



## Software Design Specification

### Z/IP Resource Directory

<b>Document No.:</b>	SDS11633
<b>Version:</b>	
<b>Description:</b>	Mechanisms for managing resource information needed for service discovery inside a Z/IP network and across IP subnets
<b>Written By:</b>	AES;ABR;JRM;JFR;BBR
<b>Date:</b>	
<b>Reviewed By:</b>	ABR;JFR;JRM;NTJ;AES;JBU
<b>Restrictions:</b>	Public

#### Approved by:

Date	CET	Initials	Name	Justification
2018-03-06	09:25:44	NTJ	Niels Thybo Johansen	

This document is the property of Silicon Labs. The data contained herein, in whole or in part, may not be duplicated, used or disclosed outside the recipient for any purpose. This restriction does not limit the recipient's right to use information contained in the data if it is obtained from another source without restriction.



**REVISION RECORD**

<b>Doc. Ver.</b>	<b>Date</b>	<b>By</b>	<b>Pages affected</b>	<b>Brief description of changes</b>
3	20160823	ABR	All	First revision for public release
4	20180306	BBR	All	Added Silicon Labs template

# Table of Contents

<b>TERMS AND ABBREVIATIONS .....</b>	<b>1</b>
<b>1 INTRODUCTION.....</b>	<b>1</b>
1.1 Purpose .....	2
1.2 Relation to classic Z-Wave.....	2
1.3 Relation to other standards .....	2
1.1 Terms used in this document .....	2
<b>2 Z/IP RD MANAGEMENT .....</b>	<b>3</b>
<b>3 Z/IP SERVICE DISCOVERY PRINCIPLES .....</b>	<b>4</b>
3.1 Terminology.....	4
3.2 The Resource Directory .....	4
<b>4 REQUIREMENTS.....</b>	<b>5</b>
4.1 Z/IP RD.....	5
4.1.1 Building the database.....	5
4.1.2 Database records.....	5
4.1.2.1 Node Name .....	7
4.1.2.2 End Point Location .....	7
4.1.2.3 End Point Name .....	7
4.1.2.4 End Point Information.....	8
4.1.2.5 End Point ID .....	9
4.1.2.6 HomeID and NodeID .....	9
4.1.2.7 Icon Types.....	10
4.1.2.8 Communication Mode .....	10
4.1.2.9 Operational Mode.....	10
4.1.3 Maintaining a valid database.....	10
4.1.4 Preventing database divergence .....	11
<b>5 INTERFACING THE Z/IP RD TO AN MDNS RESPONDER .....</b>	<b>12</b>
5.1 ResourceProbe() .....	12
5.2 ResourcePropertiesClear().....	12
5.3 ResourcePropertiesSet().....	12
5.4 ResourceAddressSet() .....	13
5.5 ResourceServicesSet() .....	14
<b>REFERENCES .....</b>	<b>16</b>

## TERMS AND ABBREVIATIONS

Term	Explanation
DNS	Domain Name System
FLiRS	Frequently Listening Routing Slave - a Z-Wave specific variant of a FLN referring to the Routing Slave library
FLN	Frequently Listening Node A battery-powered node type which may be reached within a few 100 ms. Typically used for applications such as door bells, networked smoke alarms, etc.
HAN	Home Area Network
IP service	A dedicated function in an IP host accessed via a UDP or TCP port. Z-Wave is registered as an IP service running on UDP port 4123. The IP service named "Z-Wave" is not to be confused with the term "Z-Wave service".
mDNS	Multicast DNS
NWA	Network-Wide Association
PAN	Personal Area Network. Term used for same class of networks as HAN.
RD	Resource Directory
Z-Wave resource	A Z-Wave application running in any IP based environment, e.g. Z-Wave network, Ethernet or IEEE 802.15.4. The application is accessed via IP. An IP host may implement more than one Z-Wave resource. Each Z-Wave resource may provide one or more Z-Wave services.
Z-Wave service	A Z-Wave service is accessed via a given command class In mDNS terms, this is a sub-service, the IP service "Z-Wave" is the real service.
Z/IP	Z-Wave for IP
WakeUp node	A battery operated node which wakes up autonomously at long time intervals. The WakeUp node can only be reached via a mailbox function provided by the Z/IP Router

## 1 INTRODUCTION

Z-Wave evolved from a remote control-centric idea. A customer would buy a remote control and three plug-in modules. The remote control was a portable controller knowing the entire network. When buttons on the remote control were to be controlling the plug-in modules, the setup of the buttons was manipulated from within the remote control menu system.

Support for sensor nodes was added later. The remote control would set up associations in a sensor device to make it control a plug-in module.

Going towards larger installations, there is a need for installer tools that will present all resources of an installed network; even if the system spans multiple IP subnets on heterogeneous PHYs – and even if some resources are battery operated and sleeping most of the time.

## 1.1 Purpose

The purpose of the Z-Wave Resource Directory (RD) is to ensure that sufficient data is made available to intelligent network management systems. The following requirements apply:

- PHY agnostic
- Highest possible degree of compatibility with discovery technologies used by other major technologies in the industry such as IETF CoAP
- Support for battery operated devices
- Put only a minor load on the HAN network
- Enable large installations like hotels; several buildings, wings, levels, rooms, i.e. many subnets
- Enable call-center support by allowing service providers to remotely set up remote controls and sensors for non-technical users

## 1.2 Relation to classic Z-Wave

The Z-Wave Resource Directory (RD) provides access to information regarding IP hosts which implement Z-Wave services. A rich set of node properties may be queried from the Z-Wave Resource Directory (RD).

This document makes no assumptions as to how the RD is reached or how Z-Wave resources are. Information stored in the RD MAY be accessed via mDNS or other discovery protocols.

## 1.3 Relation to other standards

The Z/IP RD description leverages on the data structures and principles defined for IETF CoRE Resource Directory [7].

For backwards compatibility with classic Z-Wave properties, there are a few differences from the CoRE RD architecture.

A CoRE RD Endpoint is an IPv6 address and a port number.  
A Z/IP RD End Point is an IP address and a Z-Wave End Point ID.

A CoRE resource delivers data to the CoRE RD in a push style.  
A Z/IP RD retrieves data from Z-Wave resources in a pull style.

Some information is converted and merged by the Z/IP RD based on limited binary information available in classic Z-Wave nodes.

### 1.1 Terms used in this document

The guidelines outlined in RFC 2119, [1] apply. Essentially, the key words "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

NOTE: This specification may refer to Internet Drafts. Internet Drafts are by definition "work in progress". An implementation SHOULD NOT reference to Internet Drafts. An implementor SHOULD make sure that the Internet Draft has become an RFC at release time.

## 2 Z/IP RD MANAGEMENT

Node name and location information managed by the Z/IP RD is mDNS compatible and **MUST** be accessed via the Z/IP Naming and Location Command Class.

The RD manages UTF-8 strings up to 63 octets long and identified by [IP address, End PointID]. The information is used for constructing DNS names for mDNS service discovery.

When a user replaces a resource, i.e. assigns the same name to a new resource, the old resource name **MUST NOT** be notified in a Goodbye message. Rather, the resource **MUST** be re-announced so that clients can re-resolve resource properties.

When receiving a Z/IP Name Set command with the Replace option, the Z/IP RD **MUST** reset the old resource name to an auto-generated name based on the product name and unique values. If the resource is End Point 1 or higher, the autogenerated name **MUST** be <End PointID>.<resource name for End PointID 0>, e.g. "1.TV Powerstrip".

### 3 Z/IP SERVICE DISCOVERY PRINCIPLES

Identification of Z-Wave resources via the Z/IP Resource Directory (RD) MAY be done using mDNS [5]. The Z/IP RD is populated via gathering of node information.

This document only addresses the management of the Z/IP RD.

#### 3.1 Terminology

An IP host may implement multiple Z-Wave resources. The IP address identifies the IP host. The UDP port 41230 provides secure access to Z-Wave resources. The Z/IP End Point identifies the individual Z-Wave resource. A Z-Wave resource implements one or more Z-Wave services via Z-Wave Command Classes.

The Z/IP End Point identifies individual Z-Wave resources within an IP host.

As an example, a power strip may be accessed via the IPv6 address and UDP port [fd06:1e14:c501:ffff::ff:fe00:1]:41230. A Z/IP message pointing to Z/IP End Point #0 controls the global device. For instance Z/IP End Point #0 may control all four outputs of the power strip and Z/IP End Points #1 thru #4 controls each individual output of the power strip. However, this behavior is device specific..

End Point numbering allows a graphical GUI to group resources within the same graphical frame. The host name is the same for all End Points.

#### 3.2 The Resource Directory

The Z/IP RD organizes data so that various discovery protocols may be used to discover Z-Wave resources in a specific HAN or to make wildcard searches for a particular Z-Wave service (supporting a particular Z-Wave Command Class).

mDNS is one such discovery protocol. The use of mDNS for discovery of Z-Wave resources is described in [5].

The Resource Directory implements cache monitoring to ensure freshness of cache contents.

## 4 REQUIREMENTS

### 4.1 Z/IP RD

#### 4.1.1 Building the database

A Z/IP RD SHOULD build its database during node inclusion; querying the properties of each Z-Wave node.

A Z/IP RD MAY monitor traffic from the HAN. When traffic from an unknown node is detected, node properties may be requested from the node.

Each Z-Wave resource MUST be identified by an IP address and a Z/IP End Point ID.

#### 4.1.2 Database records

The following information MUST be available in the Z/IP RD.

The construction of mDNS announcements from contents of a CoRE RD is defined in [8]. The CoRE properties listed in Table 1 are defined in [7].

Readers guidance for Table 1:

- Some lines of the Z/IP Property column are marked by the tag “(same)”. This indicates that this property is common to all resources (End Points) of the actual node)
- The CoRE Property column also contains pointers to the mDNS records that receive the actual information



**Table 1, Z/IP RD database entry structure and CoRE RD equivalents**

Z/IP property (same for all End Points)	Range	CoRE property (mDNS)	Range	Comment
WakeUp Interval (same)	1s.. 16777215s (24 bits)	Lifetime (lt)	60s.. 4294967295s (32 bits)	Same unit and resolution. Z/IP RD MUST limit the value to 16777215s. (194 days)
Node address (same)	IP address	Host address (AAAA)	IP address	Information available via Z-Wave Network Management
Node Name (id=0)  (same)	1..63 octets (UTF-8)	Host (h)  (SRV host)	1..63 octets (UTF-8)	Information configured via Z-Wave Network Management
End Point Location (id=>0)	1..63 octets (UTF-8)	Domain (?)  (PTR target)	1..63 octets (UTF-8)	Information configured via Z/IP RD Management
End Point Name (id>=0)	1..63 octets (UTF-8)	Instance (ins)  (PTR target)	1..63 octets (UTF-8)	Information configured via Z/IP RD Management
End Point Information	1..63 octets (binary)	Type (rt)  (TXT info=)	1..63 octets (binary)	Static information available from the Z-Wave node  Extended Node Information format; refer to 4.1.2.4.
End Point ID (id>=0)	1 octet (binary)	  (TXT epid=)		Static information available from the Z-Wave node.  Also used to create unique names.
HomeID and NodeID (same)	-	  (SRV, PTR and AAAA)		Used to create unique names in lack of a unique MAC address
Icon	2+2 octets (enumerated)	  (TXT icon=)		Icon types maintained by the Z/IP RD. One is for user GUIs the other is for installer GUIs.
Communication Mode (same)	1 octet (enumerated)	  (TXT mode=)		Communication Mode information maintained by the Z/IP RD  Communication Mode and Operational Mode are reported in one TXT record in DNS-SD

Z/IP property (same for all End Points)	Range	CoRE property (mDNS)	Range	Comment
Operational Mode  (same)	1 octet (flags)	  (TXT mode=)		Operational Mode information maintained by the Z/IP RD  Communication Mode and Operational Mode are reported in one TXT record in DNS-SD
ManufacturerID ProductType ProductID  (same)	3 x 2 octets (binary)	  (TXT productID=)		Static information available from the Z-Wave node – if implemented.

#### 4.1.2.1 Node Name

The Node Name MUST be the same for all End Points of a particular node. The Node Name MUST NOT be exposed to end users via mDNS. Rather, the End Point Name MUST be used to represent resources. The Z/IP RD MUST auto-generate a Node Name as follows:

zw<HomeID><NodeID>

The user MUST NOT leave the name field blank (zero-length).

#### 4.1.2.2 End Point Location

Named the Resource Location by the Z/IP mDNS Responder, this information is dedicated to a single End Point.

The per-End Point relation allows an end-user to use a multi-outlet power strip for a system spanning several locations, e.g.

*all-lamps.garden*      as well as      *all-lamps.swimmingpool*

The location information MAY be the same for all End Points.

The user MAY leave the location field blank (zero-length).

#### 4.1.2.3 End Point Name

Named the Resource Name by the Z/IP mDNS Responder, this information is dedicated to a single End Point.

When discovering Z-Wave resources for the first time, the Z/IP RD MUST import the available Name and Location Information stored in Z-Wave nodes. The Z/IP RD MUST auto-generate unique resource names anyway in the following cases:

- Name and Location command class is not supported
- No Name and Location information is available in the node
- Identical Name and Location information is gathered from multiple nodes

The Z/IP RD maintains Name and Location information for each resource. If the user specifies Name and Location during inclusion of a new node, the same resource name MUST be assigned to all End Points

of that node. A unique label **MUST** be appended to each End Point to achieve uniqueness of resource names. The user **MAY** subsequently change any Name and Location information.

The End Point Name **MUST NOT** be the same for multiple End Points if the End Point Location is the same.

If both End Point Name and End Point Location are empty, the Z/IP RD **MUST** auto-generate an End Point Name as follows:

```
<ManufacturerString><ProductString> [<HomeID><NodeID><End PointID>]
```

If no product information is available, the Z/IP RD **MUST** auto-generate an End Point Name as follows:

```
<DeviceType> [<HomeID><NodeID><End PointID>]
```

Users may enter identical information for more resources. If several End Point Name and End Point Location entries are identical, the Z/IP RD **MUST** auto-append unique data to each of the conflicting End Point Names:

```
<user provided resource name> [<HomeID><NodeID><End PointID>]
```

If End Point Name and End Point Location entries are unique the Z/IP RD **SHOULD NOT** auto-append any additional data.

#### 4.1.2.4 End Point Information

The End Point Information field may hold up to 63 binary octet values.

The Z/IP RD maintains an extended version of the Node Information frame format.

**Table 2, Z/IP RD format for End Point Information**

Z/IP RD Node Information format	
Generic Device Class	
Specific Device Class	
xCommand Class 1	(Support)
...	(Support)
xCommand Class m	(Support)
xCommand Class m+1	(COMMAND_CLASS_MARK)
xCommand Class m+2	(Control)
...	(Control)
xCommand Class n	(Control)

A Z-Wave node may implement only supported command classes. Thus, a Z-Wave node **MAY** omit the `COMMAND_CLASS_MARK` and following controlled command classes.

In case a Z-Wave node has no supported command classes and that node advertises command classes which can control other nodes, the command class list **MUST** start with the identifier `COMMAND_CLASS_MARK`.

The extended version of the Node Information frame format appends specific type information to the command classes such as the Multilevel Sensor Command Class.

The concept is identical to the behavior when exchanging extended command classes. One or two bytes may follow the first byte, depending on the command class in question.

Two examples are provided below:

**Table 3, Meter Command Class entry for Z/IP RD Node Information structure**

Meter Command Class entry for Z/IP RD Node Information structure
Command Class = Command Class Meter
Meter Type = Electric Meter

**Table 4, Multilevel Sensor Command Class entry for Z/IP RD Node Information structure**

Multilevel Sensor Command Class entry for Z/IP RD Node Information structure
Command Class = Command Class Multilevel Sensor
Sensor Type = Power
Sensor Scale = W

A requesting client MAY query the sensor scale. It may for instance matter if the power sensor measures electricity or gas.

It is NOT RECOMMENDED that the Sensor Scale is queried if all possible outcomes can be used. For instance Air Temperature may be returned in Celsius or Fahrenheit. Most receiving hosts will be able to handle both temperature units.

The Z/IP RD MUST offer query records for [Sensor Type] only as well as for each [Sensor Type + Sensor Scale] tuple; e.g.

- 3101: Air Temperature
- 310100: Air Temperature, Celsius

If the Multilevel Sensor supports more than one type, individual entries MUST be created for each type. If the Multilevel sensor supports more than one scale, individual entries MUST be created for each scale.

#### Meter resources

The Z/IP RD MUST offer query records for [Meter Type] only as well as for each [Meter Type + Meter Scale] tuple; e.g.

- 3201: Electric Meter
- 3201??: Electric Meter, kWh

If the Meter supports more than one type, individual entries MUST be created for each type.

If the Meter supports more than one scale, individual entries MUST be created for each scale.

#### **4.1.2.5 End Point ID**

A single byte value indicating the End Point ID. A resource is completely identified by the combination of the Host IP address and the End PointID.

The End PointID is included in the generation of unique resource names.

#### **4.1.2.6 HomeID and NodeID**

The HomeID and NodeID values are used in the reactive generation of unique resource names and for auto-generation of resource names as well as for host names.

#### 4.1.2.7 Icon Types

The Icon field contains two Icon Type values.

Each End Point has two such icons.

Octets 0..1 carry an icon type to be used for an installer tool GUI, while octets 2.3 carry an icon type to be used for an end user GUI.

#### 4.1.2.8 Communication Mode

The Communication Mode field is maintained by the Z/IP RD. The field may hold the following values:

- [nonlistening](#)
- [alwayslistening](#)
- [frequentlylistening](#)
- [mailbox](#)

The Communication Mode value **MUST** be an enumerated value chosen from the list above. In a typical implementation the Communication Mode value remains static during the lifetime of a node.

#### 4.1.2.9 Operational Mode

The Operational Mode field is maintained by the Z/IP RD. The field may hold a number of flags concurrently:

- [Low Battery](#)
- [Failing](#)

The Operational Mode value reflects the current state of the node.

The failing mode **MUST** indicate that the Z/IP RD has failed to communicate to the node.

The Low Battery mode **MUST** indicate that the Z/IP RD has received Low Battery notifications from the node.

#### 4.1.3 Maintaining a valid database

To ensure the reliability of the database it must be ensured that nodes appearing in the database are also alive in the network. In most cases, data does not change, but nodes may disappear due to low battery, technical failure or other reasons.

This section applies to resources which implement the WakeUp Command Class:

If a resource timestamp is older than three times the WakeUp Notification Interval, the resource **MUST** be marked as failing. Failing resources **MUST** be announced during service discovery.

A resource **SHOULD** be marked as operational at any time upon the reception of a new WakeUp Notification.

This section applies to resources which do not implement the WakeUp Command Class:

The Z/IP RD **MUST** send NOPs on a periodic basis to make sure that the database stays valid. If not receiving responses for 3 consecutive NOPs, the Z/IP RD **MUST** mark the resource as failing.

The interval used for periodic NOPs **SHOULD** be 70 minutes.

#### **4.1.4 Preventing database divergence**

This section presents a requirement to the Z/IP Router, but requires the use of the Z/IP RD.

In case a node is included in the Z-Wave network via “Replace Failed”, the operation SHOULD be reported to the Z/IP RD by the Z/IP Router application.

If the database contents for the resource do not match the capabilities of the new node, the resource MUST be re-announced with its new properties. As an example, a dimmer may be replaced by an on-off switch module.

## 5 INTERFACING THE Z/IP RD TO AN MDNS RESPONDER

The mDNS responder is essentially a slim interface layer to a DNS server. The DNS server must support mDNS transmissions, allow leading underscores in DNS record elements and tolerate binary data and UTF-8 characters in TXT records.

The following interface functions must be made available to the Z/IP RD to announce and de-announce resources in mDNS. Refer to [5] for details on the mDNS responder.

The mDNS responder MUST support mDNS queries for `_services._dns-sd._udp.local.` ; returning the answer `_z-wave._udp.local.`

There is a DNS limit of 63 octets for the part left of “`_z-wave._udp`”.

### 5.1 ResourceProbe()

As specified in [9], a new resource name MUST NOT be announced before the mDNS responder has ensured that the name is unique.

This call may take time. An implementation must provide a call-back that informs the Z/IP RD on the completion of the call and the result.

### 5.2 ResourcePropertiesClear()

NOT zero terminated!

// Clear TXT and SRV records for actual resource

- ResourcePropertiesClear(
  - namelen, End Pointname,
  - locationlen, End Pointlocation,
  - homeid, nodeid, End Pointid)

Form the string

```
"<resourcename>._z-wave._udp.local."
```

Use the string as key for removing SRV and TXT entries from the mDNS server database.

### 5.3 ResourcePropertiesSet()

NOT zero terminated!

// Create TXT and SRV records for actual resource

- ResourcePropertiesSet(
  - namelen, End Pointname,
  - locationlen, End Pointlocation,
  - homeid)

- nodeid
- End Pointid
- infolen, End Pointinformation
- Icon
- CommMode, OperMode
- ManufacturerID
- ProductType
- ProductID
- ManufacturerString
- ProductString
- ProductURL
- PictureURL

Create SRV entry following this instruction:

```
"<resourcename>._z-wave._udp.local.
<TAB>SRV<TAB>0 0 4123 zw<HomeID><NodeID>.local."
```

Create TXT records following this instruction:

Each <record\_len> field must be set to the number of octets occupied by the remaining record.

```
"<resourcename>._z-wave._udp.local.
<TAB>TXT<TAB>
<record_len>txtvers=1
<record_len>info=<End Pointinformation>
<record_len>epid=<End Pointid>
<record_len>icon=<InstallerIcon><UserIcon>
<record_len>mode=<CommunicationMode><OperationalMode>
<record_len>productid=<ManufacturerID><ProductType><ProductID>
<record_len>product=<ManufacturerString><ProductString>
<record_len>producturl=<ProductURL>
<record_len>pictureurl=<PictureURL>
```

Before generating the info record, the following command classes **MUST** be filtered out:

- Z-Wave Protocol command classes
- Basic Command Class
- Others?

#### 5.4 ResourceAddressSet()

// Create AAAA record for actual resource

- ResourceAddressSet(
  - nodeaddress
  - homeid
  - nodeid
  - ProductIDString

Create AAAA record following this instruction:

```
"zw<HomeID><NodeID>.local.<TAB>AAAA<TAB><nodeaddress>SRV<TAB>"
```



## 5.5 ResourceServicesSet()

### NOT zero terminated!

// Create PTR records for actual resource

- ResourceServicesSet(
  - namelen, End Pointname,
  - locationlen, End Pointlocation,
  - infolen, End Pointinformation

Create one PTR record for each Command Class (named <CmdCls>) listed in End Pointinformation:

```
"_<CmdCls>._sub._z-wave._udp.local.
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

This record type MUST also be created for each Meter, Alarm and Sensor Command Classes.

For Meter Command Class:

For each supported Meter Type, create one PTR record:

```
"_<CmdCls><MeterType>._sub._z-wave._udp.local.
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

For Multilevel Sensor Command Class:

For each supported Sensor Type, create one PTR record:

```
"_<CmdCls><SensorType>._sub._z-wave._udp.local.
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

For each supported Sensor Scale of each sensor Type, create one PTR record:

```
"_<CmdCls><SensorType><SensorScale>._sub._z-wave._udp.local.
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

For Alarm Command Class:

For each supported Alarm Type, create one PTR record:

```
"_<CmdCls><AlarmType>._sub._z-wave._udp.local.
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

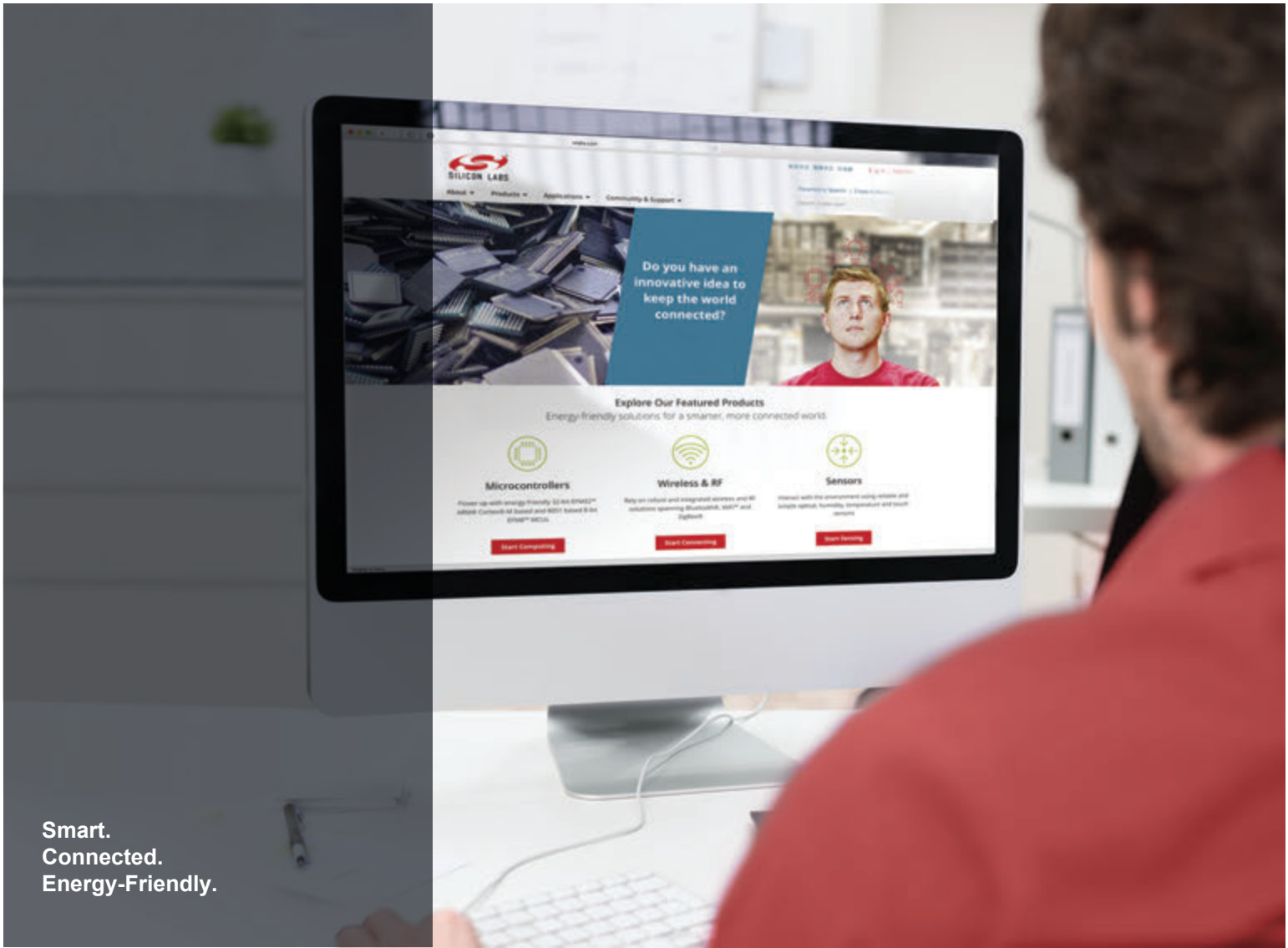
For Alarm Sensor Command Class:

For each supported Alarm Sensor Type, create one PTR record:

```
"_<CmdCls><AlarmSensorType>._sub._z-wave._udp.local.  
<TAB>PTR<TAB><resourcename>._z-wave._udp.local."
```

## REFERENCES

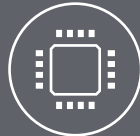
- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] IETF, draft-ietf-core-coap-02, Constrained Application Protocol (CoAP)
- [3] IETF, draft-shelby-core-link-format-00, CoRE Link Format
- [4] IETF, draft-dt-roll-rpl, RPL: IPv6 Routing Protocol for Low power and Lossy Networks
- [5] Silicon Labs, SDS11756, Z/IP DNS Discovery support (DNS-SD, mDNS)
- [6] IETF, draft-ietf-roll-p2p-rpl, RPL-P2P: Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks
- [7] IETF, draft-shelby-core-resource-directory, CoRE Resource Directory
- [8] IETF, draft-lynn-core-discovery-mapping, CoRE Link-Format to DNS-Based Service Discovery Mapping
- [9] Stuart Cheshire & Daniel Steinberg, Zero Configuration Networking: The Definitive Guide, ISBN:0596101007



Smart.  
Connected.  
Energy-Friendly.



**Products**  
[www.silabs.com/products](http://www.silabs.com/products)



**Quality**  
[www.silabs.com/quality](http://www.silabs.com/quality)



**Support and Community**  
[community.silabs.com](http://community.silabs.com)

**Disclaimer**

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice and limitation to product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Silicon Labs shall have no liability for the consequences of use of the information supplied herein. This document does not imply or express copyright licenses granted hereunder to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any Life Support System without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons.

**Trademark Information**

Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, Clockbuilder®, CMEMS®, DSPLL®, EFM®, EFM32®, EFR®, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Ember®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, ISOmodem®, Micrium, Precision32®, ProSLIC®, Simplicity Studio®, SiPHY®, Telegesis, the Telegesis Logo®, USBXpress®, Zentri, Z-Wave and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. All other products or brand names mentioned herein are trademarks of their respective holders.



**SILICON LABS**

Silicon Laboratories Inc.  
400 West Cesar Chavez  
Austin, TX 78701  
USA

<http://www.silabs.com>