Iridium **P1387-UG-002**

Certus Integrator Development Kit – Operating Guide

Issue 2

Iridium Communications Inc.
1750 Tysons Blvd, Suite 1400
McLean, VA 22102
www.iridium.com

Toll Free: +1.866.947.4348 [US Only]
International +1.480.752.5155
email: info@iridium.com

LEGAL DISCLAIMER AND CONDITIONS OF USE

This document contains information on the Iridium Certus℠ Integrator Development Kit – Installation & Operating Guide for Iridium Certus Value Added Manufacturers ("VAM") and is provided "AS IS." The purpose of providing such information is to enable Iridium Certus VAMs who are developing Iridium Certus products ("Product") to understand how to integrate such Product into a wireless voice and/or data solution.

Iridium Satellite LLC and its affiliated companies, directors, officers, employees, agents, trustees or consultants ("Iridium") have made reasonable effort to make the information in this document reliable and consistent with specifications, test measurements and other information and assume no responsibility for any typographical, technical, content or other inaccuracies in this document. Nor does Iridium certify or otherwise make any representation or warranty as to the quality, effectiveness, reliability, performance or safety of any certified Product.

Iridium reserves the right in its sole discretion and without prior written notice to any Iridium Certus VAM to change the Iridium Certus℠ Integrator Development Kit – Installation & Operating Guide and materials and/or revise this document or withdraw it at any time. The Iridium Certus VAM assumes the full risk of using all information contained in this document.

Use of the information contained in this document is restricted to the development activity by Iridium Certus VAM authorized under the relevant agreement(s) between such Iridium Certus VAM and Iridium, and is otherwise subject to all applicable terms and conditions of such agreement(s) including without limitation warranty, conditions of use and confidentiality provisions of such agreements(s).

Export Compliance Information

This document is controlled by the export laws and regulations of the United States of America. The U.S. Government may restrict the export or re-export of this document to certain individuals and/or destinations. Diversion contrary to U.S. law is prohibited.

1 Revision History

Revision	Date	Comment	Originating Document
1.0	24 August 2015	Initial Released Version	
2.0	9 February 2016	Updates for sample application	Docman 0.11

2 List of Abbreviations

Abbreviation	Description
BCX	Broadband Core Transceiver (Iridium EBBS NEXT transceiver board)
DB9	9 Way D-sub connector
EMC	Electromagnetic Compatibility
ETM	EBBS Test Modem
IDK	Integrator Development Kit
LAN	Local Area Network
LED	Light Emitting Diode
OAM	Operations And Maintenance
PSU	Power Supply Unit
RF	Radio Frequency
MCX	Micro-coaxial connector (RF)
SMA	Sub-miniature version A RF connector
TTL	Transistor-Transistor Logic
DB9	9 Way D-sub connector

3 References

1. BCX Test Interface Control Document P1387-ICD-005
2. BAE to BCX Interface Control Document P1387-ICD-001
3. BCX to BAA Interface Control Document P1387-ICD-002

4 Product Overview

The Certus Integrator Development Kit (IDK) consists of two parts, the BCX and the BCX Development Board. The BCX Development Board is an open-framed circuit board that provides developer access and basic communications to an Iridium Broadband Core Transceiver (BCX). The IDK is intended to be used by an experienced and qualified development engineer, operating in a benign and safe development environment such as an engineering office or engineering laboratory. The IDK is not suitable for other development applications and should not be used out-doors.

The IDK is designed to transmit and receive from the BCX on a cabled RF interface to other equipment, and as such is designed as a non-intentional radiator.

4.1 Software Revision

Users should read this User guide in conjunction with the communication interface documents [2] & [3] and the latest Software Release Note pertaining to the revision of the software that is installed on the BCX.

4.2 Physical Specification

The Integrator Development Kit is comprised of a BCX Development Board mated with a Broadband Core Transceiver module (BCX), which is electrically connected via a flat-flex cable (FFC). The BCX Development Board breaks out various interfaces from the BCX itself in addition to providing power and control of the BCX unit.

Figure 1 provides an annotated image of an IDK, showing its available user interfaces.

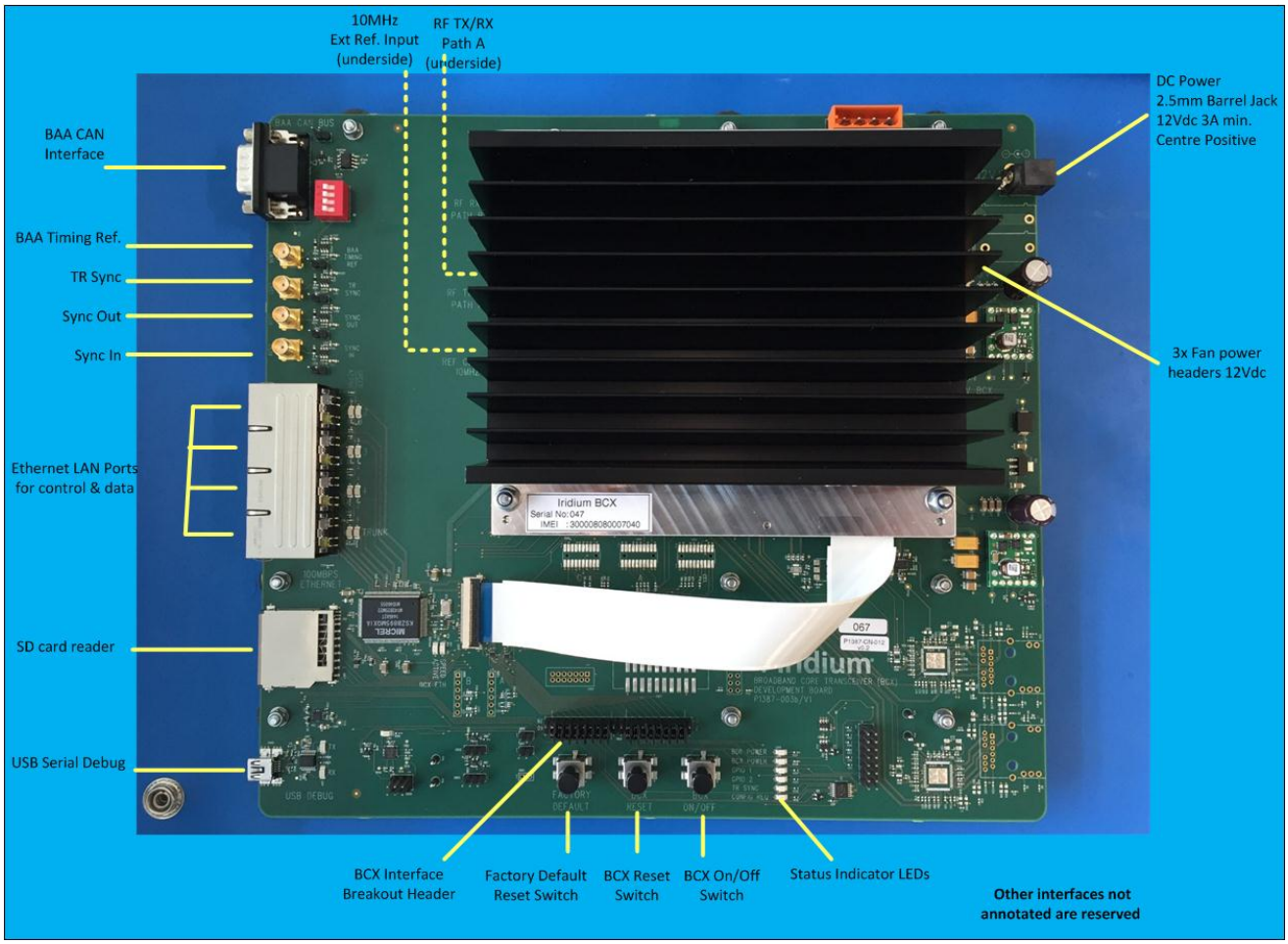


Figure 1 Integrator Development Kit

4.3 Product Dimensions

The approximate dimensions of the BCX Development Kit are represented graphically in Figure 2.

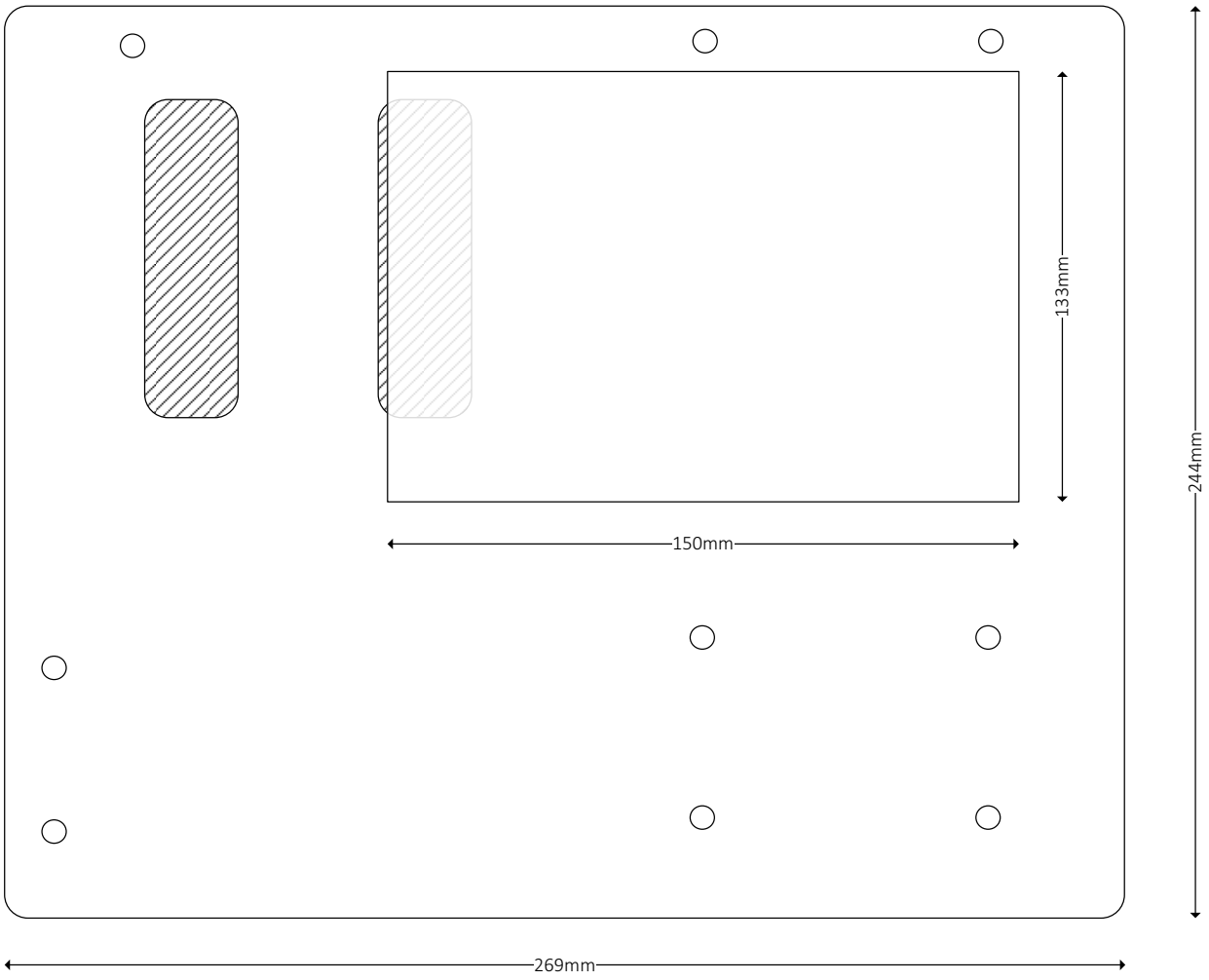


Figure 1 General Assembly and Dimensions of the BCX Development Board

(Not to scale, approximate dimensions in millimeters)

4.4 Environmental

The environmental specifications of the IDK are intended for laboratory use only and are summarized in **Table 1** below. A heatsink is fitted to the BCX allowing it to operate under these conditions, without using forced air cooling. A fan power connector is also provided on the BCX Development Kit, should the user prefer to connect a local fan to draw air across the heatsink as well.

Parameter	Value
Operating Temperature Range	+10ºC to + 35ºC

Table 1 Environmental Specifications

4.5 Physical Interface Connectors

The BCX Development Board incorporates several connectors to provide power, data and control:

- DC Power Input:
  - 2.5mm DC Barrel Jack
- RF connections (on BCX module):
  - 2x MCX RF TX/RX connections to the BCX itself
  - 1x MCX 10MHz input to the BCX
- Time synchronization connectors on the BCX Development Board:
  - 4x SMA connectors (Sync In, Sync Out, TR Sync, BAA Timing Ref.)
- 4x RJ45 Ethernet connectors
- BAA CAN interface:
  - 9-way D-Type connector
- USB Serial Debug:
  - Mini USB type B connector
  - Console port 115200 baud serial data
- SD Card Reader – *nonfunctional*
- FFC Signal Breakout Header
  - 0.1” breakout header on BCX Development Board
- External Fan Connector:
  - 3x 3-way Molex KK series header

The physical characteristics of the connectors and their electrical interfaces are described in more detail in sections 6.2 and 6.3.

All other interfaces on the IDK are reserved for other uses and should not be need.

5 IDK Installation / Configuration

Mechanically the IDK is designed for desktop development use in an ESD safe environment.

BCX software will be pre-installed on the BCX device. The current BCX software version, and the BCX self-test report can be interrogated over the Ethernet connection, using the OAM interface (See references [1] & [3] for details).

Computer network (re)configuration may be required if the host system uses different IP networks/address settings. Revised configurations should be tried and verified before they are committed to non-volatile memory. If irreversible configuration changes are made it is still possible to manually reset the IDK back to its factory default condition, by using the front-panel button marked 'FACTORY DEFAULT'.

Software release updates for the BCX together with software release note and installation instructions are available through Iridium support.

5.1 IDK Assembly Instructions

The IDK hardware is supplied as three separate packs:

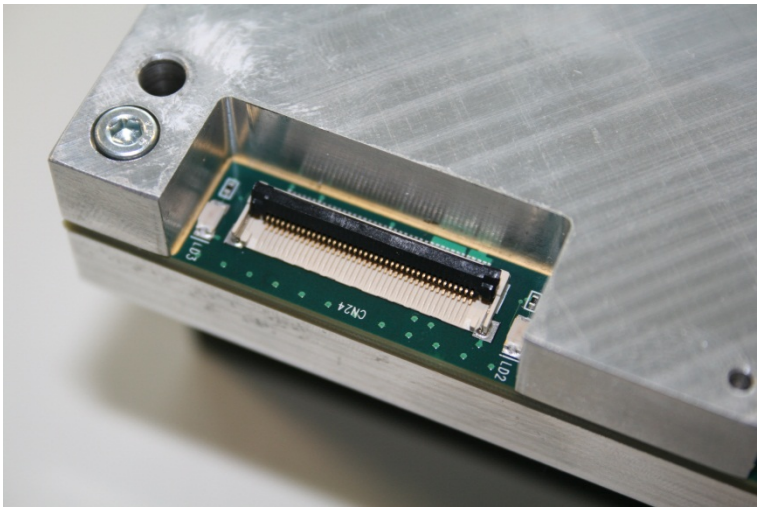
- The BCX Development Board
- The Broadband Core Transceiver module (BCX) complete with its heatsink
- A bag of items needed to connect the BCX module to the Development Board.

The bag contains the following hardware

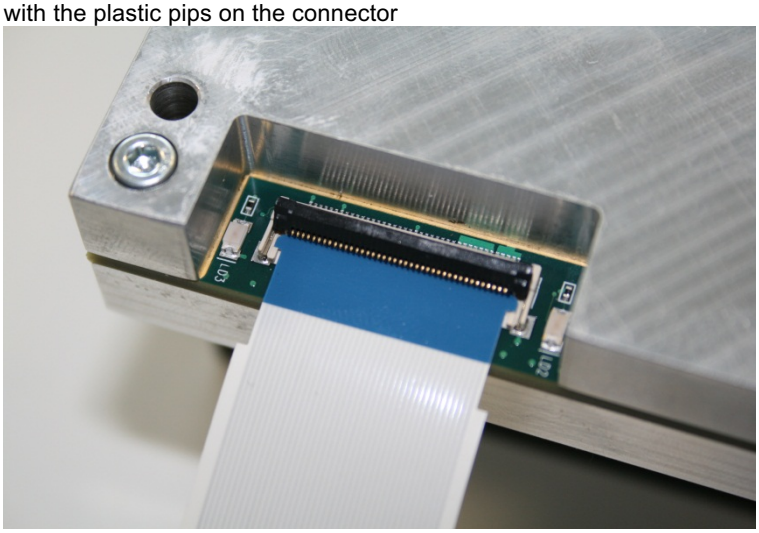
- A flat-flexi cable
- Four M4 x 30mm screws
- Four M4 nylock nuts
- An M4 x 4mm spacer
- Eight M4 plain washers

The two boards need to be assembled to each other in the following order

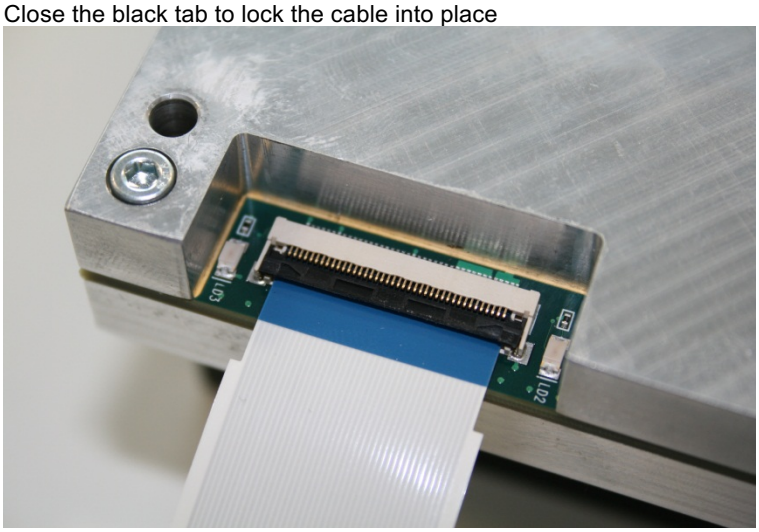
1. Insert the flat-flexi cable into the connector of the BCX module as follows:
    a. Lift the black tab on the connector



b. Insert the cable, with the blue side facing upwards, ensuring that the two 'ears' on either side are engaged with the plastic pips on the connector

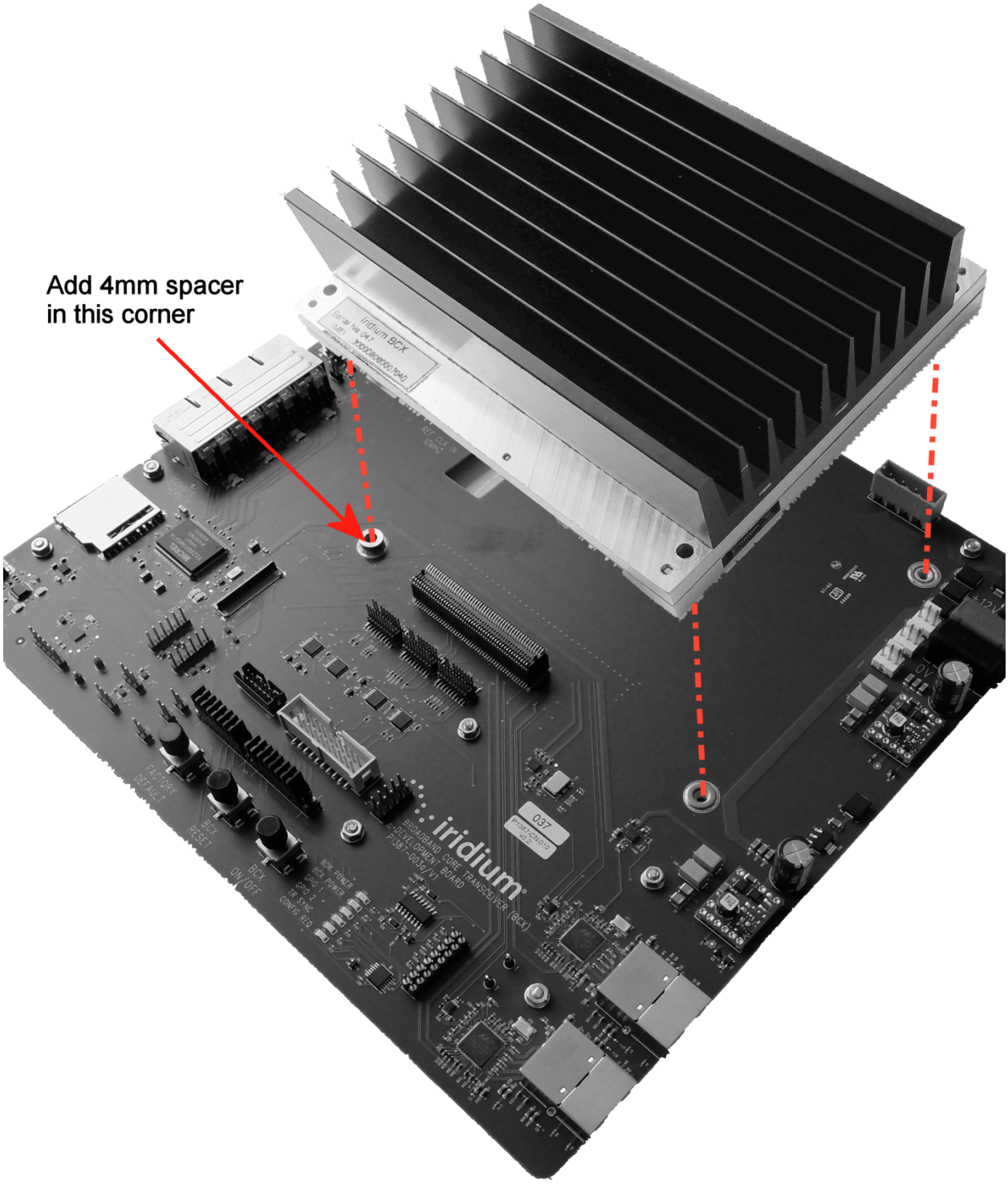


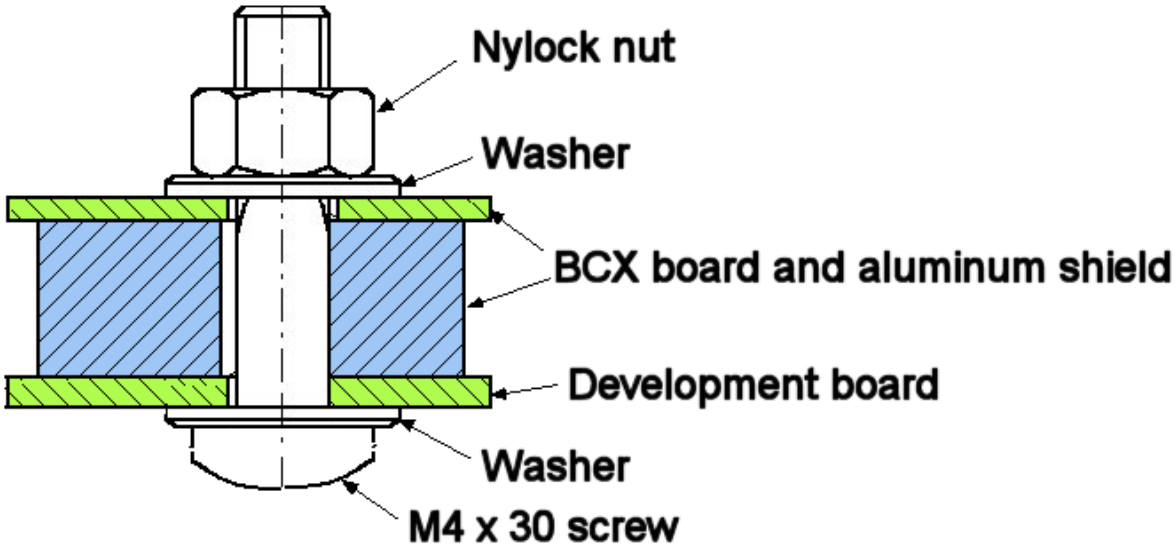
c. Close the black tab to lock the cable into place



2. Screw the BCX board to the Development Board as shown below, using the M4 screws, washers and nuts in the order shown. Take care to add the 4mm spacer in the location indicated, to reduce stress on the Development

Board as the screws are tightened.





3. Insert the flat-flexi cable into the connector on the Development Board, again ensuring that the blue side is facing upwards and that the ears are properly engaged before closing the latch.

6 Electrical Interfaces

The following subsections contain information on the electrical interfaces for the connections as shown in Figure 3 for the BCX Development Kit.

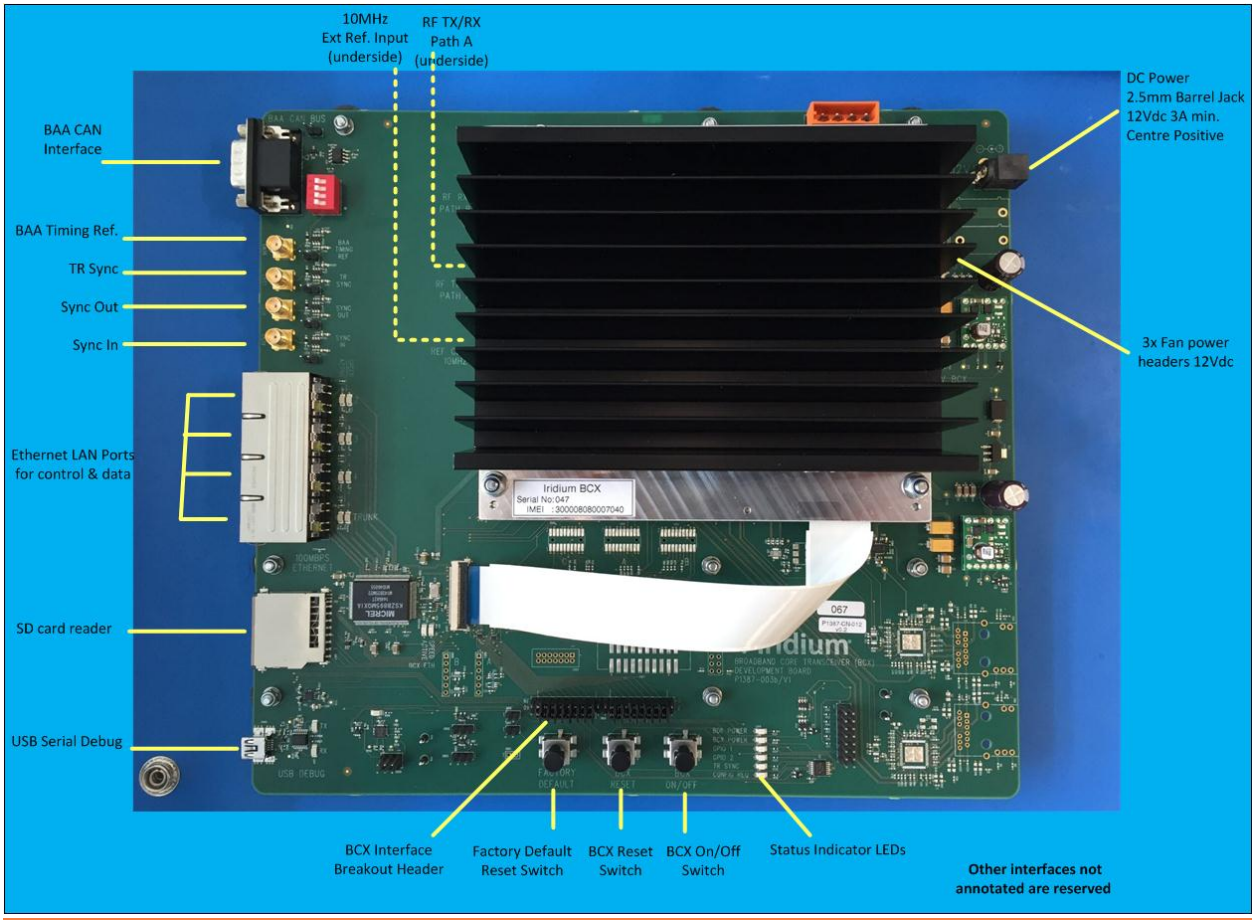


Figure 3 Integrator Development Kit

6.1 Power Supply and Control

The power supply for BCX Development Board is provided using an external DC power supply block external to the board stack. The BCX Development Board power required from the external power supply are given in Table 2.

Parameter	Value
Input Voltage	12.0 V DC (typ)
Input Current	3.0 A DC (min)

Table 2 DC Power Input Specifications for BCX Development Board

6.1.1 Power Supply Connection

The DC power connector on the BCX Development Board is a standard 2.5mm DC jack socket within its central pin connected to +12V and the outer ring at 0V.

6.1.2 Power On/Off Control

Once the external power source is applied to the IDK, the IDK enters its standby state which is shown by lighting the green 'BDK Power' LED. This shows that external power has been applied. By default, in this standby state the BCX module is held in its off state.

Pressing the 'BCX On/Off' switch powers up the BCX, as shown by lighting the blue 'BCX Power' LED. The BCX will power-on and if it contains valid software it will start its boot sequence, as indicated in Table 6. The booting process typically takes about 1 minute.

6.1.3 Reset Control

The BCX can be manually reset by pressing the 'BCX Reset' button on the BCX Development Board. On pressing and then releasing the BCX reset switch, the BCX will begin to reboot itself.

6.1.4 Factory Default Reset

It is possible to store a non-working IP network address configuration in the BCX in error. If so the user can recover the BCX back to its factory default settings, by depressing and holding the 'Factory Default' switch for approximately 5 seconds. On releasing this button the BCX will reset its internal settings to their default values, and then it will reboot itself.

6.2 Synchronization Interfaces

The BCX Development Board has a number of hardware synchronization signal interfaces.

6.2.1 External Frequency Reference

An external 10MHz frequency reference signal can be applied to the BCX using the REF IN MCX connector on the BCX unit itself. This allows the BCX to be frequency locked to an external master frequency reference.

Parameters for the BCX REF IN interface are summarized in the table below.

Parameter	Value
Connector type	MCX female
Input Frequency	10.0 MHz
Input AC Impedance	>100Ω
Assumed source impedance	50Ω
Input voltage range at connector	>0.5Vpp

Table 3 EXT REF IN Interface Parameters

6.2.2 Timing Synchronisation Input (SYNC IN)

The SYNC IN interface provides a high impedance 3.3V TTL compatible input via a female SMA connector on the BCX Development Board and may be used for test and synchronisation purposes.

When in test mode the BCX can be commanded to use the external SYNC IN signal as a 90ms frame-time reference from an external source, rather than using its own internal reference. This allows the BCX to be time-aligned to an external timing reference, which can be useful for some laboratory tests.

6.2.3 Timing Synchronisation Output (SYNC OUT)

The SYNC OUT interface provides a low impedance 3.3V TTL compatible output via a female SMA connector.

Operation of the SYNC OUT signal on the BCX Development Board is currently undefined.

6.3 Communication Interfaces

This section describes the physical characteristics of the BCX Development Board communication interfaces.

6.3.1 Local Area Network Interface (LAN)

A Local Area Network connection is provided via a 4x bank of RJ45 connector on the left hand side of the board. The Ethernet interface is compatible with 100Base-T devices.

The physical Ethernet interface is used both for controlling the BCX, and for sending/receiving data communications traffic to/from the BCX. The network interface itself logically separates the flows into OAM control plus two data flows by allocating them to specifically tagged VLANs.
Details of this VLAN split are given in references [1] & [2].

The BCX Development Board provides an on-board switch to route different VLAN traffic to different physical ports on the board. Details of the default port and VLAN configuration provided on the board are shown in Figure 4.

- VLAN-ID 2 for OAM Control messages is mapped to physical port A
- VLAN-ID 3 for Postpaid data traffic is mapped to physical port B
- VLAN-ID 4 for Prepaid data traffic is mapped to physical port C
- LAN TRUNK for all traffic is mapped to physical port D

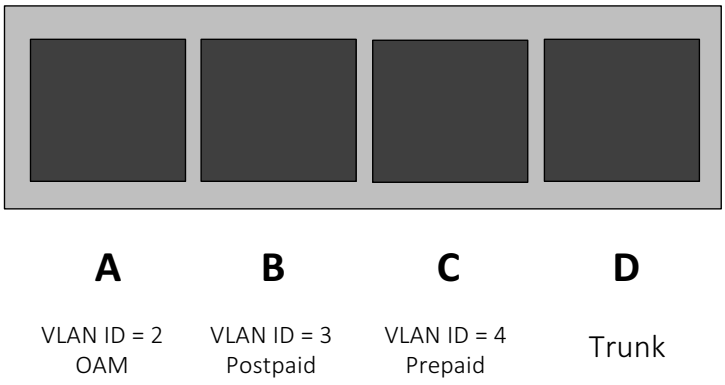


Figure 4 - Physical Network Port mapping showing VLAN Routing

6.3.2 BCX RF Interfaces for Transmit/Receive

There are two BCX RF signals provided for cabled connection to laboratory test system.

- Path A is a bidirectional TX/RX connection, to be used as the main RF connection
- Path B is a unidirectional RX connection, to be used for downlink scanning on directional antennas

The RF interface parameters are summarized in the table below.

Parameter	Value
Frequency Range	1616 MHz to 1626.5 MHz
RF input range	Approx. -110 to -70 dBm
RF output level (Path A only)	Approx. 0 dBm
Input/Output Impedance	50Ω
Connector type	MCX female

Table 4 RF Interface Parameters

NOTE: The RF connection on the BCX is intended for cabled connection to a laboratory test system and should not be directly connected to an external antenna.

6.3.3 BAA CAN-bus Interface

In order to connect to an external Broadband Platform Active Antenna, a 9-way D-sub-miniature connector is provided on the BCX Development Board. This interface uses the CAN-bus standard, plus a timing reference signal, to communicate between the BAA and BCX. Further signals can be switched onto the pins of the connector using the switch bank located adjacent to the D-sub-miniature connector (reference designator SW4).

Pin Number	Signal	SW4 Number
1	FACTORY_RESET	1
2	CANL	-
3	GND	-
4	BAA_TIMING_REF_BUF	2
5	Shield	-
6	BCX_TO_BAA_D2	3
7	CANH	-
8	BAA_TO_BCX_D2	4
9	Reserved for future use	-

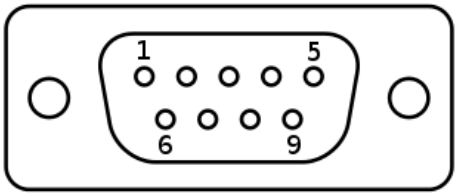


Table 5 BAA CAN-bus Interface Connector Pinout

Further information regarding this connection is found in the BCX to BAA Interface Control Document [3].

6.3.4 Debug USB Connector

A debug serial port is provided via mini USB type B connector on the IDK motherboard and operates at a speed of 115200 baud.

Although this debug interface is not needed for normal operation, it can be used for basic text diagnostics.

6.3.5 IDK Status LEDs

The BCX Development Board provides a set of 6 status LEDs on the front edge of the board relating to the status of the unit. This set of status indicators is shown in Figure 5.

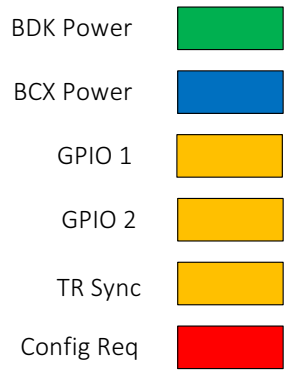


Figure 5 Indicator LEDs

'BDK power' indicates when +12Vdc power is present on the BCX Development Board.

'BCX power' indicates when the BCX module has been powered on.

The remaining status LEDs indicate the current operating state of the BCX as defined in Table 6.

BCX STATE		LEDS				
State	Sub State	BCX POWER	STATUS 1 (GPIO1)	STATUS 2 (GPIO2)	STATUS 3 (TR_SYNC)	STATUS 4 (CONF REQ)
NO POWER	-	OFF	OFF	OFF	OFF	OFF
STARTUP	Booting/self-test	ON	OFF	OFF	OFF	OFF
	Fault	ON	OFF	OFF	OFF	ON
INACTIVE	Standby	ON	OFF	ON	ON	ON after a successful power-on/boot ON/OFF as set by "bootState" command
TEST	PHY testing	ON	ON	ON/OFF Flashing with RX activity	ON/OFF Flashing with TX activity	ON/OFF as set by "bootState" command

Table 6 BCX Power-up sequence shown on status LEDs

6.4 Start-up & Shutdown

After the IDK has been plugged into its mains power supply, the IDK power LED will illuminate on the front-panel.

BCX start up is triggered by pressing the BCX ON/OFF switch.

On start up the BCX will go through the sequence shown in Figure below, reaching its “INACTIVE” state, where it ready to be controlled by the host test system. Further state transitions to either ACTIVE or TEST modes are commanded by the host test system, using the Ethernet interface.

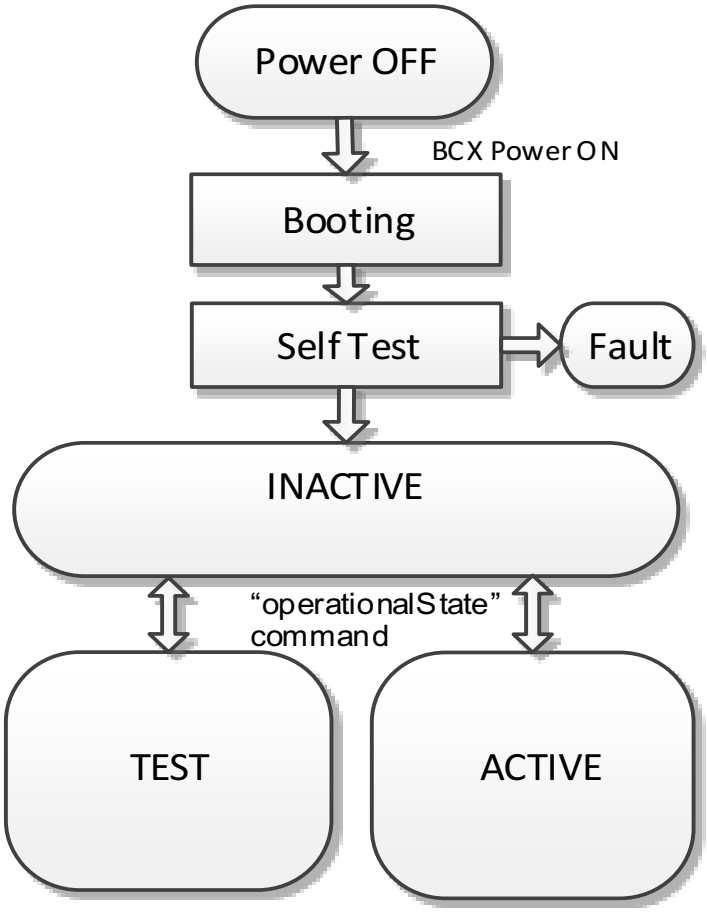


Figure 6- BCX Boot Sequence

6.4.1 Verify basic connectivity

In order to verify basic connectivity, the following command will get the operational state of the BCX. It assumes that you have ‘curl’ installed on your computer. For Windows users, you will need to install cygwin & curl.

```
$ curl  http://169.254.20.23:9091/v1/operationalState
{"state":"inactive"}
```

6.4.2 Verify basic transmit operation

This section assumes you have Python 2.7 installed as well as the http requests python package.

You can get python at: https://www.python.org/downloads

Installing the http module can be done using: python –m pip install requests

Use the following python script to get a TX waveform…

```
#!/usr/bin/env python
#!C:\Python27\pythonw.exe
#
# this is the BCX Hello World TX test
#
# install python 2.7
# https://www.python.org/downloads/
#
# install the http drivers
# C:\Python27\python.exe  -m pip install requests
#
#
# //Copyright (c) 2015 Iridium Communications Limited
#
import requests
import json
import time
headers = {'content-type': 'application/json'}
#Get the current state
url = 'http://169.254.20.23:9091/v1/operationalState'
data = requests.get(url, headers=headers)
print "-Current State: %s" %data.json()

#set to Test mode
payload = {'state':'test'}
data1 = requests.put(url, data=json.dumps(payload) , headers=headers)
print "-New State: %s" %data1.json()

#choose no BAA connected Transmit test mode
cmd = 'http://169.254.20.23:9091/v1/testContinuousTx'
payload = {'channel_number':'121'  , 'transmit_power':'300'  , 'apv_index':'3'}
data2 = requests.put(cmd, data=json.dumps(payload) , headers=headers)
print "-Start TX: %s" %data2._content

#set to Test idle
time.sleep(60)
payload = {'state':'inactive'}
data3 = requests.put(url, data=json.dumps(payload) , headers=headers)
print "-End State: %s" %data3.json()
```

The output of the BCX RF Path A should look like the screen capture shown below.