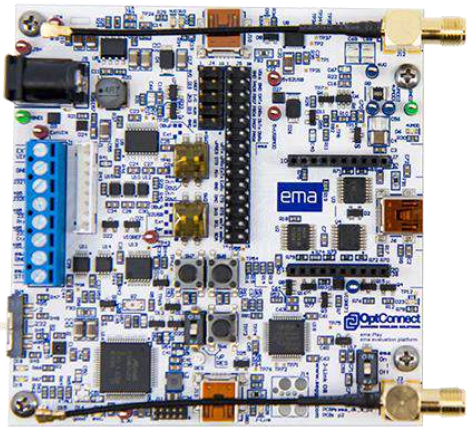
OptConnect ema™

ema:Play User Guide

V1.0 Updated May 2020



OptConnect
MANAGED WIRELESS SOLUTIONS

Table of Contents

1. Introduction

1.1 Scope

The OptConnect ema:Play Evaluation Kit is a hardware platform designed and optimized for evaluating OptConnect ema™ modems. ema:Play provides easy access to the features of OptConnect ema™ modems, and acts as a launching point for translation into custom IoT/M2M applications using OptConnect ema™. This document serves as a guide and point of reference for using ema:Play correctly and efficiently.

1.2 Contact Information

For more information regarding OptConnect ema™ contact OptConnect Sales at 1.877.678.3343 ext. 2020 during normal business hours. For technical support contact OptConnect Customer Care Center at 1.877.678-3343 ext. 2021 from 8 am till 9 pm MST Monday through Saturday.

1.3 Orderable Part Numbers

Orderable Device	Primary Module Firmware Revision	Operating Temperature	LTE Bands	3G UMTS	Network	Region
EMA-L4-1-XX-A-A	20.00.505	-40 to +85˚C	FDD B2, B4, B5, B12, B13	B2, B5	AT&T, Verizon	North America
EMA-L4-1-US-B-A	20.00.005	-40 to +85˚C	FDD B2, B4, B5, B12, B13	B2, B5	AT&T, Verizon	United States
EMA-L4-1-XX-A-A-000	20.00.506	-40 to +85˚C	FDD B2, B4, B5, B12, B13	B2, B5	AT&T, Verizon	North America
EMA-L4-1-US-B-A-000	20.00.006	-40 to +85˚C	FDD B2, B4, B5, B12, B13	B2, B5	AT&T, Verizon	United States

Unless instructed otherwise EMA-L4-1-XX… will utilize AT&T as the primary carrier and Verizon as the secondary carrier. Unless instructed otherwise, EMA-L4-1-US… will utilize Verizon as the primary carrier and AT&T as the secondary carrier.

Orderable Device	Description	Operating Temperature	Region
EMA-ZZ-1-XX-Z-B	ema:Play Evaluation Kit, OptConnect ema™ evaluation platform	-40 to +85˚C	North America
EMA-L4-1-XX-A-B	ema:Play Evaluation Kit, OptConnect ema™ evaluation platform, EMA-L4-1-XX ema modem included	-40 to +85˚C	North America
EMA-L4-1-US-B-B	ema:Play Evaluation Kit, OptConnect ema™ evaluation platform, ema EMA-L4-1-US ema modem included	-40 to +85˚C	United States

1.4 Additional Resources

OptConnect ema™ is supported by a full range of documentation, including User Guides and Application Notes as well as related code samples. The latest versions of these resources can be found at **http://optconnect.com/ema** . Suggested prerequisites for this document are the following:

- OptConnect ema™ Hardware Guide
- OptConnect ema™ Getting Started
- OptConnect ema™ emaLink AT Command Manual

1.5 Activation

If ema is not activated/registered upon receipt, please visit **https://summit.optconnect.com/ema** to activate/register ema. Once activated, ema:Play includes a free trial period that will last for three months, with 1 GB of data being available per month. Please contact OptConnect (see section 1.2) for further questions and requests in regard to the trial period.

2. Overview

2.1 Contents

The ema:Play Evaluation Kit is shipped as a ready to use, "all-in-one" platform for evaluating OptConnect ema™ modems for IoT/M2M applications. The contents of the kit are listed below.

1. ema:Play Evaluation Kit with/without ema installed (see section 1.3 for device p/n's)
2. OptConnect Gemini MIMO Antenna
3. 5VDC Power Supply
4. 2 x USB cables

2.2 Features

- On-Board Microcontroller Unit (MCU)
    - Microchip ATSAME51J20A.
    - 120MHz Core Processor
    - 1 MB Flash, 256 KB Ram
    - On Board programmer/debugger
        - JLink OB Technology
        - Cortex SWD Connector for external programmers/debuggers
    - Debug Port
        - UART to USB access
        - External Header access
    - Voltage Monitors
        - Input Voltage
        - ema Voltage
- Configurable Power Supply
    - Wide input voltage range 4.5V – 30V
    - External power input terminal for other sources (Battery, benchtop, etc.)
    - Standard barrel connector input
    - USB power input (excludes powering ema)

    - ema load switch for low power applications
- Configurable ema Communication Interfaces
    - Modem UART interface access
        - RS485/422 full-Duplex
        - RS485 half-Duplex with Auto Direction Control
        - On-Board MCU
        - External header access
        - Serial 2 USB (S2USB) for computer terminal access
    - emaLink interface access
        - On-Board MCU
        - External header access
    - ema USB interface
    - Screw terminal block access for wire to board
    - Shrouded connector access for board to board
- Built in sensors and feedback
    - Temperature and humidity
    - Light/photo
    - 2 user push buttons
    - LED cellular signal array (Blue)
    - 2 x LEDs (Red)
- User Application Headers
    - Input/Output Application Header
        - Access to On-Board MCU
            - 1 x i2c
            - 1 x CAN bus
            - 4 x UART
            - 1 x Analog to Digital (ADC)
            - 1 x Digital to Analog (DAC)
            - 6 x General Purpose Input Output (GPIO)
        - Access to ema communication interfaces
            - Modem UART
            - emaLink
    - Power Application Header
        - ema power control
        - 3.3V power control
        - ema IOVREF control
        - ema Reset Request control
        - ema On/Off control
        - 3.3V, 4.0V for external usage
- Easy grab test points for measurements
- Efficient design
    - Cost and availability optimized
- Antenna adaptors built in
    - U.FL to SMA for both main and diversity
- Electrical design CAD available for reference

3. Hardware

Section 3 outlines the hardware of ema:Play from a user perspective. *Figures 1-3* can be used as quick reference guides for navigating the various components and features of ema:Play. The hardware is designed for flexibility and configurability to support different user IoT/M2M applications and integration architectures, and to also serve as a reference for custom hardware designs integrating ema.

3.1 At a Glance

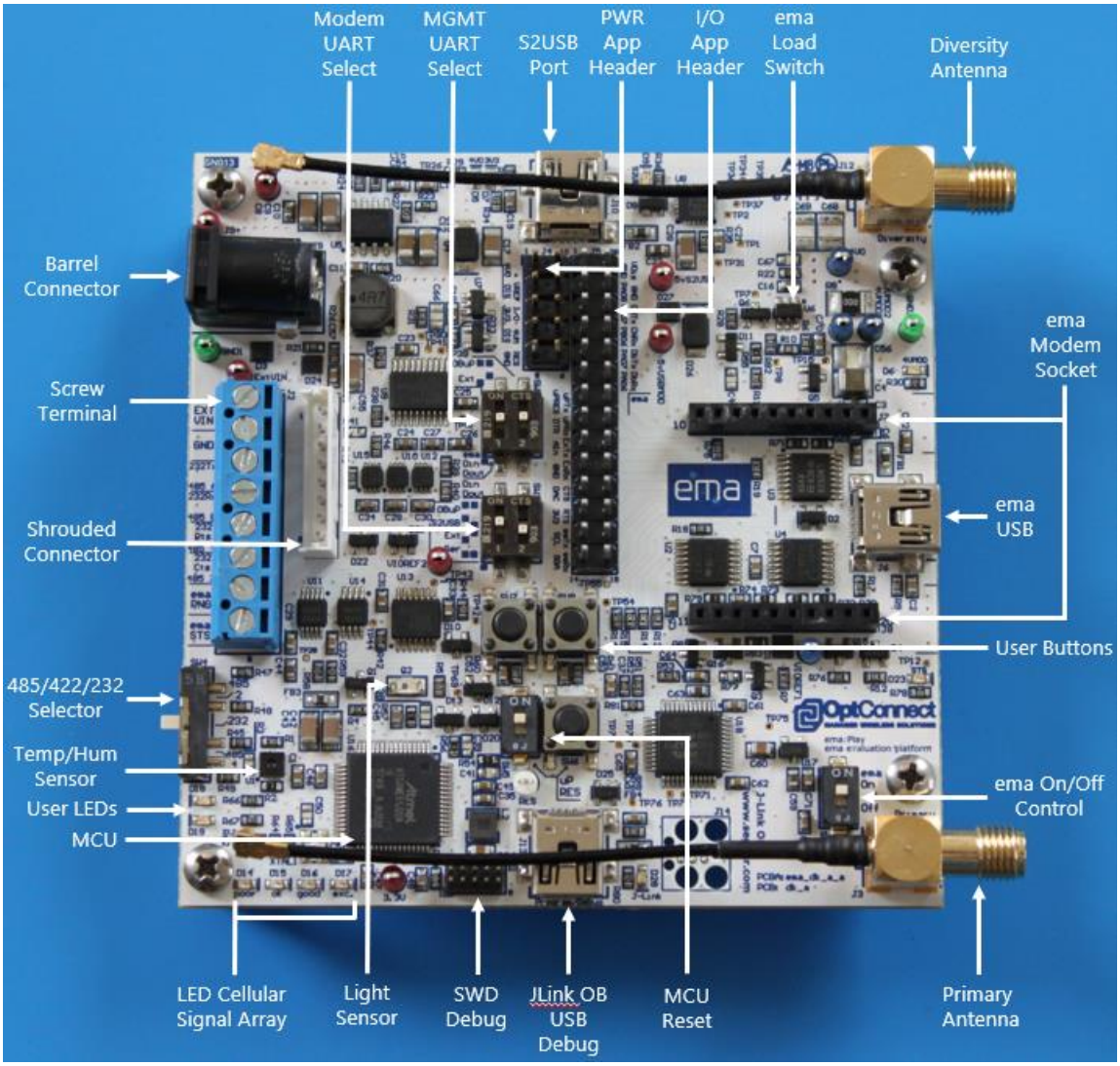


Figure 1

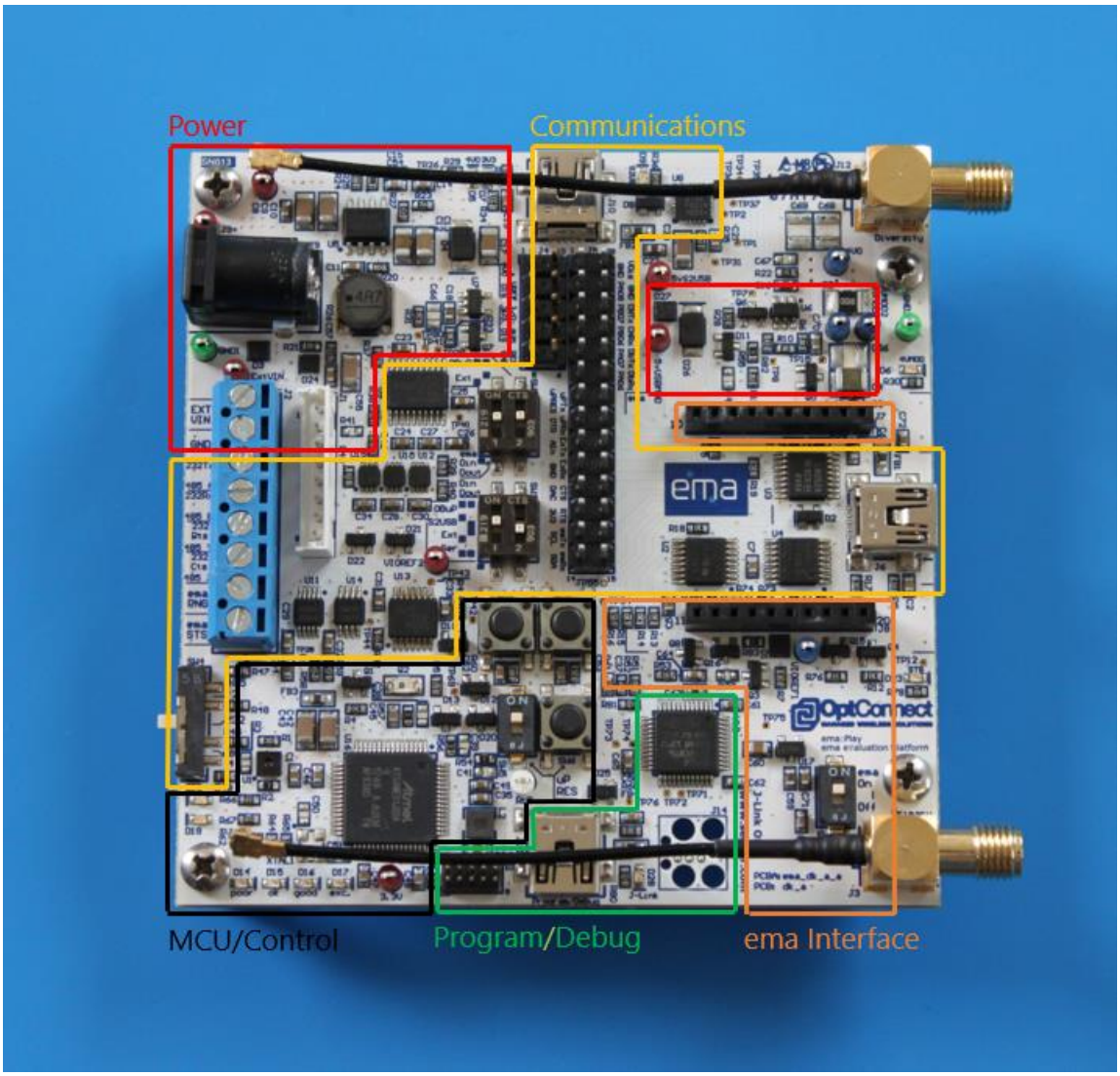


Figure 2

3.2 System Architecture

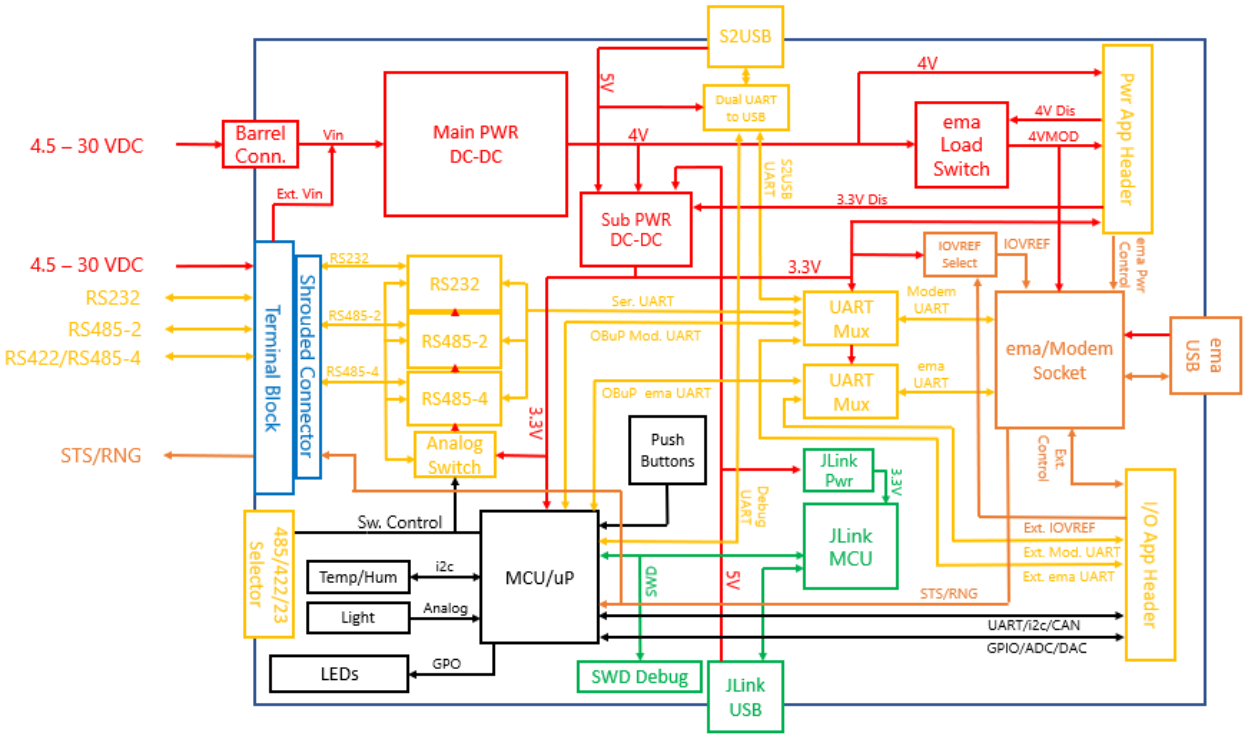


Figure 3

3.3 Power

ema:Play can be powered through multiple connectors and/or ports. Furthermore, depending on which connector/port is powering ema:Play, will also determine which sub systems of the hardware are powered. There are various feedback LED's that are provided to let the user know which sub systems are currently powered. In addition to power input configurations, ema:Play also offers power output for applications requiring off board power:

Table 1 summarizes the power input architecture and configurations.

Table 2 summarizes related power control signals.

Description	Voltage (ref)	Sources	Inputs Ports (ref)	Outputs Ports (ref)	Systems Powered	Feedback (ref)
Main Power Input	**Min 4.5V (VIN)** **Max 30V (VIN)** **Min 6.75W**	External Source	Barrel Connector (J9) External VIN (J2/J1)	Test Hook (J9+) Test Hook (ExtVIN, VIN)	4V, 4VMOD(ema), 3.3V	4V (D5) 4VMOD (D6) 3.3V (D7)
S2USB USB	5V (5V_S2USB)	External USB Port	S2USB Port (J10)	Test Hook (5vS2USB)	3.3V, S2USB	3.3V (D7) S2USB (D9)
JLink USB	5V (5V_USBJL)	External USB Port	JLink USB Port (J13)	Test Point (TP76)	3.3V, JLink	3.3V(D7) JLink (D28)
Main Regulator (U5) Output	4V (4V0) **Max 3A**	Main Power Input	na	Test Hook (4V0) Pwr App Hdr (J4.1)	4V, 4VMOD(ema), 3.3V	4V (D5) 4VMOD (D6) 3.3V (D7)
Load Switch (U6) Output	4V (4VMOD) **Max 2A**	4V (4V0)	na	Test Hook (4VMOD1, 4VMOD2) Pwr App Hdr (J4.7)	4VMOD(ema)	4VMOD (D6)
MCU & Interface Power (U7)	3.3V (3V3) **Max 1A**	4V (4V0) 5V (5V_S2USB) 5V (5V_USBJL)	na	Test Hook (3.3V) Pwr App Hdr (J4.3) IO App Hdr (J5.12)	3.3V	3.3V (D7)
ema IOVREF	**Min 1.8V** (IO_VREF) ***Max 5.5V** (IO_VREF)	3.3V (3V3) External via Pwr App Hdr (J4.9)	Pwr App Hdr (J4.9)	Test Hook (VIOREF1, VIOREF2)	ema IO interface	na

Table 1

* Onboard MCU (U16) must be held in reset if IOVREF is supplied as >4.1V

Description	Signal (ref)	Systems Disabled	Notes
ema Power disable	4VMOD_DIS (J4.2)	ema Power	This signal can be driven high to disable ema's power for power sensitive applications
MCU & interface Power disable	3V3_DIS (J4.4)	3.3V Power rail	This signal can be driven high to disable the entire 3.3V power rail. Typically, this would be used if an external host embedded system is interfacing to ema using the Application headers
Onboard IOVREF disable	OB_uP_IOVREF_DIS (J5.28)	Onboard 3.3V IOVREF	This signal can be used to set ema's IOVREF level from an external source

Table 2

3.4 Communications

ema:Play provides several useful and common communication interfaces for sending and receiving data to and from ema. Additionally, standard interfaces are offered to communicate directly with the Onboard MCU (U16) for further flexibility. Refer to the following tables for identifying which interface will work best for the application:

Table 3 summarizes the ema Modem, emaLink, and USB Interface configurations.

Table 4 summarizes the Onboard MCU (U16) interface configurations.

ema Interface	ema:Play Interface	Access Port (ref)	Configuration Switches (ref)	Use Case Example
ema Modem UART Interface	S2USB Standard Comm Port	S2USB USB Port (J10)	Dipswitch (SW2)	This interface can be used to access the ema Modem UART from a computer terminal program and manually send AT commands
	RS232	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Dipswitch (SW2) Slide Switch (SW4)	This interface can be used to access the ema Modem UART using RS232 from an external device, to send AT commands
	RS422/RS485-4	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Dipswitch (SW2) Slide Switch (SW4)	This interface can be used to access the ema Modem UART using RS422/RS485 full duplex from an external device, to send AT commands
	RS485-2	Wire to Board Terminal (J2.3, J2.4) Board to Board Terminal (J1.3, J1.4)	Dipswitch (SW2) Slide Switch (SW4)	This interface can be used to access the ema Modem UART using RS485 half-duplex from an external device, to send AT commands
	Onboard MCU (U16)	Onboard MCU (U16.44, U16.45)	Dipswitch (SW2)	This interface can be used by the Onboard MCU (U16) to access the ema Modem UART to send AT commands
	External Device	IO App Hdr (J5.17-J5.20)	Dipswitch (SW2)	This interface can be used by an external embedded device to access the ema Modem UART to send AT commands
emaLink Interface	Onboard MCU (U16)	Onboard MCU (U16.35, U16.36)	Dipswitch (SW3)	This interface can be used by the Onboard MCU (U16) to access the emaLink interface for ema management features
	External Device	IO App Hdr (J5.15, J5.16)	Dipswitch (SW3)	This interface can be used by an external embedded device to access the emaLink interface for ema management features

ema USB	ema USB Port	ema USB Port		This interface can be used to directly access the ema USB interface

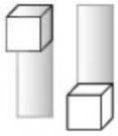
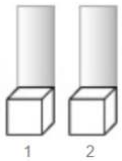
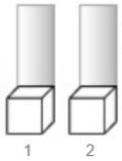
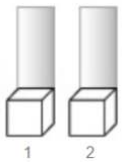
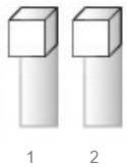
Table 3

Onboard MCU (U16) Interface	ema:Play Interface	Access Port (ref)	Configuration Switches (ref)	Use Case Examples
UART	RS232	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Slide Switch (SW4)	This interface can be used to access the Onboard MCU (U16) using RS232 from an external device
	RS422/RS485-4	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Slide Switch (SW4)	This interface can be used to access the Onboard MCU (U16) using RS422/RS485 full duplex from an external device
	RS485-2	Wire to Board Terminal (J2.3, J2.4) Board to Board Terminal (J1.3, J1.4)	Slide Switch (SW4)	This interface can be used to access the Onboard MCU (U16) using RS485 half-duplex from an external device
	External Device	IO App Hdr (J5.21-J5.24)		This interface can be used to access the Onboard MCU (U16) using an external embedded device using UART
	S2USB Enhanced Comm Port	S2USB USB Port (J10)		This interface can be used as a debug/console input/output port for the Onboard MCU (U16)
i2c	External Device	IO App Hdr (J5.13, J5.14)		This interface can be used to interface to the Onboard MCU (U16) using an external device over i2c
CAN	External Device	IO App Hdr (J5.25, J5.26)		This interface can be used to access the Onboard MCU (U16) using an external CAN transceiver
GPIO	External Device	IO App Hdr (J5.2-J5.7)		This interface can be used to interface to the Onboard MCU (U16) using GPIO and/or peripherals supported by the MCU.

Table 4

ema:Play has built in configuration switches to allow the user to easily select which communication interface to use in their application. Alternatively, the Onboard MCU (U16) can be used to control the communication interface selection via custom firmware. The architecture is such that the ema Modem UART signals and the emaLink interface signals can be directed according to *Table 5* using dipswitches (SW2, SW3), and the slide switch (SW4).

Table 5 summarizes how to configure the ema:Play hardware for the desired ema interface. Alternatively, the switch settings are also labeled on the ema:Play PCB silkscreen for quick reference.

Desired ema Interface	Desired ema:Play Interface	Access Port (ref)	Switch Configuration
ema Modem UART Interface	S2USB Standard Comm Port	S2USB USB Port (J10)	Dipswitch (SW2):  **S2USB** 1 2
	RS232	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Dipswitch (SW2) Slide Switch (SW4)   **Ser** 1 2 **232**
	RS422/RS485-4	Wire to Board Terminal (J2.3-J2.6) Board to Board Terminal (J1.3-J1.6)	Dipswitch (SW2) Slide Switch (SW4)   **Ser** 1 2 **485 4**
	RS485-2	Wire to Board Terminal (J2.3, J2.4) Board to Board Terminal (J1.3, J1.4)	Dipswitch (SW2) Slide Switch (SW4)   **Ser** 1 2 **485 2**
	Onboard MCU (U16)	Onboard MCU (U16.44, U16.45)	Dipswitch (SW2)  **OBuP** 1 2

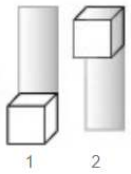
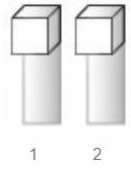
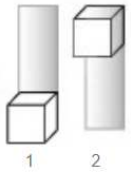
	External Device	IO App Hdr (J5.17-J5.20)	Dipswitch (SW2)  **Ext**
emaLink Interface	Onboard MCU (U16)	Onboard MCU (U16.35, U16.36)	Dipswitch (SW3)  **OBuP**
	External Device	IO App Hdr (J5.15, J5.16)	Dipswitch (SW3)  **Ext**
ema USB	ema USB Port	ema USB Port (J6)	

Table 5

3.5 Onboard MCU (U16)

ema:Play provides an onboard MCU (U16) to allow users to develop their own custom IoT/M2M applications using ema. There are various sensors, peripherals, and LED's integrated into the ema:Play hardware that can be accessed by U16. Please visit the following website for more information regarding the onboard MCU (U16):

https://www.microchip.com/wwwproducts/en/ATSAME51J20A

Table 6 summarizes the various sensors and peripherals available to U16 for customization.

Sensor/Peripheral (ref)	Interface to U16 (ref)
Temperature/Humidity (U1)	i2c (U16.29, U16.30)
Photo/Light (Q2)	A/D (U16.61)
Main Voltage Monitor (VIN_MON)	A/D (U16.63)
ema Voltage Monitor (MOD_V_MON)	A/D (U16.62)

External A/D Monitor (OB_uP_AIN)	A/D (U16.64)
External High Accuracy Crystal (XTAL1)	XIN/XOUT (U16.31, U16.32)
Push Buttons (SW7, SW8)	Digital Input (U16.19, U16.20)
Manual Reset (SW5, SW6)	Digital Input (U16.52)
External D/A Output (OB_uP_DAC)	Digital Output (U16.3)
GPIO (PA06-PA08, PB06-PB07)	GPIO (U16.15, U16.16, U16.9, U16.10, U16.17)
User LEDs (BLU_LED0-BLU_LED3, RED_LED0-RED_LED1)	GPIO (U16.23-U16.28)

Table 6

3.6 Onboard Programmer/Debugger

ema:Play provides an onboard Segger JLink In Circuit Programmer/Debugger (ICP/D) USB port (J13) for loading firmware into the onboard MCU (U16) for custom applications. This ICP/D also functions as a debugger for development and evaluation purposes. This allows the user to eliminate purchasing an external ICP/D to use with the onboard MCU (U16). Additionally, a Cortex Single Wire Debug (SWD) Connector (J11) is provided for use with any compatible external ARM Cortex ICP/D. For more information regarding the onboard Segger JLink ICP/D, please visit this website:

https://www.segger.com/products/debug-probes/j-link/models/j-link-ob/

3.7 User Application Headers and Connectors

ema:Play provides a wire to board screw terminal connector (J2) and a board to board shrouded connector (J1) to interface to external devices and allow them direct access to ema, or the onboard MCU (U16). Also provided are a power application header (J4) and an IO application header (J5), for ultimate flexibility when developing custom applications for ema.

The wire to board screw terminal connector (J2) and the board to board shrouded connector (J1) are electrically connected in parallel and can be used to allow external devices with common interfaces (RS232, RS422, RS485) to directly access ema and the onboard MCU (U16). Flow control (CTS/RTS) is also provided at these connectors where applicable. Reference section 3.4 for further details on these interfaces.

Table 7 summarizes the signals relative to J2 and J1

J2/J1 Pin Number	J2/J1 Signal Name	Usage Example
1	EXT_VIN	This signal can be used to supply 4.5V-30V to power the ema:Play/ema platform
2	DGND	Connect this to the external devices' system Ground/return
3	232TX/A	(RS232) Connect this to the external devices' RX signal (RS422/485-2/4) Connect this to the external devices' TXD+ signal
4	232RX/B	(RS232) Connect this to the external devices' TX signal (RS422/485-2/4) Connect this to the external devices' TXD- signal
5	232RTS/Y	(RS232) Connect this to the external devices' CTS signal (RS422/485-4) Connect this to the external devices' RXD+ signal
6	232CTS/Z	(RS232) Connect this to the external devices' RTS signal (RS422/485-4) Connect this to the external devices' RXD- signal
7	EMA_RNG	This signal can be monitored by an external device. It is programmable in ema via an AT command
8	EMA_STS	This signal can be monitored by an external device. It will go high (IO_VREF) when ema's communication interfaces become ready

Table 7

The power application header (J4) can be used to control various power functions on ema:Play as well as ema.

Table 8 summarizes the signals relative to J4.

J4 Pin Number	J4 Signal Name	Usage Example
1	4V0	This signal can be used to power external devices (4V @ 1.25A max)
2	4VMOD_DIS	This signal can be used by an external device to remove power from ema. Drive high (max 5.5V) to disable ema power via load switch (U6).
3	3V3	This signal can be used to power external devices (3.3V @ .75A max)
4	3V3_DIS	This signal can be used by an external device to disable the onboard 3.3V power rail. Drive high (max 5.5V) to disable the 3.3V power rail.
5	DGND	Connect this to the external devices' system Ground/return
6	MODEM_RESET	This signal can be used by an external device to request an ema reset in the event that ema becomes unresponsive. Drive high (max 5.5V) for a minimum of 1 seconds, then release.

7	4VMOD	This signal can be used to power external devices that may require the ability to have power removed by the load switch (U6). (4V @ 1.25A max)
8	MODEM_ON_OFF	This signal can be used to request ema to turn ON or OFF. Drive high (max 5.5V) to request ema to turn ON. Drive low to request ema to turn OFF.
9	MODEM_IO_VREF	This signal can be used by an external source to set ema's IO voltage levels (1.8V-5.5V)
10	Unused	Unused

Table 8

The IO application header (J5) can be used to control, monitor, and interface with various signals and systems throughout the ema:Play hardware, as well as ema.

Table 9 summarizes the signals relative to J5.

J5 Pin Number	J4 Signal Name	Usage Example
1, 10, 27	DGND	Connect this to the external devices' system Ground/return
2, 3, 4, 5, 6	PA08, PB07, PB06, PA07 PA06	These signals connect directly to the onboard MCU's (U16) GPIO's and can be used according to the limitations of U16
7	OB_uP_RESET	This signal can be used by an external device to force a hardware reset of the onboard MCU (U16)
8	MOD_DTR	This signal can be used by an external device to assert the Data Terminal Ready (DTR) signal for ema
9	OB_uP_AIN	This signal can be used by an external device to input an analog voltage (max 3.3V) for reading by the onboard MCU (U16)
11	OB_uP_DAC	This signal can be used by the onboard MCU (U16) to generate analog voltage (max 3.3V) for use by an external device
13, 14	OB_uP_SCL, OB_uP_SDA	These signals can be used to connect external i2c devices for use by the onboard MCU (U16)
15, 16	EXT_EMA_RX, EXT_EMA_TX	These signals can be used by an external device to communicate directly over the emaLink interface. See section 3.4.
17, 18, 19, 20	EXT_RTS, EXT_CTS, EXT_RX, EXT_TX	These signals can be used by an external device to communicate directly over the ema Modem UART interface. See section 3.4.
21, 22	OB_uP_APP_HDR_RX, OB_uP_APP_HDR_TX	These signals can be used by an external device to communicate directly with the onboard MCU (U16)
23, 24	OB_uP_DBG_RX, OB_uP_DBG_TX	These signals can be used by the onboard MCU (U16) to send and receive debug/console messages over UART.
25, 26	CAN/RX, CAN/TX	These signals can be used by the onboard MCU (U16) to interface to a CAN transceiver for communications

28	OB_uP_IOVREF_DIS	This signal can be used by an external device to disable the onboard 3.3V IO_VREF. Drive high (max 5.5V).

Table 9

3.8 Antennas

ema:Play provides two U.FL to SMA antenna adaptor cables. The external antennas should be connected at the SMA ends of the adaptor cables (primary-J3, diveristy-J12). The U.FL ends of these adaptors should be carefully connected to ema after it has been seated correctly in the modem socket (J7/J8).

Table 10 lists recommended antennas for use with ema:Play

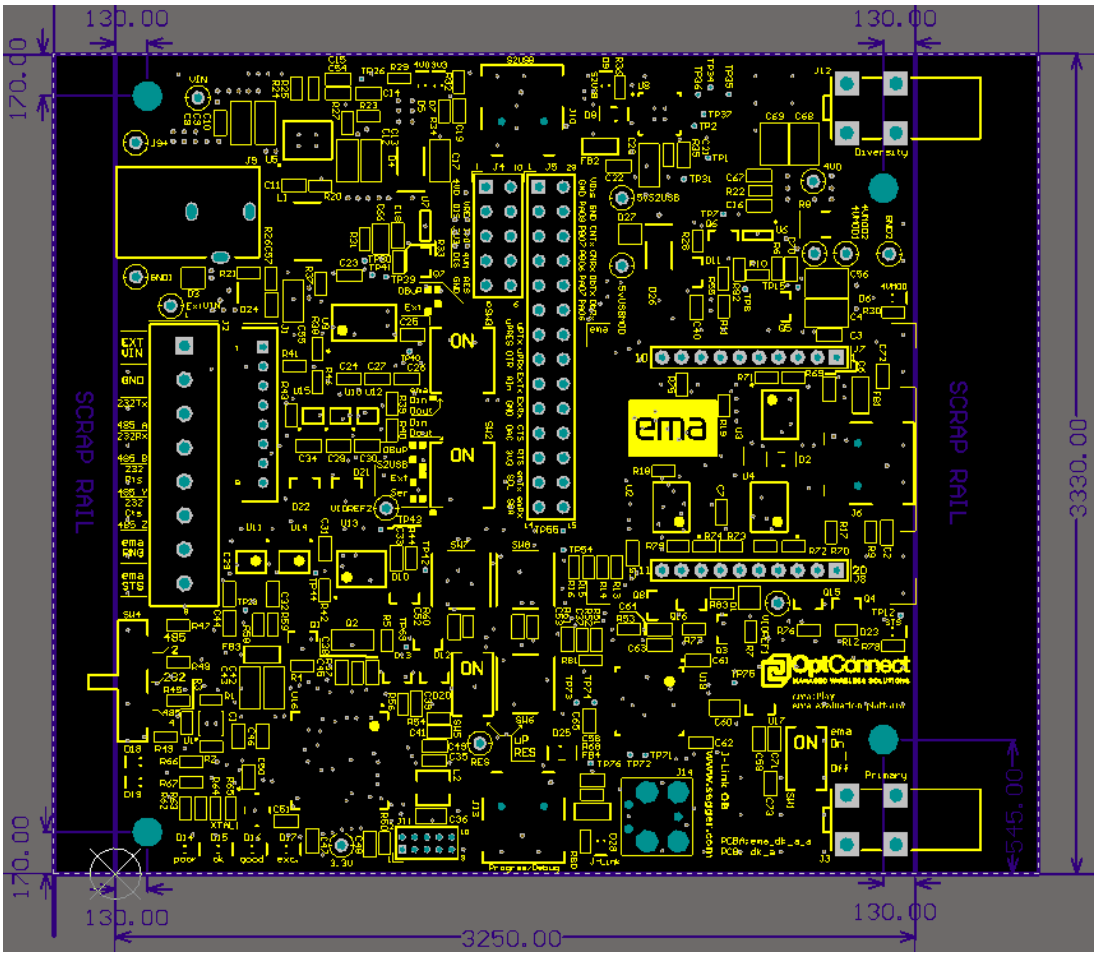
Type	Manufacturer	Part Number
External Cellular Antenna	Taoglas	TG.30.8113
External Cellular Antenna	Taoglas	GSA.8841.A.105111
External Cellular MIMO Antenna *	2J	2J7724Ma

Table 10 * Available from OptConnect

3.9 Mechanical

3.9.1 Dimensions

Width = 3.250 inches, 82.55 mm
Length = 3.330 inches, 84.58 mm
Max height, ema seated, not including standoffs = .5825 inches, 14.8 mm
Max height, ema seated, including standoffs = .8770 inches, 22.3 mm



The PCB assembly for ema:Play includes a 40 mil(.0400 inches) clearance area along both short sides(x) of the PCB assembly along the edge. This allows the PCB assembly to slide into an enclosure that is suitable for the slide mounting method.

4. Software

Section 4 outlines the software architecture and framework that is compatible with ema:Play. The methods and techniques outlined in this section represent different ways to develop software for ema:Play and ema, and are by no means the only methods and techniques that can be used.

There are four main methods for interfacing software with ema using ema:Play.

1. Develop software that runs on ema:Play onboard MCU (U16). This method closely follows the procedure outlined in section 4.1, and involves a technical understanding and knowledge of writing code for embedded systems. Typically, this method will involve a Real Time Operating System (RTOS) runtime environment, such as freeRTOS or similar, and will use the U16 UARTs to access ema, along with any other desired features that ema:Play offers. See section 3.5. Reference section 5.4.1 for ema:Play configuration to support this method.

2. Develop software that runs external to ema:Play on a host platform that has a standard serial COMM interface. This method requires that ema:Play be connected to the host platform using the S2USB USB interface (J10). The most common host platform in this scenario is a standard computer with a terminal program. This method grants access to ema via its Modem UART interface. Reference *OptConnect ema™ Getting Started with ema* and *OptConnect ema™ Application Note 001: HTTP Using Socket Dials* for guidance on this method. Reference section 5.4.2 for ema:Play configuration to support this method.
3. Develop software that runs external to ema:Play on a host platform that can access ema using USB drivers. USB drivers are available for download at **http://optconnect.com/ema** . This method involves writing software at a higher application level, with easy access to the hardware's USB drivers. The interface to ema via ema:Play would occur at the ema USB port (J6). Reference *OptConnect ema™ Application Note 002: Windows Networking Guide* for guidance on this method. Reference section 5.4.3 for ema:Play configuration to support this method.
4. Develop software that runs external to ema:Play on a different embedded development kit or custom piece of embedded hardware or machine. ema can be accessed via ema:Play at the user Application Headers (J4, J5). Additionally, built in features of ema:Play can also be accessed using this method. See section 3.5 and 3.7. Reference section 5.4.4 and 5.4.5 for ema:Play configuration to support this method.

4.1 Native Development Environment

4.1.1 Overview

ema:Play has an onboard MCU (U16) that is provided for custom user applications utilizing ema as the cellular connection. This section demonstrates how to use *Atmel Studio 7* to get up and running, developing code quickly on ema:Play. The hardware for ema:Play is setup for flexibility relative to the development environment, and *Atmel Studio 7* was chosen due to its simplicity and availability. Additionally, more in-depth documentation about *Atmel Studio 7* is available than this User Guide provides. To reduce development time, and spend more time developing the application versus the driver and register initialization code, Microchip offers the *Atmel | Start* tool. The samples used in ema:Play have been developed using this tool. For more information regarding *Atmel | Start*, visit https://start.atmel.com/.

Any development environment that supports the onboard MCU (U16, Microchip ATSAME51J20A) and a Segger JLink In Circuit Programmer/Debugger (ICP/D), or the ARM Cortex SWD Debug Connector (J11) interface can be used. Reference section 3.6 for more details on the ICP/D.

4.1.2 Setup and Building

1. **Download and install *Atmel Studio 7*:** Go to the following website for the download link:

   https://www.microchip.com/mplab/avr-support/atmel-studio-7

   Make sure to select the following options during the installation. See *Figure 4*.

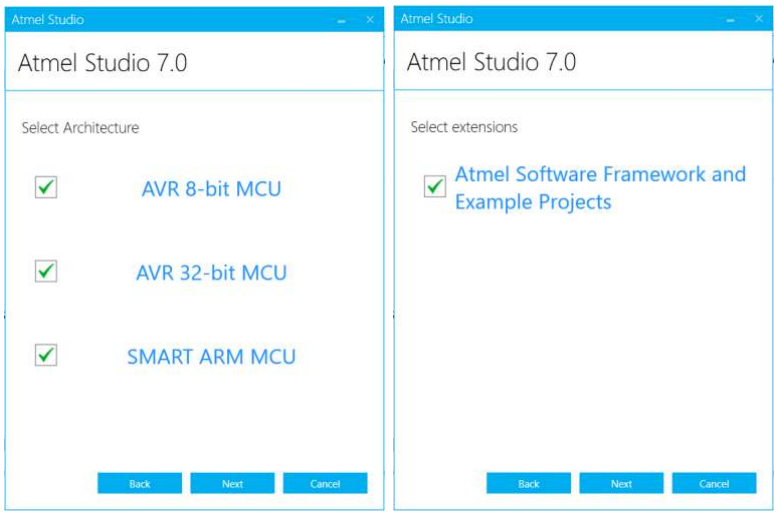


Figure 4

The installation will take several minutes to complete.

2. **Obtain the latest *GettingStartedWithemaVx.x.zip* firmware package:** Go to **http://optconnect.com/emaPlay** , or reach out to an OptConnect representative to obtain the file. Reference section 1.2 for more information.
3. **Launch *Atmel Studio 7*:** Once the software is launched the landing page should look similar to *Figure 5*.

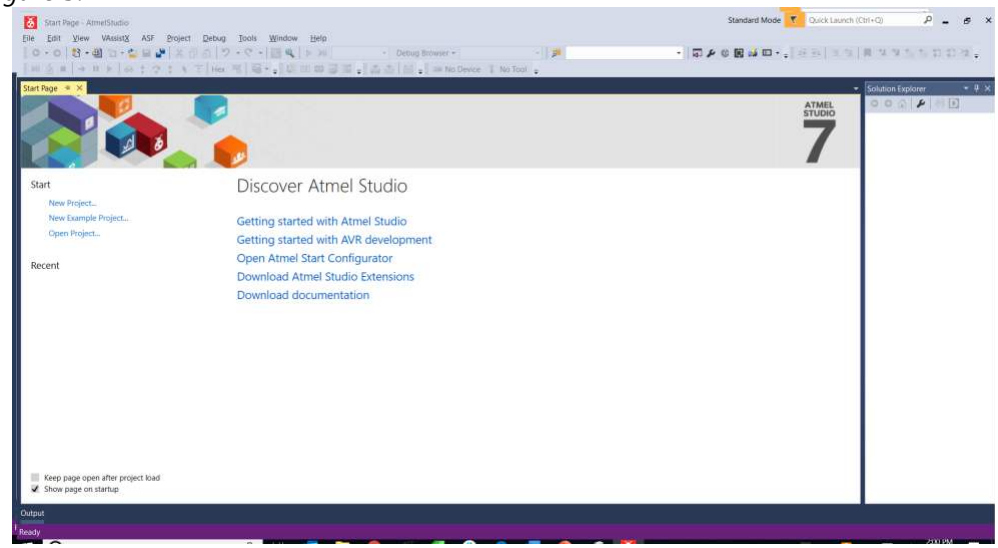


Figure 5.

4. **Open the project:** Open the *GettingStartedWithemaVx.x* sample project by clicking on the "Open Project" link or selecting "File", then "Project/Solution" and navigating to the location of the *GettingStartedWithema.atsln* file that was obtained in step 2. Once the project has been opened, the screen should look similar to *Figure 6*.

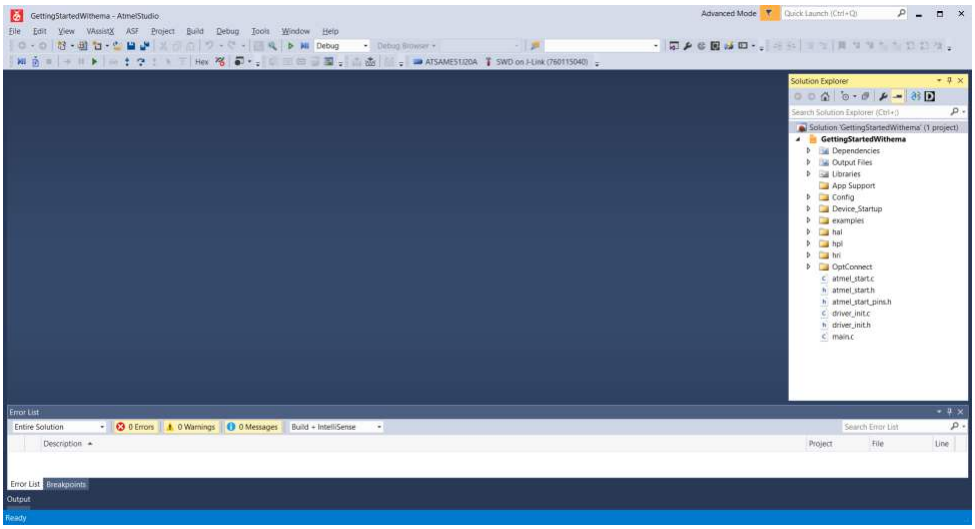


Figure 6

5. **Investigate the *Atmel* | *Start* Setup:** On the right side of the screen, navigate to the "Solution Explorer", and click on the "Reconfigure Atmel Start Project" button as shown in *Figure 7*.

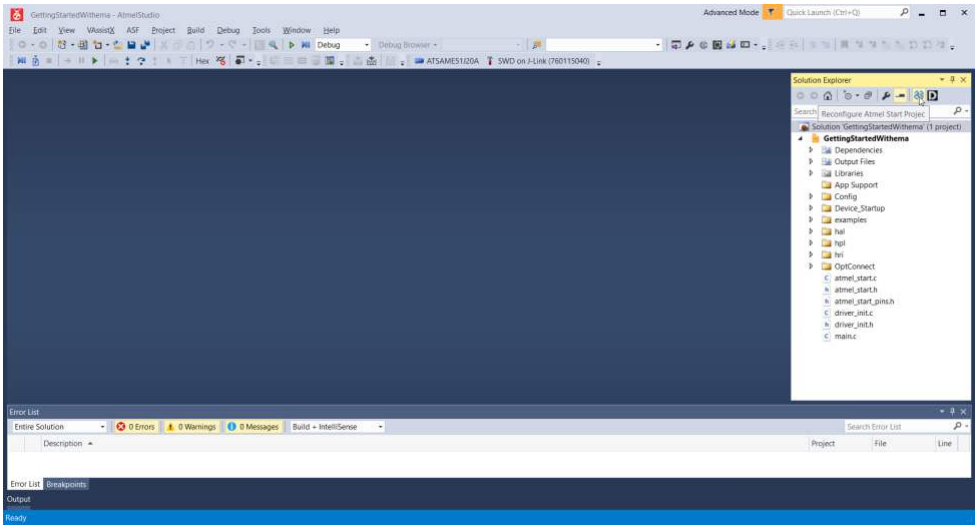


Figure 7

This will load the *Atmel* | *Start* tool setup relative to this project. This tool can be used to quickly configure the drivers and middleware for the onboard MCU (U16). For this project, the following peripherals have already been configured. Reference *Figure 8*.

ADC_0	Analog to digital converter driver used to convert voltages.
EMAPLAY_DEBUG_UART	UART communications between U16 and the S2USB enhanced COMM port (J10), used for console debug I/O.
I2C_0	i2c communications between U16 and the onboard temperature/humidity sensor (U1).
EMA_MGMT_UART	UART communications between U16 and emaLink.
EMA_MODEM_UART	UART communications between U16 and the ema Modem UART.

TIMER_0	Timer driver used to create a periodic time-based interrupt for U16.

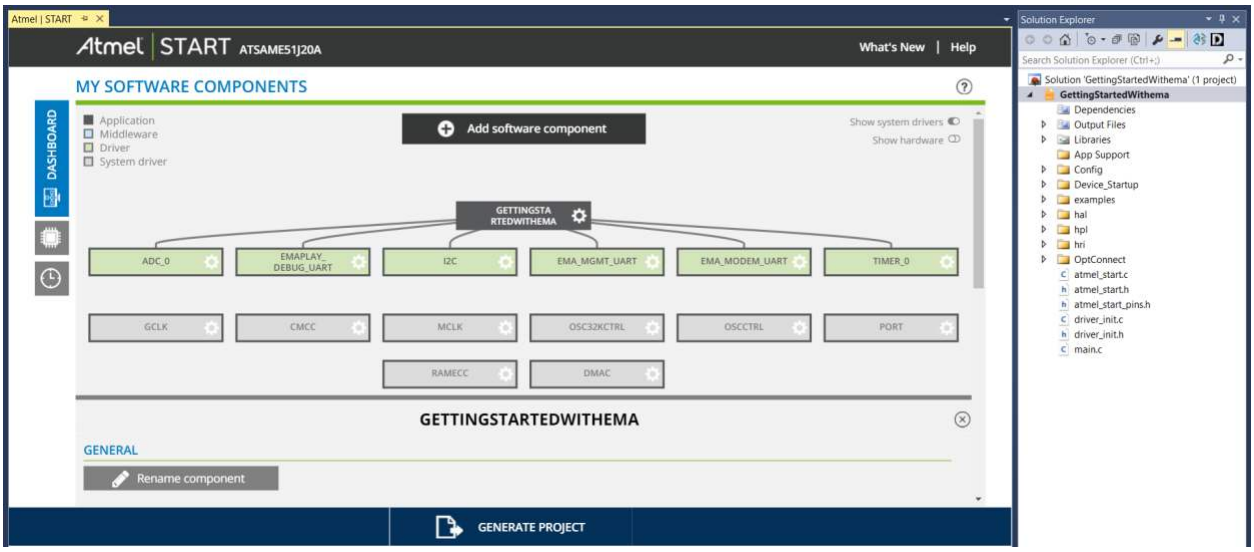


Figure 8

The "Add software component" button can be used to add various drivers and middleware as needed.

6. **Investigate the PINMUX Configurator in Atmel | Start:** On the left side of the screen, click on the button labeled "Go to PINMUX Configurator" as shown in *Figure 9*.

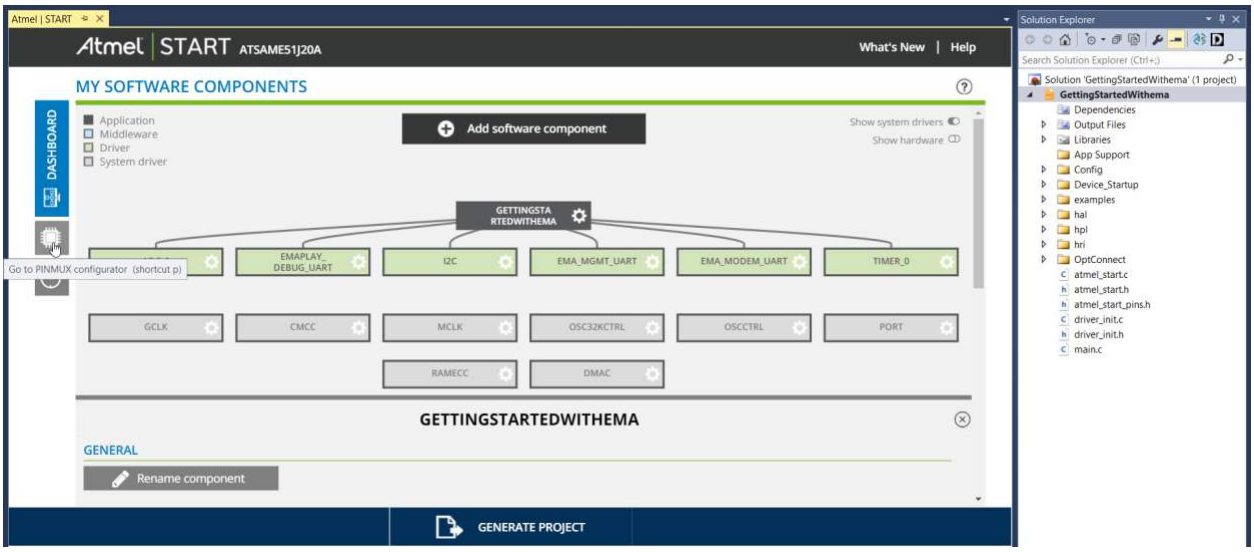


Figure 9

This will pull up the PINMUX Configurator where the ema:Play user can change the pin settings on U16. Note that the column labeled "User Label" is the reference that can be seen in the API calls from the source code. See *Figure 10*. *Atmel | Start* API documentation can be found here: http://ww1.microchip.com/downloads/en/DeviceDoc/50002633A.pdf .

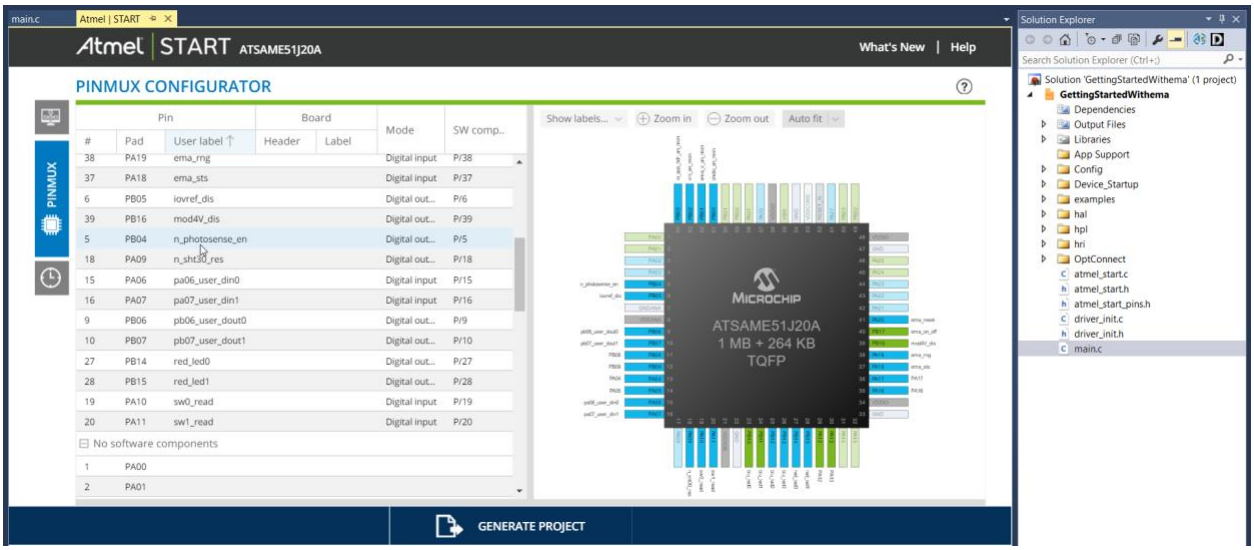


Figure 10

7. **Investigate the main.c source file:** In the "Solution Explorer" under the folder "GettingStartedWithema", locate the main.c file and open it. It can be observed that the "User Label" references are used in the API calls in the int main(void) function. This is a simple demonstration of the correlation between the *Atmel | Start* tool and the source code references. Reference *Figure 11*.

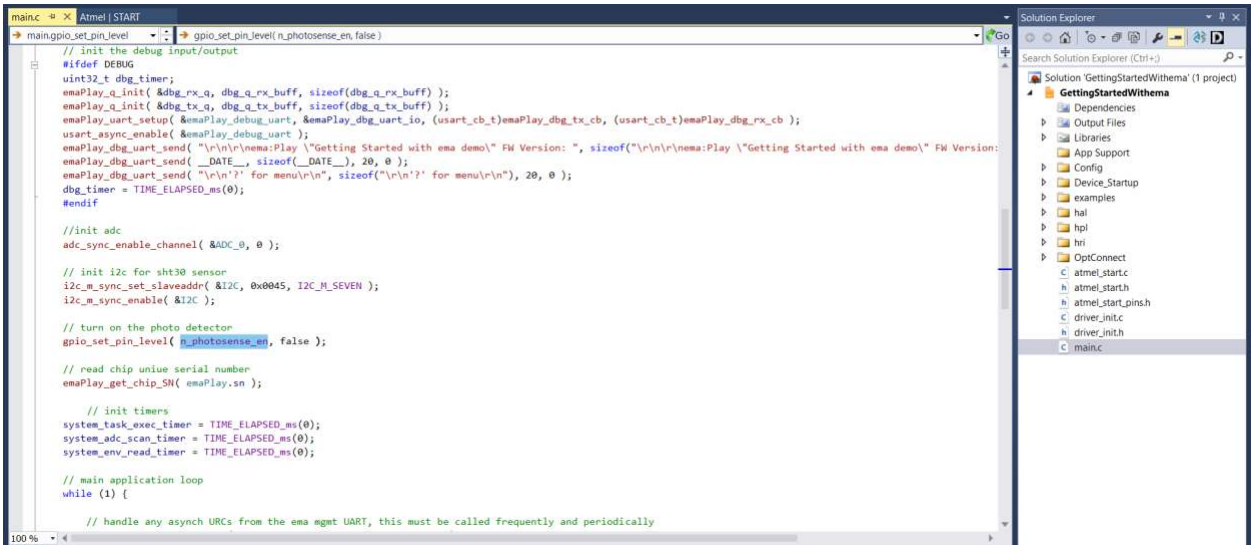


Figure 11

8. **Build the project:** Click "Build", then "Build Solution", or simply press F7 to compile the source code and build the project. The "Output" window can be observed for any errors. There should not be any errors if using the provided project. Reference *Figure 12*.

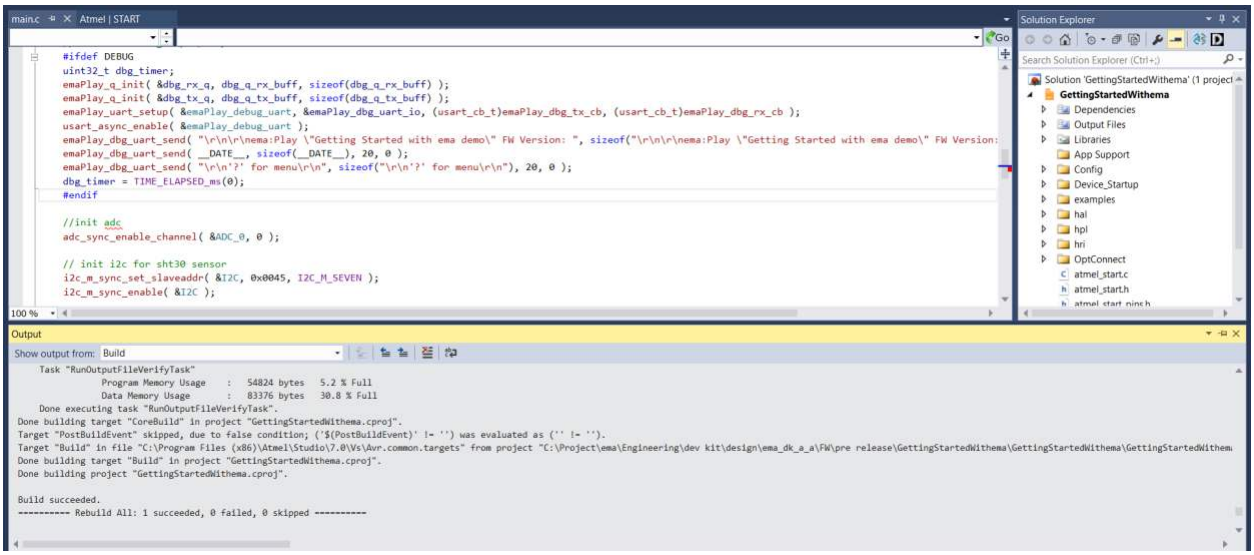


Figure 12

4.1.3 Sample Program/Debug Session

This section continues from section 4.1.2 and demonstrates how to program the firmware into the ema:Play onboard MCU (U16). *Atmel Studio 7* can also be used to step through code in a debug capacity.

1. **Connect ema:Play JLink ICP/D to computer:** Locate the onboard JLink Programmer/Debugger port (J13) and connect it to the computer running *Atmel Studio 7*. Right click on the "GettingStartedWithema" project folder and select "Properties". The project properties window appears. On the left side, click on the "Tool" tab, then select the J-Link programmer/debugger from the list as shown in *Figure 13*.

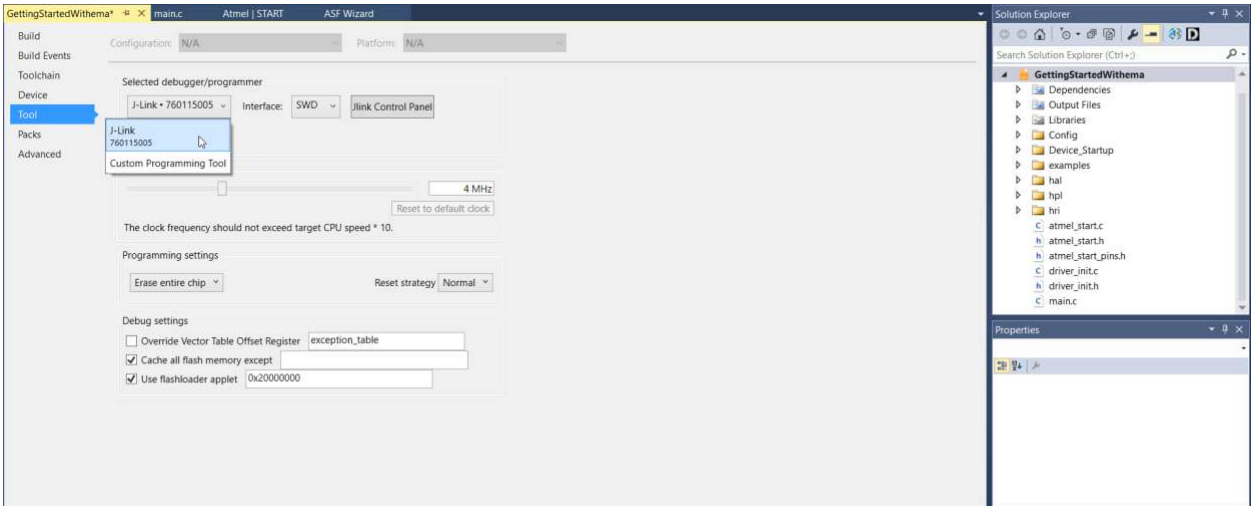


Figure 13

Note that if an external programmer/debugger were to be used, it would show up in this list as well.

2. **Program the firmware:** Click the "Tools" dropdown menu and select "Device Programming". Confirm that the "Tool" is the J-Link, and the "Device" is the ATSAME51J20A as shown in *Figure 16*. Click "Apply". Click the Blue JLink Control Utility Icon to launch the J-Link Control Panel. Confirm that the "Device" is set to ATSAME51J19* as shown in *Figure 14*. If it's not set to this, click on the "Settings" tab, and select the correct device next to the check box "Override device selection" as shown in *Figure 15*.

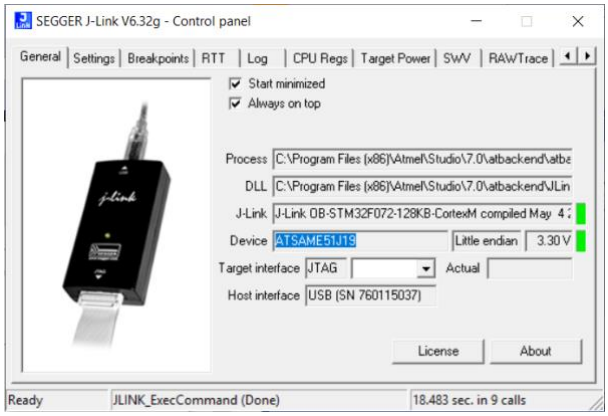


Figure 14

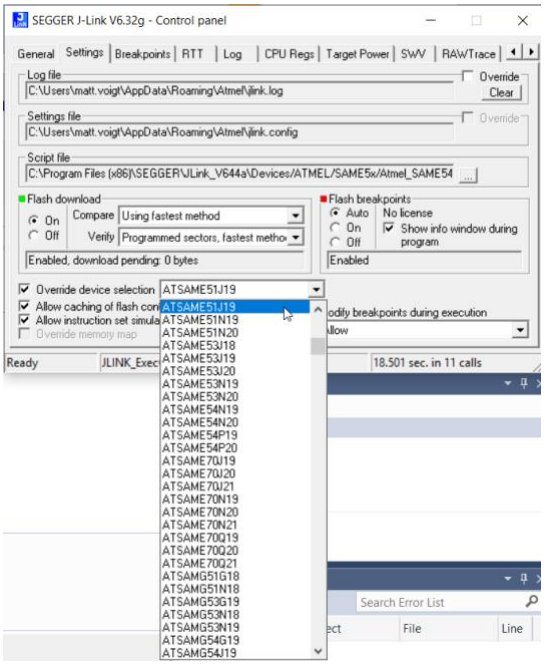


Figure 15

* There is an identified issue with *Atmel Studio 7* that does not allow the ATSAME51J20A to be used by the JLink Utility

Close the JLink Control Utility and Click "Read". The "Device Signature" and "Target Voltage" will be read from the onboard MCU (U16) as shown in *Figure 16*.

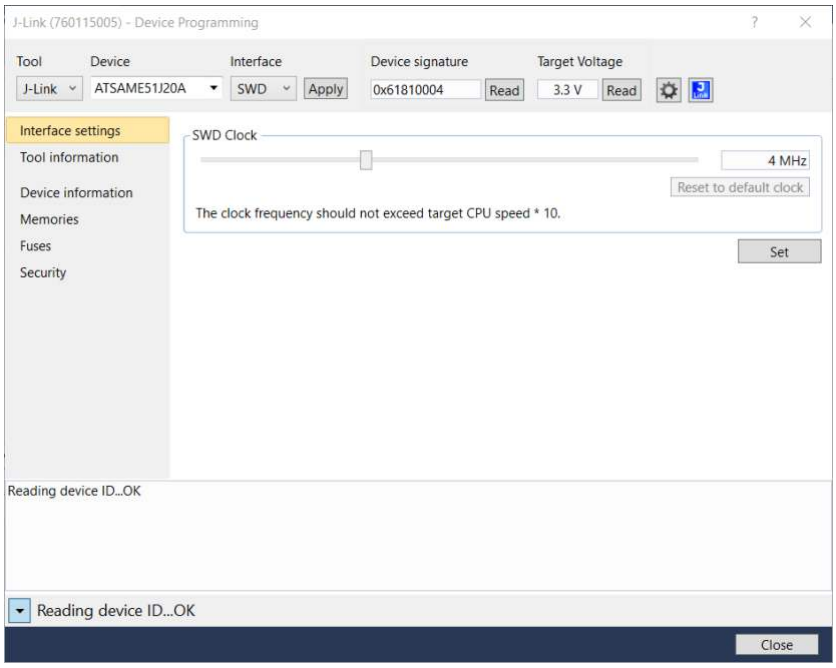


Figure 16

On the left side, click the tab labeled "Memories". Next, click the "Program" button. *Atmel Studio 7* will use the onboard JLink programmer to program the firmware into the ema:Play onboard MCU (U16) program flash. Once complete, ema:Play Blue LEDs signal array (D14-D17) should continue to scroll. Reference *Figure 17*.

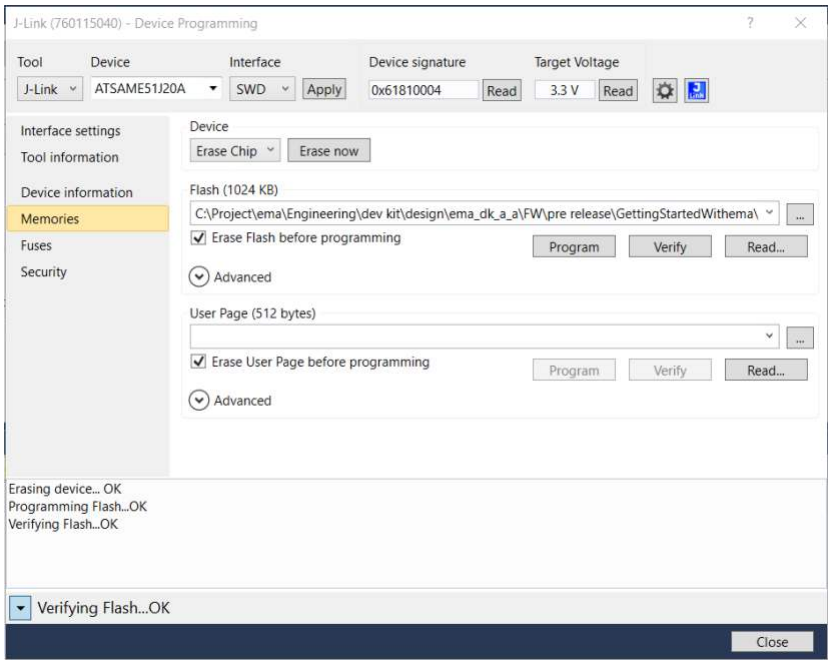
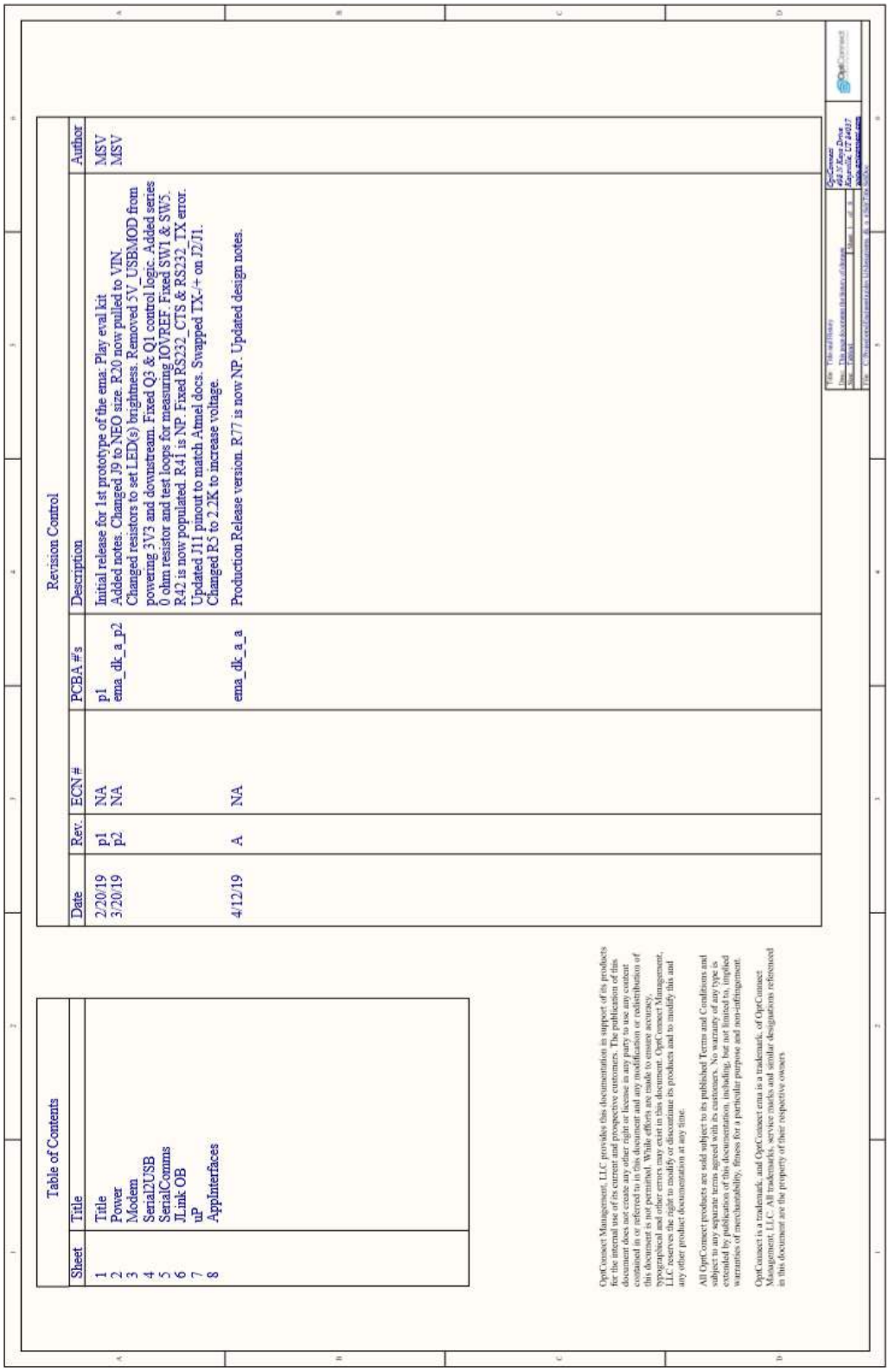


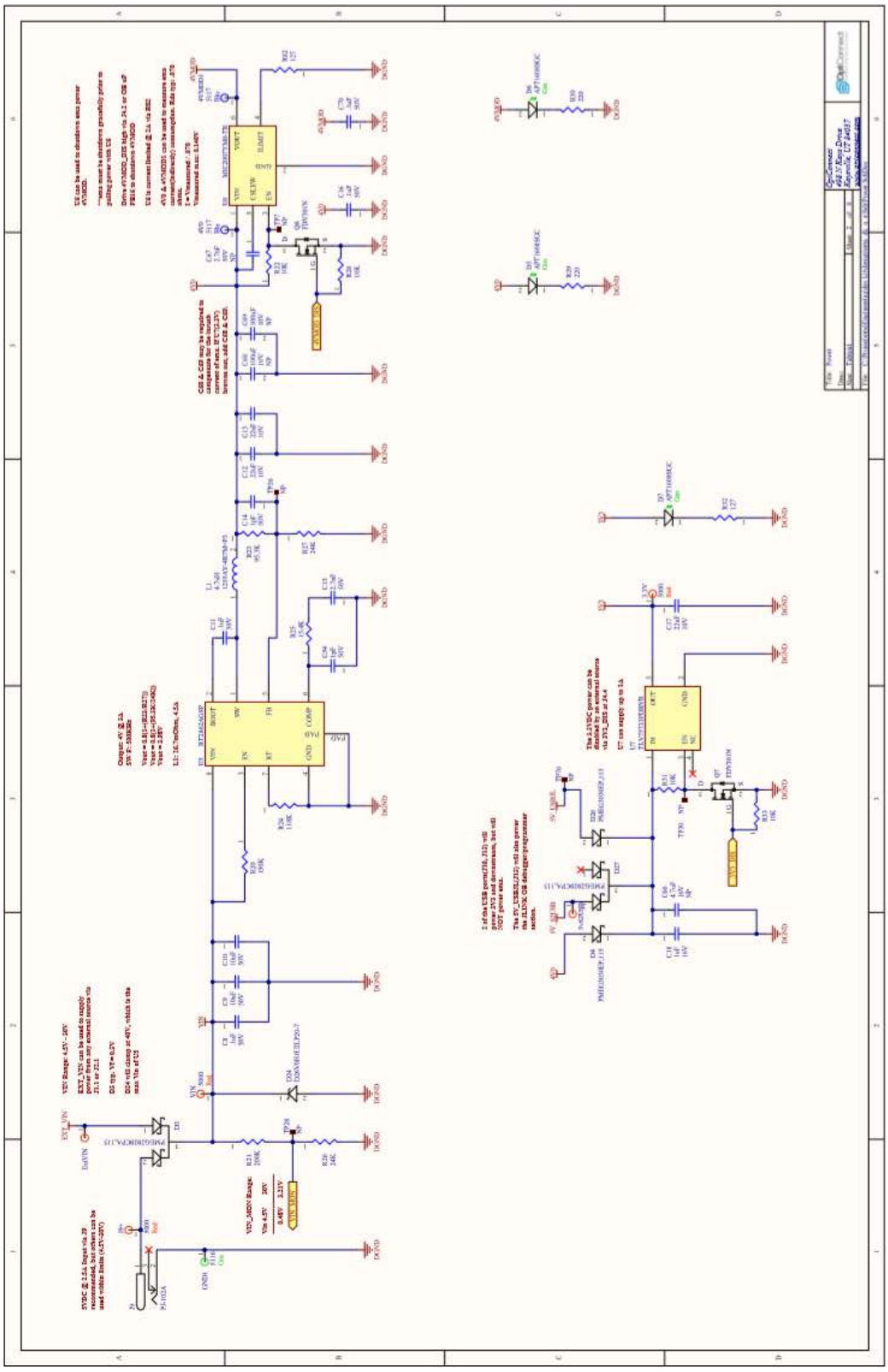
Figure 17

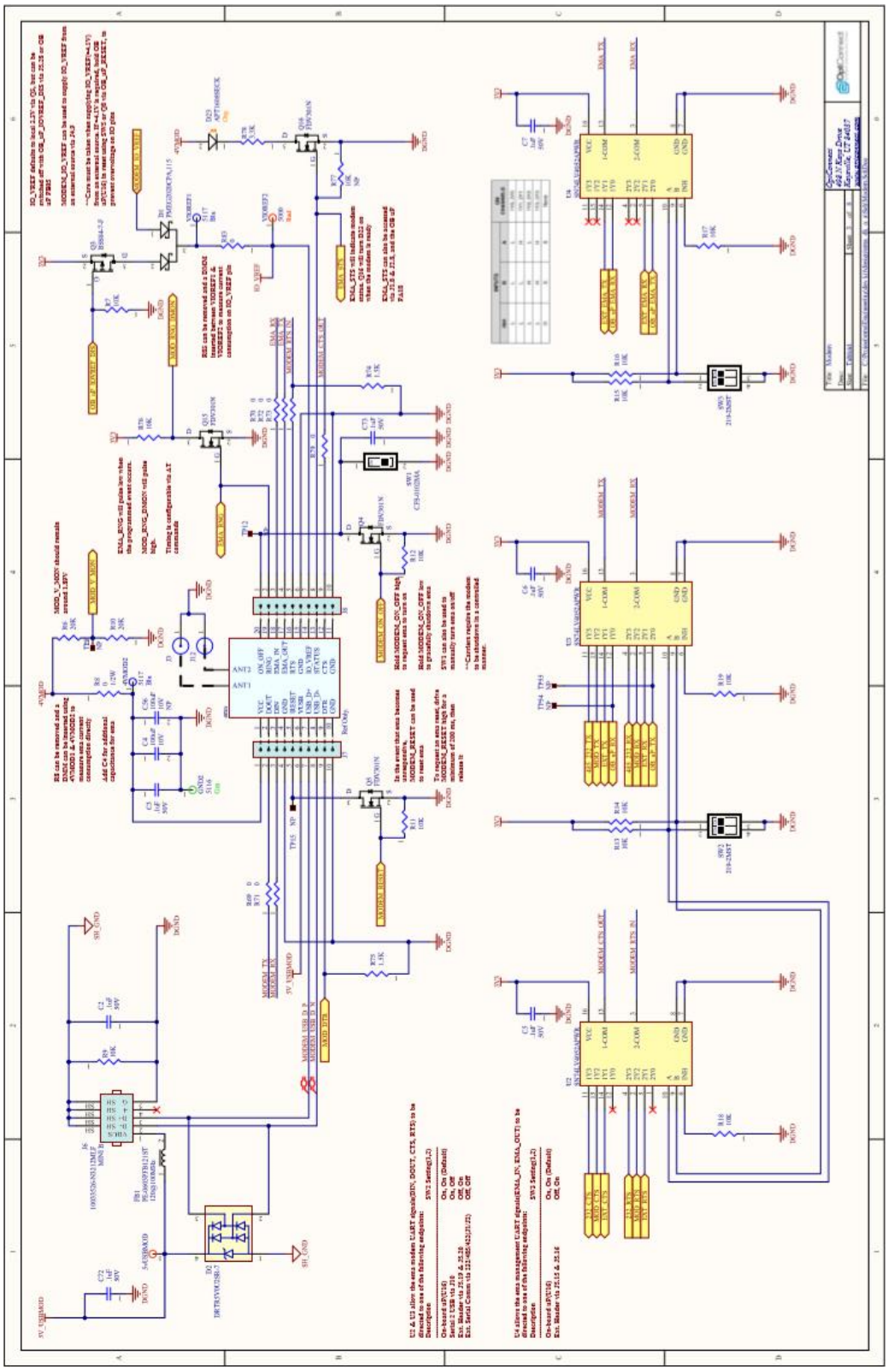
5. Reference

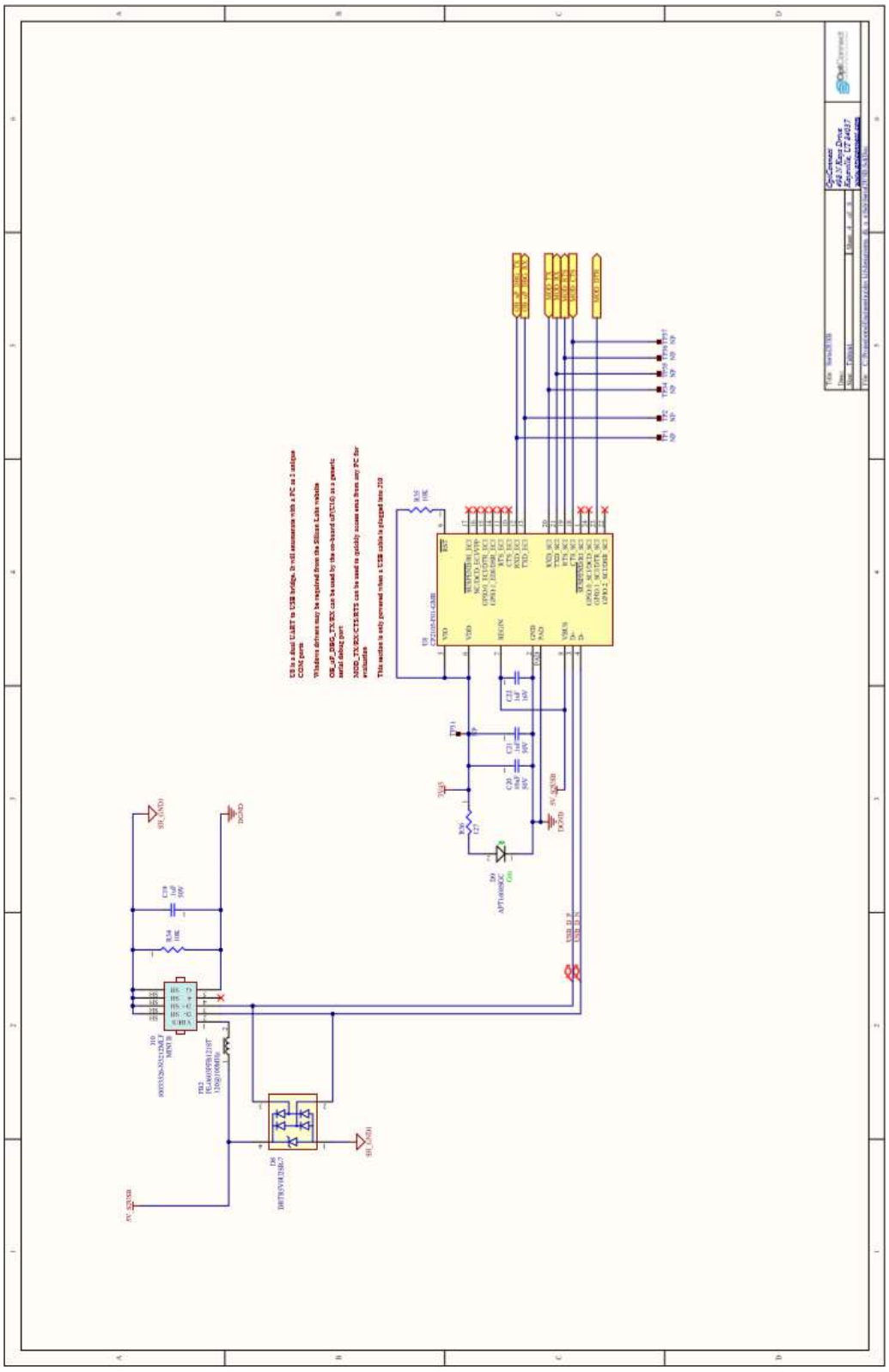
OptConnect provides several reference documents relative to ema:Play and ema that designers and developers should review carefully and use as needed to help with custom hardware and software. Schematics, PCB Layout, and sample code projects can be requested. Refer to section 1.2, or visit **http://optconnect.com/ema** for the latest versions of these documents and projects. The following sections include copies of ema:Play design files for quick reference and further design considerations.

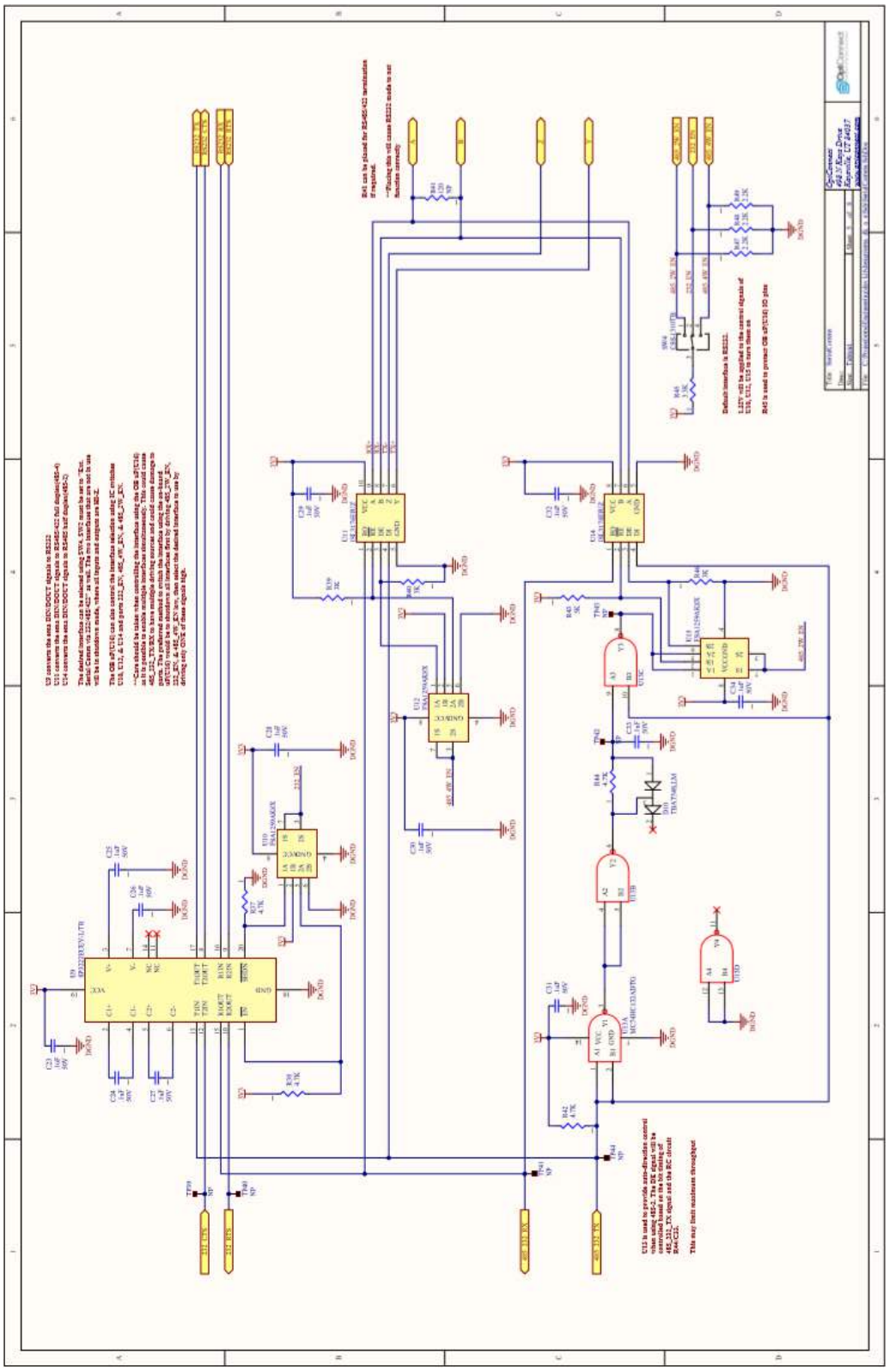
5.1 Schematics

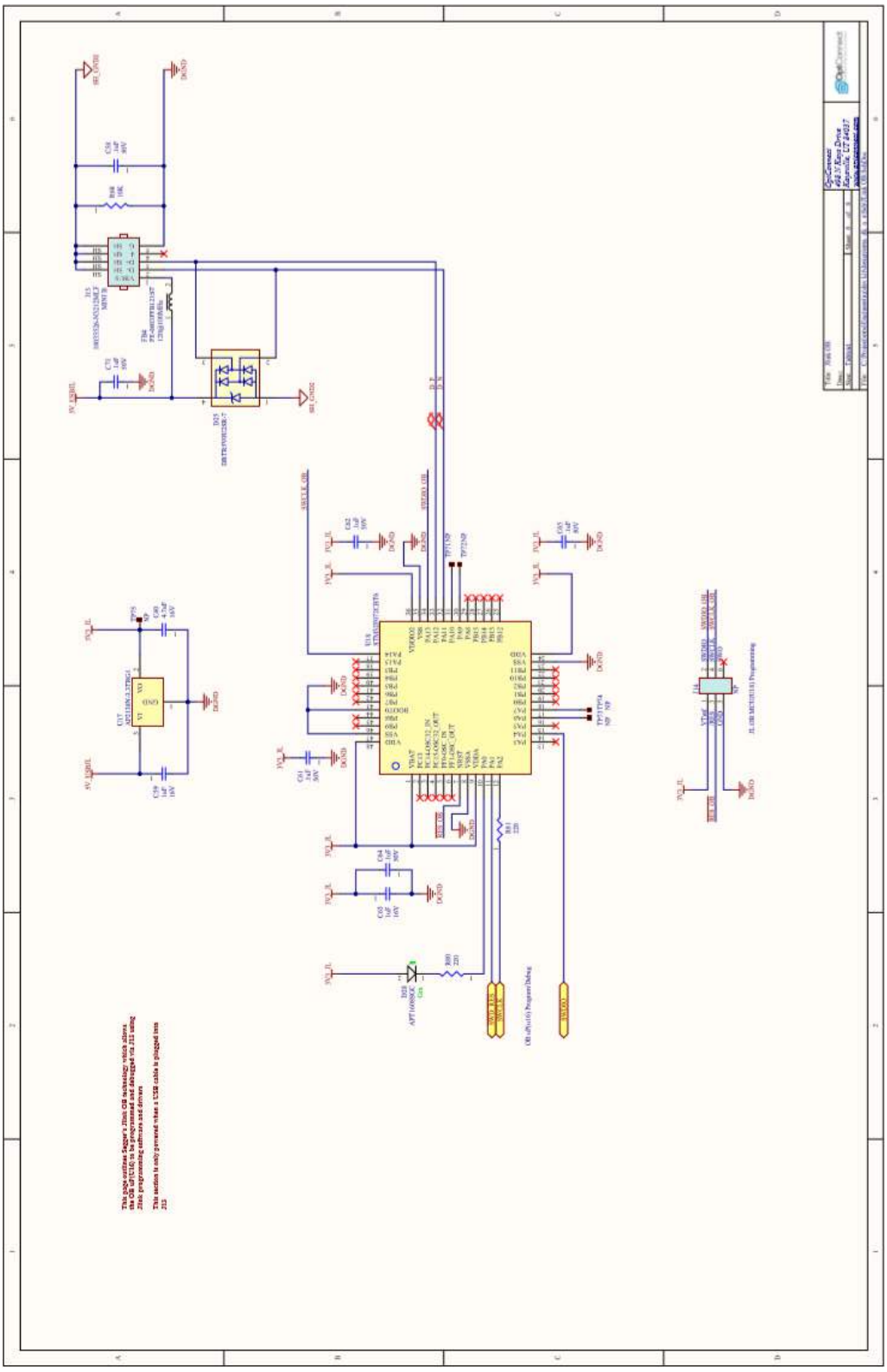


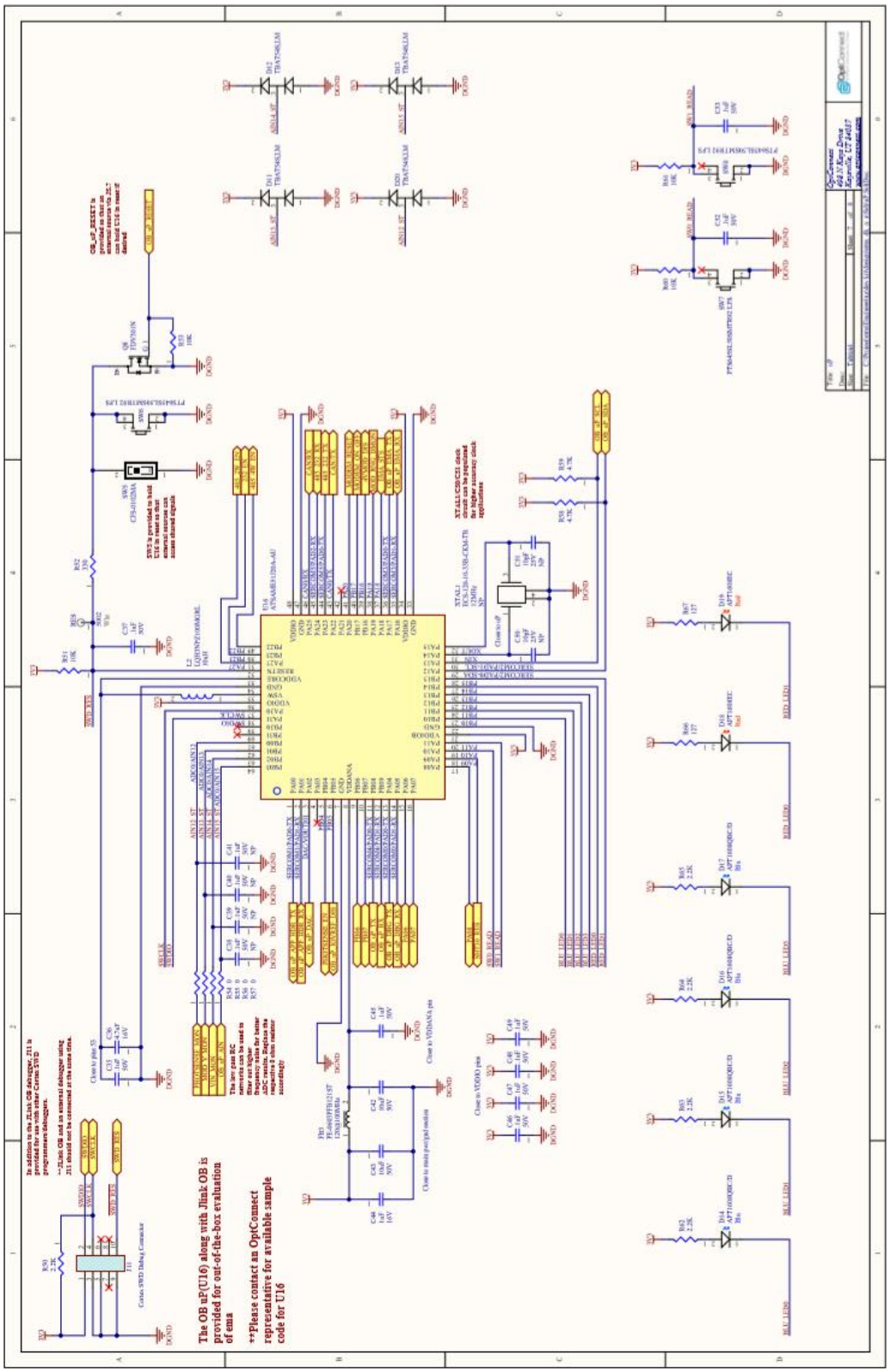


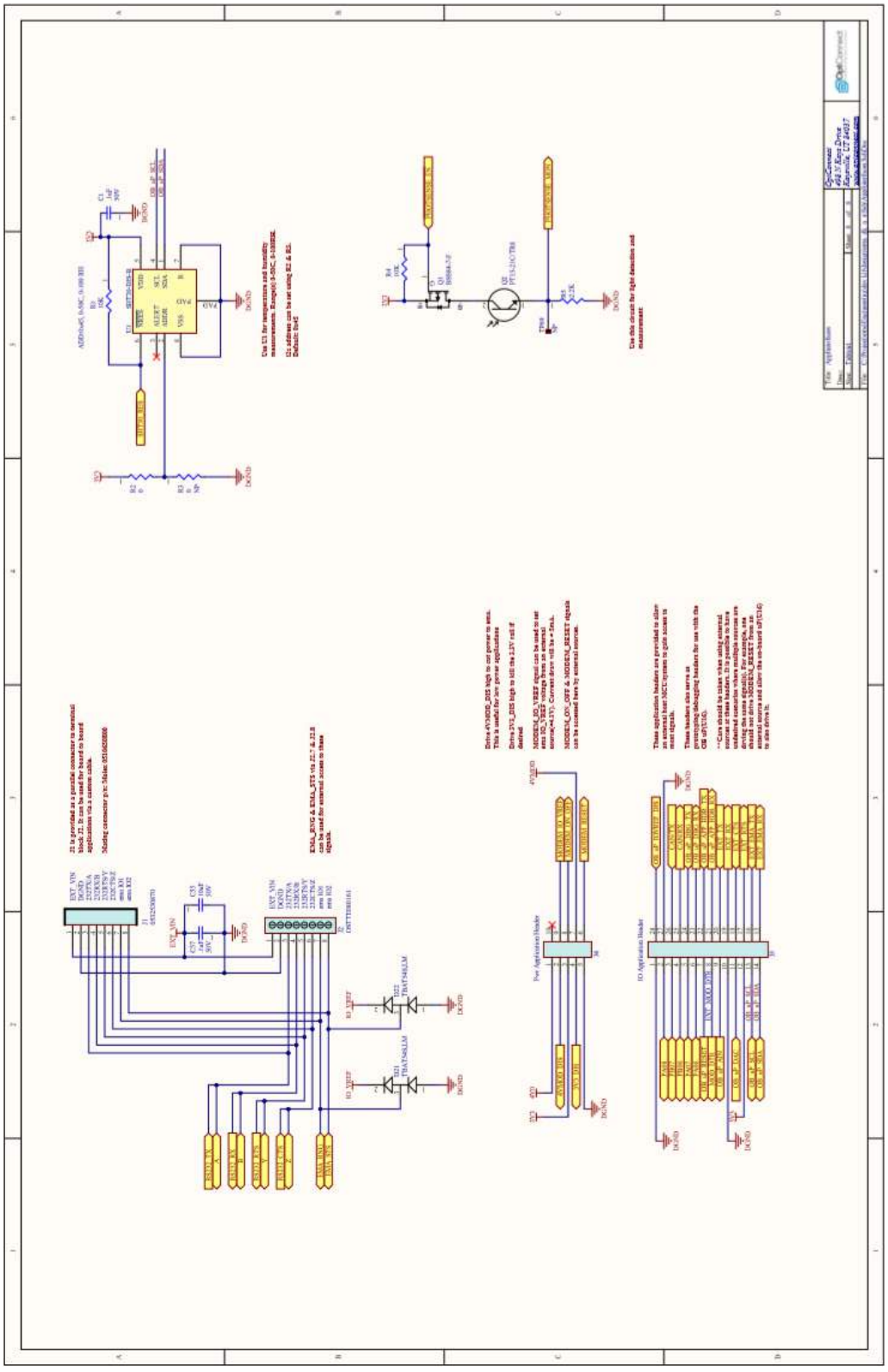












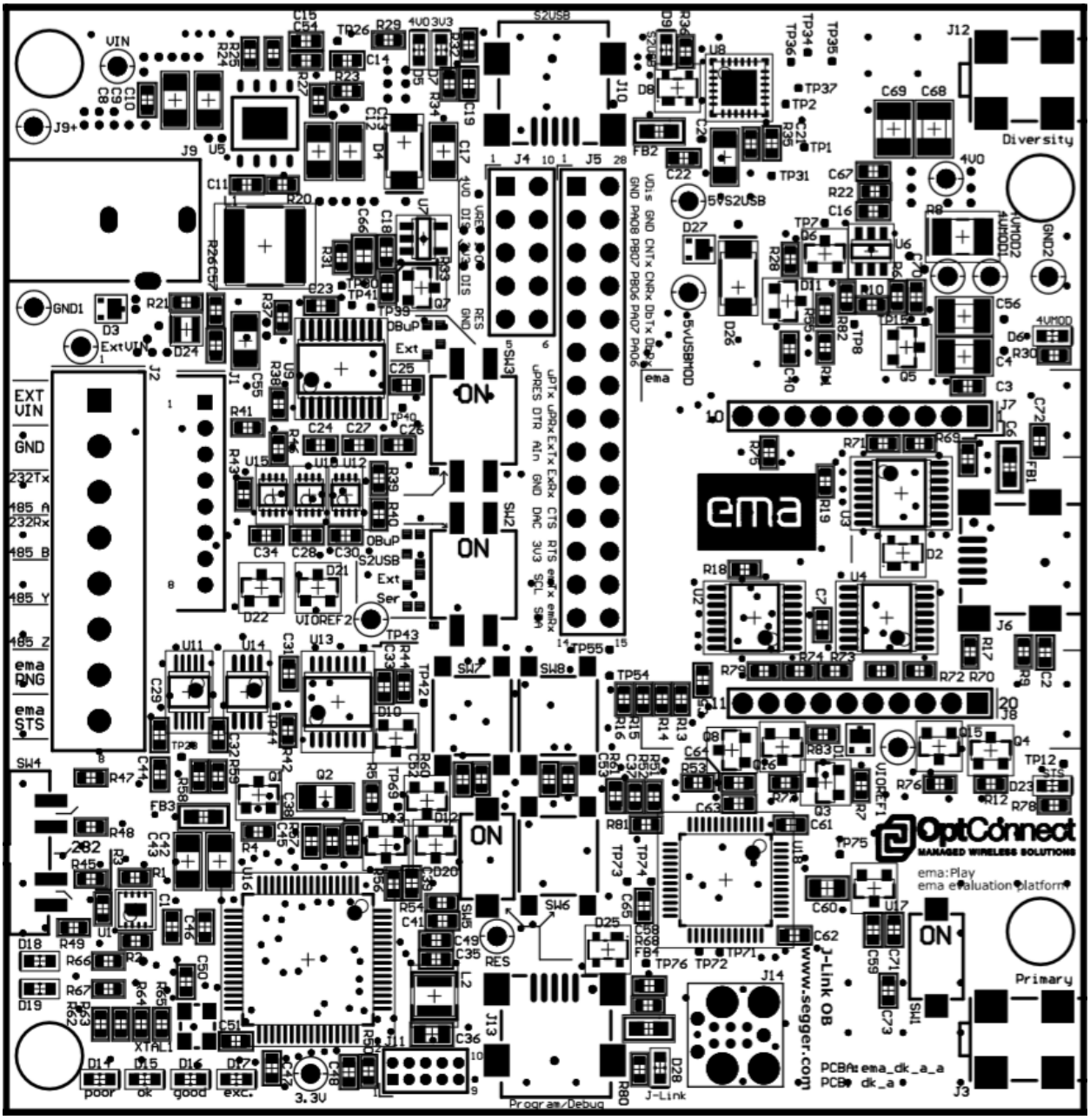
5.2 Bill of Materials (BOM)

1	ema_dk_a_a PCBA	BOM	rev 1	4/19/2019									
2													
3	Designator	Value	Population	Description	Footprint	Mfgr 1	Mfgr 1 p/n	Supplier 1	Supplier Part Number 1	DigiReel p/n	Quantity	Supplier Unit Price 1	Supplier Subtotal 1
4	3.3V, 5vS2USB, 5vUSBMOD, ExtVIN, J9+, VIN, VIOREF2			PC TEST POINT MINIATURE RED	THRU, TEST LOOP, RED	Keystone Electronics	5000	Digi-key	36-5000-ND		7	0.35	2.45
5	4V0, 4VMOD1, 4VMOD2, VIOREF1			PC TEST POINT MINIATURE BLUE	THRU, TEST LOOP, BLU	Keystone Electronics	5117	Digi-key	36-5117-ND		4	0.35	1.4
6	C1, C2, C3, C5, C6, C7, C8, C11, C16, C19, C21, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C37, C45, C46, C47, C48, C49, C52, C53, C57, C58, C61, C62, C64, C65, C70, C71, C72, C73	.1uF		CAP CER 0.1UF 50V X7R 0603	CAP0603IPCMED	Yageo	CC0603KRX7R9BB104	Digi-key	311-1344-1-ND	311-1344-6-ND	42	0.084	3.528
7	C4	100uF		CAP CER 100UF 10V X5R 1210	CAP1210IPCLARGE	Taiyo Yuden	LMK325BJ107MM-T	Digi-key	587-1965-1-ND	587-1965-6-ND	1	1.36	1.36
8	C9, C10, C20, C42, C43, C55	10uF		CAP CER 10UF 50V X5R 1206	CAP1210IPCMED	Samsung Electro-Mechanic	CL31A106KBHNNNE	Digi-key	1276-2876-1-ND	1276-2876-6-ND	6	0.65	3.9
9	C12, C13, C17	22uF		CAP CER 22UF 10V X7R 1206	CAP1210IPCMED	Samsung Electro-Mechanic	CL31B226KPHNFNE	Digi-key	1276-3145-1-ND	1276-3145-6-ND	3	0.63	1.89
10	C14, C54	1pF		CAP CER 1PF 50V C0G/NP0 0603	CAP0603IPCMED	Samsung Electro-Mechanic	CL10C010CB8NNNC	Digi-key	1276-1293-1-ND	1276-1293-6-ND	2	0.1	0.2
11	C15	2.7nF		CAP CER 2700PF 50V X7R 0603	CAP0603IPCMED	Samsung Electro-Mechanic	CL10B272KB8NNNC	Digi-key	1276-2022-1-ND	1276-2022-6-ND	1	0.1	0.1
12	C18, C22, C44, C59, C63	1uF		CAP CER 1UF 16V X7R 0603	CAP0603IPCMED	Samsung Electro-Mechanic	CL10B105MO8NNWC	Digi-key	1276-6524-1-ND	1276-6524-6-ND	5	0.1	0.5
13	C36, C60	4.7uF		CAP CER 4.7UF 16V X7R 0805	CAP0805IPCMED	Samsung Electro-Mechanic	CL21B475KOFNNNE	Digi-key	1276-2873-1-ND	1276-2873-6-ND	2	0.34	0.68
14	D1, D3, D27			DIODE ARRAY SCHOTTKY 20V 3HUSON	3-PowerUDFN	Nexperia USA Inc	PMEG2020CPA,115	Digi-key	1727-5194-1-ND	1727-5194-6-ND	3	0.56	1.68
15	D2, D8, D25			TVS DIODE 5.5V SOT143	SOT-143, TO-253-4, TO-253AA	Diodes Inc	DRTR5V0U2SR-7	Digi-key	DRTR5V0U2SR-7DICT-ND	DRTR5V0U2SR-7DIDKR-ND	3	0.39	1.17
16	D4, D26			DIODE SCHOTTKY 30V 3A SOD128	SOD-128	Nexperia USA Inc.	PMEG3030EP,115	Digi-key	1727-5324-1-ND	1727-5324-6-ND	2	0.49	0.98
17	D5, D6, D7, D9, D28			LED GREEN CLEAR CHIP SMD	0603LED	Kingbright	APT1608SGC	Digi-key	754-1121-1-ND	754-1121-6-ND	5	0.39	1.95
18	D10, D11, D12, D13, D20, D21, D22			DIODE SCHOTTKY 30V 140MA SOT23	SOT-23-3 ,TO-236-3, SC-59	Toshiba Semi	TBAT54S,LM	Digi-key	TBAT54SLMCT-ND	TBAT54SLMDKR-ND	7	0.22	1.54
19	D14, D15, D16, D17			LED BLUE CLEAR CHIP SMD	0603LED	Kingbright	APT1608QBC/D	Digi-key	754-1434-1-ND	754-1434-6-ND	4	0.49	1.96
20	D18, D19	Red		LED RED CLEAR CHIP SMD	0603LED	Kingbright	APT1608EC	Digi-key	754-1117-1-ND	754-1117-6-ND	2	0.37	0.74
21	D23			LED ORANGE CLEAR CHIP SMD	0603LED	Kingbright	APT1608SECK	Digi-key	754-1120-1-ND	754-1120-6-ND	1	0.43	0.43
22	D24			TVS DIODE 26V 40V U-DFN2020-2	2-UDFN, U-DFN2020-2	Diodes Inc	D26V0H1U2LP20-7	Digi-key	D26V0H1U2LP20-7DICT-ND	D26V0H1U2LP20-7DIDKR-ND	1	0.43	0.43
23	FB1, FB2, FB3, FB4			FERRITE BEAD 120 OHM 0603 1LN	0603X055M	Pulse Electronics	PE-0603PFB121ST	Digi-key	553-2387-1-ND	553-2387-6-ND	4	0.1	0.4
24	GND1, GND2			PC TEST POINT MINIATURE GREEN	THRU, TEST LOOP, GRN	Keystone Electronics	5116	Digi-key	36-5116-ND		2	0.35	0.7
25	J1			CONN HEADER VERT 8POS 2MM	THRU, 1X8, 2MM	Molex	0532530870	Digi-key	WM9207-ND		1	0.37	0.37
26	J2			TERM BLK 8POS SIDE ENT 3.5MM PCB	THRU, 1X8, TERM, 3.5MM	On Shore Technology	OSTTE080161	Digi-key	ED2641-ND		1	1.73	1.73
27	J3, J12			CONN, RF, SMA TO U.FL, ADAPTOR, PCB MOUNT	THRU, RFSMARA-7053	Roho Communication	RFSMARA-7053-DOWN	Roho Communication	RFSMARA-7053-DOWN		2		
28	J4			CONN HEADER VERT 10POS 2.54MM	THRU, 2X5, 2.54MM	Amphenol ICC	77313-118-10LF	Digi-key	609-4437-ND		1	0.54	0.54
29	J5			CONN HEADER VERT 28POS 2.54MM	THRU, 2X14, 2.54MM	Sullins Connector	PREC014DAAN-RC	Digi-key	S2012EC-14-ND		1	0.53	0.53
30	J6, J10, J13			CONN RCPT MINI USB B 5POS SMD RA	SMD, USB	Amphenol ICC	10033526-N3212MLF	Digi-key	609-4701-1-ND	609-4701-6-ND	3		
31	J7, J8			CONN RECEPT 2MM SINGLE STR 10POS	THRU, 1X10, 2MM	Sullins Connector Solutions	NPPN101BFCN-RC	Digi-key	S5751-10-ND		2	0.98	1.96
32	J9			CONN PWR JACK 2.5X5.5MM SOLDER	POWER JACK THRU HOLE	CUI Inc	PJ-102A	Digi-key	CP-102A-ND		1	0.64	0.64
33	J11			CONN HEADER VERT 10POS 1.27MM	THRU, 2x5, 1.27MM	Wurth Electronics Inc.	62201021121	Digi-key	732-5374-ND		1	1.4	1.4
34	L1	4.7uH		FIXED IND 4.7UH 4.5A 27.6 MOHM	SMD, INDUCTOR	Murata	1255AY-4R7M=P3	Digi-key	490-10812-1-ND	490-10812-6-ND	1	0.48	0.48
35	L2	10uH		FIXED IND 10UH 1.12A 336 MOHM	SMD-1212, INDUCTOR	Murata	LQH3NPZ100MGRL	Digi-key	490-15956-1-ND	490-15956-6-ND	1	0.44	0.44
36	Q1, Q3			MOSFET P-CH 50V 130MA SOT23-3	SOT-23-3 ,TO-236-3, SC-59	Diodes Inc	BSS84-7-F	Digi-key	BSS84-FDICT-ND	BSS84-FDIDKR-ND	2	0.27	0.54
37	Q2			PHOTOTRANSISTOR FLAT CLEAR 1206	CAP1210IPCMED	Everlight Electronics Co	PT15-21C/TR8	Digi-key	1080-1380-1-ND	1080-1380-6-ND	1		
38	Q4, Q5, Q6, Q7, Q8, Q15, Q16			MOSFET N-CH 25V 220MA SOT-23	SOT-23-3 ,TO-236-3, SC-59	ON Semi	FDV301N	Digi-key	FDV301NCT-ND	FDV301NDKR-ND	7	0.36	2.52
39	R1, R4, R7, R9, R11, R12, R13, R14, R15, R16, R17, R18, R19, R22, R28, R31, R33, R34, R35, R51, R53, R60, R61, R68, R76	10K		RES SMD 10K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0710KL	Digi-key	311-10.0KHRCT-ND	311-10.0KHRDKR-ND	25	0.024	0.6
40	R2, R54, R55, R56, R57, R69, R70, R71, R72, R73, R79, R83	0		RES SMD 0 OHM JUMPER 1/10W 0603	RES0603IPCMED	Yageo	RC0603JR-070RL	Digi-key	311-0.0GRCT-ND	311-0.0GRDKR-ND	12	0.021	0.252
41	R65	2.2K		RES SMD 2.2K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-072K2L	Digi-key	311-2.20KHRCT-ND	311-2.20KHRDKR-ND	9	0.1	0.9
42	R6, R10	20K		RES SMD 20K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0720KL	Digi-key	311-20.0KHRCT-ND	311-20.0KHRDKR-ND	2	0.1	0.2
43	R8	0		RES 0 OHM JUMPER 1/2W 1210	1210X07M	Stackpole Electronics	RMCF1210ZT0R00	Digi-key	RMCF1210ZT0R00CT-ND	RMCF1210ZT0R00DKR-ND	1	0.12	0.12
44	R20	150K		RES SMD 150K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07150KL	Digi-key	311-150KHRCT-ND	311-150KHRDKR-ND	1	0.1	0.1
45	R21	200K		RES SMD 200K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07200KL	Digi-key	311-200KHRCT-ND	311-200KHRDKR-ND	1	0.1	0.1
46	R23	95.3K		RES SMD 95.3K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0795K3L	Digi-key	311-95.3KHRCT-ND	311-95.3KHRDKR-ND	1	0.1	0.1
47	R24	118K		RES SMD 118K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07118KL	Digi-key	311-118KHRCT-ND	311-118KHRDKR-ND	1	0.1	0.1
48	R25	15.4K		RES SMD 15.4K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0715K4L	Digi-key	311-15.4KHRCT-ND	311-15.4KHRDKR-ND	1	0.1	0.1
49	R26, R27	24K		RES SMD 24K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0724KL	Digi-key	311-24.0KHRCT-ND	311-24.0KHRDKR-ND	2	0.1	0.2
50	R29, R30, R80, R81	220		RES SMD 220 OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07220RL	Digi-key	311-220HRCT-ND	311-220HRDKR-ND	4	0.1	0.4
51	R32, R36, R66, R67, R82	127		RES SMD 127 OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07127RL	Digi-key	311-127HRCT-ND	311-127HRDKR-ND	5	0.1	0.5
52	R37, R38, R42, R44, R58, R59	4.7K		RES SMD 4.7K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-074K7L	Digi-key	311-4.70KHRCT-ND	311-4.70KHRDKR-ND	6	0.1	0.6
53	R39, R40, R43, R46	3K		RES SMD 3K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-073KL	Digi-key	311-3.00KHRCT-ND	311-3.00KHRDKR-ND	4	0.1	0.4
54	R45, R78	3.3K		RES SMD 3.3K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-073K3P	Digi-key	YAG1300CT-ND	YAG1300DKR-ND	2	0.1	0.2
55	R52	330		RES SMD 330 OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07330R	Digi-key	311-330HRCT-ND	311-330HRDKR-ND	1	0.1	0.1
56	R74, R75	1.5K		RES SMD 1.5K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-071K5L	Digi-key	311-1.50KHRCT-ND	311-1.50KHRDKR-ND	2	0.1	0.2
57	RES			PC TEST POINT MINIATURE WHITE	THRU, TEST LOOP, WHT	Keystone Electronics	5002	Digi-key	36-5002-ND		1	0.35	0.35
58	SW1, SW5			SWITCH DIP SPST 100MA 6V	SMD, J-LEAD, 1 POS DIP SEITCH	Nidec Copal	CFS-0102MA	Digi-key	563-1976-5-ND		2	1.27	2.54
59	SW2, SW3			SWITCH SLIDE DIP SPST 100MA 20V 2 POS	SMD, GULL WING, 2 POS DIP SWITCH	CTS Electrocomponents	219-2MST	Digi-key	CT2192MST-ND		2	0.6	1.2

60	SW4	SP3T		SWITCH SLIDE SP3T 100MA 12V	SMD, GULL WING, SW, SP3T	Nidec Copal	CSS-1310TB	Digi-key	563-1095-1-ND	563-1095-6-ND	1	0.63	0.63
61	SW6, SW7, SW8			SWITCH TACTILE SPST-NO 0.05A 12V	SMD, TACT SW, SPST	C&K	PTS645SL50SMTR92 LFS	Digi-key	CKN9088CT-ND	CKN9088DKR-ND	3	0.16	0.48
62	U1			SENSOR HUMID/TEMP 5V I2C 3% SMD	8-DFN, 8-VFDFN EXPOSED PAD	Sensirion AG	SHT30-DIS-B	Digi-key	1649-1009-1-ND	1649-1009-6-ND	1	3.93	3.93
63	U2, U3, U4			IC MULTIPLEXER DUAL 4X1 16TSSOP	16-TSSOP	Texas Instruments	SN74LV4052APWR	Digi-key	296-3831-1-ND	296-3831-6-ND	3	0.48	1.44
64	U5			IC REG BUCK ADJ 3A SYNC 8SOP	8-SOIC (0.154", 3.90mm Width) Exposed Pa	Richtek USA Inc	RT2862AGSP	Digi-key	1028-1372-1-ND	1028-1372-6-ND	1	2.17	2.17
65	U6			IC SW DISTRIBUTION ADJ SOT23-6	SOT-23-6, IPCMED	Microchip	MIC2007YM6-TR	Digi-key	576-1460-1-ND	576-1460-6-ND	1	0.48	0.48
66	U7			IC REG LINEAR 3.3V 1A SOT23-5	SOT-23-5, SC-74A, SOT-753	Texas Instruments	TLV75733PDBVR	Digi-key	296-50414-1-ND	296-50414-6-ND	1	0.88	0.88
67	U8			IC SGL USB-DL UART BRIDGE 24QFN	24-QFN, 24-WFQFN Exposed Pad	Silicon Labs	CP2105-F01-GMR	Digi-key	CP2105-F01-GMRCT-ND	CP2105-F01-GMRDKR-ND	1	1.56	1.56
68	U9			IC TXRX RS232 ESD TRUE 20TSSOP	20-TSSOP (0.173", 4.40mm Width)	MaxLinear Inc	SP3222EUEY-L/TR	Digi-key	1016-1795-1-ND	1016-1795-6-ND	1	1.92	1.92
69	U10, U12, U15			IC SWITCH DUAL SPST US8	8-VFSOP(2.30MM)	ON Semi	FSA1259AK8X	Digi-key	FSA1259AK8XCT-ND	FSA1259AK8XDKR-ND	3	0.7	2.1
70	U11			IC TXRX RS422/485 20MBPS 10MSOP	10-MSOP, 10-TFSOP	Renesas	ISL3176EIUZ	Digi-key	ISL3176EIUZ-ND	ISL3176EIUZ-TDKR-ND	1		
71	U13			IC GATE NAND SCHMITT 4CH 14TSSOP	14-TSSOP	ON Semi	MC74HC132ADTG	Digi-key	MC74HC132ADTGOS-ND	MC74HC132ADTR2GOSDKR-ND	1	0.38	0.38
72	U14			IC TXRX RS422/485 20MBPS 8MSOP	8-MSOP, 8-TSSOP	Renesas	ISL3178EIUZ	Digi-key	ISL3178EIUZ-ND		1		
73	U16			IC MCU 32BIT 1MB FLASH 64TQFP	64-TQFP(10X10MM)	Microchip	ATSAME51J20A-AU	Digi-key	ATSAME51J20A-AU-ND	ATSAME51J20A-AUT-DKR-ND	1	5.4	5.4
74	U17			IC REG LINEAR 3.3V 250MA SOT23-3	SOT-23-3 ,TO-236-3, SC-59	Diodes Inc	AP2138N-3.3TRG1	Digi-key	AP2138N-3.3TRG1DICT-ND	AP2138N-3.3TRG1DIDKR-ND	1	0.35	0.35
75	U18			IC MCU 32BIT 128KB FLASH 48LQFP	48-LQFPIPCMED	STMicroelectronics	STM32F072CBT6	Digi-key	497-14645-ND		1	3.56	3.56
76													
77	**Other**												
78	PCB			PCB, ema: Play eval kit, ema_dk_a_a		OptConnect	dk_a	Advanced Circuits			1	13.62	13.62
79	STANDOFFS			ROUND STANDOFF #4-40 ALUM 3/8"		Keystone Electronics	2026	Digi-key	36-2026-ND		4	0.22	0.88
80	SCREWS			MACHINE SCREW PAN PHILLIPS 4-40		Keystone Electronics	9900	Digi-key	36-9900-ND		4	0.04	0.16
81	FIRMWARE1			FW, U18, JLINK OB		Segger	J-Link-OB-STM32F072-128KB	Segger	J-Link-OB-STM32F072-128KB		1	0.1	0.1
82	FIRMWARE2			FW, U16, ema: Play App Code		OptConnect	ema_dk_fw_1	OptConnect	ema_dk_fw_1		1		

88	**Not Populated**												
89	C38, C39, C40, C41	.1uF	NP	CAP CER 0.1UF 50V X7R 0603	CAP0603IPCMED	Yageo	CC0603KRX7R9BB104	Digi-key	311-1344-1-ND	311-1344-6-ND	4	0.12	0.48
90	C50, C51	10pF	NP	CAP CER 10PF 25V X7R 0603	CAP0603IPCMED	AVX Corp	06033C100KAT2A	Digi-key	478-11366-1-ND	478-11366-6-ND	2	0.27	0.54
91	C56, C68, C69	100uF	NP	CAP CER 100UF 10V X5R 1210	CAP1210IPCLARGE	Taiyo Yuden	LMK325BJ107MM-T	Digi-key	587-1965-1-ND	587-1965-6-ND	3	1.36	4.08
92	C66	4.7uF	NP	CAP CER 4.7UF 16V X7R 0805	CAP0805IPCMED	Samsung Electro-Mechanic	CL21B475KOFNNNE	Digi-key	1276-2873-1-ND	1276-2873-6-ND	1	0.34	0.34
93	C67	2.7nF	NP	CAP CER 2700PF 50V X7R 0603	CAP0603IPCMED	Samsung Electro-Mechanic	CL10B272KB8NNNC	Digi-key	1276-2022-1-ND	1276-2022-6-ND	1	0.1	0.1
94	ema		NP	EMA	EMA OUTLINE	OptConnect	EMA-1XXX-L4-A				1		
95	J14		NP	CABLE ADAPTER 6 POS	TAG CONNECT, PROG	Tag-Connect LLC	TC2030-IDC	Digi-key	TC2030-IDC-ND		1	43.71	43.71
96	R3	0	NP	RES SMD 0 OHM JUMPER 1/10W 0603	RES0603IPCMED	Yageo	RC0603JR-070RL	Digi-key	311-0.0GRCT-ND	311-0.0GRDKR-ND	1	0.1	0.1
97	R41	120	NP	RES SMD 120 OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-07120RL	Digi-key	311-120HRCT-ND	311-120HRDKR-ND	1	0.1	0.1
98	R77	10K	NP	RES SMD 10K OHM 1% 1/10W 0603	RES0603IPCMED	Yageo	RC0603FR-0710KL	Digi-key	311-10.0KHRCT-ND	311-10.0KHRDKR-ND	1	0.1	0.1
99	TP1, TP2, TP7, TP8, TP12, TP15, TP26, TP28, TP30, TP31, TP34, TP35, TP36, TP37, TP39, TP40, TP41, TP42, TP43, TP44, TP54, TP55, TP69, TP71, TP72, TP73, TP74, TP75, TP76		NP	TEST POINT	TEST POINT, .4MM			Digi-key			29		
100	XTAL1	12MHz	NP	CRYSTAL 12MHZ 10PF SMD	4-SMD, NO LEAD	ECS Inc.	ECS-120-10-33B-CKM-TR	Digi-key	XC2426CT-ND	XC2426DKR-ND	1	0.59	0.59

5.3 Assembly Reference



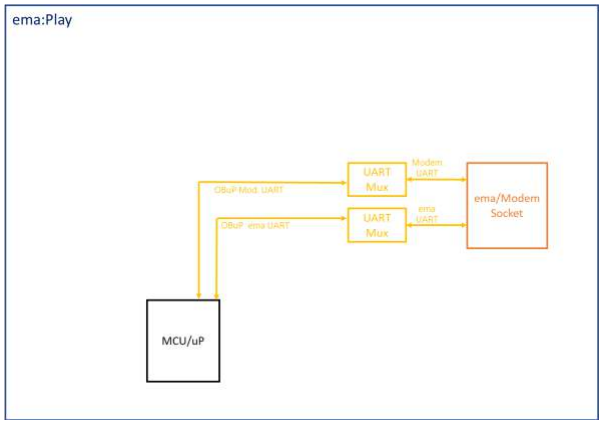
5.4 Sample ema:Play configurations

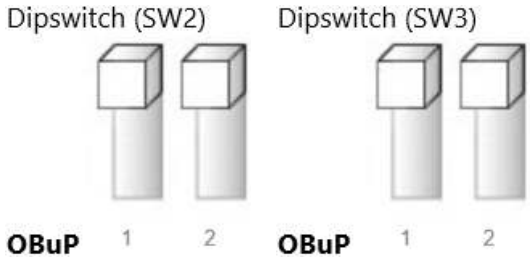
Sections 5.4.1 – 5.4.5 demonstrate how to configure the ema:Play dipswitches (SW2-SW4) for the most common use cases. A snippet of the used ema:Play hardware components is provided as well for reference.

Note: If the on-board MCU (U16) is not required for use, it should be held in reset using SW5.

5.4.1 On-board MCU control (default)

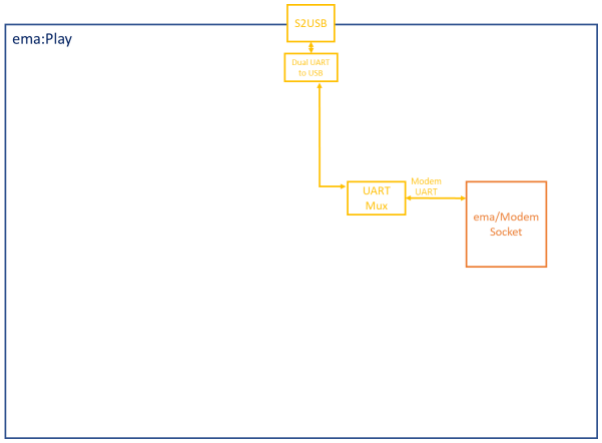
In this configuration, the on-board MCU (U16) is running an application and using ema as the cellular internet connection. Both the Modem UART and emaLink interfaces are connected directly to the MCU as shown below. AT commands are used to control ema.

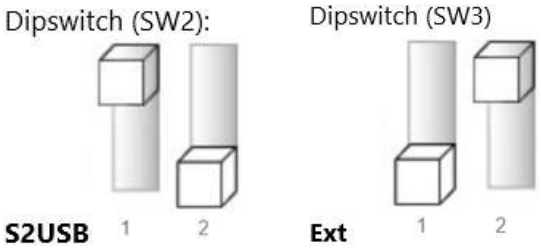




5.4.2 External S2USB control

In this configuration, an external host such as a Windows or Linux environment controls ema through use of ema:Play's S2USB port (J10). This configuration can also be used to manually send AT commands via a Terminal program. The Modem UART interface (standard COM port, 115200,8,N,1) can be accessed directly using this method. Additionally, and with the help of two short jumpers, the emaLink interface (enhanced COM port, 19200,8,N,1) can also be accessed through the S2USB port (J10). Reference *Figure 22* for how to connect the jumper wires. AT commands are used to control ema.





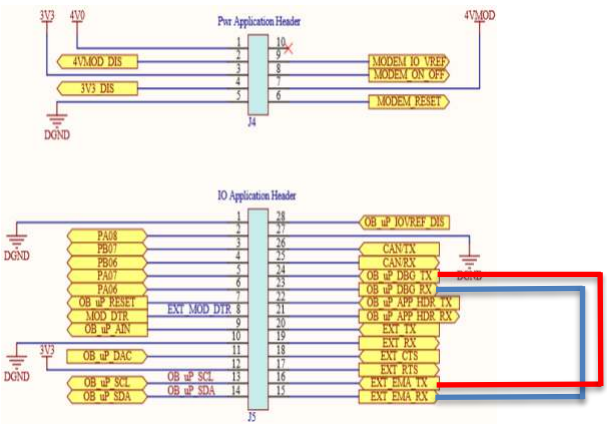


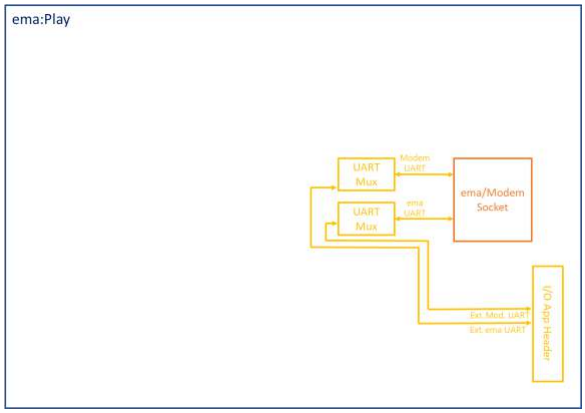
Figure 22

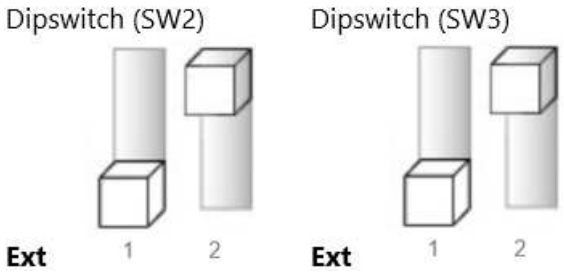
5.4.3 External USB control

In this configuration, an external USB host driver controls ema through use of ema's direct USB interface via ema:Play's USB port (J6). This configuration requires the host system to support ema USB host drivers. These drivers are available from OptConnect. Refer to section 1.2 for the best way to obtain these drivers.

5.4.4 External MCU control

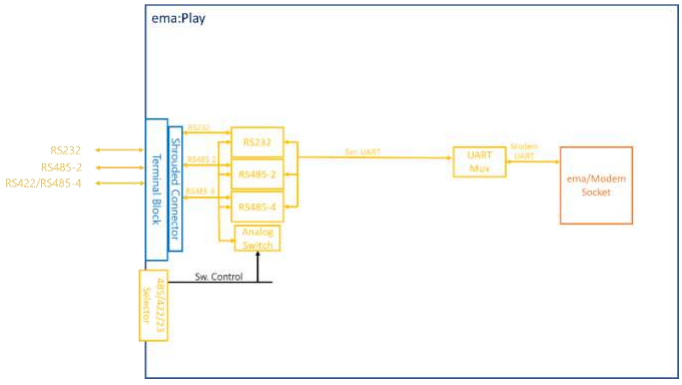
In this configuration, an external MCU is running an application and using ema as the cellular internet connection. Both the Modem UART and emaLink interfaces are connected directly to the external MCU via ema:Play's IO Application Header (J5). AT commands are used to control ema.

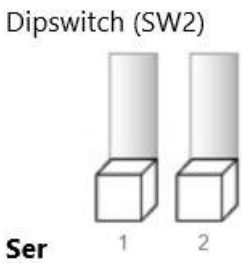


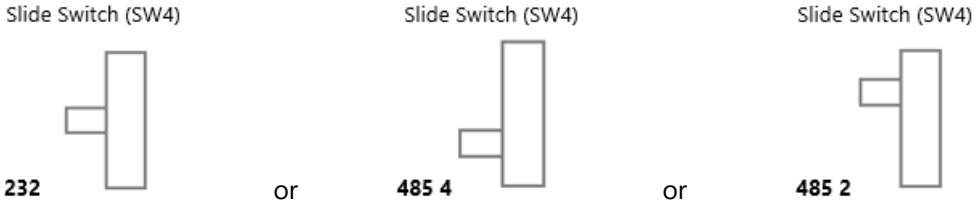


5.4.5 External RS232/RS485/RS422 control

In this configuration, an external host device that supports RS232/RS485/RS422 can be used to access ema's Modem UART interface via ema:Play's screw terminal block (J2) or shrouded connector (J1). This configuration is useful in industrial and commercial applications where the desired host system is a machine, PLC or similar. If the interface used is selected to be RS485/RS422 then the host system can be located long distances from ema and ema:Play. Auto-direction control is built into ema:Play so there is no need to control the ema:Play driver enable (DE) signal when using RS485. This configuration does not allow access to the emaLink interface. AT commands are used to control ema.







5.5 Sample Project – Getting Started with ema

5.5.1 Overview

This section aims to provide an overview and detailed description of the *Getting Started with ema* sample project. Go to **http://optconnect.com/emaPlay** , or reach out to an OptConnect representative to obtain the latest project package. Reference section 1.2 for more information. Additionally, this section will instruct the user on how to configure the ema:Play hardware to support the project. The overall goal of this sample project is to demonstrate a simple IoT application, where the ema:Play is using ema as its cellular internet connection, to send and receive ema:Play telemetry data to an IoT cloud platform. OptConnect partner's with Banyan Hills Technologies, and leverages their award-winning Canopy IoT Platform (**https://banyanhills.com/canopy-iot-platform/**). This sample project integrates Canopy's endpoint agent (Leaf) to provide the link between the ema:Play hardware and the Canopy IoT Platform. Autonomously, ema will also be managing the cellular connection. The OptConnect Summit portal (**https://summit.optconnect.com/login**) can be used to track ema's management activities.

5.5.2 Features Supported/Demonstrated

- ema startup and initialization, recovery, and shutdown
- ema microFOTA
- ema Board Notify
- ema Glimpse
- ema Board ID
- Temperature, Humidity, Light, and Voltage sensing
- 4 user digital outputs (2 LED, 2 GPIO)
- 3 user inputs (1 ADC, 2 digital)
- Banyan Hills Canopy Dashboard
- Device to cloud communications
- Cloud to device communications
- HTTPS POST
- JSON messaging

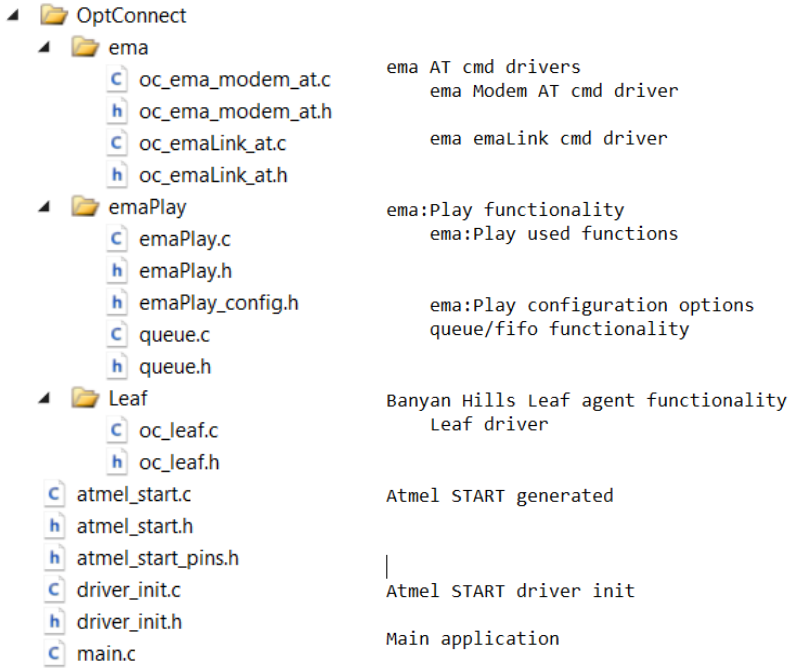
5.5.3 Hardware Configuration

The ema:Play hardware must be configured properly to accommodate this sample project. The hardware should be configured to the default configuration as shown in section 5.4.1. Additionally, verify the following settings are as follows:

- MCU reset released (SW5 set to OFF)
- ema cellular module OFF/released (SW1 set to OFF)
- Power applied through barrel connector (J9) or EXT_VIN (J1.1, J2.1)

5.5.4 Software Project Hierarchy

The sample project main application has the following folder and file hierarchy:



5.5.5 Software Configuration

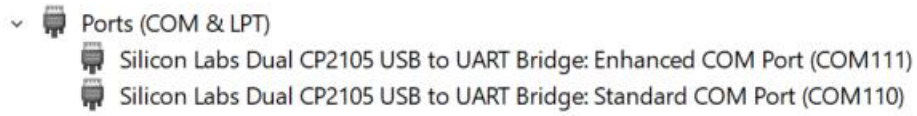
The sample project has several build time options that can be modified for different functionality of the project. The options are editable in *emaPlay_config.h*. The project must be re-built if changing these options. See below snippet for configurable options and description:

```
#define ENABLE_LEAF_NOOP                1       // 1 = send leaf noop to get data from the server synchronously(higher system latency)
                                                // 0 = use ema Board Notify to get data from the server asynchronously(not supported)
#define LEAF_ATTEMPTS                   1       // number of attempts for a leaf query(if a failure occurs)
#define DIN_D_TIME                      2000    // ms. Digital input debounce time

#ifdef DEBUG
    // function specific console debug control
    #define     DBG_MODEM_AT                0       // console output of ema Modem UART interface traffic
    #define     DBG_MODEM_AT_HEX            0       // console output of ema Modem UART interface traffic in hex format
    #define     DBG_EMA_LINK_AT             0       // console output of ema Management UART interface traffic
    #define     DBG_EMA_LINK_AT_HEX         0       // console output of ema Management UART interface traffic in hex format
    #define     DBG_EMA_LINK_URC            0       // console output of ema URCs
    #define     DBG_EMA_LINK_URC_HEX        0       // console output of ema URC hex format
    #define     DBG_APPLICATION             1       // console output of ema main application debug messsages
#endif

// functionality
#define     OPT_RECOVER_EMA             1       // 1 = forces the code to recover ema if any errors are encountered.
                                                // 0 = stops on an error
#define     OPT_AUTO_START              0       // 1 = force the code to automatically start the demo without user interaction
                                                // 0 = user must press and hold sw7 for 2 seconds or 's' in the terminal to start the demo
#define     OPT_PERIODIC_STATISTIC      0       // 1 = Periodic Mode => sends the statistic(temp, hum, adc, etc) data to the cloud periodically(~30
seconds).
                                                // WARNING, Periodic Mode consumes a lot more cellular data.
                                                // 0 = Delta Mode => sends the statistic data when a change of defined deltas(below) occurs
#if OPT_PERIODIC_STATISTIC == 0
    #define TEMP_DELTA                  .5f     // change in deg F
    #define R_HUM_DELTA                 2       // change in rel hum %
    #define EMA_V_DELTA                 .1f     // change in ema voltage
    #define SYS_V_DELTA                 .5F     // change in sys voltage
    #define SIG_Q_DELTA                 1       // change in sig q or "bars"
    #define LIGHT_DELTA                 75      // change in light adc %
    #define USER_ADC_DELTA              75      // change in user adc %
#endif
```

Console debug input and output is provided through the use of ema:Play's S2USB port (J10). This will help the user understand how the project is executing in real time and provides verbose debug messaging as per the enabled console outputs. See "*function specific console debug control*" in file *emaPlay_config.h* . The COM port settings for the console debug interface are **115200,8,N,1**. The COM port will enumerate with the computer as an Enhanced COM Port (COMxxx):



Pressing '?' in the console will list the currently active console Cmds as shown:

```
------ Cmd Menu ------
'd' = ema:Play and ema status data
's' = Start/Stop demo
```

Current console Cmd descriptions:

'd' = ema:Play and ema status data. This cmd will list the status of all relevant data to the project:

Example:

```
----ema status----
uptime:          155
pri. carrier:    AT&T
act. carrier:    AT&T
cell fw ver:     20.00.505
failover:        Disabled
failover time:   0
context ID:      1
context status:  1
acc tech:        E-UTRAN
reg status:      1
calc sig q:      2
mobile ip:       10.175.46.154
board ID:        emaPlay-4056B769384337532020203 74D2702FF-ema-n1e00d800010
ema fw ver:      v1.1.0-0-g06d50ee BUILD=dev A
sn:              ema-n1e00d800010
ema model:       EMA-L4-1-XX-A-A
OC Services:     Connected

----emaPlay status----
temp:            71.3f
rel hum:         30.8%
light ADC:       68 (12 bit)
ema V:           0.0V
sys V:           5.2V
user ADC:        1903 (12 bit)
sw 1 state:      0
sw 2 state:      0
din 1 state:     1
din 2 state:     1
led 1 duty       0 (mS)
led 1 period     0 (mS)
led 2 duty       0 (mS)
led 2 period     0 (mS)
dout 1 duty      0 (mS)
dout 1 period    0 (mS)
dout 2 duty      0 (mS)
dout 2 period    0 (mS)
sn:              4056B769384337532020203 74D2702FF
leaf init:       1
auto reboots     0
manual reboots   0
```

's' = Start/Stop demo. This cmd will start and stop (graceful ema shutdown) the demo project.

5.5.6 Software Architecture & Description

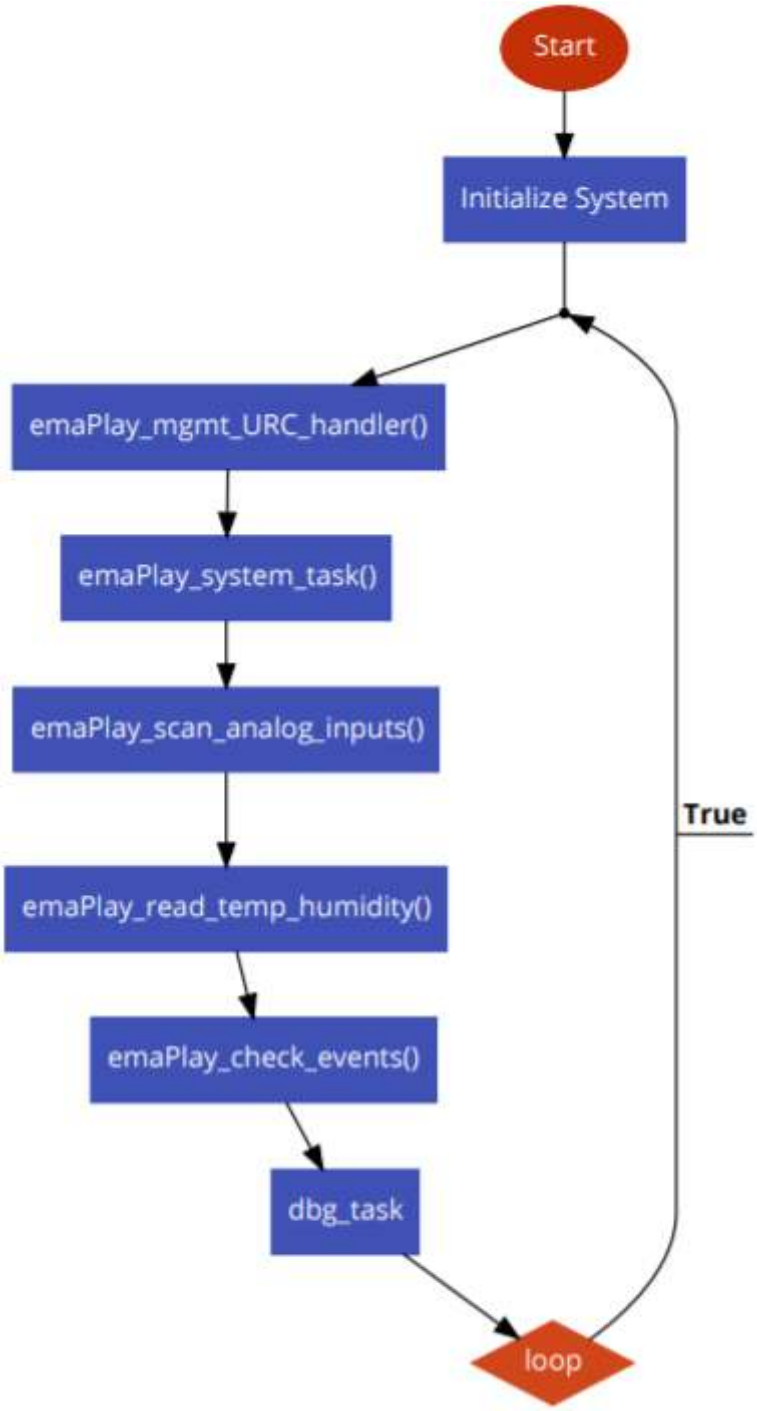
The *Getting Started with ema* sample project is architected in such a way to demonstrate how easy it is to collect local sensor/IO information and act upon it. The Canopy IoT platform allows IoT campaigns to be created that can automate tasks. For example, if Digital Input 1 goes high, then turn ON LED 1. This sample project processes the following input and outputs, that can be used with the Canopy IoT Campaigns. Reference section 5.5.10 for further information and usage example of the Canopy IoT dashboard.

Input/Type	Output/Type
Temperature/sensor	LED 1/red LED (D18)
Relative Humidity/sensor	LED 2/red LED (D19)
Light Intensity/sensor	Digital Output 1/IO app header pin (J5.4)
System Voltage/analog	Digital Output 2/IO app header pin (J5.3)
ema Voltage/analog	Signal Quality/LED bar graph/State (D14 – D17)
User ADC/analog	
Push Button 1/switch (SW7)	
Push Button 2/switch (SW8)	
Digital Input 1/IO app header pin (J5.6)	
Digital Input 2/IO app header pin (J5.5)	

D14 – D17 LED State	Description
Scrolling left to right	Idle, waiting for user input (hold SW7 for 2 seconds)
D14 Flashing	Searching for cellular network, and/or acquiring signal
All Flashing	An error has occurred
1 – 4 LEDs ON solid	Indicating current signal quality

D14 – D17 LED State Reference Table

The runtime environment is bare metal to demonstrate that the most resource constrained designs can work easily with ema. A simplified flow chart of the software is shown.



This various software components or functions that are used in this sample project are detailed below:

emaPlay_mgmt_URC_handler()*:* This function is called as frequently as possible. It handles any asynchronous ema URC's that have been received over the emaLink interface. The data is written to a FIFO queue(*mgmt_uart_async_rx_q*) at the interrupt level during the receive data callback function (*emaPlay_mgmt_rx_cb*). The entire URC verbose message is retrieved. The function *emaPlay_mgmt_process_URC()* will process the URC according to the requirements of the application.
emaPlay_system_task()*:* This function is the main state machine that handles the control of ema, as well as transacting any relevant data to and from the IoT cloud platform. Once the demo is started, the state machine powers on ema and the cellular module, checks network status, checks/starts the data session, initializes the Leaf agent, then enters a normal operations state. User events and periodic/delta statistic messages will create and build JSON messages to be sent to the cloud.
emaPlay_scan_analog_inputs()*:* This function scans and debounces the system's analog inputs.
emaPlay_read_temp_humidiity()*:* This function communicates over i2c to a temperature/humidity sensor and converts the data into degrees F and relative humidity %.
emaPlay_check_events()*:* This function processes any of the digital events that can take place from either the push buttons or the user digital inputs. The low-level reading and debouncing of the actual MCU pin(s) occurs on a timer interrupt.
dbg_task()*:* This function handles any queued console debug output as well as reading any console input.

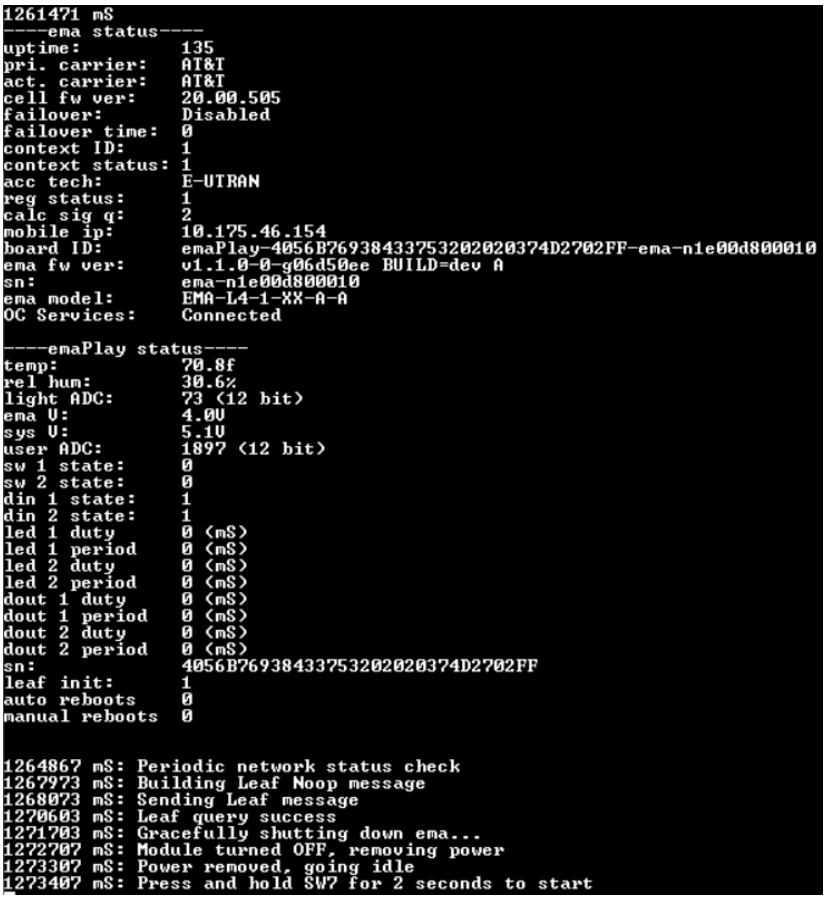
5.5.7 Console Debug Output Sample

```
ema:Play "Getting Started with ema demo" FW Version: May 11 2020
'?' for menu

761 mS: ema power detected
861 mS: Press and hold SW7 for 2 seconds to start

11202 mS: Initializing ema, applying power. Waiting.....
14211 mS: ema Board ID set: emaPlay-4056B76938433753202020374D2702FF-ema-n1e00d800010
14211 mS: ema Powered ON. Turning module ON. Waiting..........................
29858 mS: Module ON and ready
29858 mS: Checking network status, searching...
41708 mS: Registered on the AT&T network
41708 mS: Checking data session
41754 mS: No session, starting session
41897 mS: Data session active
41915 mS: Starting Leaf
41954 mS: Leaf Started
42015 mS: Delta change detected
42353 mS: No session, starting session
44298 mS: Building Leaf Statistic message
44398 mS: Sending Leaf message
46852 mS: Leaf query success
57057 mS: Building Leaf Noop message
57157 mS: Sending Leaf message
59434 mS: Leaf query success
61939 mS: Periodic network status check
69641 mS: Building Leaf Noop message
69741 mS: Sending Leaf message
72177 mS: Leaf query success
81989 mS: Periodic network status check
82990 mS: Building Leaf Noop message
83090 mS: Sending Leaf message
85422 mS: Leaf query success
86603 mS: Event detected
86703 mS: Building Leaf Event message
86803 mS: Sending Leaf message
89345 mS: Leaf query success
```

```
142182 mS: Periodic network status check
147192 mS: Building Leaf Noop message
147292 mS: Sending Leaf message
149762 mS: Leaf query success
149869 mS: Board Notification: control,4,500,1000
159969 mS: Building Leaf Noop message
160069 mS: Sending Leaf message
162467 mS: Leaf query success
162473 mS: Periodic network status check
172689 mS: Building Leaf Noop message
172789 mS: Sending Leaf message
175137 mS: Leaf query success
```



The console output shown above is typical to what would be seen when running this sample project for a short period of time. Each console line starts with a timestamp(mS) since the code execution started.

The first message outputted is the welcome message that includes the "*Project Name" FW Version: Mo Day Year,* followed by an input tip; *'?' for menu.* Reference the below table for line by line description.

Timestamp(mS)	Description
761	ema:Play has detected that power for ema is available.
861	The user is instructed to hold SW7 to start the demo.
11202	After holding SW7, ema:Play is applying power to ema and waiting for the appropriate URC.
14211	URC detected, ema:Play has set its Board ID via ema to be reported to Summit.
14211	ema:Play detected ema is powered on, and is now powering on the cellular module and waiting for the appropriate URCs.
29858	URCs detected. Cellular module is now powered on and ready.
29858	ema:Play is now checking the carrier network status for registration.
41708	ema has been registered on the AT&T network and obtained adequate signal.
41708	ema:Play is now checking for an active data connection.
41754	No data session active, ema:Play is activating the data session.
41897	The data session is now active.
41915	ema:Play is starting the Leaf agent.
41954	The Leaf agent is ready.

42015	ema:Play has detected a change in telemetry data
44298	ema:Play is building the Leaf statistic message.
44398	ema:Play is sending the Leaf statistic message via HTTPS.
…	
57157	ema:Play is checking for data from the Canopy/Leaf servers.
61939	ema:Play is periodically checking the carrier network status using OptConnect Glimpse.
…	
86603	ema:Play detected an event (button pressed).
86703	ema:Play is building the Leaf event message.
86803	ema:Play is sending the Leaf event message via HTTPS.
…	
…	
142182	ema:Play is periodically checking the carrier network status using OptConnect Glimpse.
…	
…	
149869	ema:Play received a Board Notify message from the cloud, with LED control data.
…	
…	
1261471	ema:Play outputted status data, as a result of the user inputting the ‘d’ cmd.
…	
…	
1271703	The user has instructed ema:Play to gracefully shutdown ema by pressing and Holding SW7. ema:Play is waiting for the appropriate URCs
1272702	ema:Play has gracefully shutdown ema’s cellular module.
1273307	ema:Play has removed power from ema.
1273407	ema:Play is idle and waiting for the user to (re)start the demo.

5.5.8 OptConnect Summit Portal

5.5.8.1 Overview

The OptConnect Summit portal is an online tool that allows users to manage and monitor their entire fleet of OptConnect cellular devices. This section describes how to use the Summit portal for basic ema monitoring and management. This section assumes that the users' Summit account has been setup successfully, and the user can login to the Summit portal. For questions regarding Summit account creation and login credentials, refer to section 1.2.

5.5.8.2 Login and Find ema

1. Confirm that ema:Play and ema are powered ON, and allow ema up to 30 seconds to establish a connection to OptConnect services.
2. Navigate to **https://summit.optconnect.com/login** and login to the portal. The landing page should look similar to *Figure 23.* All of the OptConnect cellular devices associated with this account will be quantified by the categories shown.

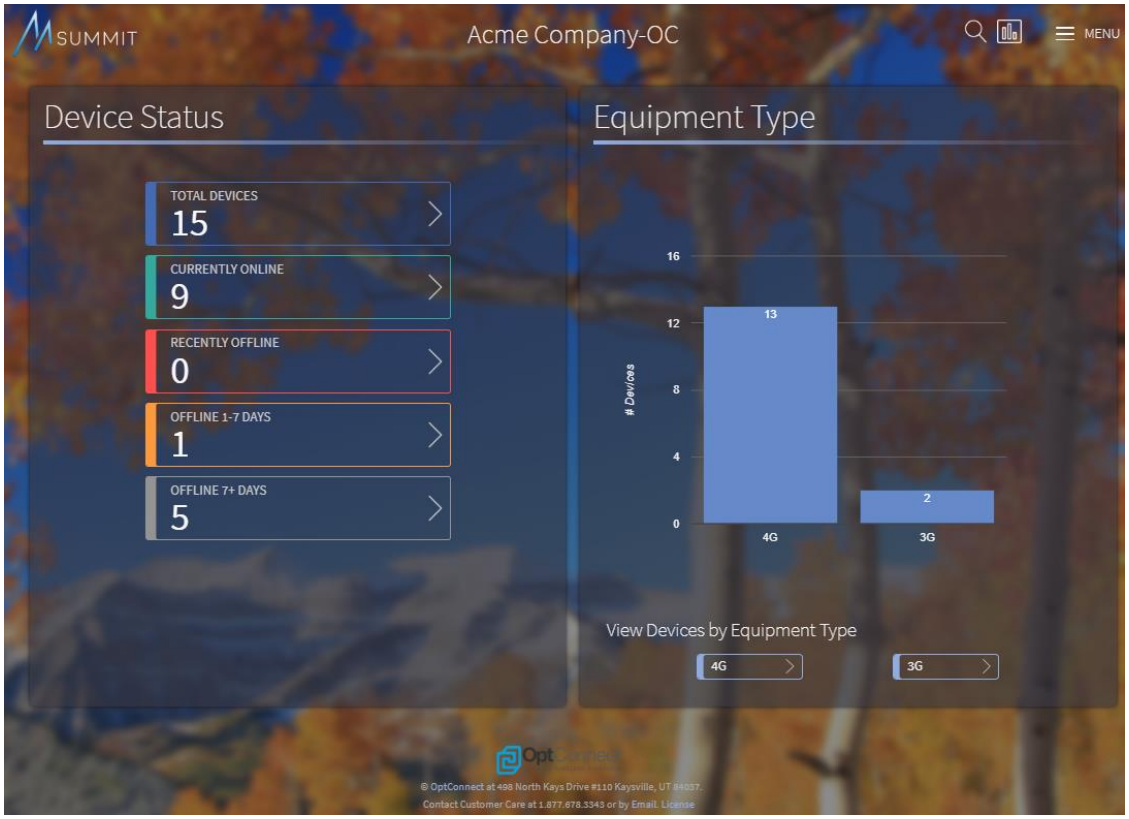


Figure 23

3. Click the "CURRENTLY ONLINE" button. The view will switch and list all OptConnect cellular devices associated with this account that are currently online. Locate the ema in question by serial number. Click on the serial number to monitor ema as shown in *Figure 24*.

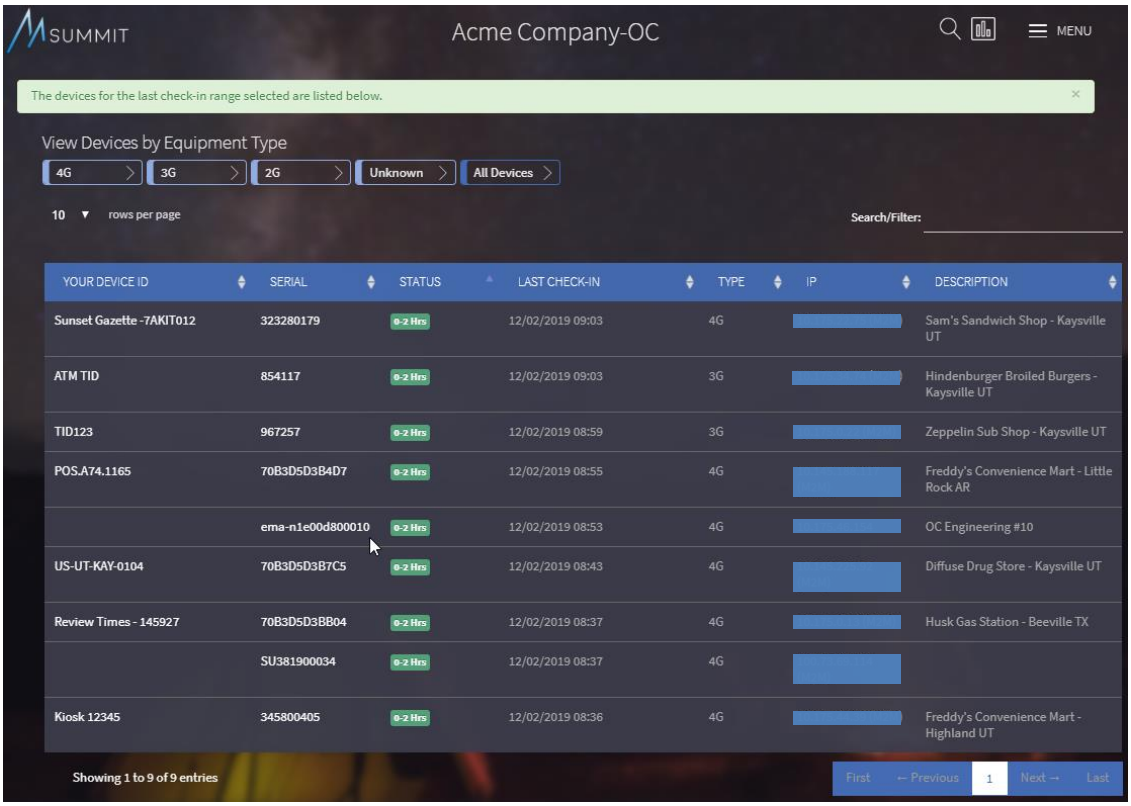


Figure 24

5.5.8.3 Device Attributes and Location

1. The "DEVICE ATTRIBUTES" section can be used to investigate ema's Board ID, IP address, Last check-in, etc. Certain fields (Device ID, and Description) can also be edited accordingly. Reference *Figure 25*. Note the board ID that was set by ema:Play in the sample log in section 5.5.7.

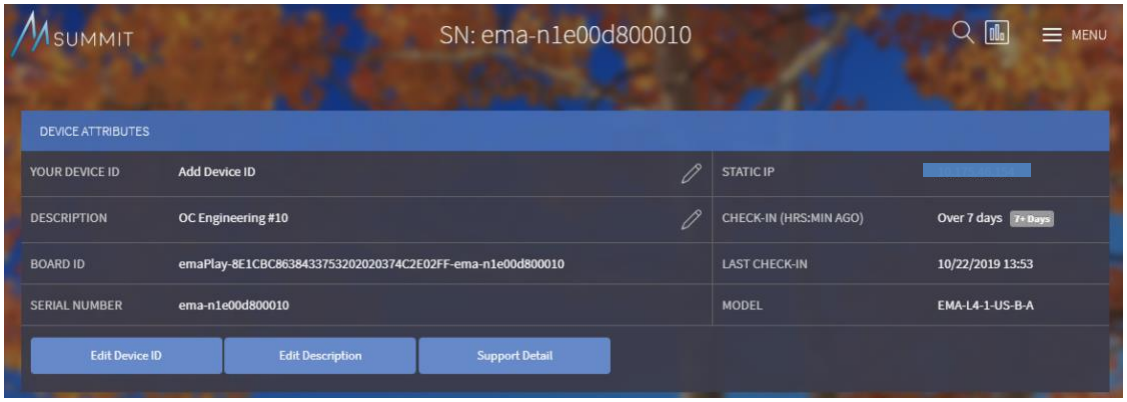


Figure 25

2. The "APPROXIMATE DEVICE LOCATION" section can be used to view ema's location. Reference *Figure 26*.

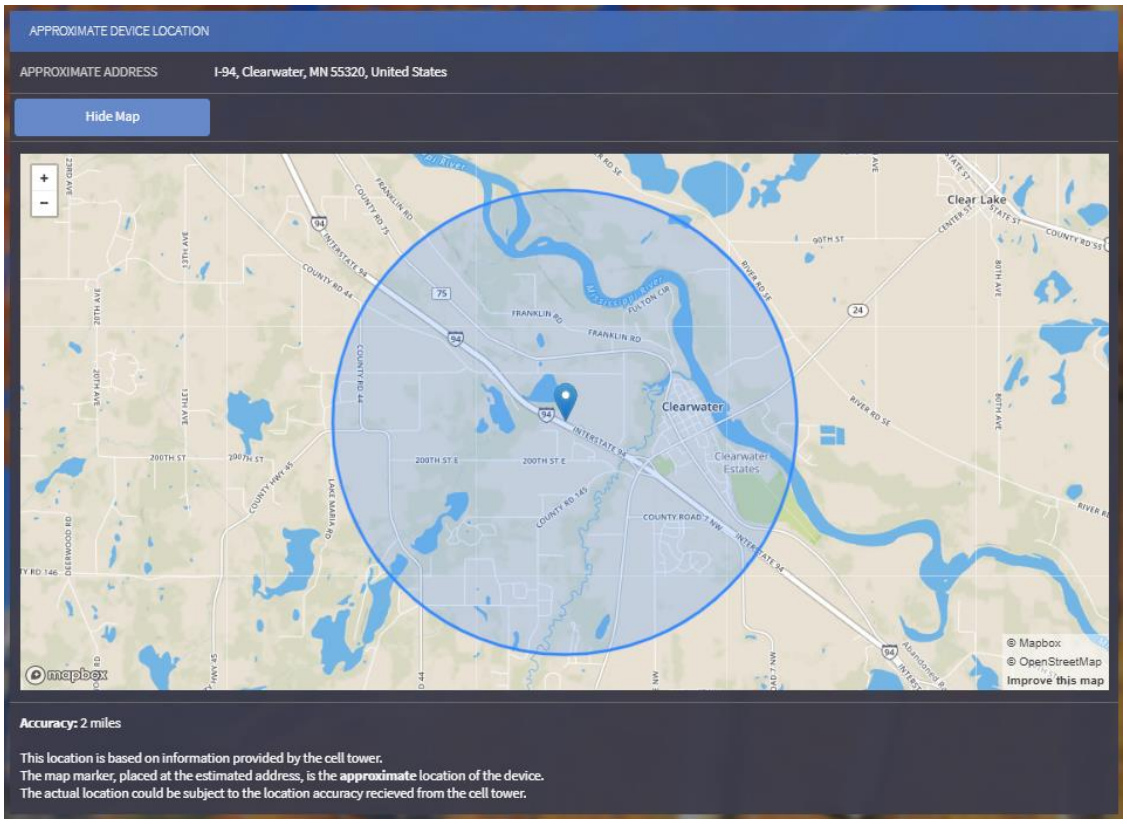


Figure 26

5.5.8.4 Events, Usage and Analytics

1. The "EMA EVENTS" section can be used to closely monitor ema's behavior over time. Reference *Figure 27*.

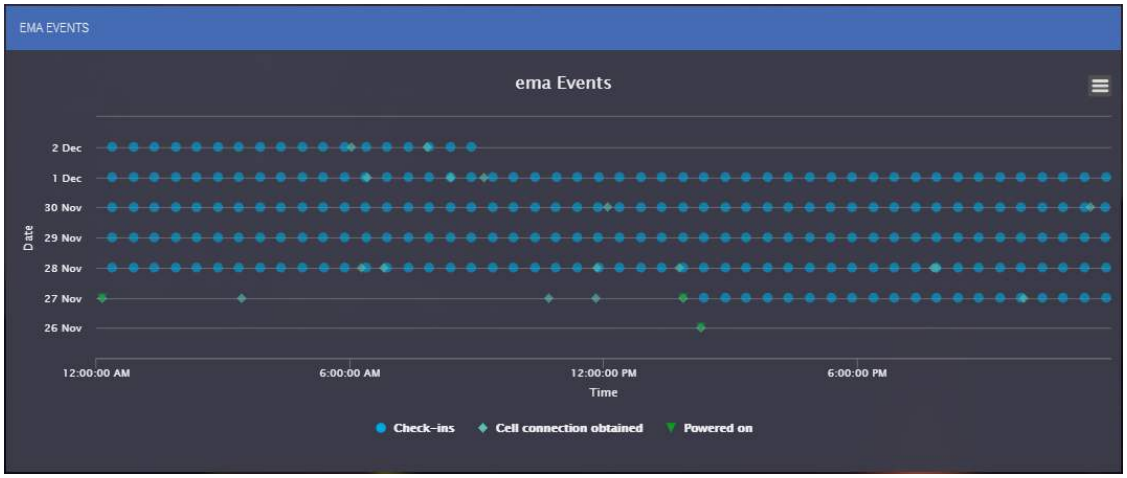


Figure 27

2. The "DATA USAGE HISTORY" and "SIGNAL ANALYTICS" sections can be used to monitor ema's usage and signal data. Reference *Figure 28*.

Figure 28

5.5.9 Board Notify via Summit API

5.5.9.1 Overview

The OptConnect Summit portal exposes the Summit Application Programming Interface (API) that can be accessed by any internet connected device. Any user application that wishes to use the Summit API must meet the following pre-requisites:

- Has a valid Summit account and login credentials
- Has the Summit API access enabled for the account

Refer to section 1.2 for information on contacting OptConnect for troubleshooting Summit API access.

The Summit API allows applications to access information and data relevant to the users Summit account and associated devices. Reference the Summit API Docs webpage at
https://docs.optconnect.com/documentation

This section demonstrates ema's Board Notify feature using ema:Play. Board Notify is a powerful but simple feature that allows any internet connected device to send small chunks of data to ema using the Summit API secure channel. This allows the user's host application utilizing ema to remain mostly idle, and not have to periodically send data out or poll for data from the server. This inherently will save on data usage and power consumption. The Board Notify feature is initiated from the Summit API and is presented to the user application and hardware via the emaLink interface as a URC. Reference the *OptConnect ema™ emaLink AT Command Manual* for further information on URC's

Confirm ema:Play is powered on and configured as shown in section 5.5.3.

5.5.9.2 Create a New API Application in Summit

1. Navigate to **https://summit.optconnect.com/login** and login to the portal. In the top right area of the screen, click on the "MENU" button. In the menu, click on "Manage API Apps" as shown in *Figure 29*.

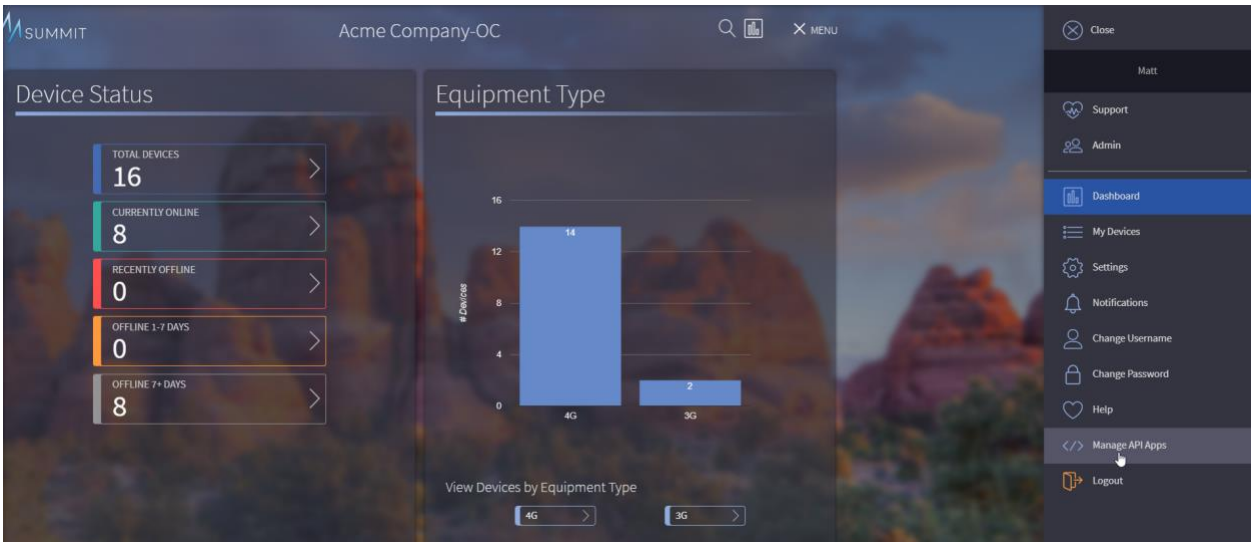


Figure 29

2. Click on the "New API Application" button as shown in *Figure 30*.

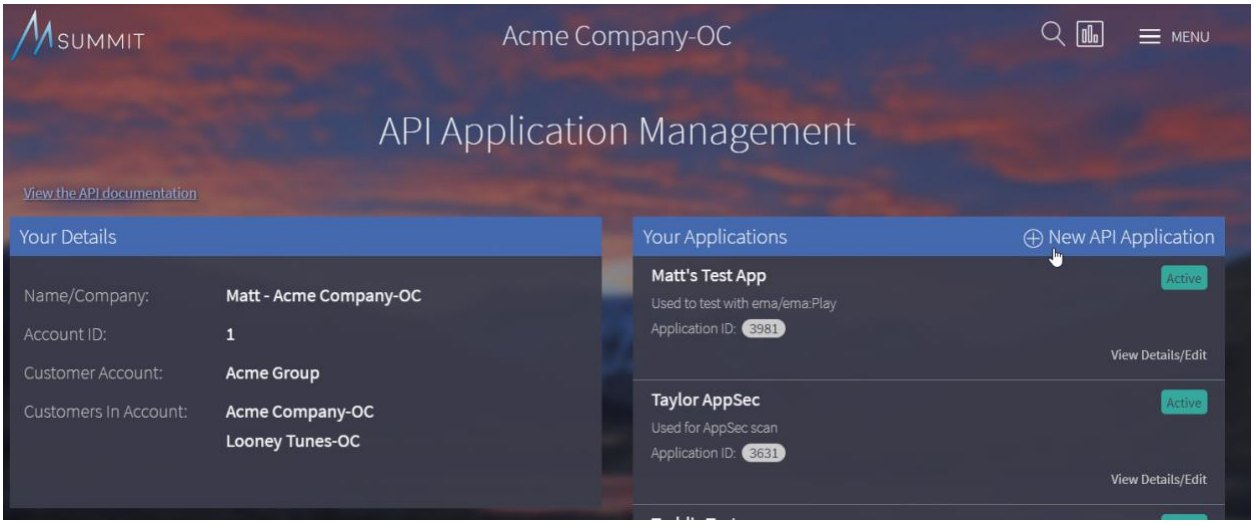


Figure 30

3. The "Create New Summit API Application" page will load. Enter a Name and Description for this API application. Select the appropriate Application Roles and Account Access settings for this API Application, then click "Create" as shown in *Figure 31*.

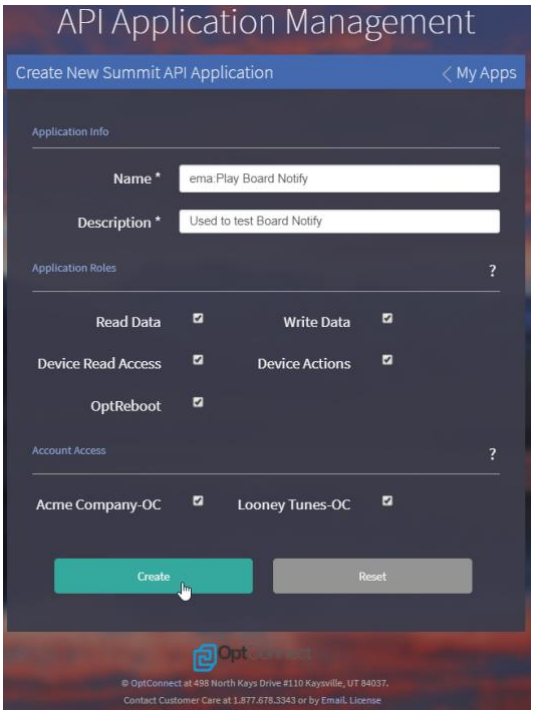


Figure 31

4. Once the "Create" button is clicked, a window will pop up asking to copy the App Secret. Copy the App Secret and save it in a secure location for later use. Close the App Secret pop up page. The "Application Details" page should now be displayed as shown in *Figure 32*. Note the Account ID and the Application ID for later use.

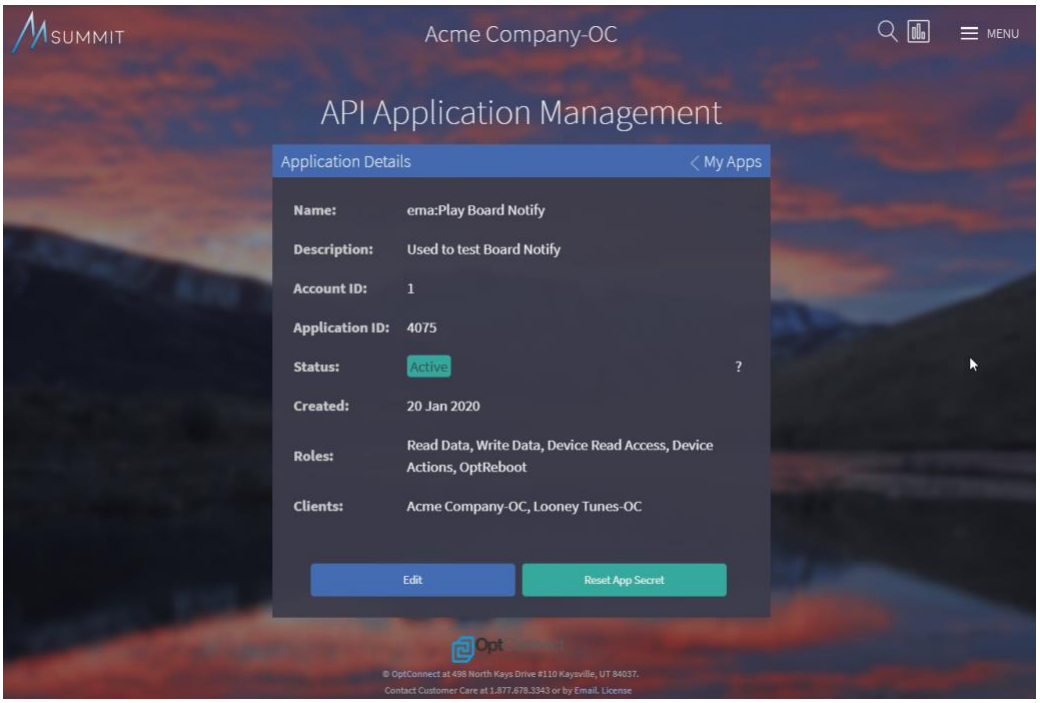


Figure 32

5. At this point, the API Application has been created for the user's Summit Account. Additionally, the App Secret, Account ID, and Application ID should all be noted and saved for the next section.

5.5.9.3 Authenticate User Application with Summit API

1. Navigate to **https://docs.optconnect.com/documentation** . OptConnect provides an interactive Summit Developers online tool that can be used to test any Summit API. In order to initiate the Board Notify feature and access the Summit API Application that was created in the previous section, follow the steps below.
2. Any application with access to the internet can use the Summit API. The application must authenticate with the Summit API using the Account ID, Application ID, and APP Secret. On the Summit Developers landing page, scroll down and click on the "POST" button next to the Authentication endpoint "/accounts/login/app_secret" as shown in *Figure 33*.

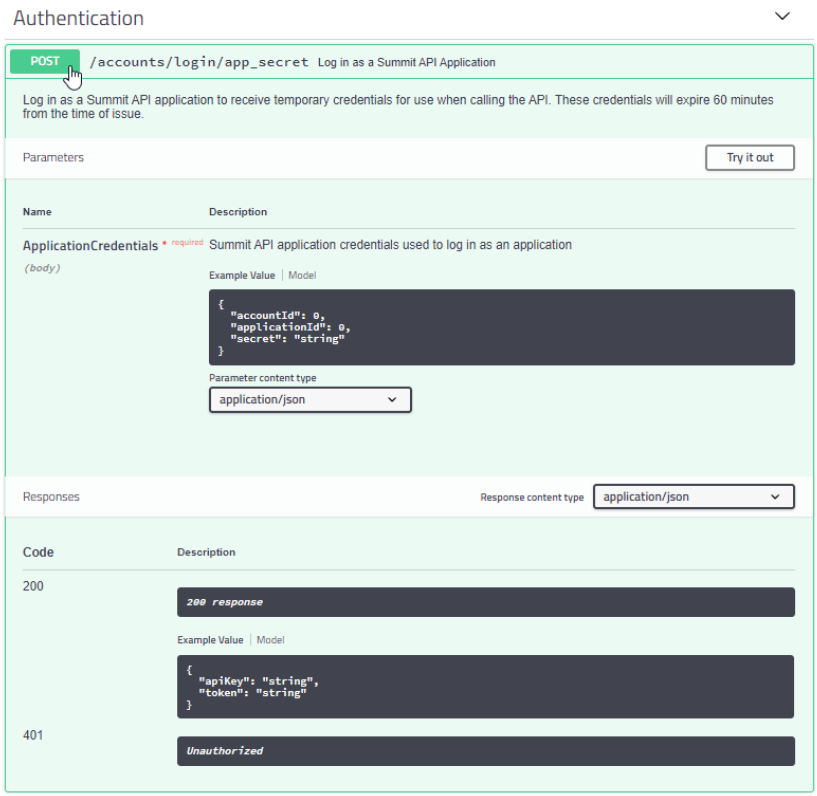


Figure 33

3. Click the "Try it out" button on the right side to make the tool interactive. Retrieve and enter the Account ID, Application ID, and App Secret directly into the JSON structure. Click the blue "Execute" button to make the request. A successful request will return an API Key and an SDK Token (JSON Web Token), which should be saved and used in subsequent Summit API calls. Reference *Figure 34*.

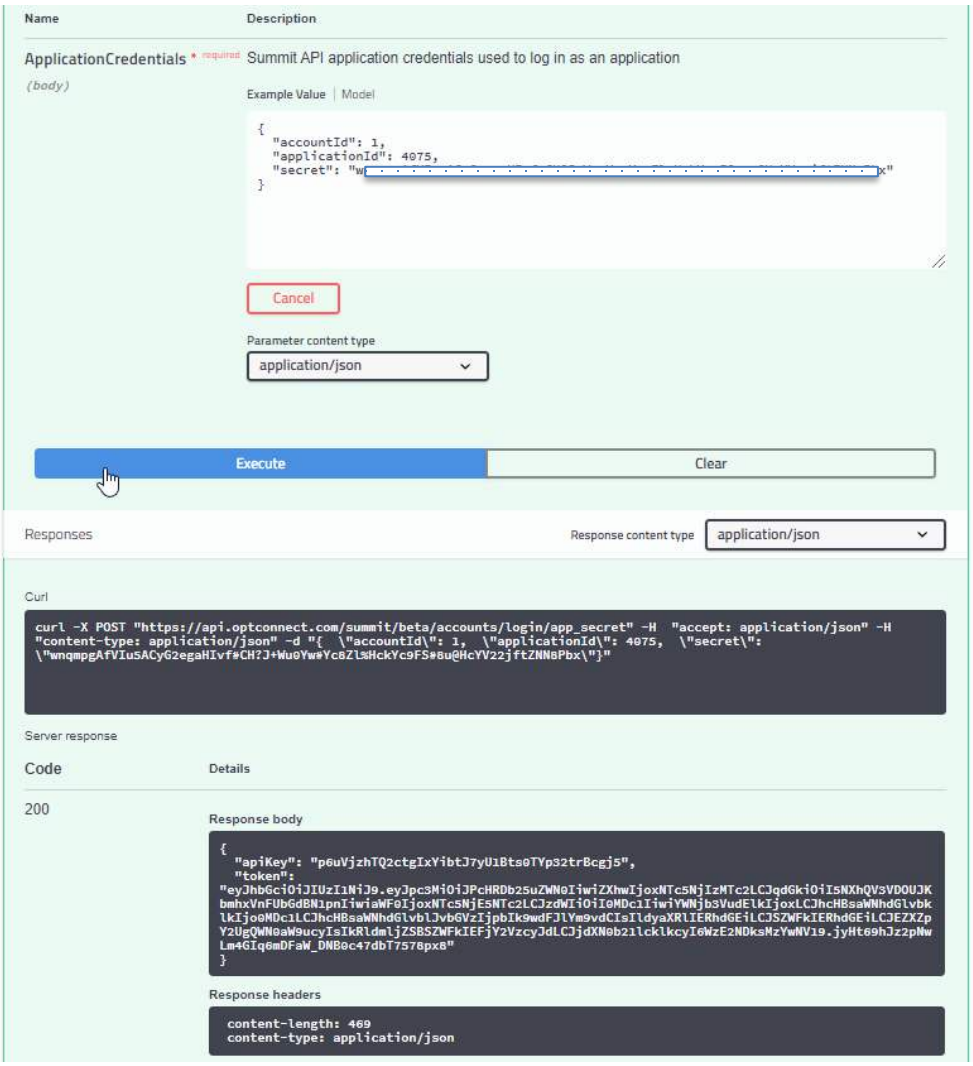


Figure 34

5.5.9.4 Send Data to ema Using Board Notify

1. Scroll down on the Summit Developers page to the Summit API Endpoint labeled "/devices/ema/boardnotify/serial/{serial}" and click the "Post" button. Next, click the "Try it out" button to make the tool interactive. Enter the ema Serial Number (found on the ema label), the API Key (from above), and the SDK Token (from above). Reference *Figure 35*.

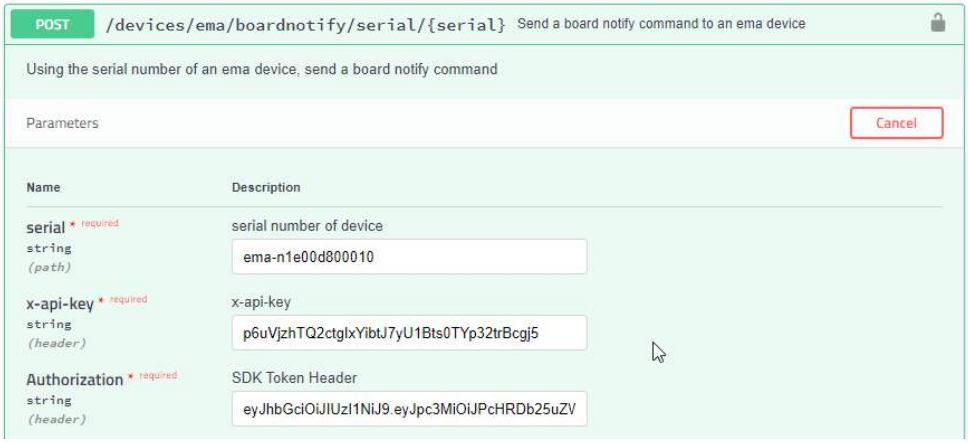


Figure 35

2. The ema:Play demonstration project will process pre-configured (reference section 5.5.6) command data adhering to the following protocol:

Cmd/Control	States	Command	Parameters (idx, duty(ms), period(ms))
LED 1	Duty On/period	"control"	"4","duty(0-65535)","period(0-65535)"
LED 2	Duty On/period	"control"	"5","duty(0-65535)","period(0-65535)"
Digital Output 1	Duty On/period	"control"	"6","duty(0-65535)","period(0-65535)"
Digital Output 2	Duty On/period	"control"	"7","duty(0-65535)","period(0-65535)"
Reboot ema		"reboot_ema"	
Reboot ema:Play		"reboot_ema_play"	

In this example, this Summit API endpoint is used to instruct the ema:Play red LED (D18) to flash with a duty cycle of 500 ms and a period of 1000 ms. Enter the command data as shown in *Figure 36* and click on the blue "Execute" button. Within a few seconds the query should respond and the ema:Play red LED (D18) should start flashing. The ema:Play debug console can also be monitored to view the Board Notify data that was received by ema:Play as shown in *Figure 37*.

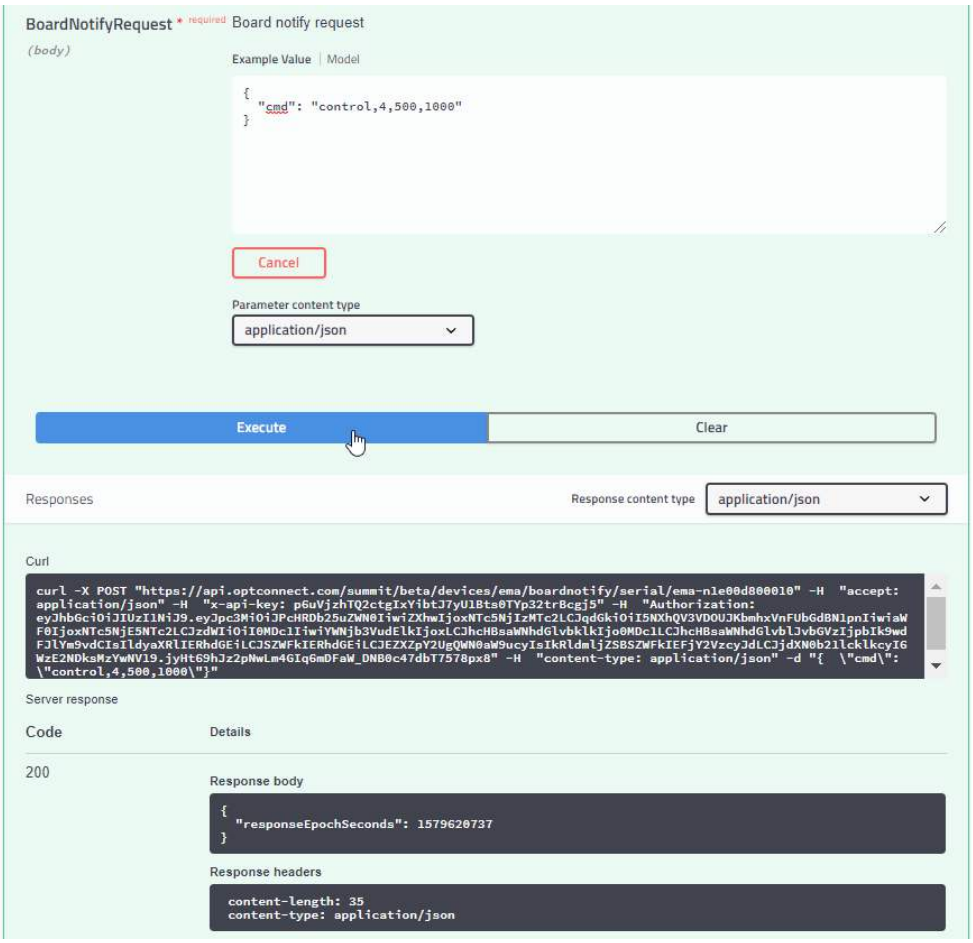


Figure 36

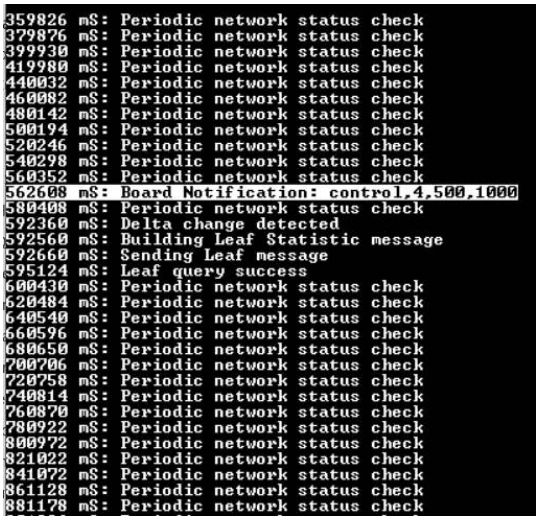


Figure 37

5.5.10 Banyan Hills Canopy IoT™ Platform

5.5.10.1 Overview

Every purchase of an OptConnect ema:Play Evaluation Kit includes a trial of Banyan Hills Canopy™ enterprise IoT platform. Its real-time dashboard, advanced features and configurable KPIs give you complete visibility into your network of data. Manage the overall health of your devices, address service issues before problems arise and delight customers with entirely new experiences. Canopy™ is coupled with Banyan Hills Leaf Agent, running on ema:Play's demonstration project, and using ema as the internet connection. This allows for a simple device to cloud round trip relationship utilizing ema:Play's onboard sensors, digital inputs, and digital outputs.

Canopy™ includes the very powerful IoT Campaigns feature which allows for custom automation tasks that can be setup and controlled from the Canopy™ dashboard. Furthermore, the ema:Play demonstration project is running the Leaf agent which connects the ema:Play endpoint hardware to the Canopy™ dashboard. The following section will briefly outline the Canopy™ dashboard relative to ema:Play and demonstrate a simple IoT Campaign to automate a task.

This section assumes that the users' Canopy account has been setup successfully, and the user can login to the Canopy dashboard. For questions regarding Canopy account creation and login credentials, refer to section 1.2. For questions about Canopy and how best to use it, contact **info@banyanhills.com** or go to **https://banyanhills.com** .

5.5.10.2 Login and Find ema:Play

1. Confirm that ema:Play and ema are powered ON, and ema:Play is configured according to section 5.5.3. Allow ema up to 30 seconds to establish a connection to the Canopy servers.
2. Navigate to **https://portal.my-canopy.com** and login to the portal. The landing page should look similar to *Figure 38*. All of the ema:Play devices associated with this account that have connected to the servers will be listed under the "Operations" tab.
   Note: The ema:Play devices are listed by the inserted ema's serial number.

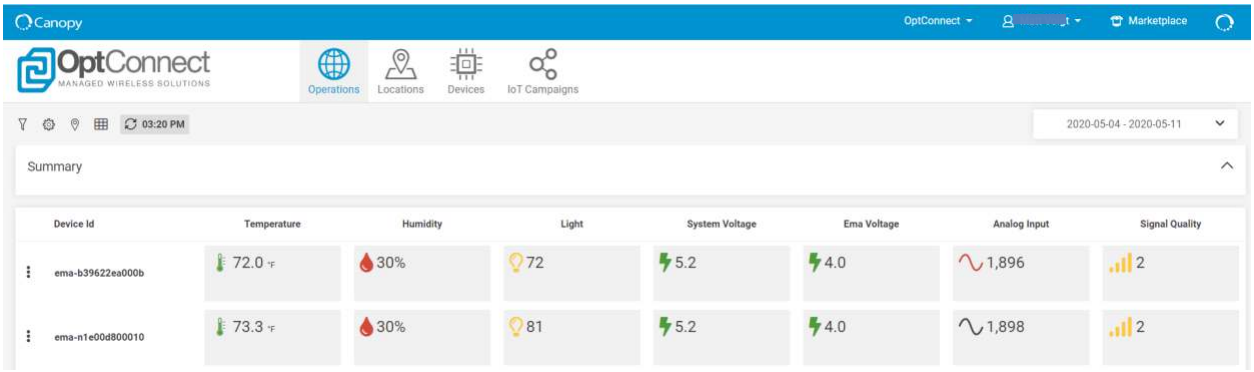


Figure 38

ema:Play's real time telemetry data can be viewed and monitored on this page.

5.5.10.3 Device Operations and Leaf Messaging

A useful tool that can be accessed from the Canopy dashboard is the JSON structure Leaf messaging data that is being sent over the cellular connection from the ema:Play hardware. To access this information, follow these steps:

1. Click on the associated ema serial number to go to the "Devices" tab as shown in *Figures 39 & 40*.

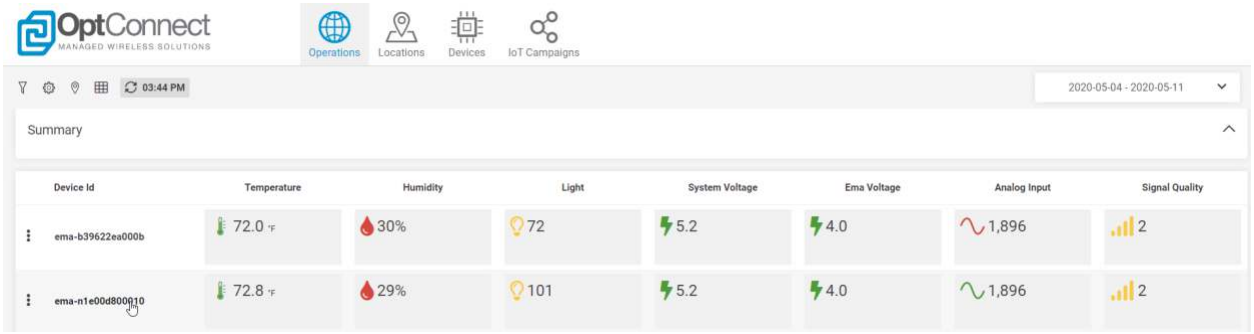


Figure 39

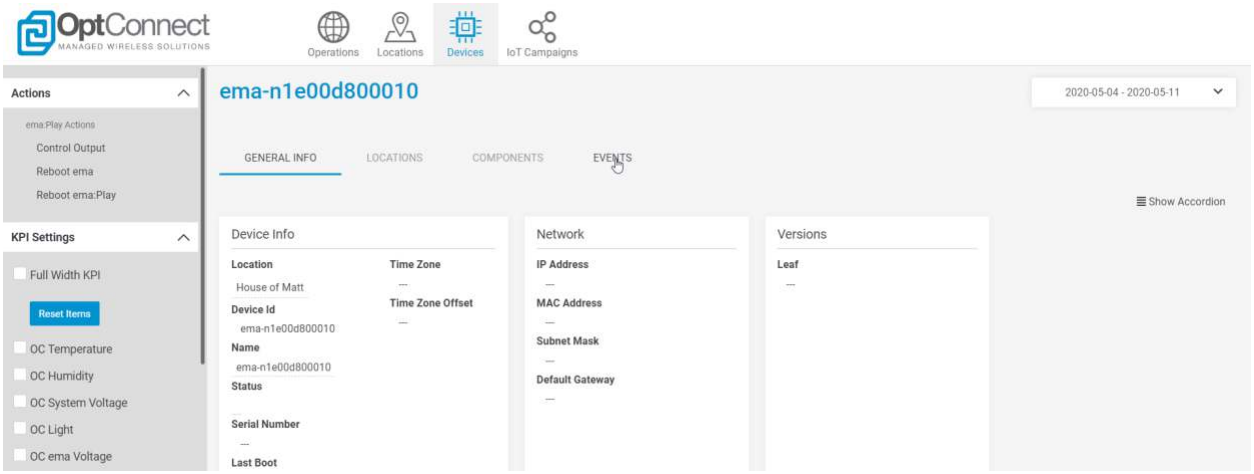


Figure 40

2. Click on the "Events" sub tab as shown in *Figure 40*. This will load and list all Leaf messages that are being transacted between the ema:Play hardware and the Canopy dashboard. *Figure 41* (below) illustrates what this could look like. This information can be used for debugging purposes to gain insight on the raw data that is flowing.
   Note: All messages transacted between ema:Play and Canopy are in JSON format.

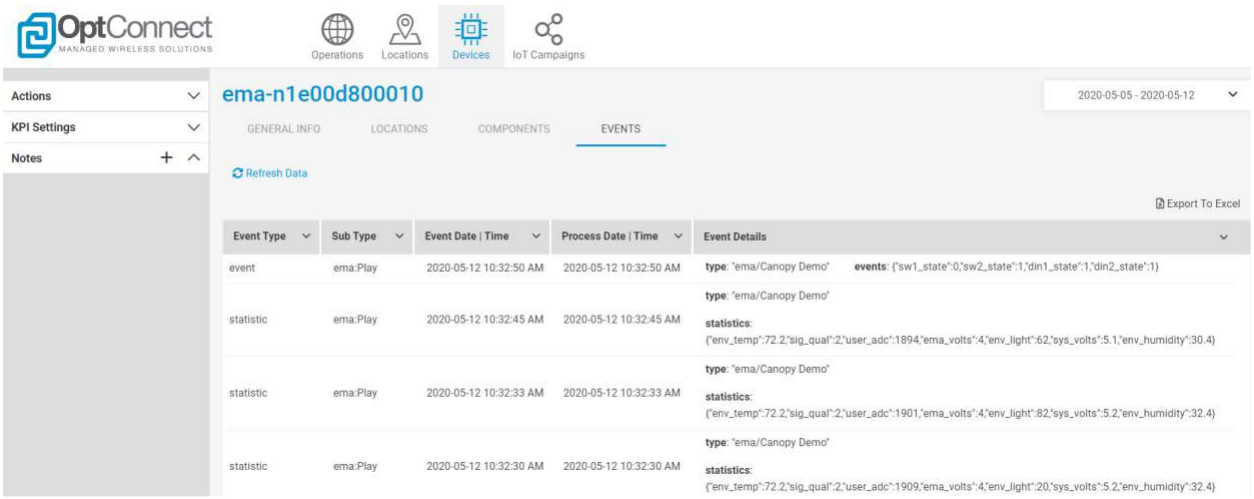


Figure 41

It can be observed from *figure 41* that an *event* type message was received on 5/12/2020 at 10:32:50 AM. The JSON data shows that the event that triggered this message was "*sw2_state":1*. This indicates that a button(ref des SW8) was pressed and released on the ema:Play hardware. Furthermore, *statistic* type messages have also been received updating the KPI state of the telemetry data that can be viewed on the "Operations" tab.

5.5.10.4 Control Actions

The ema:Play demonstration project combined with the Leaf agent is pre-programmed to handle and process commands from the Canopy servers. These are known as *Actions*. The following steps will demonstrate how easy it is to command ema:Play to control an output.

1. From the "Devices" tab, expand the "Actions" section which is located towards the upper left of the page as shown in *Figure 42*.

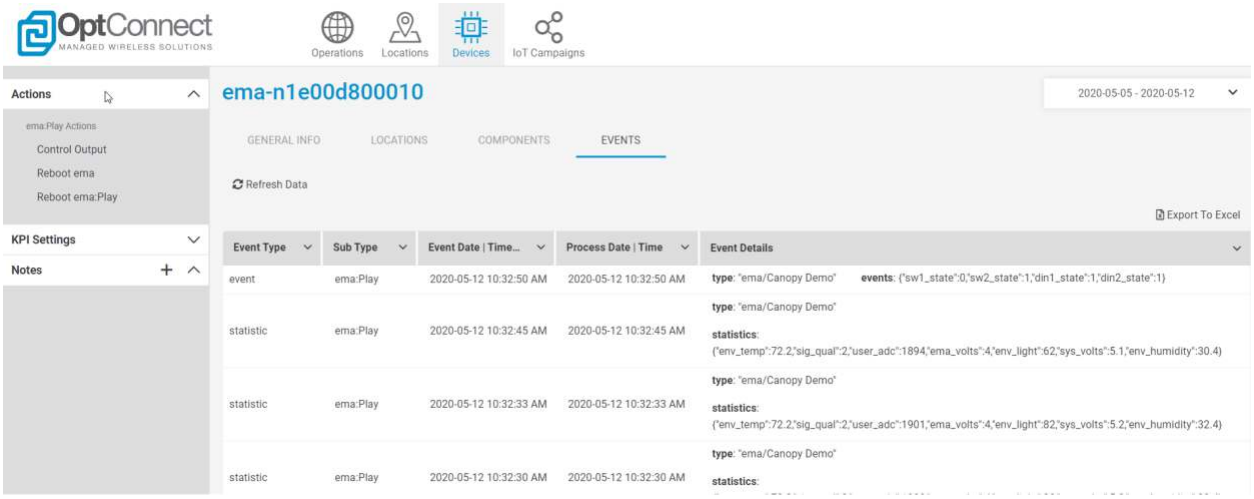


Figure 42

2. Click on the "Control Output" button to open the Control Output window. Select "LED 1" from the Output drop down, enter **500** for Duty, and enter **1000** for Period as shown in *Figure 43*. This will command ema:Play to control it's LED 1 (ref des D18) to flash at a frequency of ~2 Hz.

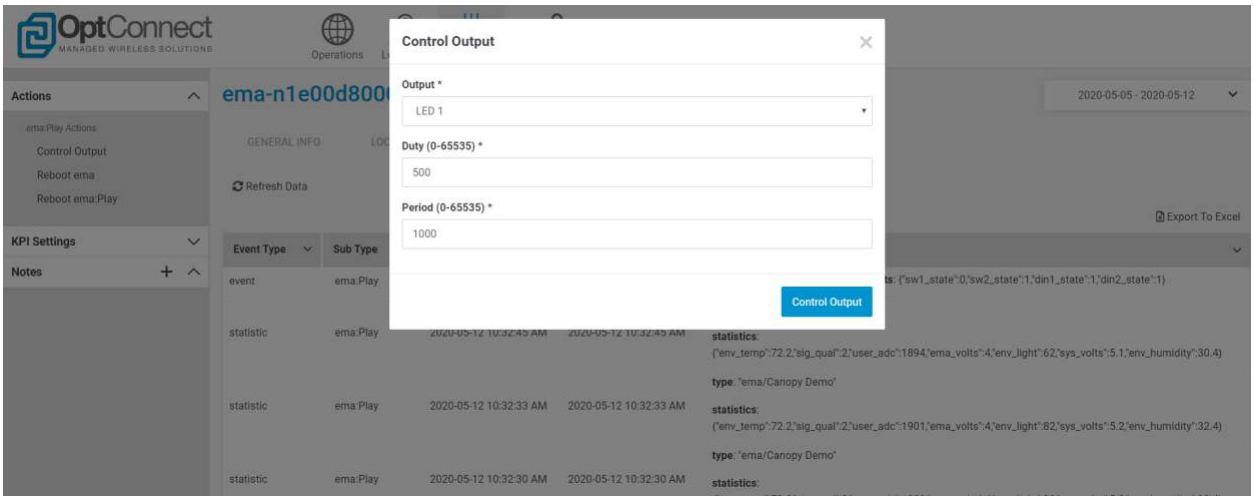


Figure 43

3. Click the "Control Output" button to close this window. Click the "Control Output" confirmation button to send the command as shown in *Figure 44*.

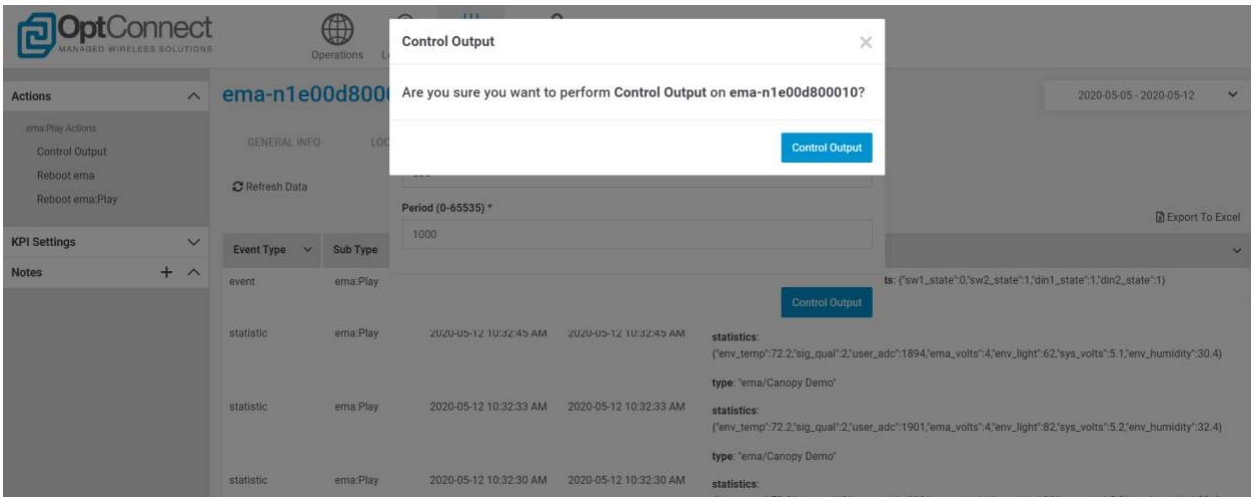
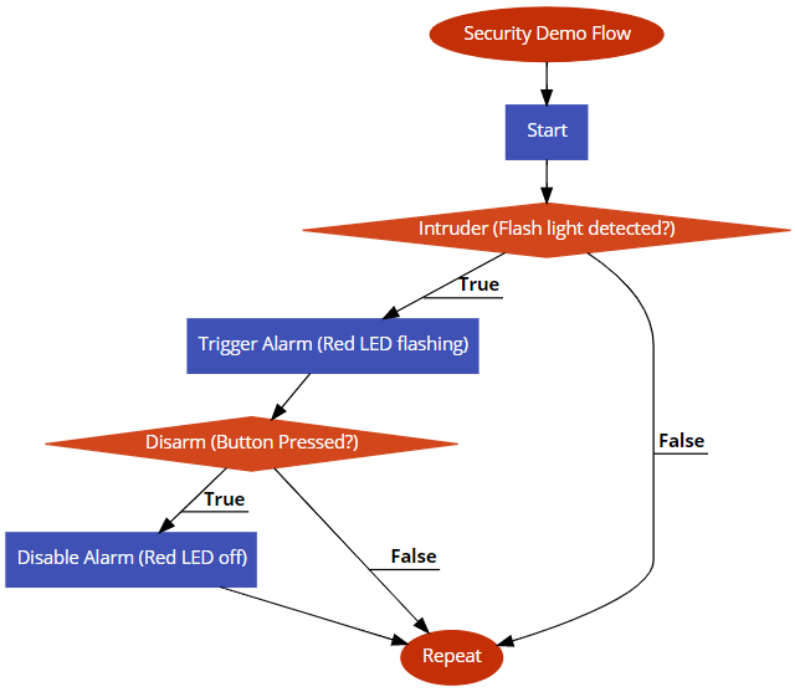


Figure 44

At this point the Canopy servers will post the command data for retrieval by the next ema:Play Leaf request. This can take up to 10 seconds. Observe on the ema:Play hardware that the RED LED D18 is flashing.

5.5.10.5 Automation using IoT Campaigns

The Canopy dashboard is displaying ema:Play's real time telemetry data and can control ema:Play's outputs, as the previous 3 sections have demonstrated. Now, it's time to tie it all togethers using IoT Campaigns. IoT Campaigns are a simple way to automate tasks using the Canopy dashboard, without having to make any changes at the endpoint hardware, ema:Play. This section will demonstrate how to setup and use IoT Campaigns. To do this we will use ema:Play's built-in light sensor, push button, and one of the two controllable LED's to simulate a security application. The high-level goal is to toggle an output when light intensity goes above a certain threshold, and to be able to press a button to disable the output. Reference section 5.5.6 for the available input and outputs of the ema:Play hardware. The scenario for this demonstration is outlined below:



1. From the Canopy dashboard, navigate to the "IoT Campaigns" page as shown in *Figure 45*.

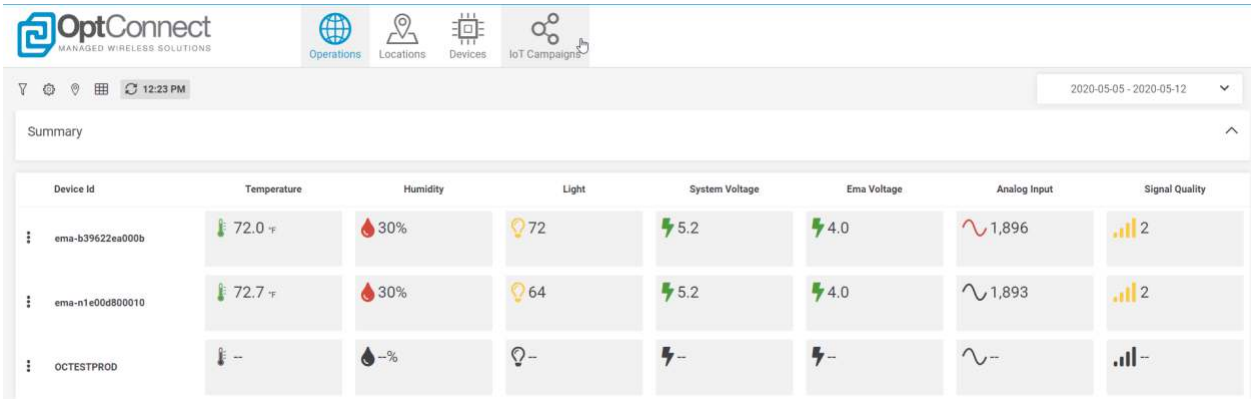


Figure 45

2. On the right side of the screen click the "Add New" button to create a new IoT Campaign. Enter **Intruder – Flash Light Detected** in the Name Field and **Alarm Trigger** in the Description Field and click "Add" as shown in *Figure 46*.

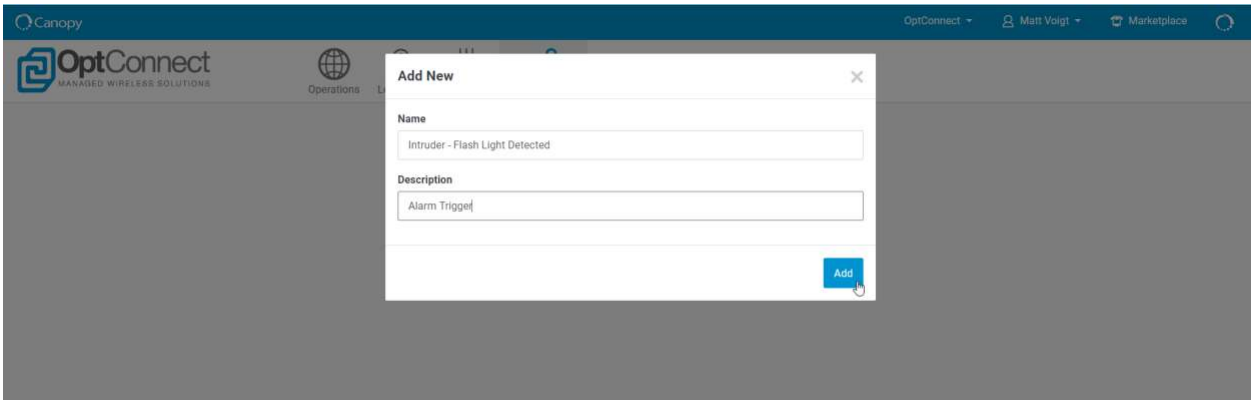


Figure 46

3. The IoT Campaign canvas page will load and should look similar to *Figure 47*. The canvas is where the logic for the **Intruder – Flash Light Detected** IoT Campaign can be designed. The left side of the screen includes the modules that can be dragged onto the canvas and interconnected with other modules.

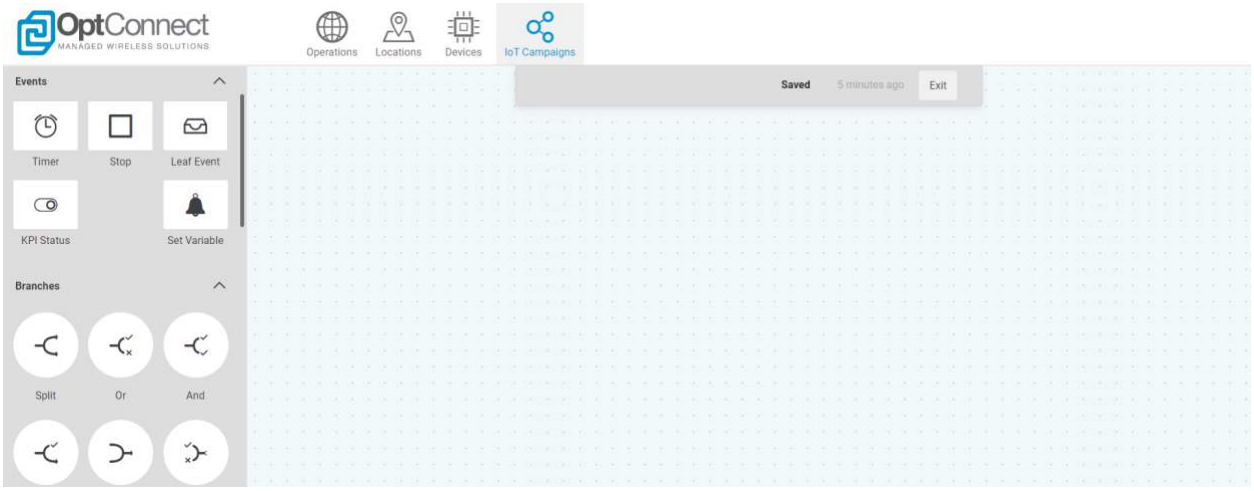


Figure 47

4. Design the IoT Campaign according to *Figure 48*, making sure to click on each module and set the appropriate fields to the correct Organization, Device, etc. Click on "Start Campaign" as shown.

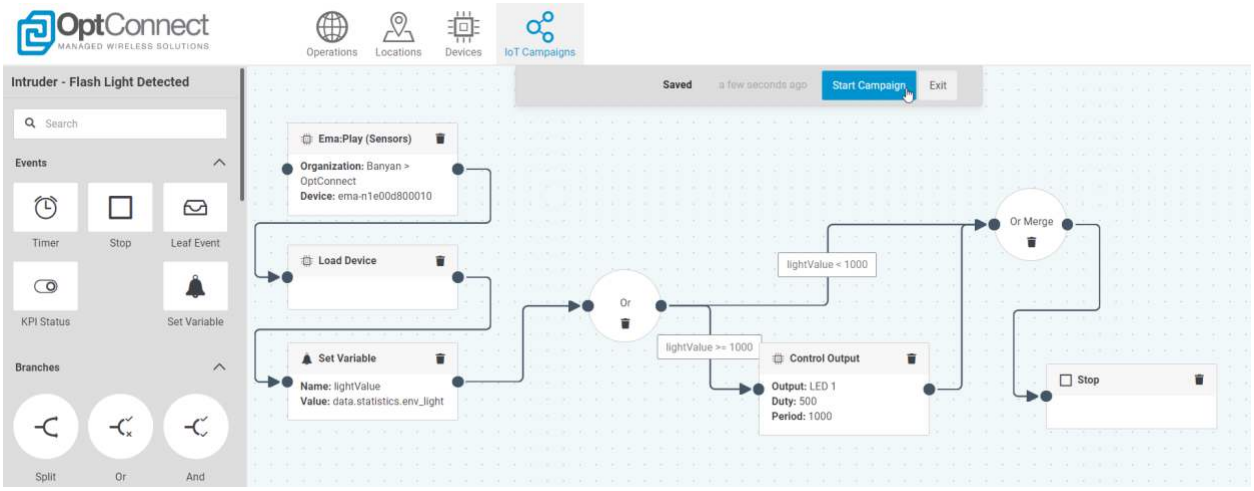


Figure 48

5. Repeat steps 2 – 4 above to create another IoT Campaign named **Disarm System** with description **Disable Alarm**. Use *Figure 49* as reference for the logic. Start the Campaign and return to the "IoT Campaigns" page.

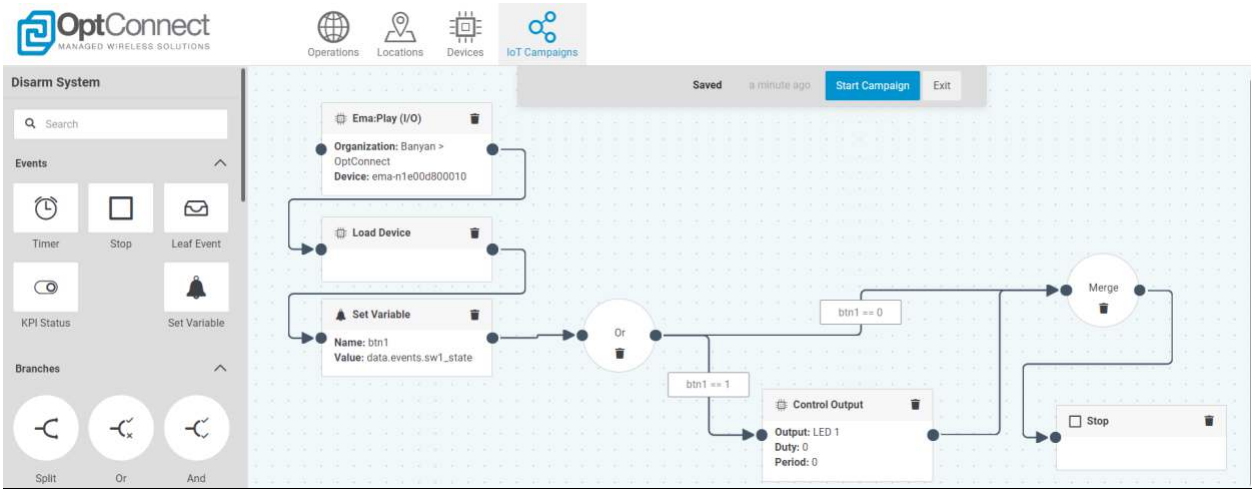


Figure 49

6. The two campaigns that were just created should be listed as shown in *Figure 50*. They should show *Active* in the "Status" column.

Figure 50

7. Navigate back to the "Operations" page and observe the real time ema:Play "Light" column for the device in question as shown in *Figure 51*.
8.

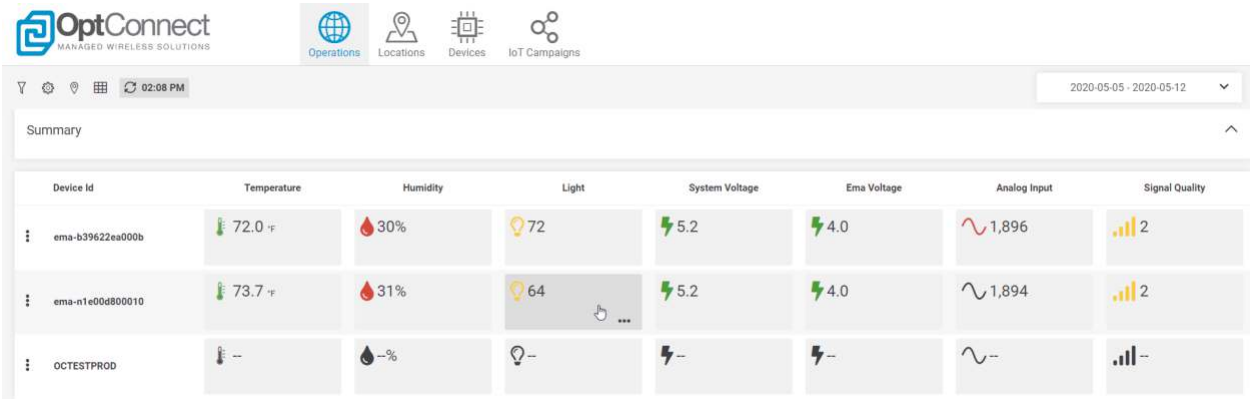


Figure 51

9. According to the logic of the IoT Campaign **Intruder – Flash Light Detected,** when this value goes above 1000, the LED 1 (ref des D18) should begin to flash at a frequency of ~2 Hz. Use a light source and shine it at the light detector (Q2) on ema:Play, while observing the "Light" column on the "Operations" page for the appropriate device. Confirm that within 10 seconds of the value crossing the threshold (1000), LED 1 begins to Flash. Reference *Figures 51 & 52*.

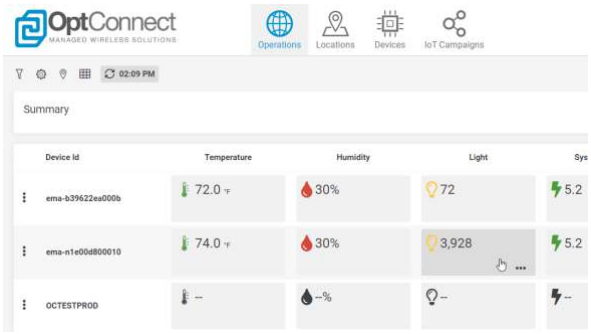


Figure 51

Figure 52

10. The alarm has been triggered. It turns out it was a false alarm, so now it's time to disable the alarm. Use the second IoT Campaign **Disarm System** to disable the alarm and stop LED 1 from flashing. According to the logic of this IoT Campaign, press SW 1 (ref des SW7) on ema:Play to stop the LED from flashing. Allow 10 seconds for the LED to stop flashing.

5.5.10.6 Summary

The demonstration above may not seem all that useful in a real-world application, but it could easily be adjusted using the IoT Campaign canvas and logic to monitor and react to any of ema:Play's preprogrammed input and outputs. Reference section 5.5.6 for a complete list of these. For example, ema:Play could be placed inside of a piece of equipment, provide the internet connection via ema, and monitor the temperature. Using Canopy and IoT Campaigns, if the temperature crossed a threshold, an email could be sent to an authority to take corrective action. All of this would require no technical work and demonstrates how ema:Play, coupled with Canopy provides a full soup-to-nuts IoT experience.

6. Revision History

Revision	Date	Description	Author
1.0	5/14/2020	Initial Release	MSV