Introduction

This document describes the protocol that is used by the Accsense gateway in Accsense Direct Web (ADW) mode to communicate to a customer host computer.

Intended Audience

This document is intended to be as a guide for writing software that resides on a host computer that interfaces to the Accsense B01-02 Gateway. Typically, the host computer software intercepts, formats, displays and stores the data from the Accsense gateway. This document is also suitable for individuals that are familiar with web site configuration because the Accsense gateway protocol is based on common internet concepts.

Readers of this document should have a basic understanding of Hypertext Transfer Protocol (HTTP) 1.1.

System Overview

The system is comprised of a wireless mesh/gateway and a host computer. The wireless mesh consists of wireless sensor pods. These pods transmit information to the gateway, primarily sensor data. The gateway collects data from the wireless sensor pods and then formats this data and forwards it to the host computer using an Ethernet connection. Status information (for example, error messages) is also sent to the host computer in a similar fashion.

The gateway communications protocol is based on the REST (Representational State Transfer) architecture. Specifically, the gateway sends HTTP GET strings to the host and the host computer responds with an XML object. The host must always respond to a gateway HTTP GET string with an XML object before the next gateway HTTP GET string is sent out. The XML object that is sent to the gateway contains information that is intended to configure the gateway's operation, such as current timestamp and the measurement interval.

Gateway Requests

As mentioned previously, the gateway initiates all requests to the host computer using an HTTP GET string. Various parameters are attached to this request in the form of query parameters. The operation of the gateway request string is always represented by the OP parameter.

The basic format of the HTTP GET string is as follows:

> **GET <URL PATH>?<Operation Code>&<other query string parameters>…\r\n**
> **Host: <URL or IP address>\r\n**
> **User-Agent: accsense-gateway; linux\r\nAccept: */*\r\n**
> **\r\n**

The characters "\r" and "\n" are ASCII control characters that represent carriage return and line feed. Additionally, the protocol used is always HTTP 1.1 and, as is customary in HTTP, the HTTP packet is terminated with an additional "\r\n" character sequence.

There are 4 different types of requests: Hello, heartbeat, data, and pod configuration.

Example Gateway Requests

The following examples illustrate the gateway-generated query strings. The "\r" and "\n" characters have been replaced by carriage returns for readability.

Hello Request Example

The hello request is generated whenever a gateway has experienced a reset condition.

GET /customer/index.php?OP=40&GWMAC=0014A0004C23&GWHW=0&GWFW=0.2.3.1-P2.10&GWRADIOFW=00.03.02.03&GWRESET=48 HTTP/1.1
Host: 192.168.1.134
User-Agent: accsense-gateway; linux
Accept: */*

Heartbeat Request Example

The heartbeat request is generated by the gateway during idle periods. In other words, if the gateway has not generated any other messages to the host computer in a certain time period, the heartbeat message is generated. This message allows the host computer to ascertain that a gateway is still operational. Additionally, this message gives the host computer software an opportunity to change operational aspects

GET /customer/index.php?OP=30&GWMAC=0014A0004C23 HTTP/1.1
Host: 192.168.1.134
User-Agent: accsense-gateway; linux
Accept: */*

Data Request Example

The data request is generated whenever the gateway receives sensor data transmitted by a wireless sensor pod. The amount and type of data contained in this message (comma delimited) is dependent on the type of pod that transmitted the data message.

GET
/customer/index.php?OP=10&GWMAC=0014A0004C23&PODEUI=0014A000000014D0&PODDATA=0040,0001,0105,0243&DATATS= 1147285909&PODLQI=00F0&&PODBATT=1 HTTP/1.1
Host: 192.168.1.134
User-Agent: accsense-gateway; linux
Accept: */*

Pod Configuration

The pod configuration message is generated by the gateway whenever a pod is first associated (introduced) to the gateway's wireless mesh or upon reset or power-up of a given pod that has already been associated in the wireless network.

GET
/customer/index.php?OP=20&GWMAC=0014A0004C23&PODEUI=0014A000000014D0&PODTYPE=0103&PODHW=0000&PODFW=040206 HTTP/1.1
Host: 192.168.1.134
User-Agent: accsense-gateway; linux
Accept: */*

Error String

If the gateway encounters an error, it will generate a message similar to the following:

GET
/customer/index.php?OP=50&GWMAC=0014a0000abc&GWFW=0.4.7.3&GWERRTS=1150323750&GWERRTS=ERROR…WSM%msg%is%NULL HTTP/1.1
Host: 192.168.1.134
User-Agent: accsense-gateway; linux
Accept: */*

Request Query String Format

As explained previously, the gateway relays information to the host computer in the form of an HTTP GET string. The details specifying the message content are conveyed using a query string. The query string variable format is detailed below:

GWMAC

This parameter is contained in every gateway-generated request and identifies the source of the request.

OP

This parameter details the operation for the query string and is contained in every gateway-generated request. The values for this field include:

- 10: Request is a pod data message. The following sub-parameters are contained in a data request:
    - **PODEUI** – This 16 digit hexadecimal number identifies the sensor pod that is reporting data
    - **PODDATA** – This parameter contains a list of comma delimited, 4 digit hexadecimal (16 bit) numbers. These numbers represent the raw A/D values that are captured in the sensor pod. The coefficients necessary to convert this raw data into meaningful data must be applied by the host computer software. The order and number of parameters is determined by the Pod Configuration message.
    - **DATATS** – This parameter represents the time, in Unix epoch seconds, that the sensor data was captured.
    - **PODLQI** – The Link Quality Indication (LQI) of the pod's radio receiver. This parameter can be used to indicate the relative strength of the radio signal at the pod. This number has a range from 0000 to 00FF hex, where 00FF represents the best signal strength.
    - **PODBATT** – This parameter contains a value of 1 when the sensor pod is on battery power, 0 if it is line-powered.
- 20: Request is a pod configuration message. Each pod reports its configuration whenever it is associated with a wireless mesh or upon reset. The following sub-parameters are contained in a pod configuration request:
    - **PODEUI** – This 16 digit hexadecimal number identifies the sensor pod that is reporting data
    - **PODTYPE** – This parameter indicates the pod type. The host computer should make an association between the pod EUI and the pod type.
    - **PODHW** – Indicates the hardware version of the pod
    - **PODFW** – Indicates the firmware version of the pod
- 30: Request is a gateway heartbeat. As mentioned above, this request is generated during idle periods
- 40: Request is a gateway hello. This request is generated each time the gateway resets.
    - **GWHW** – Indicates the gateway hardware version
    - **GWFW** – Indicates the gateway firmware version
    - **GWRADIOFW** – Indicates the firmware version of the gateway radio
    - **GWRESET** – Indicates the reason that the gateway reset. Possible values include:
        - 42: Reset Button
        - 44: Power Up
        - 46: Software crash
        - 50: Brownout
        - 52: Clock failure
        - 54: Software reset

- 50: Request is an error/status string. This request is generated whenever an important system event has occurred (e.g. suspected radio control failure or memory allocation problems). The following sub-parameters are contained in an error/status request:
    - **GWFW** – The firmware version of the gateway
    - **GWERRTS** – The approximate time that the error occurred
    - **GWERRSTR** – An ASCII string which contains a description of the error/status

Host Computer Responses

After receiving a request from the gateway, the host computer must transmit a response to the gateway. The response is formatted as an HTTP message with an XML object attached in the body. The typical response should look similar to the following example:

```
HTTP/1.1 200 OK\r\n
Date: Sun, 14 May 2006 23:44:41 GMT\r\n
Server: Apache/2.0.58 (Win32) PHP/5.1.4\r\n
X-Powered-By: PHP/5.1.4\r\n
Content-Length: 187\r\n
Content-Type: text/html\r\n
\r\n
<?xml version="1.0" encoding="utf-8" ?>\n
<hostrsp>\n
  <GW_MAC>0014a000000c</GW_MAC>\n
  <TIME>5/14/06 11:44:41 PM</TIME>\n
  <MEASUREMENT_PERIOD_MS> 1000 </MEASUREMENT_PERIOD_MS>\n
</hostrsp>\n
```

The first 7 lines of the response represent the HTTP part of the message and the last 6 lines of the message represent the XML part of the message. All XML responses to the gateway must start with the XML prolog and (<?xml version="1.0" encoding="utf-8" ?>). The body of the XML object then starts with the root element (<hostrsp>…</hostrsp>).

Host Gateway XML Response Element Definitions

GW_MAC XML Element

This element contains the hex-ASCII representation of the MAC (serial number) of the gateway that originated the request. This element is used by the gateway for validation purposes and, therefore, is required in every XML response.

TIME XML Element

This element contains the host computer's time when the gateway request was received. It is only required when the gateway hello request (OP=40) is received by the host computer. Otherwise, it is optional. However, it is recommended that the host computer periodically respond (approximately once per 30 minutes) with the timestamp in order to prevent clock drift between the gateway and the host computer. Additionally, it is recommended, but not required, that the time that is generated is a representation of GMT in order to prevent potential problems associated with Daylight Savings Time.

MEASUREMENT_PERIOD_MS XML Element

This element contains the desired measurement period in milliseconds. Currently, the minimum measurement period is 1000 milliseconds. This element is only required when:

1. Gateway hello message is received by the host computer
2. The desired measurement interval has changed

Please note that the measurement interval may require between 30 seconds to 5 minutes depending on the previous measurement interval selected.

DISCRETE_CFG XML Element

This element contains the configuration for discrete I/O. It is only applicable to pods which support configurable discrete I/O (A1-08 through A1-12). Discrete I/O configuration consists of two 16 bit masks, expressed as ASCII hexadecimal numbers:

1. Input/output mask. If a bit is set to 0, the corresponding I/O is configured as an input bit. Otherwise, it is configured as an output.
2. Output value mask. If a discrete I/O is configured as an output in the Input/Output mask, this value indicates the value that should be output on the corresponding output pin. A '1' value will output a high level whereas a '0' value will output a low level.

Note, for example, the following XML object:

```
<?xml version="1.0" encoding="utf-8" ?>\n
<hostrsp>\n
  <GW_MAC>0014a000000c</GW_MAC>\n
  <DISCRETE_CFG>0014a0000001123e 0002 0000</DISCRETE_CFG>
  <DISCRETE_CFG>0014a00000011255 0003 0002</DISCRETE_CFG>
</hostrsp>\n
```

This XML object configures the discrete I/O of two separate pods. The first pod with serial number **0014a0000001123e** is configured with pin 1 as an input and pin 2 as an output with a low level. Pod **0014a00000011255** is configured with both pins as output: Pin 1 is configured with a low level and pin 2 is configured with a high level.

NOTE: Each DISCRETE_CFG element describes only one pod I/O configuration.
NOTE: Discrete I/O configurations are persistent per gateway/pod power cycle. Therefore, it is recommended that the host application refresh these states (preferably per measurement interval) in case of gateway/pod reset or power cycle.

APPENDIX A: Recommended Pod Configuration

Pod Measurement Period vs. Power Supply

If the configured measurement period is less than 30 seconds, it is recommended that the pods in the mesh be AC powered due to battery life and pod wakeup latency considerations.

Number of Pods in the Mesh

Due to the limitations of the wireless mesh bandwidth, there is a strong dependency between the measurement period and the maximum number of pods that the mesh can reliably support. This dependency is summarized in the following table:

# of Pods	20	15	10	5	2
16	100%	99%	95%	95%	75%
12	100%	99%	95%	95%	94%
8	100%	99%	96%	96%	96%
4	100%	99.5%	99%	99%	99%
2	100%	100%	99%	99%	99%
1	100%	100%	100%	100%	100%

A1-01 Pod

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
4-20ma	0	0.020345	0	0	mA
0-5V	0	0.0048828	0	0	Volts
Digital 1	0	1	0	0	
Digital 2	0	1	0	0	
Event Counter LSB	0	1	0	0	
Event Counter MSB	0	1	0	0	
Temperature	-39.6	0.01	0	0	°C
Humidity	-4	0.0405	-0.0000028	0	%RH
Vibration	100	-0.097752	0	0	%(of 6 Gs)
Acoustic	100	-0.097752	0	0	%
Supply Voltage	0	0.0048828	0	0	Volts
Light	0	0.097752	0	0	%

A1-02

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
RTD	-244.806034	0.714371	8.28744E-05	1.63706E-08	°C
Thermistor	113.719	-0.33	0.000438	-2.573E-07	°C
Ambient Temp	158.36	-0.20743	0	0	°C
Thermocouple	1240.74	-1.61559	0.000663263	-4.50566E-07	°C
0-5V	0	0.0048828	0	0	Volts
Supply Voltage	0	0.0048828	0	0	Volts
Digital	0	1	0	0	

A1-03

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
Temperature	-39.6	0.01	0	0	°C
Humidity	-4	0.0405	-0.0000028	0	%RH
Vibration	100	-0.097752	0	0	%(of 6 Gs)
Acoustic	100	-0.097752	0	0	%
Supply Voltage	0	0.0048828	0	0	Volts
Light	0	0.097752	0	0	%

A1-04

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
4-20ma	0	0.020345	0	0	mA
0-5V	0	0.0048828	0	0	Volts
Digital 1	0	1	0	0	
Digital 2	0	1	0	0	
Event Counter LSB	0	1	0	0	
Event Counter MSB	0	1	0	0	
Ambient Temp	158.36	-0.20743	0	0	°C
Supply Voltage	0	0.0048828	0	0	Volts

A1-05

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
Temperature	-39.6	0.01	0	0	°C
Humidity	-4	0.0405	-0.0000028	0	%RH
Supply Voltage	0	0.0048828	0	0	Volts

A1-06

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
RTD	-244.806034	0.714371	8.28744E-05	1.63706E-08	°C
Ambient Temp	158.36	-0.20743	0	0	°C
Supply Voltage	0	0.0048828	0	0	°C
Digital	0	1	0	0	

A1-07

Sensor	Coeff 1	Coeff 2	Coeff 3	Coeff 4	Units
Digital 1	0	1	0	0	
Digital 2	0	1	0	0	
Event Counter LSB	0	1	0	0	
Event Counter MSB	0	1	0	0	
Ambient Temp	158.36	-0.20743	0	0	°C
Supply Voltage	0	0.0048828	0	0	Volts

The pod reports a single, 4-diget hex value. In order to convert that value into a standard measurement, you must first convert it to decimal, and then use the following equation:

Value = Coeff1 + Coeff2*[ReportedValue] + Coeff3*[ReportedValue]^2 + Coeff4*[ReportedValue]^3

If you have any questions about how to apply these coefficients, please contact us at:
support@accsense.com