7040 Avenida Encinas #104211, Carlsbad, CA 92011
+1 (760) 814-1575

Cypherbridge Systems, LLC.

Demo Kit Guide

for the

Cypherbridge®
Trusted, safe and secure

SDKPac™

Powered by

RENESAS synergy™

www.cypherbridge.com
+1 (760) 814-1575

Table of Contents

1 INTRODUCTION

1.1 Getting Started

The SDKPac Demo Kit is supported on the Renesas Synergy PK-S5D9 Promotion Kit. This Demo Kit Guide has step-by-step instructions to install and run the SDKPac Demo on the Renesas Synergy Kit.

To quickly get started with the demo:

1. Review the Overview section.
2. Next go to the Install section to install the SDKPac demo
3. Then go to the Run section to run the demo.

1.2 References

[1] Renesas PK-S5D9 v1.01 Quick Start Guide March 23, 2017

[2] Renesas PK-S5D9 User's Manual Rev 1.01 April 2017

1.3 Technical Support

To submit a technical support issue, please include details including a reproducible test case. This includes the project build options, debug options, screen shots, and other test setup information as needed. Capture the UART debug message output log with teraterm, and include it with your problem report.

Technical support questions should be sent to Renesas Electronics America through the Synergy Gallery support page, where you can also visit forums, knowledge base and other resources including ***Chat Now*** with a Synergy Specialist:

https://synergygallery.renesas.com/support

1.4 Next Steps

Contact Cypherbridge Systems customer support for detailed information on the SDKPac features, software license, pricing and options.

Order the full function SDKPac Synergy Developer Kit to start project integration.

1.5 SDKPac Premium Edition

The SDKPac Premium Edition includes priority technical support, response time and design services to fast track your project. Please contact Cypherbridge Systems to add a Support Agreement to your project. Support for

purchased SDKs and toolkits under your Software License Agreement is typically 90 days, after which you can purchase annual maintenance and support. Contact Cypherbridge Systems or your distributor for support options.

Cypherbridge Systems is based in the USA in the PDT time zone. Our support process is by email and phone.

To submit a technical support issue, please include details including a reproducible test case. This includes the project build options, debug options, screen shots, and other test setup information as needed. Capture the UART debug message output log with teraterm, and include it with your problem report email to the email address listed in your Support Agreement.

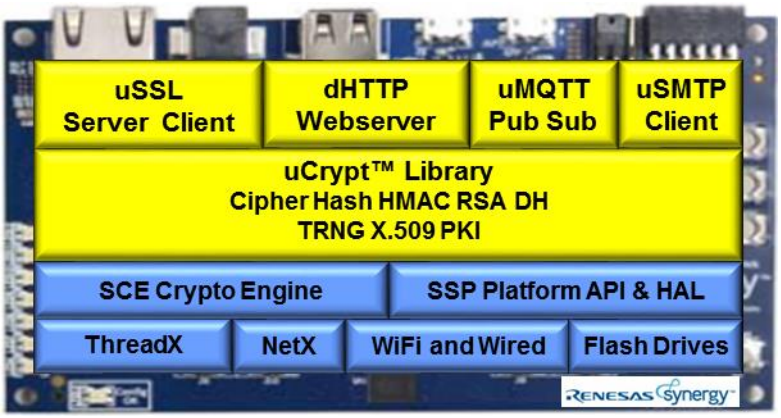
2 SDKPac OVERVIEW

2.1 What is SDKPac

The Cypherbridge Systems SDKPac is a collection of embedded device SDKs and Toolkits that can be used out of box to add secure device connectivity to a target project. SDKPac includes features and standards based protocols for secure IoT connect-to-cloud, gateway, embedded servers and clients, secure file transfer protocols, and electronic data privacy.

This SDKPac Demo Kit contains binary demo applications built for S5D9 including:

- ✓ uSSL SDK micro-content HTTPS server, and client sample applications
- ✓ uMQTT pub/sub client integrated with TLS for secure Internet of Things connection to IoT clouds, including Cyclone, Medium One, and AWS IoT.
- ✓ dHTTP Embedded WebServer
- ✓ uSMTP TLS secure email client
- ✓ Cyclonite GUIX based GUI for SDKPac setup and monitoring
- ✓ SDKPac for Synergy Essentials: debug serial port, Ethernet MAC configuration, Hard Fault Handler, RTC clock management
- ✓ Synergy SCE crypto engine support
- ✓ Built with Renesas e2s studio
- ✓ Integrated with Express Logic ThreadX RTOS and NetX TCP/IP, including DHCP and DNS support.
- ✓ SSP and HAL peripheral library
- ✓ Synergy VSA approved – compatible and tested with SSP on Synergy Development Kits



2.2 Demo System Configuration

The demo system is integrated with the following system, software and toolchain components:

- SDKPac Demo Kit distribution zip file
- Windows 10 PC development machine
- Renesas Synergy Cortex-M4 Synergy Kits: PK-S5D9
- e2studio 5.4.0.018
- SSP 1.2.1
- J-Link ARM GDB Debugger
- Segger J-Link USB driver using Standard-A Male to Micro-B male cable
- USB Serial port connection to starter kit 115.2K/N/8/1
- CAT5 network cable for LAN testing
- ThreadX 5.7
- NetX 5.9
- GUIX Studio 5.3.3.0 using GUIX Library 5.3.2

3 INSTALL THE DEMO

3.1 Synergy Gallery Registration

SDKPac installation uses the Renesas e2studio toolchain. Use the following links to register on Renesas Synergy Gallery, then download, install and activate e2studio:

http://am.renesas.com/products/embedded_systems_platform/synergy/gallery/index.jsp

https://synergygallery.renesas.com/auth/login

https://synergygallery.renesas.com/

3.2 Download, Install and Activate e2studio

https://synergygallery.renesas.com/isde

https://synergygallery.renesas.com/ssp

Click 'Create a Developer/Product License'

3.3 Synergy Starter Kit

The SDKPac demo kit uses the Synergy PK-S5D9 Promotion Kit. These kits use essentially the same PCB but run different MCUs.

3.3.1 Ordering

The PK- S5D9 can be ordered directly from Renesas. These kits are also available through distributors including DigiKey Part Number:

PK-S5D9: YSPKS5D9E10-ND

3.3.2 Setup

Unpack the Synergy kit and cable the board as shown in the following diagram:

- Blue: Connect a Standard-A to Micro-B M/M USB cable to J-Link J19 connector. This provides the power and Debug connection.
- Yellow: CAT5 network cable to switch or router
- Magenta: USB serial port debug connection to header row J10

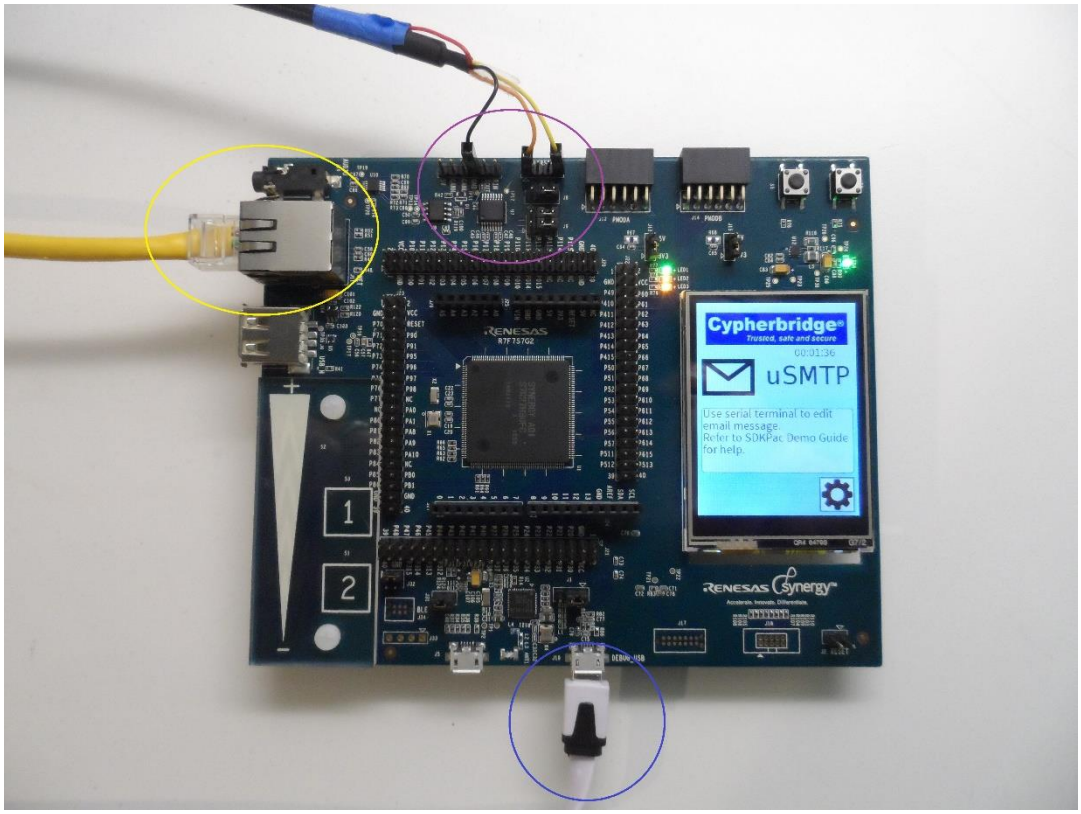


Figure 1: Synergy Kit Setup (SK-S7G2 Shown)

3.3.3 USB Serial Port

The SDKPac Demo uses SCI3 to output debug and status messages to confirm the operation of the Synergy target and demo application. These debug messages are also used for technical support to verify normal operation and troubleshoot problems. This design allows the Demo Kit serial port to run independently, fully stopped in e2studio debugger breakpoint, vs. an on-Synergy USB CDC device.

The Synergy Kit offers several serial port configuration options set by jumpers. Refer to Reference [2] PK-S5D9 User Manual, Table 6.1 Page 17, showing jumper configuration for serial port SCI3 TTL operation. As shown in Table, use jumper settings J8 2-3, and J9 3-5 4-6

Connect a USB to TTL serial cable to target J10. Both of the following cables are FTDI based and use in-box Windows drivers, for zero driver install.

3.3.3.1 Sparkfun FTDI CAB-12977 USB to TTL Serial Cable

The Sparkfun cable has each interface connector broken out and labeled with a 1 pin connector. Connect the cable as follows:

Synergy PK-S5D9 signal	Sparkfun cable signal
J10.1 P7.6 RXD3	TXD Orange Transmit Async Data output
J10.4 P7.7 TXD3	RXD Yellow Receive Async Data input
J7 GND	GND Black

3.3.3.2 FTDI TTL-232R-3V3 TTL

Refer to FTDI TTL-232R TTL to USB serial converter datasheet FT_000054.

Pin connections are shown in the following table:

Synergy PK-S5D9 signal	FTDI cable signal
J10.1 P7.6 RXD3	Pin 4 TXD Orange Transmit Async Data output
J10.4 P7.7 TXD3	Pin 5 RXD Yellow Receive Async Data input
J7 GND	Pin 1 GND Black

3.4 Network Connection

SDKPac startup includes network initialization. NetX requires the ETH1 CAT5 connector to be connected to LAN in order to complete driver initialization, at which point link LED is on.

3.5 Register and Download the SDKPac Demo Kit

Next, register and download the SDKPac Demo Kit. There are two ways to get it, from Gallery or Cypherbridge.

3.5.1 From Synergy Gallery

Log into your registered account on Renesas Synergy Gallery, navigate to VSA Gallery, Cypherbridge Systems. Click the SDKPac Demo Kit link, review and agree to terms of End User License Agreement (EULA), then download and save the SDKPac Demo Kit on your Windows PC

https://synergygallery.renesas.com/addon

Click the Cypherbridge Systems Logo

https://synergygallery.renesas.com/addon/detail/3

3.5.2 From Cypherbridge Systems

From your browser, navigate to www.cypherbridge.com/Renesas.html
Register and submit your contact information and request the ***Renesas Synergy SDKPac Demo Kit***. You will receive an email with a link to download the SDKPac Demo Kit. Your registration must include company information and email. Personal email address on gmail, yahoo, etc. will not be accepted.

3.6 Install the SDKPac Demo Kit

Along with various project files, the SKDPac Demo Kit includes documentation and pre-built binary applications.

Download the SDKPac Demo Kit. Unzip it in the Windows development machine directory shown here:

C:\Renesas\sdkpac_demokit_pk

3.6.1 SDKPac Demo Kit Files

Depending on your Synergy kit, the SDKPac Demo Kit includes the following files.

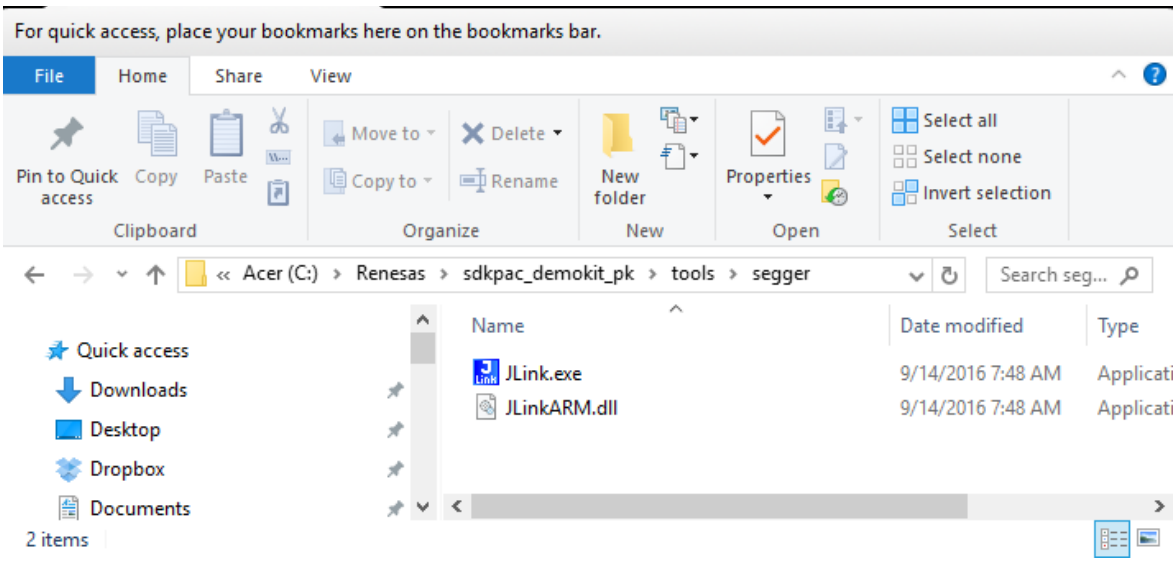
```
-- sdkpac_demokit
   |-- docs
   |   |-- SDKPac_DemoKit_Guide_Synergy_PKS5.pdf
   |-- demos (for kit PK-S5D9)
   |   |-- sdkpac_pks5d9_dhttp.hex
   |   |-- sdkpac_pks5d9_umqttp.hex
   |   |-- sdkpac_pks5d9_umqtts.hex
   |   |-- sdkpac_pks5d9_usmtp.hex
   |   |-- sdkpac_pks5d9_ussl_client.hex
   |   |-- sdkpac_pks5d9_ussl_server.hex
   |   |-- sdkpac_pks5d9_umqtt_m1.hex
   |   |-- sdkpac_pks5d9_umqtt_awsiot.hex
   |-- tools
   |   |-- program_sdkpac_dhttp.bat
   |   |-- program_sdkpac_umqttp.bat
   |   |-- program_sdkpac_umqtts.bat
   |   |-- program_sdkpac_usmtp.bat
   |   |-- program_sdkpac_ussl_client.bat
   |   |-- program_sdkpac_ussl_server.bat
   |   |-- program_sdkpac_umqtt_m1.bat
   |   |-- program_sdkpac_umqtt_awsiot.bat
```


The following table shows the SDKPac pre-built binary demo applications, file name, and demo description:

Demo Name	**Hex File Name** ***Flash Program Batch File Name***	**Demo Description**
uSSL/TLS HTTPS server	sdkpac_kit_ussl_server.hex *program_sdkpac_ussl_server.bat*	HTTPS Connect to the Synergy kit from desktop browser using the static target IP address
uSSL/TLS HTTPS client	sdkpac_kit_ussl_client.hex *program_sdkpac_ussl_client.bat*	HTTPS GET home page at www.redhat.com
dHTTP Webserver	sdkpac_kit_dhttp.hex *program_sdkpac_dhttp.bat*	dHTTP embedded Webserver with sample content
uMQTT subscriber client	sdkpac_kit_umqtts.hex *program_sdkpac_umqtts.bat*	Connect the Synergy kit to Cyclone MQTT server using TLS/MQTT protocol to subscribe to test topic.
uMQTT publisher client	sdkpac_kit_umqttp.hex *program_sdkpac_umqttp.bat*	Connect the Synergy kit to Cyclone MQTT server using TLS/MQTT protocol to publish messages to test topic.
uSMTP email client	sdkpac_kit_usmtp.hex *program_sdkpac_usmtp.bat*	TLS secure email client. Uses USB serial port to edit and send email messages.
uMQTT Medium One Client	sdkpac_kit_umqtt_m1.hex *program_sdkpac_umqtt_m1.bat*	Connect Synergy kit to Medium One MQTT server using TLS/MQTT protocol.
uMQTT AWS IoT Client	sdkpac_kit_umqtt_awsiot.hex *program_sdkpac_umqtt_awsiot.bat*	Connect Synergy kit to AWS IoT MQTT server using TLS/MQTT protocol.

3.7 Configure e2studio JLink

The SDKPac Demo Kit batch files are based on Segger JLink v6.10 and configured so that a local copy of Jlink.exe and JLinkARM.dll are the SDKPac/tools/segger folder as shown. Copy these files as needed from your e2studio DebugComp directory:



3.7.1 Connect Target J-Link

Refer to Reference [1] Renesas PK-S5D9 Quick Start Guide for overview of the programming steps.

Confirm the Segger JLink USB drivers are installed, and connect USB mini cable to the USB connector J19 on the target board.

3.7.2 Program the Demo Application

In Windows Explorer, in the sdkpac_demokit/tools directory, **double click on the batch file to JLink program the demo application** as shown in the table above. For example, to program PK-S5D9 sdkpac_pks5d9_ussl_server.hex, click the file "program_sdkpac_ussl_server.bat. If windows security error pops up, use Run As Administrator and Run Anyway to override security warnings.

Once the demo application is programmed in the Starter Kit MCU flash memory, it is saved in the target flash memory. The target can be reset with Reset J2 any time to re-launch the demo.

3.8 Network Configuration

3.8.1 DHCP

SDKPac includes DHCP and DNS support for dynamic IP address and default gateway negotiation. Network configuration messages are output to the debug serial port terminal during target initialization. Configuration can also be viewed in the SDKPac Cyclonite GUI Setup page.

3.8.2 Static IP

SDKPac is configured with default static IP and gateway address, for use in an environment without DHCP, or direct cable connection between laptop PC and target. Default configuration is: static IP: 192.168.1.8 netmask: 255.255.255.0 gateway: 192.168.1.1

Plug in a CAT5 cable between laptop and target. Then configure a static IP address for laptop PC Ethernet port, for example 192.168.1.44. This configures the PC on the same LAN subnet address as the Synergy target.

Find Windows Network settings, right click Properties, Change adapter settings. Select adapter Ethernet, Properties. Scroll and right click to highlight 'Internet Protocol Version 4 (TCP/IPv4). Click Properties again. Now click the radio button 'Use the following IP address', and enter 192.168.1.44, subnet mask 255.255.255.0. Click OK, Close to save settings.

The Ethernet connection can be tested from a command prompt window. Press Windows key, type CMD, then in the command window type 'ping 192.168.1.8' The target will respond to the ping packet:

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\test>ping 192.168.1.8
Pinging 192.168.1.8 with 32 bytes of data:
Reply from 192.168.1.8: bytes=32 time<1ms TTL=128
Reply from 192.168.1.8: bytes=32 time<1ms TTL=128
Reply from 192.168.1.8: bytes=32 time<1ms TTL=128
Reply from 192.168.1.8: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3.8.3 Default Gateway

The uSSL client, uMQTT subscriber and Publisher demos connect to the internet via default gateway. If your gateway is configured differently that described above, and you are unable to configure your network to match these settings, contact technical support for guidance.

3.8.4 MAC Address

Typically only one Synergy Starter Kit will be tested on LAN. To test multiple kits, the network MAC address must be unique. SDKPac includes MAC address generator and configuration APIs.

The MAC address can be viewed and edited in the SDKPac Cyclonite GUI Setup page.

4 RUN THE DEMO

Now that the demo platform is setup and the installation steps completed, run the selected demo.

The SDKPac demo outputs diagnostic and trace messages on the RS232 connection. Open the Windows Teraterm terminal emulator, File, New connection to target board, selecting the Windows USB serial adapter COM port such as "COM8: USB Serial Port (COM8)"

The demos are tested using Teraterm version 4.80 (SVN#5451). Setup, Terminal settings New-line Receive: AUTO, Transmit CR, Local Echo.

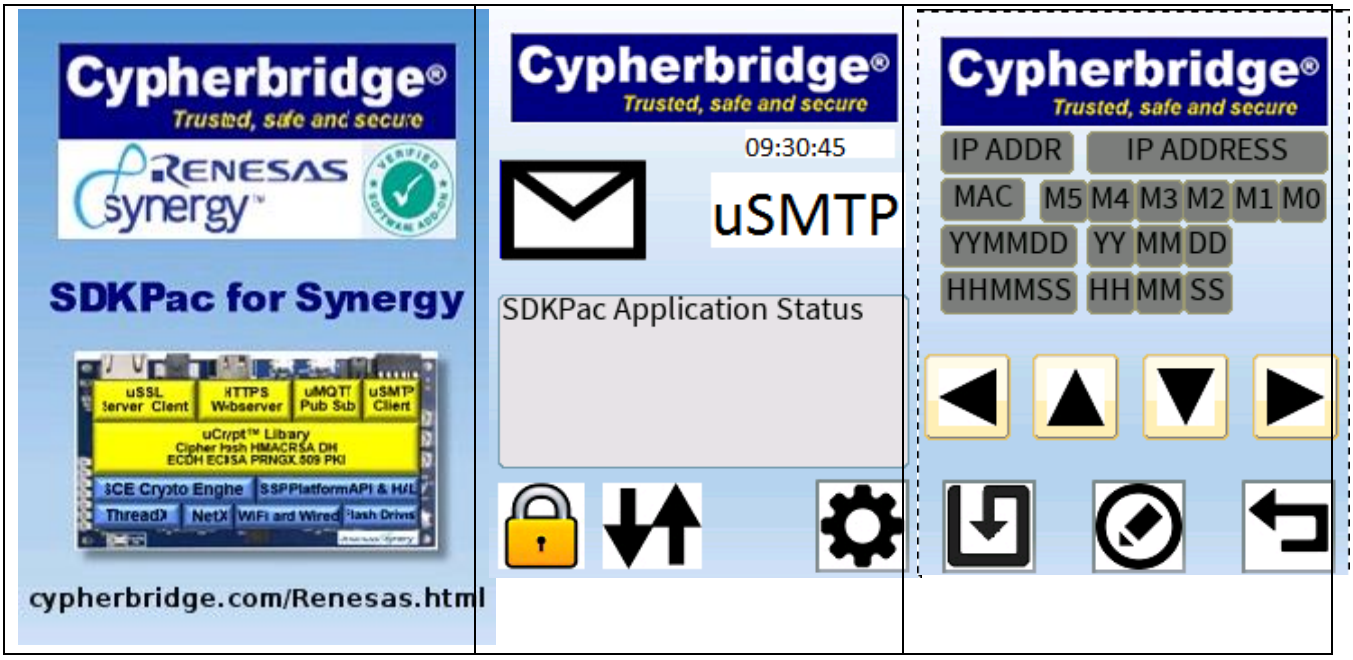
4.1 Network Initialization

SDKPac startup includes network initialization. NetX requires ETH1 CAT5 connector on main board to be connected to LAN in order to complete driver initialization, at which point link LED is on. With the CAT5 cable connected and NetX link up, the demo application will run.

SDKPac outputs debug message on UART serial port confirming the run-time static or DHCP assigned IP address. It is also shown in the SDKPac GUI in the Setup Screen. Use this IP address for the following demos to connect to the PK-S5D9.

4.2 SDKPac GUI

The SDKPac Cyclonite interactive GUI is based on Synergy GUIX and is available in the SDKPac Developer Kit. This application includes a working GUIX project that can be used to quick start your Synergy project. It includes the Splash screen, Main Screen, and Setup.



4.2.1 Main Screen

The Main Screen displays currently running Demo application, TLS lock status, network activity. Application messages are displayed in the message box.

4.2.2 Setup

From Main page press the Gear icon to go to the Setup view. This displays current system IP address (DHCP assigned or default static), Ethernet MAC address, and RTC calendar date and time. Press the Edit button to change settings. Each press of Edit cycles between MAC and RTC. Press Save button to save settings.

4.3 uSSL SDK

The uSSL SDK provides common crypto library, network protocols, and platform support for SDKPac including:

- SSL3, TLS 1.0, 1.1 and 1.2 server and client protocols
- Sample server and client applications
- Crypto library including: RSA, 3DES, AES, ARC4, SHA1, SHA2, MD2, MD4, MD5, ECC
- Common crypto library integrated with SCE Crypto engine
- Synergy platform support: debug log, BSD oriented secure sockets, system time and RTC.
- 2WAY Device Certificate mutual authentication
- X.509 PKI certificate processing, parse, generate, signing, verification, RSA, ECDSA
- Integrated memory manager zero heap solution
- Certbuilder toolkit option to generate, manage and embed X.509 device certificates

4.3.1 HTTPS Server

The uSSL HTTS server initializes and waits for a browser connection from the laptop. From the laptop browser connect to uSSL HTTPS server at the target IP address. The IP address can be confirmed in the GUI Setup page, for example 192.168.1.76. Enter in the browser address bar https://192.168.1.76

4.3.2 HTTPS Client

The uSSL client demo initializes, opens an SSL connects to WAN internet Apache Cyclone IOT server, then writes an HTTPS GET request to the server to get the home page. The client repeats this cycle. Similar to how an embedded HTTPS client application would interface to a public or private HTTPS server, this demonstrates general purpose use of uSSL secure client for project applications for HTTPS protocols. This mechanism is easily adapted to proprietary TCP protocols using the uSSL client sample application and secure socket API to transact with private back office systems or cloud services.

The following shows the uSSL client initialization, SSL connection to https://www.cycloneiot.com, followed by HTTP GET:

```
0:00:00.000:m_init pool size 30720 OK
00:00:05.530:ip_changed
```

```
00:00:05.530:dhcp_changed BOUND
00:00:06.030:netx_init_2 network status x0 rqst x3d actual x3d
00:00:06.030:netx_init_2 net0 pool size 51840 status=0
00:00:06.030:netx_init_2 UP 192.168.0.139 255.255.255.0 192.168.0.254 mss:1460
00:00:06.080:sdkpac_application_entry 00.04.A1 start #1
00:00:06.080:usslclient connecting to www.cycloneiot.com:443 on:
net0 192.168.0.139 mask 255.255.255.0 gw 192.168.0.254

00:00:06.090:ussl_stack_control uSSL.00.03.46
00:00:06.100:SERVER cert loaded
00:00:06.310:SERVER private key loaded
00:00:06.320:CA cert loaded
00:00:06.410:new_metactx mctx 0x2006d48c sz=3120
00:00:06.420:client session connect#1 sock=0x0
00:00:06.420:(0094): client hello, max TLS version: [3:3]
00:00:06.440:(0243): server hello, chosen TLS version [3:1]
00:00:06.440:(0317): server hello, chosen cipher 0x2f RSA_AES_CBC_128_SHA1
00:00:06.480:(0277): cipher is RSA_AES_CBC_128_SHA1
00:00:06.500:ussl_client_start_session result=0 state=15
00:00:06.500:usslclient connected to server www.cycloneiot.com:443
00:00:06.510:POLICY: WARN CERT not trusted
00:00:06.510:> client GET request 110 bytes
00:00:06.520:client read from server
00:00:06.530:usslclient packet=1 len=187 total=187
00:00:06.530:usslclient server response:
HTTP/1.1 200 OK
Date: Thu, 15 Jun 2017 01:31:16 GMT
Server: Apache
Last-Modified: Wed, 25 Nov 2015 00:53:02 GMT
Accept-Ranges: bytes
Content-Length: 1007
Content-Type: text/html


00:00:06.560:usslclient packet=2 len=1007 total=1194
00:00:06.560:usslclient server response:
<html>
<head><meta http-equiv="Content-Type" content="text/html; charset=us-ascii">

        <title></title>
</head>
<body bgcolor=#FFD800>
```

```
<p style="text-align: center;"></p>

<br>
<br>
<br>

<p style="text-align: center;"><span
00:00:06.590:usslclient html end total: packets=2 len=1194 protocol: TLS 1.0
00:00:06.600:net_close closing socket 0
00:00:07.600:net_close socket_disconnect err=0x41
00:00:07.600:destroyed mctx 0x2006d48c
00:00:08.600:sdkpac_application_entry 00.04.A1 start #2
00:00:08.600:usslclient connecting to www.cycloneiot.com:443 on:
net0 192.168.0.139 mask 255.255.255.0 gw 192.168.0.254
```

4.4 dHTTP Webserver

The SDKPac dHTTP webserver is a full-function embedded HTTPS webserver solution. It includes HTTP 1.1 POST, GET, CGI, ENV, SSI, CSS media plugins, redirect, and client side javascript. An offline ROM compiler builds content files into the embedded HTTP content file system used by dHTTP.

The dHTTP demo shows sample forms and navigation. From the Windows browser connect to dHTTPS at the target IP address, for example https://192.168.1.76

4.5 uMQTT Toolkit

The uMQTT Toolkit adds TLS secure MQTT 3.1 connectivity to connect your Internet of Things application to the cloud. The toolkit includes subscribe and publish APIs that can connect to industry leading cloud services and visualization applications.

- MQTT subscribe and publish APIs
- Integrated TLS secure transfer of MQTT messages between embedded target and cloud broker
- Interoperates with industry leading MQTT broker services
- Mosquitto broker compatible
- Zero-threaded, RTOS and TCP neutral

The uMQTT Toolkit includes demo applications, startup APIs and protocol library for embedded MQTT based subscribe and publish cloud messaging.

The uMQTT IoT embedded application, called Cyclone Device Client, or uCloud, exchanges TLS secure uMQTT messages with the Cypherbridge Cyclone IoT cloud at io.cycloneiot.com. A test account is used to access an example IO sensor project and topic. MQTT topics are organized like a file path hierarchy, for example project1/sim1 refers to project1 that includes topic sim1. Topics can be written by publishers and read by subscribers. Updates to the topic are replicated in Cyclone IoT cloud and propagate to other uMQTT device clients.

When the demo starts, uCloud connects to Cyclone and subscribes to topic update messages. Other IoT devices running the demo, publish sensor IO point messages to the Cyclone cloud.

The uMQTT platform layer is interfaced to the S5D9 target board. The subscriber demo receives sensor data messages from Cyclone IoT cloud, and writes to the platform actuator IO point. The publisher demo reads the sensor IO point, and sends the value to the Cyclone IoT cloud for replication. The sensor/control IO point can be simulated in the platform layer, or may be interfaced to a hardware peripheral on the S5D9, target board or shield.

Options in the form of option lists are used to configure and start the demo session. The following sections show the startup APIs and default options for the subscriber and publisher demos. The demo startup can be optionally compiled with interactive prompt enabled, so the user can edit the defaults such as IoT Cloud IP address, and login access name and access code.

4.5.1 Publisher

The publisher demo application connects to the Cyclone IoT cloud and publishes to topic project1/sim1. The demo application runs on a one second timer cycle, automatically publishing an updated sensor value each cycle. The Cyclone IoT cloud replicates the sensor messages to subscribers.

Debug messages are output to the teraterm window as shown in the following subscriber session output. Default session options including cloud server IP, device access username and passphrase, and other default settings are output, then the subscriber connects to the IoT Cloud. When the cloud connection is established, publisher updates IoT Sensor1 values each second:

```
00:00:00.000:m_init pool size 30720 OK
00:00:05.530:ip_changed
00:00:05.530:dhcp_changed BOUND
00:00:06.030:netx_init_2 network status x0 rqst x3d actual x3d
00:00:06.030:netx_init_2 net0 pool size 51840 status=0
00:00:06.030:netx_init_2 UP 192.168.0.139 255.255.255.0 192.168.0.254 mss:1460
00:00:06.080:sdkpac_application_entry 00.04.A1 start #1
default session options:
-h 192.249.115.140 -p 8883 -t project1/sim1 -m test -u synergy -a syn10101 -l 1:/tmp/upub.txt
--t00:00:06.090:umqtt_publisher start session
00:00:06.160:ussl_stack_control uSSL.00.03.46
00:00:06.170:SERVER cert loaded
00:00:06.310:SERVER private key loaded
00:00:06.320:CA cert loaded
00:00:09.480:new_metactx mctx 0x2006d684 sz=3120
00:00:09.480:client session connect#1 sock=0x0
00:00:09.490:(0094): client hello, max TLS version: [3:3]
00:00:09.510:(0243): server hello, chosen TLS version [3:3]
00:00:09.510:(0317): server hello, chosen cipher 0x3d RSA_AES_CBC_256_SHA2
00:00:09.530:sandv_verify RSA OK
00:00:09.540:sandv_verify RSA fail ret -1040
00:00:09.540:(0277): cipher is RSA_AES_CBC_256_SHA2
```

```
00:00:09.570:ussl_client_start_session result=0 state=15
00:00:09.570:POLICY: WARN CERT not trusted
00:00:09.570:syn-XoOBMJLd connected OK to 192.249.115.140:8883
00:00:09.590:umqtt_loop syn-XoOBMJLd time 6 message ready
00:00:09.590:pub_connect_callback Connection Accepted.
00:00:09.590:io_device_init ID 1
00:00:09.600:osal_task_create rc=0
00:00:09.600:pio_read_sim_1 {"SENSOR1": -888}
00:00:10.600:pio_read_sim_1 {"SENSOR1": -883}
00:00:11.600:pio_read_sim_1 {"SENSOR1": -878}
00:00:12.600:pio_read_sim_1 {"SENSOR1": -873}
00:00:13.600:pio_read_sim_1 {"SENSOR1": -868}
00:00:14.600:pio_read_sim_1 {"SENSOR1": -863}
00:00:15.600:pio_read_sim_1 {"SENSOR1": -858}
```

4.5.2 Subscriber

The subscriber demo application connects to the Cyclone IoT cloud and subscribes to topic project1/sim1. uMQTT publishers, running on different target boards, publish updates to /project1/sim1 topic. The Cyclone IoT cloud replicates the MQTT messages to the subscriber, and these replicated values are written to the platform layer.

Debug messages are output to the teraterm window as shown in the following subscriber session output. Default session options including cloud server IP, device access username and passphrase, and other default settings are output, then the subscriber connects to the IoT Cloud. When the cloud connection is established, subscriber reads IoT Sensor1 values each time they are updated:

```
00:00:00.000:m_init pool size 30720 OK
00:00:05.530:ip_changed
00:00:05.530:dhcp_changed BOUND
00:00:06.030:netx_init_2 network status x0 rqst x3d actual x3d
00:00:06.030:netx_init_2 net0 pool size 51840 status=0
00:00:06.030:netx_init_2 UP 192.168.0.139 255.255.255.0 192.168.0.254 mss:1460
00:00:06.080:sdkpac_application_entry 00.04.A1 start #1
default session options:
-h io.cycloneiot.com -p 8883 -t project1 -u synergy -a syn10101 -l 1:/tmp/usub.txt --tls-
enable t00:00:06.090:umqtt_subscriber start session
00:00:06.160:ussl_stack_control uSSL.00.03.46
00:00:06.170:SERVER cert loaded
00:00:06.310:SERVER private key loaded
00:00:06.320:CA cert loaded
00:00:06.370:new_metactx mctx 0x2006d660 sz=3120
00:00:06.370:client session connect#1 sock=0x0
00:00:06.380:(0094): client hello, max TLS version: [3:3]
```

```
00:00:06.400:(0243): server hello, chosen TLS version [3:3]
00:00:06.400:(0317): server hello, chosen cipher 0x3d RSA_AES_CBC_256_SHA2
00:00:06.430:sandv_verify RSA OK
00:00:06.430:sandv_verify RSA fail ret -1040
00:00:06.440:(0277): cipher is RSA_AES_CBC_256_SHA2
00:00:06.460:ussl_client_start_session result=0 state=15
00:00:06.460:POLICY: WARN CERT not trusted
00:00:06.470:syn-CD2tDFSD connected OK to io.cycloneiot.com:8883
00:00:06.490:umqtt_loop syn-CD2tDFSD time 6 message ready
00:00:06.490:io_device_init ID 1
00:00:06.490:osal_task_create rc=0
00:00:06.520:umqtt_loop syn-CD2tDFSD time 6 message ready
00:01:06.540:umqtt_loop syn-CD2tDFSD time 66 message ready
```

4.6 uSMTP Client

The uSMTP client provides TLS secure email. An essential form of cloud connectivity, many existing systems use SMTP to email device settings, alarms, and telemetry data. uSMTP Toolkit supports TLS implicit, explicit, and non-TLS connections. It supports ASCII in-line text, as well as text and binary MIME attachments. The design includes application callback functions to generate and write binary attachment data on the fly, allowing for large file uploads with minimal RAM impact.

The uSMTP demo application uses the USB serial port to configure the email session, edit and send email. Refer to the USB Serial Port section above for details.

In the following sample session, edit the session with your email account settings, including username, password, and SMTP server. You can check your desktop email client, such as Outlook or Thunderbird to get the account settings:

```
------------------------------
uSMTP session settings:
------------------------------

HERE ENTER EMAIL ACCOUNT SETTINGS AND CONFIRM BY ENTER 'y'.
PRESS ENTER TO ACCEPT DEFAULTS

sender: (synergy.mydomain.com):
from: (sam@mydomain.com):
to: (fred@mydomain.com):

ENTER your email account login username and password here:

username (sam@mydomain.com):
password (): xxxxxx [NOTE character input does not echo]

subject: (sdkpac for synergy usmtp):

ENTER your email server and port number.  Use default TLS mode:
```

```
smtp server (smtpout.secureserver.net):
port (465):
tls mode 0-none, 1-explicit, 2-implicit (2):

confirm settings to connect and edit email y/n (): y

07:50:31.868:usslio_client_open connect to smtpout.secureserver.net:465 tlsmode 2
07:50:31.869:ussl_stack_control uSSL.00.03.45
07:50:31.869:SERVER cert loaded
07:50:31.879:SERVER private key loaded
07:50:31.880:CLIENT cert loaded
07:50:31.883:CLIENT private key loaded
07:50:31.884:CA cert loaded
07:50:31.891:new_metactx mctx 0x20067454 sz=3460
07:50:31.891:client session connect#1 sock=0x0
07:50:31.892:(0094): client hello, max TLS version: [3:3]
07:50:31.894:(0242): server hello, chosen TLS version [3:3]
07:50:31.894:(0316): server hello, chosen cipher 0x2f RSA_AES_CBC_128_SHA1
07:50:31.917:sandv_verify RSA OK
07:50:31.933:(0277): cipher is RSA_AES_CBC_128_SHA1
07:50:31.936:ussl_client_start_session result=0 state=15
07:50:31.936:usslio_client_open mctx 0x20067454 connected to server
smtpout.secureserver.net:465
```

HERE ENTER EMAIL TEXT LINES. TERMINATE MESSAGE WITH PERIOD (.)

```
enter email text (. to end) (): hello from sdkpac for synergy test
enter email text (. to end) (): .

07:50:54.231:usslio_client_close mctx 0x20067454
07:50:55.332:net_close socket_disconnect err=41x
07:50:55.332:destroyed mctx 0x20067454
07:50:55.332:smtp returned 0
07:50:55.332:dynapool.stats valid=1 alloc=10964 free=25892 max=23736, get=457
rel=386
```

HERE CHECK YOUR EMAIL RECIPIENT TO CONFIRM THE MESSAGE

4.7 uMQTT Medium One Client

The uMQTT Medium One client connects to the Medium One IoT cloud server. Similar to the uMQTT Publisher demo above, it publishes messages to the Medium One cloud.

```
00:00:00.000:m_init pool size 30720 OK
00:00:05.530:ip_changed
00:00:05.530:dhcp_changed BOUND
00:00:06.030:netx_init_2 network status x0 rqst x3d actual x3d
00:00:06.030:netx_init_2 net0 pool size 51840 status=0
```

```
00:00:06.030:netx_init_2 UP 192.168.0.139 255.255.255.0 192.168.0.254 mss:1460
00:00:06.080:sdkpac_application_entry 00.04.A1 start #1
00:00:06.080:umqtt_new
00:00:06.090:CA cert loaded
00:00:06.090:cert_x509_load cert 2 loaded
00:00:06.090:umqttapi_demo_start initialized umqtt session version v0.1.1E
00:00:06.100:umqtt_new
00:00:06.100:cert_x509_load cert 2 already loaded
00:00:06.110:umqttapi_demo_start registered callbacks
00:00:06.110:demo_loop
00:00:06.110:ussl_stack_control uSSL.00.03.46
00:00:06.120:ussl_stack_control USSLCERT_PRELOAD_AUTO disabled
00:00:06.290:new_metactx mctx 0x2006e570 sz=3120
00:00:06.290:client session connect#1 sock=0x0
00:00:06.370:(0094): client hello, max TLS version: [3:3]
00:00:07.210:(0243): server hello, chosen TLS version [3:3]
00:00:07.210:(0317): server hello, chosen cipher 0x2f RSA_AES_CBC_128_SHA1
00:00:07.220:sandv_verify RSA OK
00:00:07.230:sandv_verify RSA fail ret -1024
00:00:07.240:(0277): cipher is RSA_AES_CBC_128_SHA1
00:00:07.350:ussl_client_start_session result=0 state=15
00:00:07.350:POLICY: WARN CERT not trusted
00:00:07.360:16:Client xyz123 sending CONNECT
00:00:07.360:umqtt_connect rc 0
00:00:07.360:demo_loop connect 167.114.77.233 0
00:00:08.370:umqtt_loop xyz123 time 6 message ready
00:00:08.370:16:Client xyz123 received CONNACK
00:00:08.370:connect_callback 167.114.77.233:61620 rc:0
00:00:08.380:demo_cb_connect rc=0
00:00:08.380:dynapool.stats valid=1 alloc=24888 free=5824 max=5824, get=82 rel=6
00:00:08.390:16:Client xyz123 sending SUBSCRIBE (Mid: 1, Topic:
1/AfzvMXSqmrM/4KEHYifLft0/synergy/event, QoS: 0)
00:00:08.400:umqtt_subscribe rc 0
00:00:08.400:16:Client xyz123 sending PUBLISH (d0, q0, r0, m2,
'0/AfzvMXSqmrM/4KEHYifLft0/synergy', ... (32 bytes))
00:00:08.410:publish_callback mid 2
00:00:08.410:demo_cb_publish 2 rc=0
00:00:09.420:umqtt_loop xyz123 time 8 message ready
00:00:09.420:16:Client xyz123 received SUBACK
00:00:09.420:subscribe_callback mid 1:1:0
00:00:09.430:demo_cb_subscribe 1 rc=0
00:00:09.430:16:Client xyz123 sending PUBLISH (d0, q0, r0, m3,
'0/AfzvMXSqmrM/4KEHYifLft0/synergy', ... (32 bytes))
```

```
00:00:09.440:publish_callback mid 3
00:00:09.440:demo_cb_publish 3 rc=0
00:00:10.450:16:Client     xyz123     sending     PUBLISH     (d0,     q0,     r0,     m4,
'0/AfzvMXSqmrM/4KEHYifLft0/synergy', ... (32 bytes))
00:00:10.450:publish_callback mid 4
00:00:10.460:demo_cb_publish 4 rc=0
00:00:11.460:16:Client     xyz123     sending     PUBLISH     (d0,     q0,     r0,     m5,
'0/AfzvMXSqmrM/4KEHYifLft0/synergy', ... (32 bytes))
00:00:11.460:publish_callback mid 5
00:00:11.470:demo_cb_publish 5 rc=0
```

4.8 uMQTT Amazon AWS IoT Client

The uMQTT AWS IoT client connects to the Amazon AWS IoT cloud server. Similar to the uMQTT Publisher demo above, it publishes messages to AWS IoT:

```
00:00:00.000:m_init pool size 30720 OK
00:00:05.530:ip_changed
00:00:05.530:dhcp_changed BOUND
00:00:06.030:netx_init_2 network status x0 rqst x3d actual x3d
00:00:06.030:netx_init_2 net0 pool size 51840 status=0
00:00:06.030:netx_init_2 UP 192.168.0.139 255.255.255.0 192.168.0.254 mss:1460
00:00:06.080:sdkpac_application_entry 00.04.A1 start #1
00:00:06.080:umqtt_new
00:00:06.090:CA cert loaded
00:00:06.090:cert_x509_load cert 2 loaded
00:00:06.100:CLIENT cert loaded
00:00:06.240:CLIENT private key loaded
00:00:06.240:cert_x509_load cert 1 loaded
00:00:06.240:umqttapi_demo_start initialized umqtt session version v0.1.1E
00:00:06.250:umqtt_new
00:00:06.250:cert_x509_load cert 2 already loaded
00:00:06.260:cert_x509_load cert 1 already loaded
00:00:06.260:umqttapi_demo_start registered callbacks
00:00:06.270:demo_loop
00:00:06.270:ussl_stack_control uSSL.00.03.46
00:00:06.270:ussl_stack_control USSLCERT_PRELOAD_AUTO disabled
00:00:06.360:new_metactx mctx 0x2006d2f0 sz=3120
00:00:06.360:client session connect#1 sock=0x0
00:00:06.440:(0094): client hello, max TLS version: [3:3]
```

```
00:00:06.600:(0243): server hello, chosen TLS version [3:3]
00:00:06.600:(0317): server hello, chosen cipher 0x2f RSA_AES_CBC_128_SHA1
00:00:06.610:sandv_verify RSA OK
00:00:06.620:sandv_verify RSA OK
00:00:06.630:(0277): cipher is RSA_AES_CBC_128_SHA1
00:00:06.780:ussl_client_start_session result=0 state=15
00:00:06.790:POLICY: WARN CERT expired
00:00:06.790:16:Client xyz123 sending CONNECT
00:00:06.800:umqtt_connect rc 0
00:00:06.800:demo_loop connect ahuy0gy6pjpj4.iot.us-west-2.amazonaws.com 0
00:00:07.800:umqtt_loop xyz123 time 6 message ready
00:00:07.800:16:Client xyz123 received CONNACK
00:00:07.800:connect_callback ahuy0gy6pjpj4.iot.us-west-2.amazonaws.com:8883 rc:0
00:00:07.810:demo_cb_connect rc=0
00:00:07.810:dynapool.stats valid=1 alloc=24020 free=6692 max=6692, get=99 rel=28
00:00:07.820:16:Client xyz123 sending SUBSCRIBE (Mid: 1, Topic: $aws/things/Device/sensor,
QoS: 0)
00:00:07.830:umqtt_subscribe rc 0
00:00:07.830:16:Client xyz123 sending PUBLISH (d0, q0, r0, m2, '$aws/things/Device/sensor',
... (32 bytes))
00:00:07.840:publish_callback mid 2
00:00:07.840:demo_cb_publish 2 rc=0
00:00:08.850:umqtt_loop xyz123 time 7 message ready
00:00:08.850:16:Client xyz123 received SUBACK
00:00:08.850:subscribe_callback mid 1:1:0
00:00:08.860:demo_cb_subscribe 1 rc=0
00:00:08.860:16:Client xyz123 sending PUBLISH (d0, q0, r0, m3, '$aws/things/Device/sensor',
... (32 bytes))
00:00:08.870:publish_callback mid 3
00:00:08.870:demo_cb_publish 3 rc=0
00:00:09.880:umqtt_loop xyz123 time 8 message ready
00:00:09.880:16:Client xyz123 received PUBLISH $aws/things/Device/sensor (sz32 d0, q0, r0,
m32)
00:00:09.890:message_callback 0:$aws/things/Device/sensor:{"event_data":{"temperature":1}}
00:00:09.890:demo_cb_message $aws/things/Device/sensor:{"event_data":{"temperature":1}} rc=0
00:00:09.900:16:Client xyz123 sending PUBLISH (d0, q0, r0, m4, '$aws/things/Device/sensor',
... (32 bytes))
00:00:09.910:publish_callback mid 4
00:00:09.920:demo_cb_publish 4 rc=0
00:00:10.920:umqtt_loop xyz123 time 9 message ready
00:00:10.920:16:Client xyz123 received PUBLISH $aws/things/Device/sensor (sz32 d0, q0, r0,
m32)
00:00:10.930:message_callback 0:$aws/things/Device/sensor:{"event_data":{"temperature":2}}
00:00:10.930:demo_cb_message $aws/things/Device/sensor:{"event_data":{"temperature":2}} rc=0
```



```
00:00:10.940:16:Client xyz123 sending PUBLISH (d0, q0, r0, m5, '$aws/things/Device/sensor',
... (32 bytes))
00:00:10.950:publish_callback mid 5
00:00:10.960:demo_cb_publish 5 rc=0
```

5 Additional Information

5.1 Copyright and License

Visit us on the web at: www.cypherbridge.com/Renesas.html

Email us at: renesas@cypherbridge.com

5.2 Cypherbridge Product Catalog

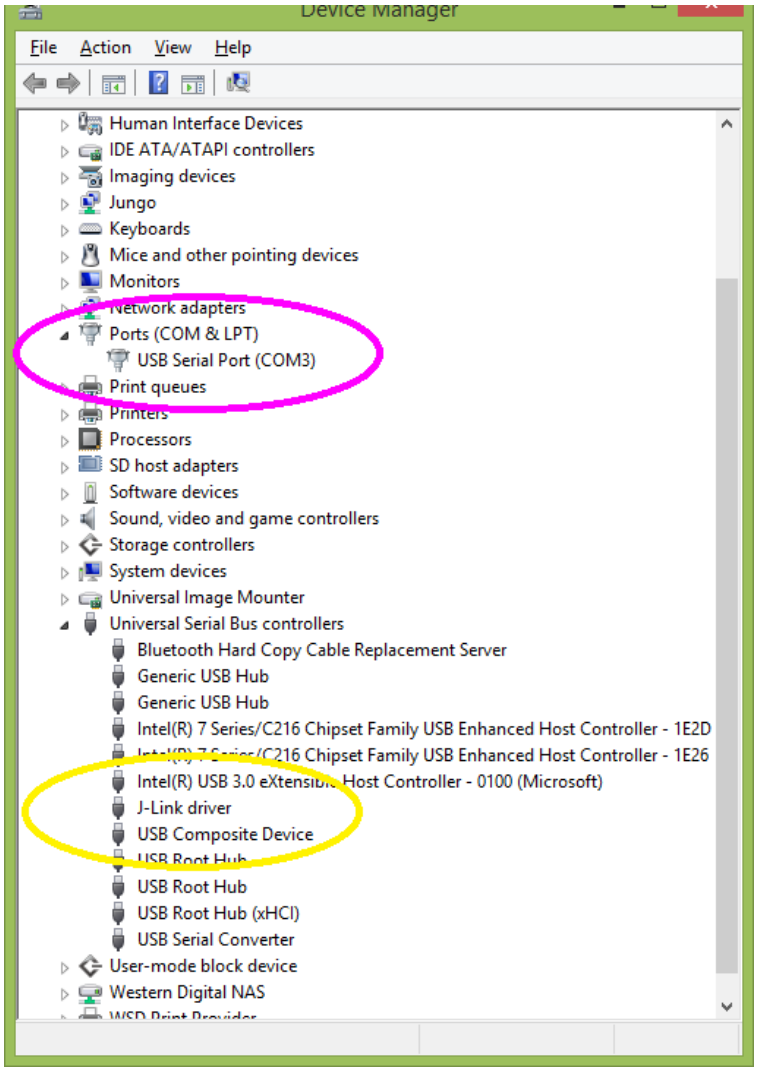
Visit Cypherbridge Systems on the web at http://cypherbridge.com/ProductsServices.html and check out our wide range of embedded SDKs and Toolkits:

- ✓ uSSL SSL/TLS SDK, client and server, uCrypt crypto library,
- ✓ dHTTPS webserver
- ✓ uSSH embedded SSH SDK, client and server, uSCP option
- ✓ uFTP FTPS client based on uSSL SDK and uCrypt common crypto library
- ✓ uLoad Install Defender SDK – secure loader and safe software installer
- ✓ uVPN SDK – IPsec and IKEv1/2 initiator and responder, tunnel and transport modes
- ✓ uFile Toolkit – encrypted file system Toolkit based on uCrypt common crypto library
- ✓ uCloud Device Client SDK – IoT direct-to-cloud file system replicates, scales and synchronizes across cloud and target platforms.
- ✓ uMQTT Toolkit – low footprint MQTT 3.1.1 protocol library with subscribe/publish APIs integrated with uSSL TLS secure network transport. Built in support for IoT clouds including Cyclone, Medium One, and AWS IoT.
- ✓ uSMTP TLS secure email client
- ✓ FIDO eU2F and eUAF Toolkits – Universal Two Factor Authentication message and crypto library toolkits

6 Appendix A: Troubleshooting

If the SDKPac Demo Kit application fails to download or run, here are some recommended checks and troubleshooting steps.

- Start by reviewing the installation section of this Guide to verify the 5V power, RJ45 ethernet, J-Link USB, and RS232 UART is properly connected.
- Confirm J-Link USB cable connected to Starter Kit J19, Windows Device Manager, confirm J-Link USB driver is correctly installed and shown in the USB devices section
- Confirm that the FTDI USB serial adapter is enumerated in Windows as shown in the COM port section



- For problems browsing your Windows PC browser to target board, confirm network settings that Windows PC is on same LAN subnet as the starter kit using static IP address 192.168.1.8.

- Open a Windows command window, then enter the command ‘ping 192.168.1.8’ to confirm network connection

After these steps if problems continue, Contact Renesas technical support at the support contacts listed in Section 1.

7 Appendix B: uMQTT Platform IO Interface

Contact Cypherbridge Systems for the uMQTT Toolkit Quick Start Guide. The SDKPac source code project distribution uMQTT Toolkit includes platform interface and JSON components. These files can be modified to interface SDKPac to your target platform actuators and sensors, such as LEDs, serial channels, thermal sensors, accelerometers, etc.

Wired or wireless local area networks such as CANbus, Zigbee, BLE, Enocean, and so on can be implemented in this layer to add IoT gateway to cloud functionality using SDKPac uMQTT.

8 Appendix C: uMQTT Test Client

The SDKPac uMQTT Publisher section above shows how to start sdkpac umqttp to publish simulated sensor data messages to the Cyclone IoT Cloud

These published MQTT messages can be monitored with MQTT client tools and GUIs including these recommendations and reviews:

http://www.hivemq.com/seven-best-mqtt-client-tools/

8.1 MyMQTT Android Setup

MyMQTT for Android is an easy to use subscribe and publish app. Install it from Google Play store:

https://play.google.com/store/apps/details?id=at.tripwire.mqtt.client&hl=en

To configure it:

Open MyMQTT

On top toolbar, left arrow, go to Settings

Enter Cyclone IoT Cloud server and credentials

MQTT Broker: io.cycloneiot.com

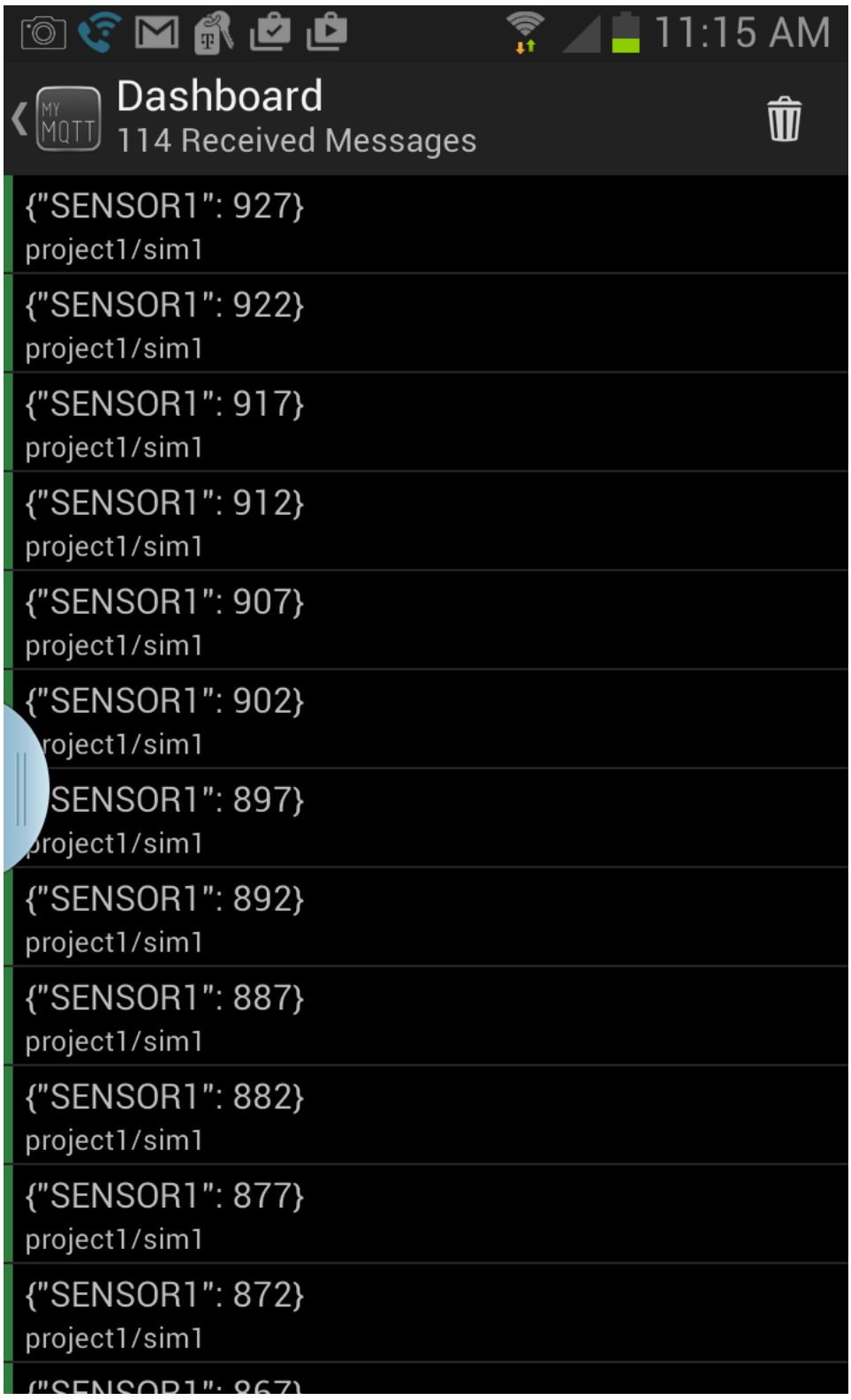
Port: 1883

User: mobile1

Pass: mcycle1

Next Subscribe to topic project1/sim1

Then go to Dashboard to monitor the dashboard to watch topics as they are published in real-time from the Synergy kit as show in this screen shot:



9 Revision History

Revision	Date	Description
0.1.0	21 Sep 2015	First draft
0.1.1	25 Nov 2015	SDKPac Beta 2 updates
0.1.2	15 Jan 2016	SDKPac Gallery Release 00.01.A0
0.1.3	29 Mar 2016	SDKPac for SK-S7G2
1.0.1	5 May 2016	Updated for SDKPac 00.00.1D installation section, Network initialization, Cyclonite GUI
1.0.2	28 May 2016	SDKPac 00.02.A1 GUI updates and added uSMTP application demo
1.0.3	5 June 2016	SDKPac 00.02.A2 GUI updates and added uSMTP application demo
1.0.4	28 June 2016	SDKPac 00.02.C1 QA release
1.0.5	12 July 2016	Touchups to file names in section 3.6.1
1.0.6	27 July 2016	SDKPac ECC is supported for project specific applications, however demo kit excludes ECC. Block diagram updated.
1.0.7	14 June 2017	Updated for SDKPac 00.04.A1 with PK-S5D9 support

