Developer's Guide

for the

Wisplet® S2W IOT Engine

WWW.IOTARCHITECTUREGROUP.COM

IOTAG is an operating unit of Silicon Engines.

SILICON ENGINES

3550 W. SALT CREEK LANE, SUITE 105

ARLINGTON HEIGHTS, IL 60005 USA

847-637-1180

CONTENTS

1. INTRODUCTION

1.1 Scope

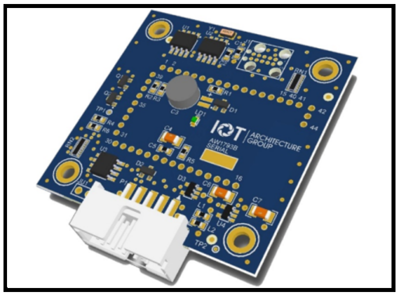
This document is a supplement to the **Wisplet S2W User's Guide**. The User's Guide gives an overview of the operation of the Wisplet S2W using the **Wisplet S2W Eval Kit**.

In this document we cover:

- How to use the **IOT Architect** utility to specify the **IOT Rules**.
- Technical details on the Wisplet hardware.
- How to interface the user's target system to the Wisplet S2W.

1.2 Review

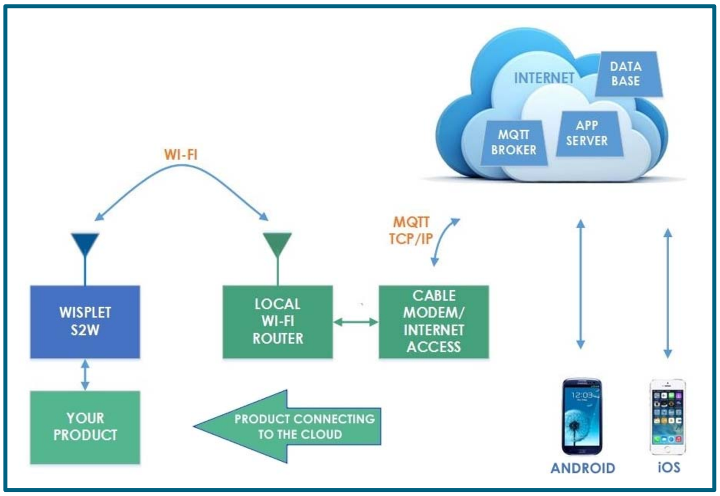
The Wisplet S2W is a small, low-cost, internet connectivity hardware board specifically designed for Internet-of-Things applications.



WISPLET S2W HARDWARE MODULE

FIGURE 1.2.1

The Wisplet S2W provides internet connectivity on one side, and connectivity to a hardware product on the other side, for the purpose of providing a way for that hardware product to connect to a cloud application.



WISPLET S2W CONNECTS YOUR PRODUCT TO THE CLOUD

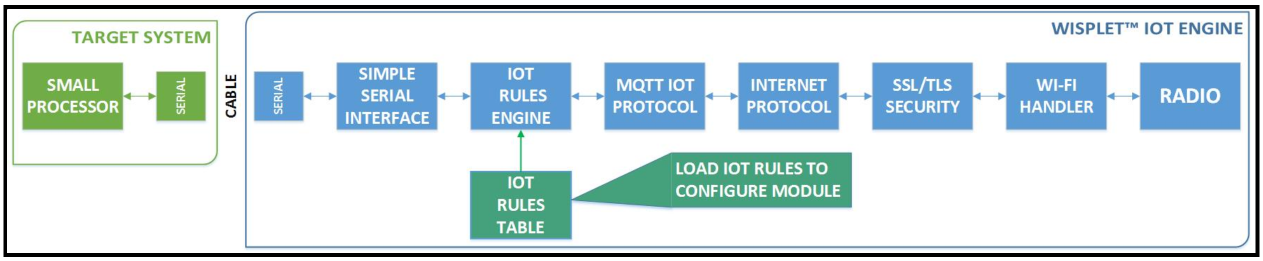
FIGURE 1.2.2

2. IOT RULES CONCEPT

2.1 IOT Rules Engine

The **IOT Rules Engine** is a firmware utility embedded into each Wisplet device.

- The IOT Rules Engine in the Wisplet device handles nearly all the IOT processing required at the edge of the IOT network. This **offloads the target system**, allowing a designer to add IOT functionality to a low-cost system without excessive burden on the target system's hardware and firmware. The only extra firmware needed within the target device is a simple serial interface handler that can communicate sensor values to the Wisplet device.



IOT RULES ENGINE IN WISPLET DEVICE

FIGURE 2.1.1

- The IOT Architect utility allows you to specify the operation of an IOT system with a **minimum of custom firmware**.

The IOT Rules Engine also optimizes data traffic flow over the network. The Wisplet device generates alerts only when a sensor value is out of bounds, according to the IOT rules in effect. This can greatly reduce the amount of data traffic sent over the network, which **can offer considerable savings.** Cloud hosting fees are typically based on data traffic in kilobytes per unit per month.

- Once the Wisplet sends information over the internet to the server in the cloud, it has already been screened for significance. It is already of interest to the subscriber—the home-owner, service technician, or system manager. This **simplifies the required server software**. For the most part the server needs only to act as a broker, routing messages to the appropriate subscribers, and backing traffic up in a data base.

2.2 Alerts, Reports, and Controls

The IOT Rules Engine concept is based on a simple but effective data model for IOT event processing. Nearly all IOT processing can be implemented using three types of event:

- **Alerts:** Automatic alerts when sensor values fall out of range.
- **Reports:** Scheduled, routine reports on system status.
- **Controls:** Commands allowing users to control the device remotely using a mobile app, laptop, or desktop system.

2.3 Wisplet at the edge

The Wisplet **IOT Rules Engine** contains all the firmware necessary to handle **Alerts**, **Reports**, and **Controls** at the edge of the IOT network.

The target system need only expose a simple serial interface that can communicate its sensor and system status to the Wisplet device.

The **IOT Architect** tool configures the IOT Rules without the need to write custom firmware.

3. IOT ARCHITECT TOOL

3.1 Availability

Note that the **IOT Architect™** tool is currently, as of the release of this document, a PC-based tool for Windows® only. All functionality of the IOT Architect tool will soon be available via the Wisplet server web interface.

3.2 Overall functionality

The **IOT Architect™** tool is the starting point for integrating the Wisplet S2W with your hardware product. So when you are ready to disconnect the Eval Kit's Sensor Board and connect your own hardware product, the IOT Architect is what you will use to reconfigure the Wisplet for use with your hardware.

The IOT Architect tool allows you to define all the parameters (sensors and values) that should be exposed to the Wisplet S2W via the 3.3V serial interface over the ribbon cable.

This tool also lets you define behaviors the Wisplet S2W should implement, such as rules that define the sensor reporting intervals. It also lets define which values for all sensors should be considered to be alert conditions.

3.3 IOT Architect functionality

The IOT Architect tool allows you to:

- Define which parameters are to be pushed to the cloud
- Define the rules applicable to each parameter
- Set other values, such as the reporting interval for reports

- Generate a C header file that you can include in the firmware for your target product. By using the IOT rules specified in this header file, you will ensure that your product's code provides the sensor values to the Wisplet in a location and manner from which the Wisplet can read them.

3.4 Target firmware tasks

The firmware running in your hardware product has to implement these tasks:

- Scan its sensors regularly, and place the current values in a location where they can be read over the 3.3V serial interface whenever the Wisplet device asks for them.
- Respond to requests for the latest sensor values, when the Wisplet device sends a sensor data request.
- Respond to requests to change parameter values, when the Wisplet device sends a control command.

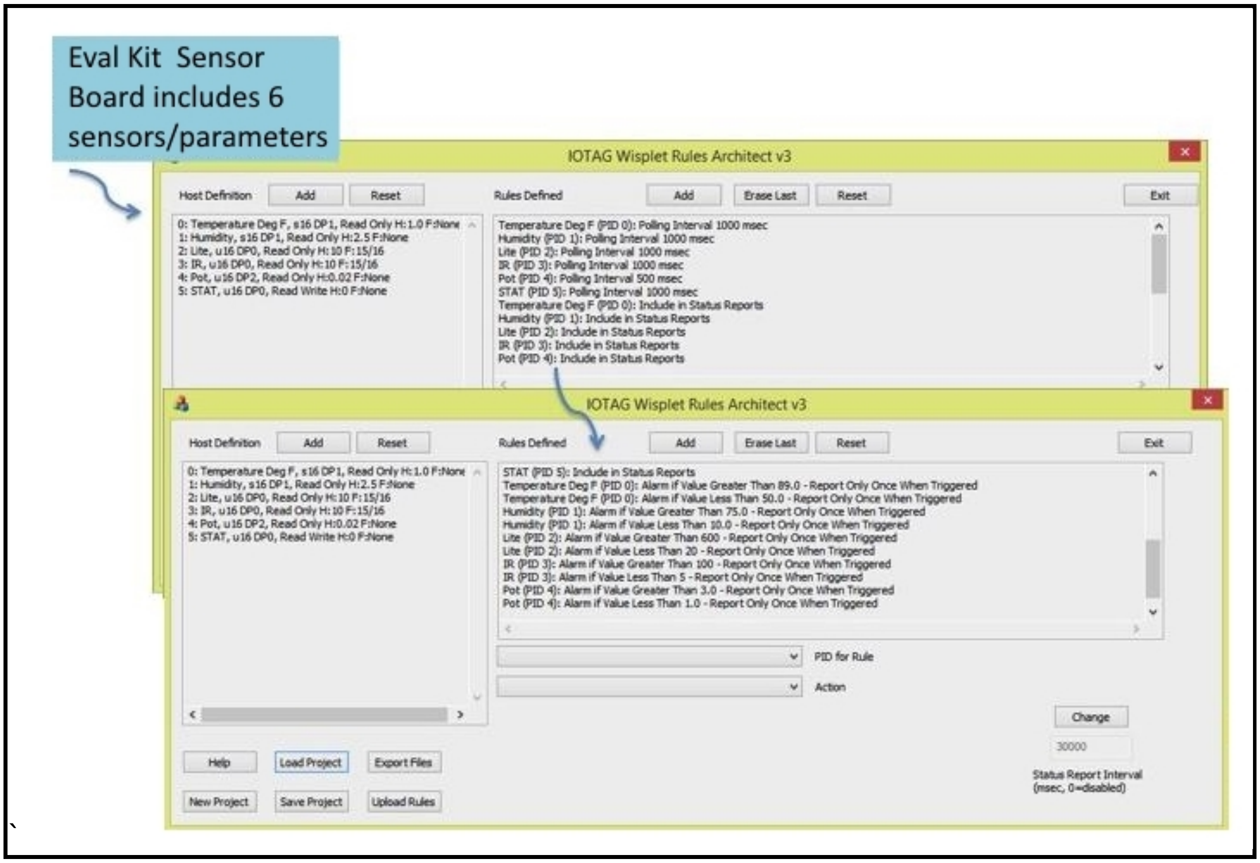
3.5 Wisplet firmware tasks

The tasks of the Wisplet device are to:

- Ask the target device for current sensor values on a periodic basis. The scan rate is programmable using the IOT Architect tool.
- Analyze sensor values to determine whether they are in normal range, based on the **IOT Rules** currently in effect.
- When an out-of-range condition exists, generate an **Alert** to the cloud.
- Include current sensor values in routine **Reports** which the Wisplet sends over the cloud**.** The rules for generating routine reports—which sensors are included, and how often the reports are sent—are defined by the IOT Architect tool, and these **IOT Rules** are downloaded to the Wisplet.
- Receive **Control** messages over the cloud from an authorized user—for example a consumer who is programming new set-points using an app on his or her smart phone.

3.6 Wisplet S2W Eval Board Rules

Here is the set of IOT Rules for the Wisplet S2W Eval Board, serving as a design example.



IOT RULES SET FOR THE WISPLET S2W EVAL KIT

FIGURE 3.6.1

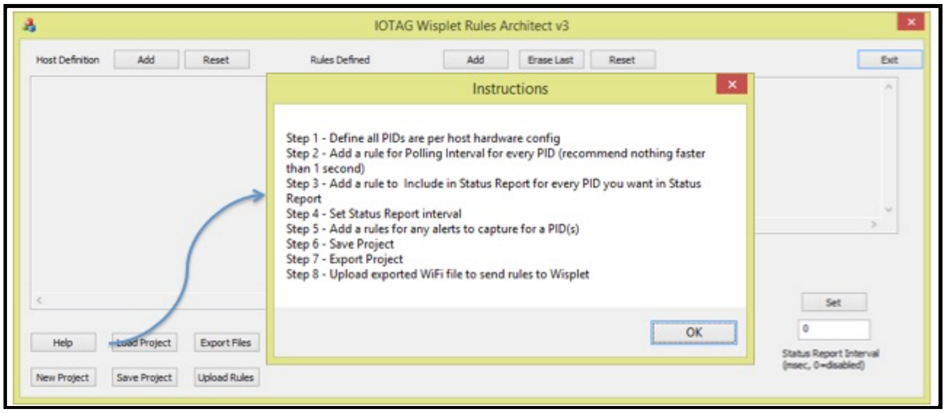
When these rules entered, the IOT Architect utility assigned parameter numbers from 0 to 5, starting at 0, in the order in which they were entered.

By applying these IOT Rules, the Wisplet IOT Rules Engine automatically handles the desired Alerts, Reports, and Controls for this set of sensors.

3.7 Instructions dialog box

We will now go through the steps to build a new set of IOT Rules using the IOT Architect tool.

The **HELP** button on the IOT Architect main screen exposes the following steps to specify a set of IOT Rules:



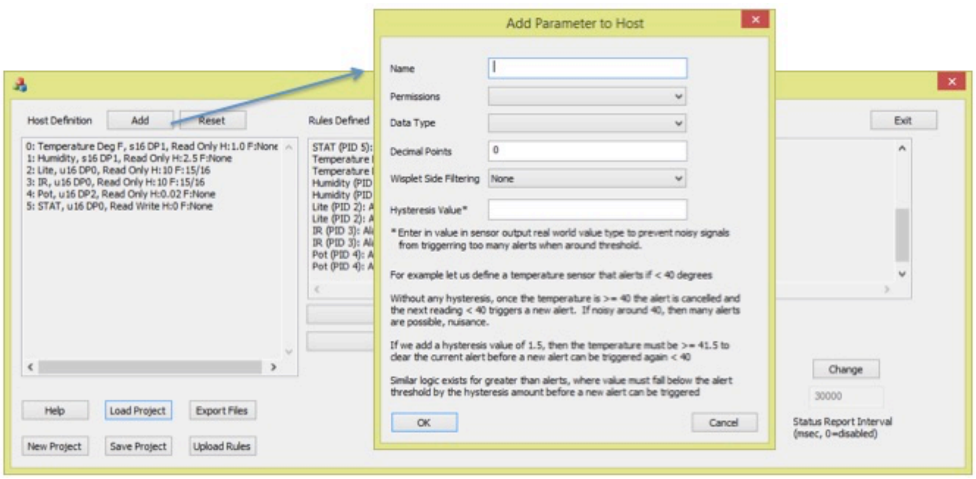
STEPS IN SETTING UP IOT RULES

FIGURE 3.7.1

3.8 Adding a parameter (PID)

A PID (parameter ID) is used to identify a specific data value for the IOT system. Typically a PID represents a sensor value, but a PID may also represent another data object—for example, the results of a functional test.

The following shows the procedure to define and add a parameter.



DEFINING A PARAMETER

FIGURE 3.8.1

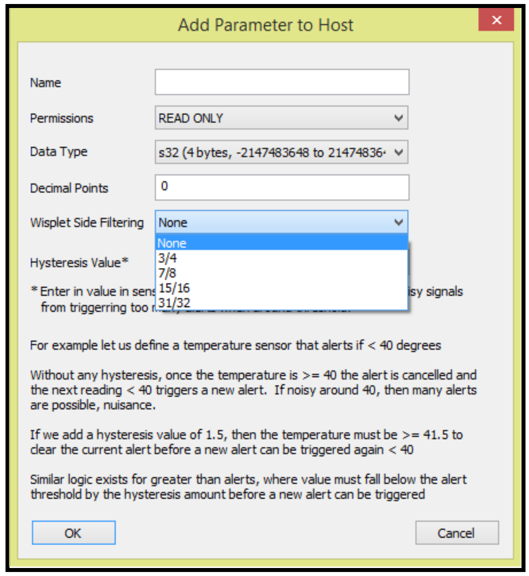
Start by clicking the *ADD* button in the *HOST DEFINITION* area.

3.9 Parameter name

For each parameter, you need to specify the parameter name. Typically this is the name of a parameter measured by a sensor—for example, *Temperature*. Can also be another type of data object, for example, Self-Test Status.

3.10 Parameter permissions

This field controls access to the parameter permissions value.



PARAMETER PERMISSIONS

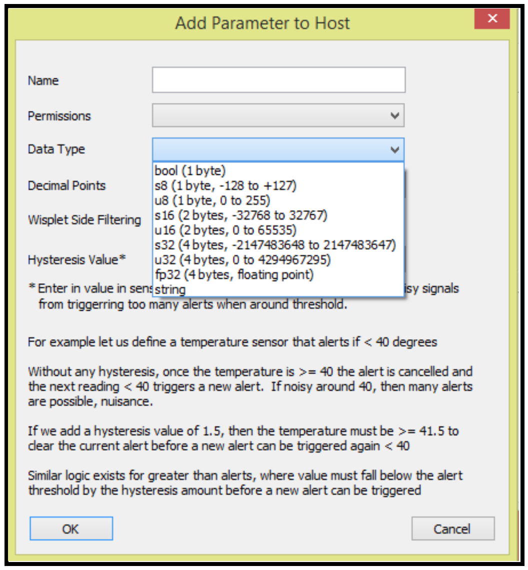
FIGURE 3.7.1

Allowable values are *READ ONLY*, *WRITE ONLY*, and *READ WRITE*. The sense of the terms *read* and *write* are from the point of view of the Wisplet device:

- **Read only:** A parameter whose value is determined solely by the target hardware. For example, *temperature*. The Wisplet can read this value, but cannot over-write it.
- **Write only:** A parameter whose value is determined by an authorized user gaining access over the cloud. For example, set-points contained in IOT Rules. Or control messages, such as: *Perform Self-Test*.
- **Read write:** A parameter that both can be maintained by the target device, and also can be overwritten by an authorized user over the cloud. For example, current date and time.

3.11 Data type

This field specifies the numerical data type for a parameter value. Specifying the data type in the IOT Rules is necessary to assure consistent parameter handling from the target board, through the Wisplet, over the MQTT protocol, and into the cloud.



PARAMETER DATA TYPES

FIGURE 3.9.1

Allowable values are:

- **Bool:** A single-bit value, 1 or 0, representing such values as *On* or *Off*; or *Enabled* or *Disabled*; or *Yes* or *No*. This value will be stored and communicated as one byte.
- **s8:** A signed 8-bit value, which can range from -128 to +127. One byte.
- **u8:** An unsigned 8-bit value, which can range from 0 to 255. One byte.
- **s16:** A signed 16-bit value, which can range from -32,768 to 32,767. Two bytes.
- **u16:** An unsigned 16-bit value, which can range from 0 to 65,535. Two bytes.
- **s32:** A signed 32-bit value, which can range from -2,147,483,648 to 2,147,483,647. Four bytes.
- **u32:** An unsigned 32-bit value, which can range from 0 to 4,294,967,295. Four bytes.
- **fp32:** A 32-bit floating-point number (IEEE 754). Consists (MSB to LSB) of the following: one sign bit, 8 exponent bits (s8 format), and 23 value bits. Can represent a very wide range of whole and fractional numbers, from 1.18×10^{-38} to 3.402823×10^{38}. Four bytes.

3.12 Decimal points

This field allows you to set the placement of a decimal point in a numerical value. The number you enter indicates the number of digits to the right of the decimal point. For example, for a temperature reading in Fahrenheit, using *s8* for the Data Type, and *0* for the Decimal Points value, we could handle a temperature sensor with a range from -40 to +120 degrees Fahrenheit.

3.13 Wisplet Side Filtering

The Wisplet IOT Rules Engine contains a built-in capability to filter sensor values. The purpose is to provide sensor data averaging, which in many cases can avoid false alerts caused by fluctuating sensor values. Incorporating this filter in the Wisplet firmware can ease the task of adding IOT reporting capabilities while minimizing the need for additional firmware in the target system.

Choices are:

- **None.** No filtering in effect.
- **3/4 filter.** Each time a new sensor data value is transferred from the target to the Wisplet device, the IOT Rules Engine recalculates a running average as follows,

$$Avg_{new} = \frac{3 * Avg_{old} + Sample_{new}}{4}$$

- **7/8 filter.** The average is calculated as:

$$Avg_{new} = \frac{7 * Avg_{old} + Sample_{new}}{8}$$

- **15/16 filter.** The average is calculated as:

$$Avg_{new} = \frac{15 * Avg_{old} + Sample_{new}}{16}$$

- **31/32 filter.** The average is calculated as:

$$Avg_{new} = \frac{31 * Avg_{old} + Sample_{new}}{32}$$

Each successive filter value, moving from 3/4 out to 31/32, provides more smoothing action, and is in effect a slower filter. In other words a step change in a sensor value takes longer to change the average.

When a filter is in effect, the IOT Rules for that sensor value will act on the filtered value.

Engineers may recognize this as a low-pass IIR (infinite impulse response) digital filter. The effective response time to a change in sensor values is a function both of the filter order, and the rate at which the Wisplet device polls the target system for the sensor values.

3.14 Hysteresis value

The Wisplet IOT Rules Engine contains a built-in capability to add *hysteresis* to sensor values. Example: a system is programmed to cause an alert when the temperature falls below 40 degrees. We enter a hysteresis value of 1.5. The alert will still be caused when the temperature falls below 40.0. But now when the temperature starts to rise again, it must reach 41.5 before the alert condition is removed.

The purpose is to provide protection against multiple alerts when sensor values are close to set-point boundaries. Small fluctuations in sensor values could cause erratic behavior in generating alerts.

Continuing this example, the system is also programmed to cause an alert when the temperature rises above 100 degrees. In this case the alert happens when the temperature exceeds 100.0. When the temperature starts to fall again, it must reach 98.5 degrees before the alert condition is removed.

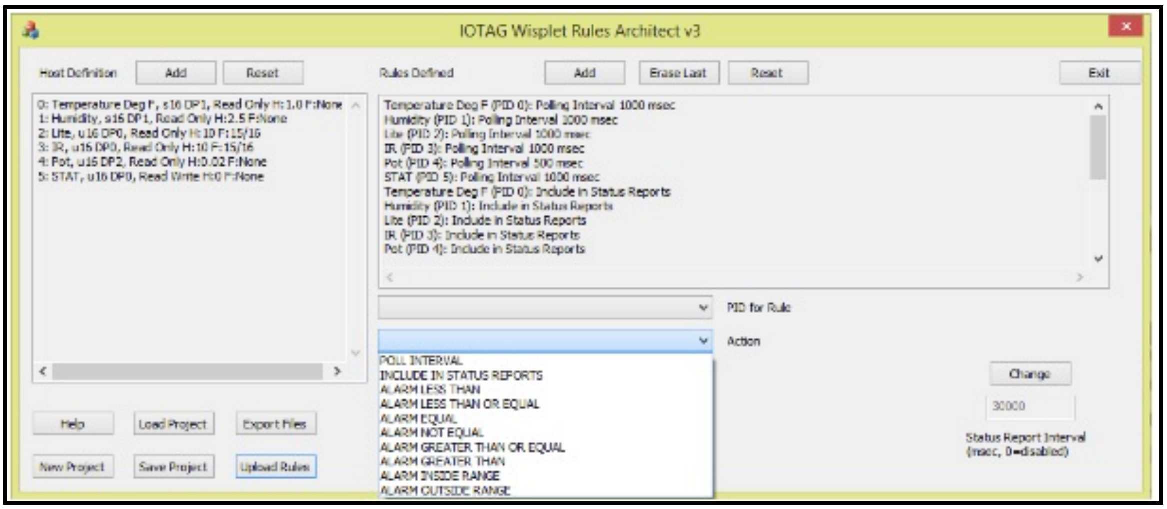
When a hysteresis value is in effect, the IOT Rules for that sensor will act on the value as modified by the hysteresis value.

3.15 Adding actions for a parameter

Once one or more parameters have been defined, the next step is to set up the **Alerts** and **Reports** rules.

To add an action, first select the parameter in question.

Then choose an action for that parameter.

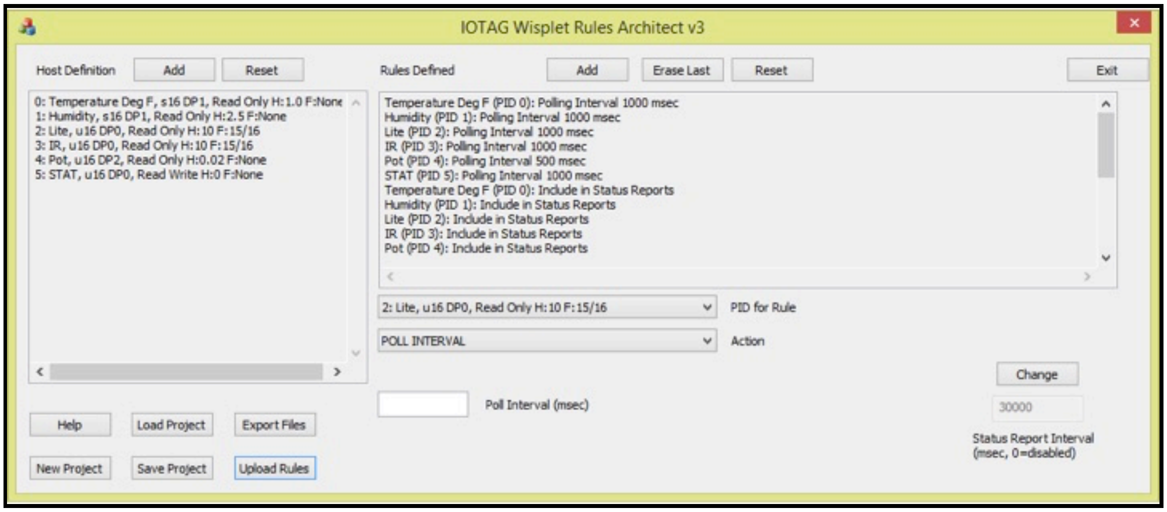


CHOOSING ACTIONS THAT APPLY TO A SPECIFIC PARAMETER

FIGURE 3.15.1

3.16 Poll interval

To activate the IOT Rules for a given parameter, you must specify the *poll interval*. This sets how often the Wisplet device will request the latest sensor value from the target system.



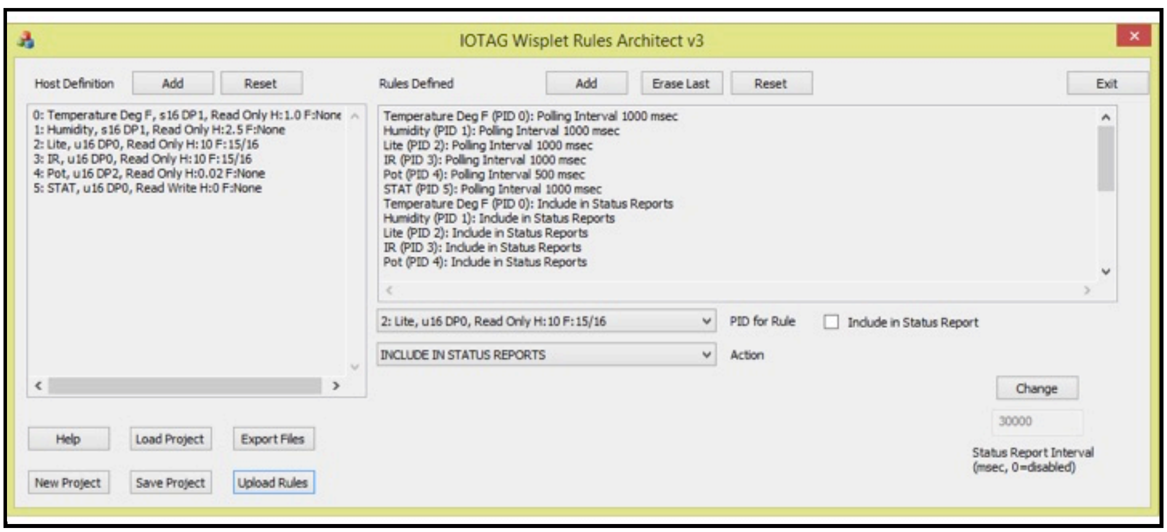
CHOOSING A POLL INTERVAL

FIGURE 3.16.1

Enter in milliseconds (0.001 second). The recommended minimum is 1,000 milliseconds, or 1 second. To deactivate polling, set the value to 0.

3.17 Include in status reports

The next step is to choose whether to include this parameter in status reports.



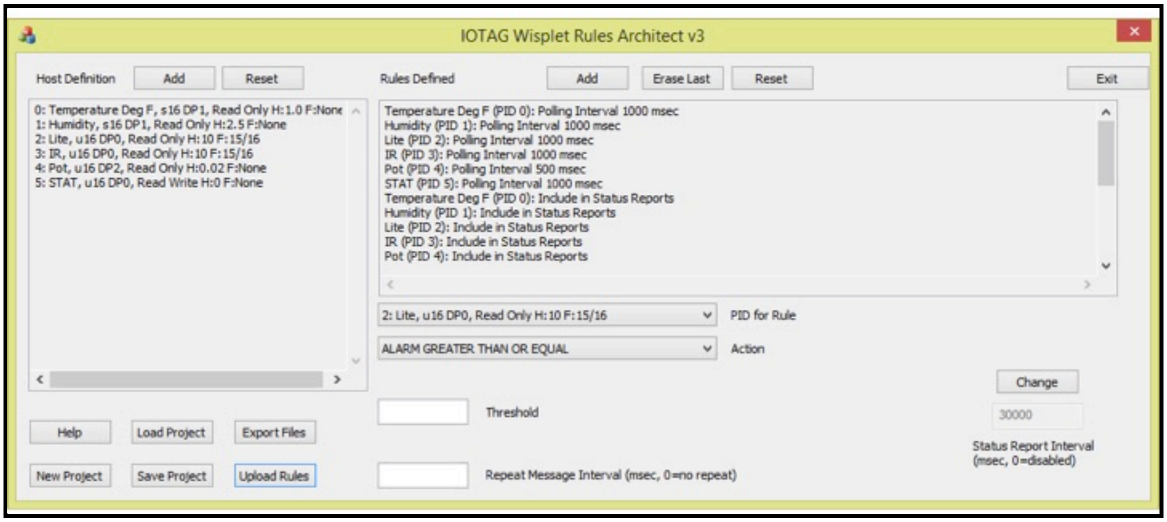
INCLUDE IN STATUS REPORTS

FIGURE 3.17.1

To activate this option, check the box, *Include in Status Report.*

3.18 Setting IOT Rules for Alerts

Next we define rules for generating alerts.



RULES FOR GENERATING ALERTS

FIGURE 3.18.1

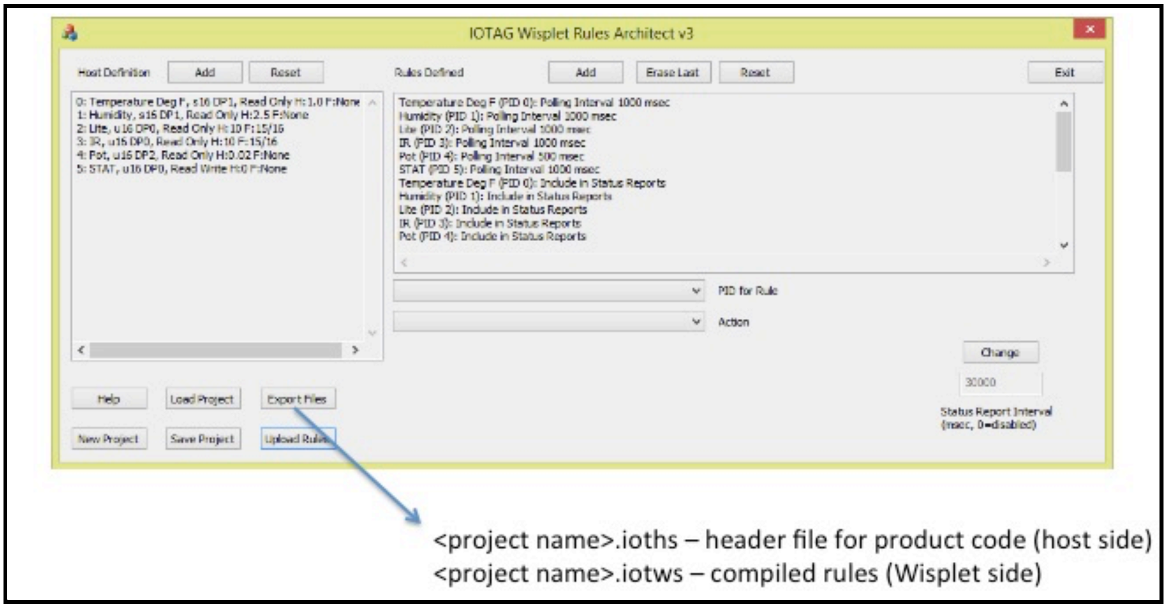
Here we have a number of logical operators to choose from. These include:

- ALARM LESS THAN
- ALARM LESS THAN OR EQUAL
- ALARM EQUAL
- ALARM NOT EQUAL
- ALARM GREATER THAN OR EQUAL
- ALARM GREATER THAN
- ALARM INSIDE RANGE
- ALARM OUTSIDE RANGE

Once you have selected one of these logical operators, you are prompted to fill in the specific value or values needed to complete the alert rule.

3.19 Generating compiled files

Once you have completed filling in the IOT Rules for your system parameters, the next step is to generate and save compiled files for use by the Wisplet unit and the target unit.



GENERATING FILES THAT CONTAIN THE IOT RULES

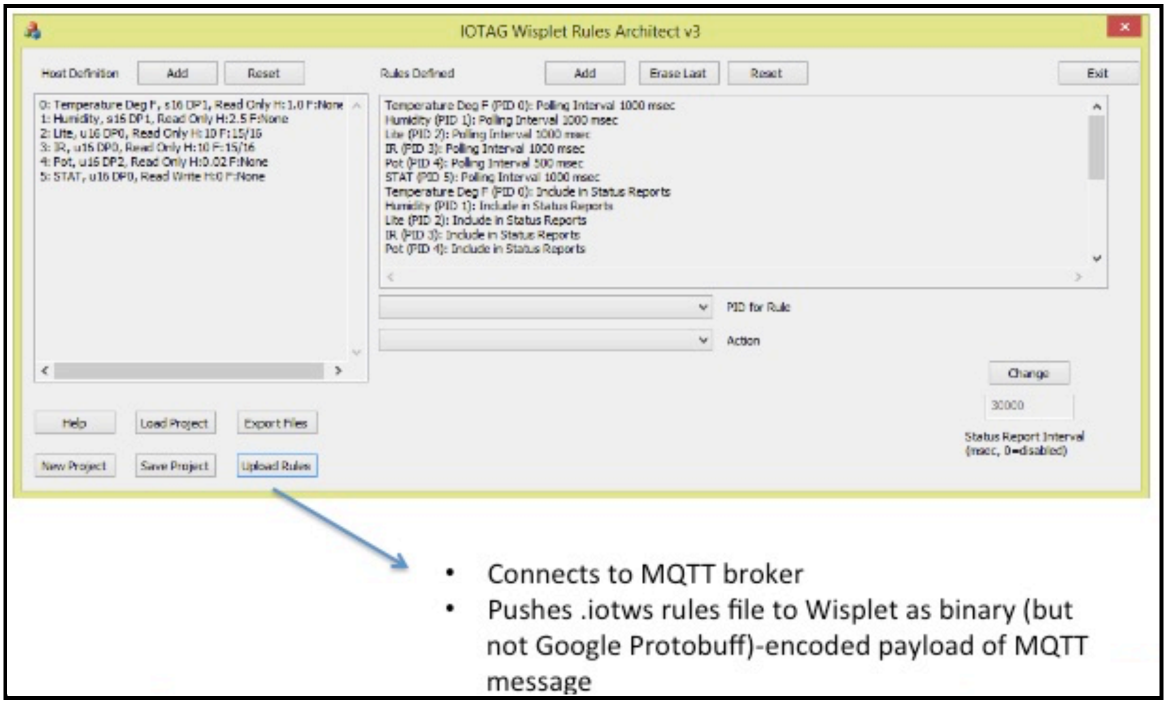
FIGURE 3.19.1

The IOT Architect tool will generate the following files:

- **<project name>.ioths** is a header file for use by the serial handler in the target system. (See Part 5 below for target system firmware guidelines.)
- **<project name>.iotws** is the set of compiled IOT rules for use by the Wisplet device.

3.20 Pushing IOT Rules to the Wisplet device

In order to push a new set of IOT Rules to your Wisplet device, use the *Upload Rules* button in the IOT Architect tool.



PUSHING IOT RULES TO THE WISPLET DEVICE

FIGURE 3.20.1

To push a rule set to your Wisplet device, your Wisplet unit must be online and connected to the MQTT broker and Wisplet server. Check the connection status via the Wisplet server web interface or app before attempting to push a ruleset out to it.

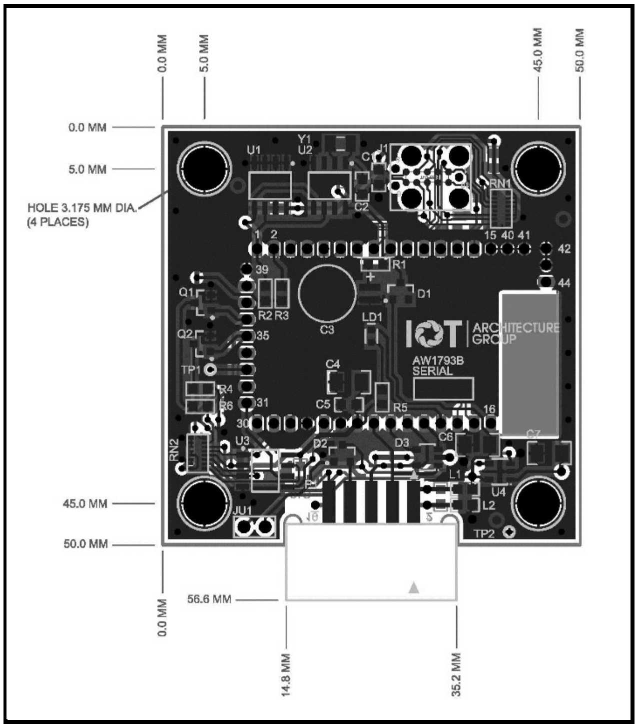
You will again need to provide the MAC address of your Wisplet when pushing out a new ruleset to it. You can find this on the sticker on your Wisplet Eval kit.

4. HARDWARE

4.1 3D CAD file

A 3D CAD file for the Wisplet S2W is available in STEP format. Contact IOT Architecture Group for details.

4.2 Mechanical dimensions

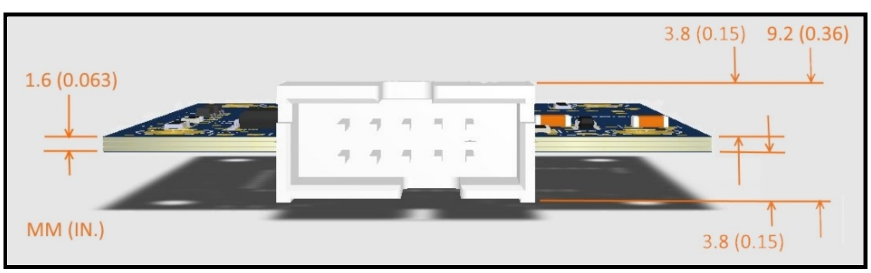


DIMENSIONAL DRAWING—TOP VIEW

FIGURE 4.2.1

The four mounting holes have a diameter of 3.175 mm (0.125 in.). The PCB can be mounted into the target system using:

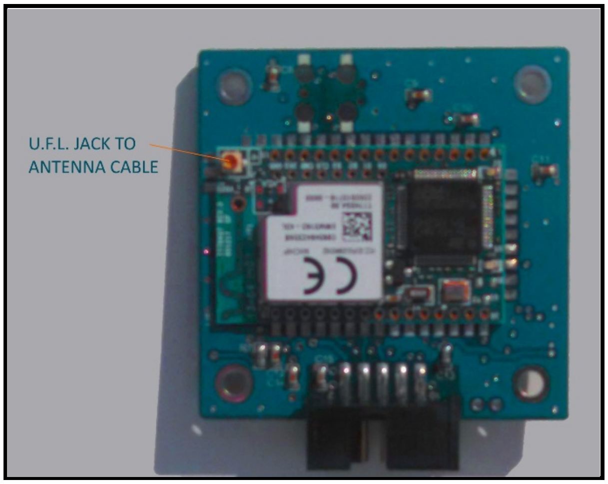
- Snap-lock nylon board support spacers, such as Keystone 9020.
- Metric M3 spacers and screws
- English #4-40 spacers and screws



DIMENSIONAL DRAWING—SIDE VIEW

FIGURE 4.2.2

The picture below shows the location of the U.F.L. jack, on the bottom side of the module. The U.F.L. antenna cable plugs in here.



BOTTOM VIEW—SHOWING U.F.L. CONNECTOR

FIGURE 4.2.3

4.3 Antenna options

There are three antenna options for the Wisplet S2W:

- **On-board antenna.** A foil pattern on the circuit board serves as the Wi-Fi antenna. This option provides the most compact installation. It is recommended only when the Wisplet module is installed in a product with a plastic case. Please test the positioning of the module within your product in order to provide good Wi-Fi performance.
- **Short in-box antenna.** A short antenna with a U.F.L. connector. Recommended for a product in a plastic case. Better performance than the on-board antenna. Please test the mounting and positioning of the module and this antenna within your product in order to provide good Wi-Fi performance.

SMALL ANTENNA WITH U.F.L. CONNECTOR

FIGURE 4.3.1

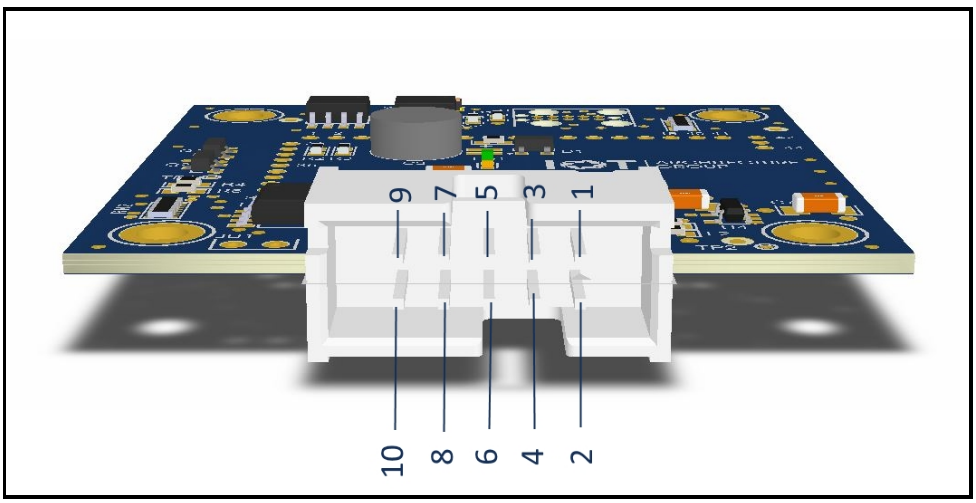
- **External antenna.** A plastic whip antenna with a U.F.L. connector. Recommended for a product in either a metallic or plastic case. Best performance. Mounts in a 9.5 mm (0.375 inch) diameter opening in the enclosure

LARGE ANTENNA WITH U.F.L. CONNECTOR

FIGURE 4.3.2

4.4 Ribbon cable

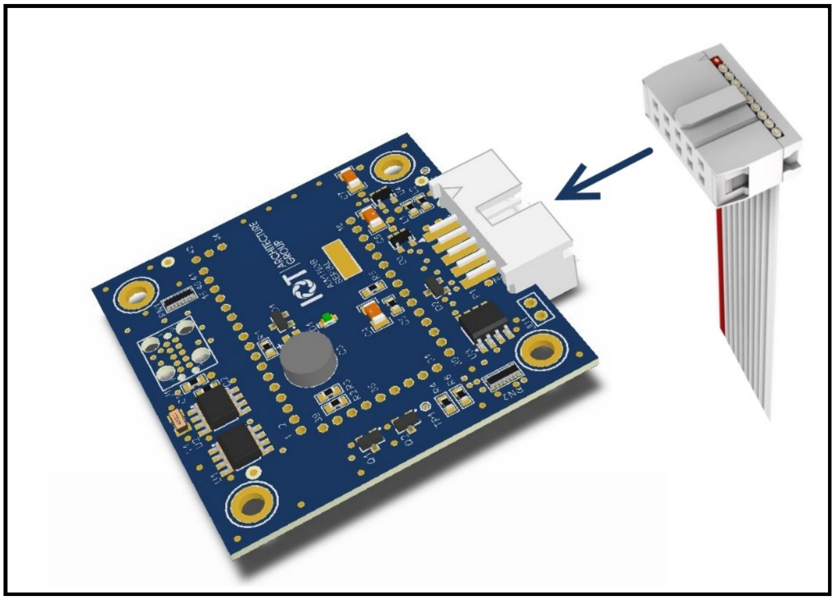
The ribbon cable connector, which accepts a ribbon cable that leads to the target system, is centered on one of the four edges of the 50-mm-square PCB.



PIN NUMBERING ON RIBBON CABLE CONNECTOR

FIGURE 4.4.1

The ribbon cable should provide polarized receptacles at both ends to prevent accidental incorrect insertion. The recommended cable style, shown below, has a red stripe to mark the conductor that connects to pin 1, and a polarizing key on each cable receptacle which fits into a matching slot in the PCB header.



RIBBON CABLE POLARIZATION

FIGURE 4.4.2

Gold plating on the cable's receptacle pins is recommended for long-term resistance to corrosion. Cables with tin plating on the receptacle pins are not recommended.

An example of an inexpensive compatible ribbon cable:

ITEM	PARAMETER	VALUE	UNITS
1	CONDUCTORS	10	
2	WIRE PITCH	0.050 (1.27)	INCHES (MM)
3	WIRE GAUGE	28	AWG
4	INSULATION	PVC	
5	POLARIZATION	KEY	
6	PIN SPACING	0.100 (2.54)	INCHES (MM)
7	CONTACT PLATING	GOLD FLASH OVER NICKEL	
8	GOLD THICKNESS	15 (0.38)	µIN (µm)
9	CABLE LENGTH	6 (152)	INCHES (MM)
10	MANUFACTURER	ASSMANN	
11	PART NUMBERS	AWP10-7240-T, AWG28-10/G/300	
11	VENDOR	DIGI-KEY	
12	VENDOR NUMBER	H3CCH-1006G	

EXAMPLE RIBBON CABLE

FIGURE 4.4.3

4.5 Power specifications

ITEM	PARAMETER	MIN.	TYP.	MAX.	UNITS
1	POWER VOLTAGE	3.13	3.30	3.47	VOLTS
2	PEAK CURRENT			350	MILLIAMPS

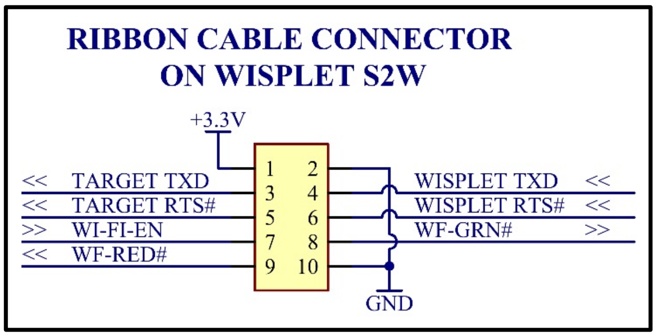
POWER SPECIFICATIONS

FIGURE 4.5.1

5. HARDWARE INTERFACE TO TARGET

5.1 Connector circuitry on Wisplet S2W

The following is the schematic diagram for the 10-pin ribbon cable connector on the Wisplet S2W:



SCHEMATIC FOR WISPLET RIBBON CABLE CONNECTOR

FIGURE 5.1.1

Here is a description of the functions of each of these circuits.

PIN	CIRCUIT	DESCRIPTION	DIRECTION
1	+3.3V	POWER SUPPLY FOR WISPLET S2W	TARGET TO WISPLET
2	GROUND	POWER AND SIGNAL COMMON	COMMON
3	TARGET TXD	TRANSMIT DATA LINE FROM TARGET	TARGET TO WISPLET
4	WISPLET TXD	TRANSMIT DATA LINE FROM WISPLET	WISPLET TO TARGET
5	TARGET RTS#	REQUEST TO SEND FROM TARGET	TARGET TO WISPLET
6	WISPLET RTS#	REQUEST TO SEND FROM WISPLET	WISPLET TO TARGET
7	WIFI-EN	POWER SWITCH TO WISPLET	TARGET TO WISPLET
8	WF-GRN#	WI-FI GREEN LED CONTROL	WISPLET TO TARGET
9	WF-RED#	WI-FI RED LED CONTROL	WISPLET TO TARGET
10	GROUND	POWER AND SIGNAL COMMON	COMMON

SIGNAL FUNCTIONS

FIGURE 5.1.2

5.2 Four-wire serial interface

The **TARGET TXD** and **WISPLET TXD** signals are serial data signals running at 3.3V levels. No RS-232 level shifter is required to connect the Wisplet unit to the target board. These signals implement an asynchronous serial data interface typically handled by a UART (universal asynchronous receiver-transmitter) peripheral within a microcontroller. This is a full-duplex connection—in other words, messages are handled independently over the two separate signals.

The **TARGET TXD** signal comes from the TXD (transmit data) output of the UART on the target module. It connects to the RXD (receive data) input of the UART in the Wisplet module.

The **WISPLET TXD** signal comes from the TXD output of the UART on the Wisplet module. It connects to the RXD input of the UART in the target module.

Serial data bits are sent in 10-bit words in 8N1 format—consisting of one START bit, eight data bits (least significant bit first), no parity bit, and one STOP bit. Data and voltage levels on the TXD and RXD signal lines will be at as follows:

SIGNAL CONDITION	LOGIC LEVEL	VOLTAGE LEVEL
IDLE LINE	1	3.3V
START BIT	0	0V
1 DATA BIT	1	3.3V
0 DATA BIT	0	0V
STOP BIT	1	3.3V

SERIAL DATA LEVELS ON TXD AND RXD LINES

FIGURE 5.2.1

The data rate is 38,400 bps (bits per second).

The **TARGET RTS#** and **WISPLET RTS#** signals are flow control signals, also operating at 3.3V levels. These are active-low signals (as indicated by the # character).

The TARGET RTS# signal comes from the RTS# (request to send) output of the UART on the target module. It connects to the CTS# (clear to send) input of the UART in the Wisplet module.

The WISPLET RTS# signal comes from the RTS# output of the UART on the target module. It connects to the CTS# input of the UART in the target module.

SIGNAL NAME	LOGIC LEVEL	VOLTAGE LEVEL	CONDITION
TARGET RTS#	1	3.3V	TARGET BOARD IDLE
	0	0V	TARGET BOARD READY
WISPLET RTS#	1	3.3V	WISPLET BOARD IDLE
	0	0V	WISPLET BOARD READY

RTS# AND CTS# SIGNALS

FIGURE 5.2.2

For further details on these signals, see the protocol description below.

5.3 Wisplet module power control

The signal **WIFI-EN** is used to control power to the Wisplet module.

SIGNAL NAME	LOGIC LEVEL	VOLTAGE LEVEL	CONDITION
WIFI-EN	1	3.3V	WISPLET MODULE TURNED ON
	0	0V	WISPLET MODULE TURNED OFF

POWER CONTROL SIGNAL

FIGURE 5.3.1

This signal allows the target module the option to conserve power by shutting off the Wisplet module to implement a low-power sleep mode. This signal controls a solid-state switch on the Wisplet module that shuts down the Wi-Fi circuitry completely.

If the target application does not require sleep mode, this signal should be connected to +3.3V.

5.4 Wi-Fi status LED signals

The Wisplet S2W provides two signals intended to drive LED indicators on the target system to show Wi-Fi connection status.

SIGNAL NAME	SIGNAL LEVEL	CONDITION
WIFI-RED#	HIGH	RED LED OFF
	LOW	RED LED ON
WIFI-GRN#	HIGH	GREEN LED OFF
	LOW	GREEN LED ON

WI-FI LED CONTROL SIGNALS

FIGURE 5.4.1

The active-low LED driver transistors on the Wisplet module provide 220Ω current-limiting resistors. The exact voltage level on these signal lines when the LEDs are turned on depends on the forward voltage of the specific LEDs chosen.

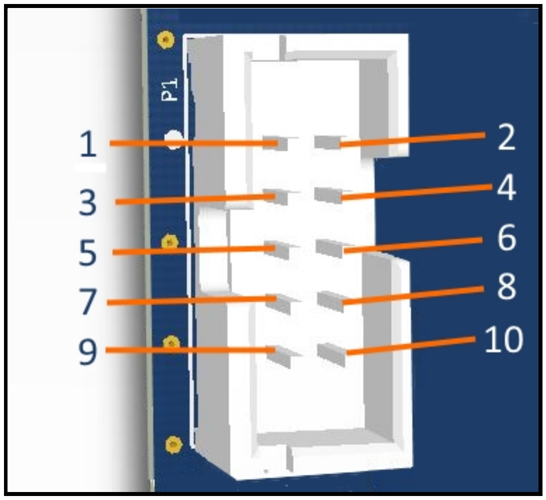
See the section below on target board hardware for typical connections of these LED signals.

5.5 Interface circuitry on target board

TBD

5.6 Ribbon cable connector on target board

The target system typically provides a 10-pin vertical header for compatibility with the ribbon cable leading to the Wisplet S2W.



PIN-OUTS ON VERTICAL HEADER ON TARGET BOARD

FIGURE 5.3.1

The header should be a protected header (with side walls). It should have a polarization slot, as shown between pins 3 and 5 in the illustration above. These features prevent accidental reverse insertion of the ribbon cable.

Gold plating on the header pins is recommended for long-term resistance to corrosion. Headers with tin plating on the pins are not recommended.

An example of an inexpensive vertical header:

ITEM	PARAMETER	VALUE	UNITS
1	CONDUCTORS	10	
2	CONSTRUCTION	PROTECTED	
3	POLARIZATION	SLOT	
4	PIN SPACING	0.100 (2.54)	INCHES (MM)
5	CONTACT PLATING	GOLD FLASH OVER NICKEL	
6	GOLD THICKNESS	1.2 (0.03)	μIN (μm)
7	MANUFACTURER	ON-SHORE TECHNOLOGY	
8	PART NUMBER	302-S101	
9	VENDOR	DIGI-KEY	
10	VENDOR NUMBER	ED1543-ND	

EXAMPLE 10-PIN HEADER FOR TARGET BOARD

FIGURE 5.3.2

6. SENSOR BOARD REFERENCE DESIGN

IOT Architecture Group has designed the Sensor Board in the Wisplet S2W Eval Kit as a reference design for Wisplet product users.

Schematic diagrams and firmware source code in C can be made available to IOTAG customers who have engaged with us on a system design.

Please contact IOTAG for assistance.

7. PARAMETER FILE STRUCTURE

TO BE COMPLETED.

8. MQTT BASICS

TO BE COMPLETED.

9. CLOUD SERVICES AND APPS

TO BE COMPLETED.

10. WISPLET ROAD MAP

10.1 OTHER MODELS

TO BE COMPLETED.

10.2 CELLULAR

TO BE COMPLETED.

11. REVISION HISTORY

11.1 Rev. A

First release