperature Message – contains: reading from a wireless temperature sensor
- Exp Message – contains Expansion port relay and input status

RF XML

XML data will contain these nodes/data:

```
<XML>
      <WEBIO DATA>
            <IP> IP Address of WebIO</IP>
            <MAC> MAC Address of WebIO </MAC>
            <RF>
                  <CHANNEL> + Sensor Channel </CHANNEL>
                  <ID> + Sensor ID as Hex Value </ID>
                  <EVENTCODE> + Sensor Event </EVENTCODE>
            </RF>
      </WEBIO DATA>
</XML>
```

Example XML file:

```
<XML>
        <WEBIO DATA>
                <IP>192.168.1.78</IP>
                <MAC>00:00:80</MAC>
                <RF>
                        <CHANNEL>6</CHANNEL>
                        <ID>01</ID>
                        <EVENTCODE>OFF</EVENTCODE>
                </RF>
        </WEBIO DATA>
</XML>
```

XML Temperature:

Example XML file:

```
<XML>
        <WEBIO DATA>
                <IP>192.168.1.77</IP>
                <MAC>10:01:75</MAC>
                <TEMPERATURE>
                        <CHANNEL>1</CHANNEL>
                        <ID>9C1</ID>
```

```
            <CELCIUS>16.3</CELCIUS>
            <FAHRENHEIT> 61.34</FAHRENHEIT>
        </TEMPERATURE>
    </WEBIO DATA>
</XML>
```