IOTYS
Internet Of Things You Search

IOTCape

User manual

TABLE OF CONTENT

1 Introduction 5

1.1 Scope 5

1.2 Product description 5

2 Safety 6

3 Hardware guide 7

3.1 Board presentation 7

3.2 Block diagram 8

3.3 Pins usage 8

3.4 Hardware Features 9

3.4.1 Power Supply 9

3.4.2 Modem 9

3.4.3 SIM Card interface 10

3.4.4 BlueTooth 10

3.4.5 USB 10

3.4.6 EEPROM 10

4 Software guide 12

4.1 System 12

4.2 Init 12

4.3 GPIOs 12

4.4 Modem 12

4.5 Bluetooth 14

4.5.1 Install Service Pack 14

4.5.2 Install Bluez 14

4.5.3 Using Bluez 15

5 Technical specifications 17

FIGURES

Figure 1 : Board presentation 7

Figure 2 : Block Diagram 8

Figure 3 : Pins used 8

Figure 4 : Switch antenna 10

DOCUMENT HISTORY

Revision	Date	Description
1.0	2015-03-20	First release

1 INTRODUCTION

1.1 Scope

Scope of this document is to present the features of IOTCape.

It provides a description of hardware and software features to use the product with BeagleBone and Linux.

1.2 Product description

IOTCape is a BeagleBone Cape for IOT which can be used with white and black versions of BeagleBone.

With its connectivity capabilities, the IOTCape easily transforms your BeagleBone in an IOT gateway prototype.

By adding a MiniPcie compatible modem on this cape, you can connect your gateway to internet everywhere. No modem choice is decided by IOTYS, you can choose the modem solution according to the specific requirements of your application.

This cape provides also a BlueTooth up to date dual-mode (2.1 BDR/EDR, LE 4.1) link with its embedded CC2564MODN module. This connectivity link can be used to:

- manage BlueTooth Low Energy devices
- configure the gateway by smartphone

IOTCape provides also two USB 2.0 host ports to add some supplementary USB devices (mass storage, ZigBee, ZWave, etc.).

2 SAFETY

The Products shall be used for engineering or laboratory purpose only and shall never be considered End User Product.

Make sure that the user of the Product has received sufficient electronics training and observe good engineering practice standards.

User must read BeagleBone’s manual and follow voltage and temperature instructions before using the IOTCape.

3 HARDWARE GUIDE

3.1 Board presentation

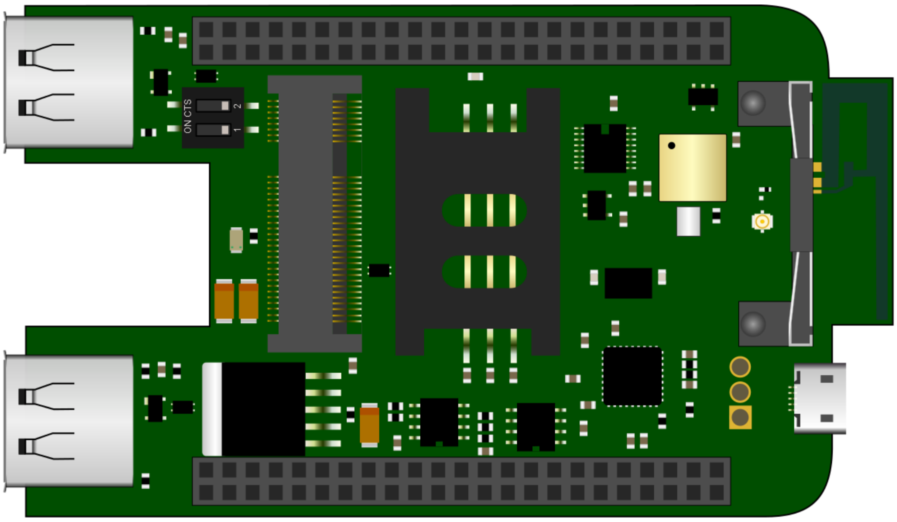


Figure 1 : Board presentation

The Cape embeds:

- One emplacement for MiniPci Express card for modem
- A SIMCard connector with 6 pins
- A Bluetooth module
- Two USB connectors
- One switch to configure I2C address of embedded cape EEPROM
- Two 46 connectors for stacking another cape

All components are soldered on the top of board to avoid any contact with the lower part of the BeagleBone board.

3.2 Block diagram

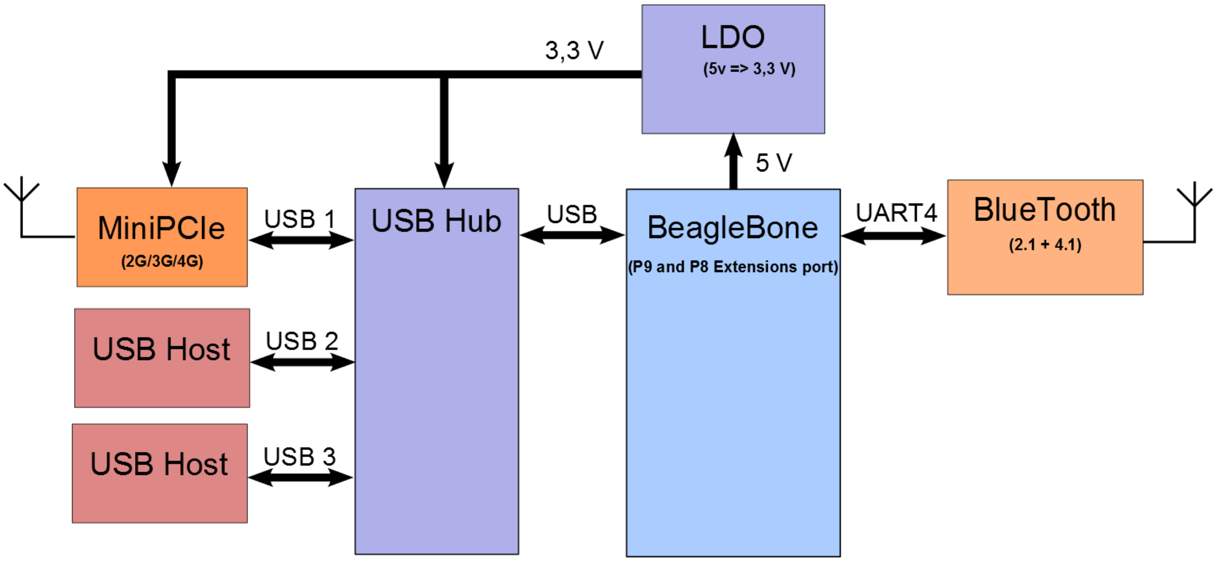


Figure 2 : Block Diagram

3.3 Pins usage

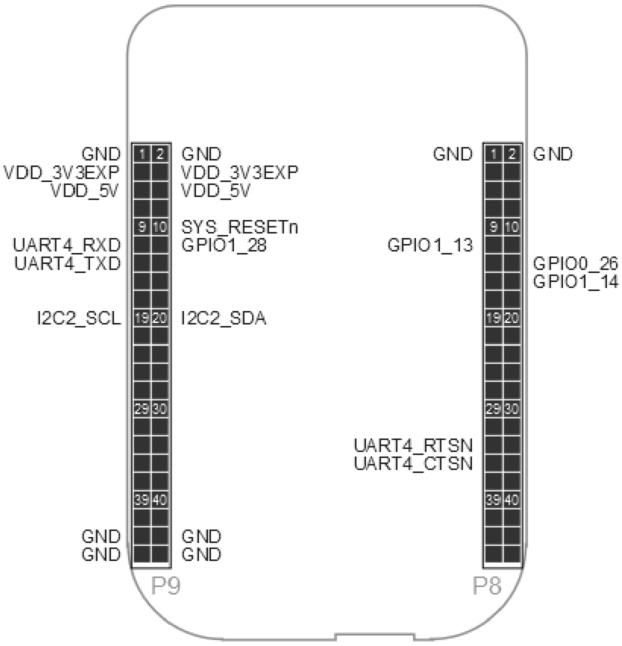


Figure 3 : Pins used

PIN Name	Header	Pin Number	Pin Usage
GPIO1_13	8	11	Modem reset
GPIO0_26	8	14	Disable Modem (LowPower)
GPIO1_14	8	16	Shutdown Bluetooth
UART4_RTS	8	33	RTS serial used by BlueTooth
UART4_CTS	8	35	CTS serial used by BlueTooth
SYS_RST	9	10	System Reset of BeagleBone
UART4_RXD	9	11	RXD serial used by BlueTooth
GPIO1_28	9	12	Indicate the connection state of modem
UART4_TXD	9	13	TXD serial used by BlueTooth
I2C2_SCL	9	19	I2C clock for EEPROM Cape
I2C2_SDA	9	20	I2C data for EEPROM Cape

3.4 Hardware Features

3.4.1 Power Supply

The Cape requires an external 5V power supply provided by BeagleBone on VDD_5V with at least 2A to enable MiniPcie and USB functions.

However, the user can program and use BlueTooth functionality connecting only the usb mini connector on BeagleBone.

3.4.2 Modem

The modem is connected to USB and power supply through an LDO 5V => 3.3 V. Then a 5V external power supply is required to power the BeagleBone as well as USB link between BeagleBone and IOCape.

No modem choice is decided by IOTYS. User can choose any MiniPcie modem which is compliant with the standard Mini PCI express connector.

However, for your convenience, IOTYS have tested this modems:

Manufacturer	Model
Sierra Wireless	MC7710
	MC7304
Telit	HE910_MiniPcie
Huawei	M909u-521
	MU609
	MU709s-2p

Some MiniPcie modem provides also GPS functionality, refer to the modem user manual to use it.

3.4.3 SIM Card interface

IOTCape embedded a 6 pins SIM card connector compliant with the 3GPP standards.

3.4.4 BlueTooth

The BlueTooth module provides Dual-Mode connectivity to BeagleBone via an HCI interface.

The power supply is provided through VDD_3V3EXP from BeagleBone.

BlueTooth module uses the UART4 serial link with control flow mechanism (RTS/CTS).

By default, the Bluetooth module is connected to the PCB antenna mounted on the cape.

If user wants to integrate the cape in a metal case, the use of an external antenna is recommended.

In this case, IOTCape provides also a UFL connector. The use can switch to the UFL connector unsoldering the R24 resistor and soldering it to R23 place.

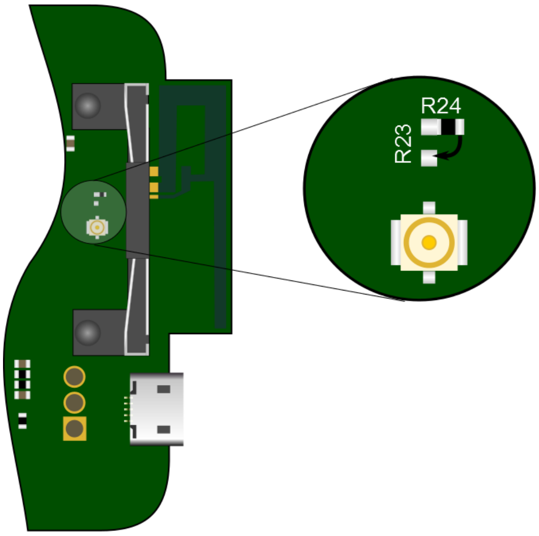


Figure 4 : Switch antenna

3.4.5 USB

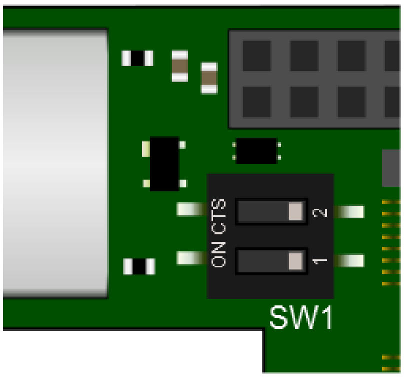
IOTCape embeds a Microchip USB2513 three port USB 2.0 hub supporting LS/FS/HS USB connections. The USB hub is used to provide additional USB ports for the Beaglebone Black while allowing for direct communication between the Beaglebone Black and cellular modem.

Connect the Beaglebone’s P3 USB-A port to the mini USB IOTCape connector.

3.4.6 EEPROM

IOTCape embeds EEPROM containing the boards EEPROM information per the Beaglebone.

The address of the EEPROM is set via switch as shown below.



Address	1	2
0x54	OFF	OFF
0x55	ON	ON
0x56	OFF	OFF
0x57	ON	ON

4 SOFTWARE GUIDE

4.1 System

IOTCape has been tested on BeagleBone with Linux 3.8.14 and Debian wheezy 7.8. The implemented examples below were made with these versions.

4.2 Init

To be compliant with the standard configuration of BeagleBone Capes, IOTYS provides a DTBO binary device tree file **BB-IOTCape-00A0.dtbo** available for download at **www.iotys.com**.

Copy this file under **/lib/firmware** and reset the board.

According to the EEPROM information, the system loads the file and configures the BeagleBone according to the IOTCape requirements.

4.3 GPIOs

Like describe before IOTCape uses some GPIOs for modem and BlueTooth:

PIN Name	Header	Pin Number	I/O	Pin Usage
GPIO1_13	8	11	Output	Modem reset
GPIO0_26	8	14	Output	Disable Modem (LowPower)
GPIO1_14	8	16	Output	Shutdown Bluetooth
GPIO1_28	9	12	Input	Indicate the connection state of modem

The file BB-IOTCape-00A0.dtbo provided by IOTYS on web site configures the GPIO in output or input.

However, the user has to set the GPIO level for outputs:

```
echo 1 > /sys/class/gpio/gpio46/value

echo 1 > /sys/class/gpio/gpio26/value

echo 1 > /sys/class/gpio/gpio45/value
```

Remark: Do not forget to set these outputs before using the IOTCape.

To set automatically the GPIO outputs, the user can define and add a service which sets these at startup (see : **update-rc.d**).

4.4 Modem

The following examples are based on using a MC7710 Sierra Wireless modem.

The modem is detected at startup by Linux.

You can verify its presence as show below by typing:

```
lsusb
```

The system will show you the USB devices:

```
Bus 001 Device 002: ID 0424:2513 Standard Microsystems Corp.

Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

Bus 001 Device 003: ID 1199:68a2 Sierra Wireless, Inc.
```

In this example, a Sierra Modem MC7710 is installed on IOTCape.

To verify the proper operation of the modem, install minicom:

```
apt-get install minicom
```

Test the system to find out what ttyUSB are created by modem:

```
ls /sys/bus/usb-serial/devices/
```

The system will show you the ttyUSB serial port. For example on MC7710:

```
ttyUSB0  ttyUSB1  ttyUSB2
```

Refer to modem constructor documentation to know which ttyUSB can receive the AT commands.

Launch minicom on the right ttyUSB port. For example on MC7710:

```
minicom –D /dev/ttyUSB2
```

Minicom displays this screen:

```
Welcome to minicom 2.6.1


OPTIONS: I18n
Compiled on Feb 11 2012, 18:45:56.
Port /dev/ttyUSB2


Press CTRL-A Z for help on special keys
```

Type AT command, the modems says OK:

```
at
OK
```

To establish a TCP/IP connexion with modem install and use ppp (see https://wiki.archlinux.org/index.php/3G_and_GPRS_modems_with_pppd).

4.5 Bluetooth

The bluetooth chipset embedded on IOTCape is a dual-mode (BR/EDR/BLE) CC2564MODN which provides a HCI interface on UART4.

At startup, the device tree file automatically configures the UART4 on **/dev/ttyO4**.

4.5.1 Install Service Pack

Before, using Bluez, you need to download and install the ServicePack V1.1 for CC2564MODN.

Follow these instructions:

```
wget http://processors.wiki.ti.com/images/7/7b/CC2564B_BT_BLE_SP_BTS.zip
unzip CC2564B_BT_BLE_SP_BTS.zip
cd CC2564B_BT_BLE_SP_BTS
cp bluetooth_init_cc2564B_1.1_BT_Spec_4.1.bts  /lib/firmware/ti-connectivity/TIInit_6.7.16.bts
```

4.5.2 Install Bluez

BlueZ provides support for the core Bluetooth layers and protocols.

However, due to a bug of timeout of hciattach tool, you need to recompile the package before installing.

Follow these instructions:

1. Install the compiling tools

```
apt-get update

apt-get install make gcc sudo glib2.0-dev libdbus-1-dev libudev-dev libical-dev libreadline-dev
```

2. Get Bluez sources and untar them

```
wget http://www.kernel.org/pub/linux/bluetooth/bluez-5.29.tar.xz

tar -xvf bluez-5.29.tar.xz
```

3. Edit hciattach_ti.c to increase the value of timeout response for commands from 15 to 150.
    a. Edit file (**vi tools/hciattach_ti.c**)
    b. Modify the line 256 :
       if (hci_send_req(fd, &rq, ***15****) < 0)* {
       to *if (hci_send_req(fd, &rq,* ***150****) < 0)* {
    c. Save file
4. Configure makefile, build Bluez and install it.

```
./configure

make install

reboot
```

5. Restart the system

```
reboot
```

4.5.3 Using Bluez

To start Bluez with IOTCape, follow these instructions:

1. Release the shutdown pin of Bluetooth chipset

```
echo 1 > /sys/class/gpio/gpio46/value
```

2. Attach the ttyO4 to bluetooth:

```
hciattach -t 60 /dev/ttyO4 texas 115200
```

The console system displays:

```
Found a Texas Instruments' chip!
Firmware file : /lib/firmware/ti-connectivity/TIInit_6.7.16.bts
Loaded BTS script version 1
texas: changing baud rate to 115200, flow control to 1
Device setup complete
```

3. Use **hciconfig** to bring up the associated USB device manually:

```
hciconfig hci0 up
```

4. Use **hcitool** to discover the Bluetooth-enabled devices within range:

```
hcitool scan
```

The console system shows you the discovered devices. For example an android smartphone with **30:75:12:C9:A2:64** address.

```
Scanning ...
    30:75:12:C9:A2:64    Xperia Z3 Compact
```

5. Use **hcitool** to discover the Bluetooth-LowEnergy devices within range:

```
hcitool lescan
```

The console system shows you the discovered devices. For example a **SensorTag** from Texas Instrument with **BC:6A:29:AE:CC:D4** address.

```
LE Scan ...
BC:6A:29:AE:CC:D4 SensorTag
```

5 TECHNICAL SPECIFICATIONS

Description	**Value**
Power Supply	5V (via expansion header)
	3.3V (via expansion header)
Max current usage on 5V	1 A
Max current usage on 3.3V	150 mA
BlueTooth Tx. power Level (max.)	10 dBm
Indicators	one led for modem status
Connectors	Two USB 2.0 Host
	One micro USB 2.0 OTG
	Two 46-positions headers for P8 and P9
Switch	I2C EEPROM selector
Size	95mm x 56mm
Layers	4
PCB Thickness	1.6 mm