

# USER'S MANUAL

FOR

**Interfacing of lift elevator with  
BeagleBone Black**

## Introduction:

The element14 BeagleBone Black is identical in technical design and functionality as the specified BeagleBoard.org product (BeagleBone Black) and runs on the version of the software provided by BeagleBoard.org to element14. General support for this board is available from the BeagleBoard.org community.

## Setup:

1. Connect beaglebone black to PC with +5v supply and USB cable.
2. Open terminal in ubuntu.
3. Type `sudo su` ---enter  
Type password--enter
4. Then type `minicom -s`---enter
5. Goto serial port setup-□
  - Press A
  - Change `/dev/ttyUSBxx` to `/dev/ttyACM0`—enter
  - Press G
  - Press Enter
6. Save setup as dfl – enter
7. Goto EXIT – enter
8. Now system will boot and then type username and password and you booted into beaglebone black.
9. Type `su` –enter.

## CONNECTOR DETAILS:

Connector Number	PIN Number	PIN Description	PIN connects to board	Function
P9	1,2	GND	GND(25pin)	GND
P9	3,4	VCC(3.6V)	VCC(26pin)	VCC
P8	7	GPIO2[2]	FRC-6pin	In
P8	8	GPIO2[3]	FRC-5pin	In

P8	9	GPIO2[5]	FRC-4pin	In
P8	10	GPIO2[4]	FRC-3pin	In
P8	11	GPIO1[13]	FRC-13pin	Out
P8	12	GPIO1[12]	FRC-14pin	Out
P8	13	GPIO0[23]	FRC-11pin	Out
P8	14	GPIO0[26]	FRC-12pin	Out
P8	15	GPIO1[15]	FRC-9pin	Out
P8	16	GPIO1[14]	FRC-10pin	Out
P8	17	GPIO0[27]	FRC-7pin	Out
P8	18	GPIO2[1]	FRC-8pin	Out
P8	19	GPIO0[22]	FRC-23pin	Out
P8	21	GPIO1[30]	FRC-24pin	Out
P9	11	GPIO0[30]	FRC-21pin	Out
P9	12	GPIO1[28]	FRC-22pin	Out
P9	13	GPIO0[31]	FRC-19pin	Out
P9	14	GPIO1[18]	FRC-20pin	Out
P9	15	GPIO1[16]	FRC-17pin	Out
P9	16	GPIO1[19]	FRC-18pin	Out
P9	23	GPIO1[17]	FRC-16pin	Out

P9	24	GPIO0[15]	FRC-15pin	Out
----	----	-----------	-----------	-----

TABLE 1

### How to get GPIO pin number:

- Once you have decided the pin number which you would like to use as a GPIO, you need to find out its corresponding reference number.
- For example, if you would like to use pin 12 on P8 expansion header, Then find out its default function. Note down the entire signal name. In this case, pin 12 is GPIO1\_12. So any GPIO you come across would be referenced as GPIOX\_Y. Identify X,Y.
- Use the formula below to find the corresponding reference number:  
**Reference number = ( X\*32 ) + Y )**  
Hence, pin 12 would be referenced as gpio 44 in the kernel.

### To use GPIO pin as GPIO in programs follow the steps :

- To make GPIO pin xx as output type command in terminal(xx—pin number)  
**echo xx > /sys/class/gpio/export -- press enter**  
**echo out > /sys/class/gpio/gpioxx/direction --press enter**  
for example:  
echo 44 > /sys/class/gpio/export  
echo out > /sys/class/gpio/gpio44/direction
- To make GPIO pin xx as input type command in terminal(xx—pin number)  
**echo xx > /sys/class/gpio/export -- press enter**  
**echo in > /sys/class/gpio/gpioxx/direction --press enter**  
for example:  
echo 44 > /sys/class/gpio/export  
echo in > /sys/class/gpio/gpio44/direction
- After making GPIO pin as output, To change the initial value of output pin type command in terminal(x—1/0,xx—pin number)  
**echo x > /sys/class/gpio/gpioxx/value --press enter**

for example:

```
echo 0 > /sys/class/gpio/gpio44/value
```

**OR**

```
echo 1 > /sys/class/gpio/gpio44/value
```

### **PROCEDURE:**

1. Make a connection as shown in above Table1.
2. Make pin as input or output as told in function block of Table1.

#### **Export pins :**

```
echo 45 > /sys/class/gpio/export
echo 44 > /sys/class/gpio/export
echo 23 > /sys/class/gpio/export
echo 26 > /sys/class/gpio/export
echo 47 > /sys/class/gpio/export
echo 46 > /sys/class/gpio/export
echo 27 > /sys/class/gpio/export
echo 22 > /sys/class/gpio/export
echo 62 > /sys/class/gpio/export
echo 65 > /sys/class/gpio/export
echo 66 > /sys/class/gpio/export
echo 67 > /sys/class/gpio/export
echo 68 > /sys/class/gpio/export
echo 69 > /sys/class/gpio/export
echo 30 > /sys/class/gpio/export
echo 60 > /sys/class/gpio/export
echo 31 > /sys/class/gpio/export
echo 50 > /sys/class/gpio/export
echo 48 > /sys/class/gpio/export
echo 51 > /sys/class/gpio/export
echo 15 > /sys/class/gpio/export
echo 49 > /sys/class/gpio/export
```

#### **Set pin direction:**

```
echo out > /sys/class/gpio/gpio45/direction
echo out > /sys/class/gpio/gpio44/direction
echo out > /sys/class/gpio/gpio23/direction
echo out > /sys/class/gpio/gpio26/direction
echo out > /sys/class/gpio/gpio46/direction
echo out > /sys/class/gpio/gpio47/direction
```

```
echo out > /sys/class/gpio/gpio27/direction
echo out > /sys/class/gpio/gpio65/direction
echo out > /sys/class/gpio/gpio22/direction
echo out > /sys/class/gpio/gpio62/direction
echo out > /sys/class/gpio/gpio30/direction
echo out > /sys/class/gpio/gpio60/direction
echo out > /sys/class/gpio/gpio31/direction
echo out > /sys/class/gpio/gpio50/direction
echo out > /sys/class/gpio/gpio48/direction
echo out > /sys/class/gpio/gpio51/direction
echo out > /sys/class/gpio/gpio15/direction
echo out > /sys/class/gpio/gpio49/direction
echo in > /sys/class/gpio/gpio66/direction
echo in > /sys/class/gpio/gpio67/direction
echo in > /sys/class/gpio/gpio68/direction
echo in > /sys/class/gpio/gpio69/direction
```

3. Make New folder with name Lift elevator and go to that folder using command  
**mkdir lift\_elevator—press enter**  
**cd lift\_elevator—press enter**
4. Create and open a file with command  
**vim lift\_elevator.cpp**
5. Write a program and save it.
6. Now compile and run the program using command  
**g++ lift\_elevator.cpp -o lift\_elevator —press enter**  
**./lift\_elevator —press enter**