Black is the colour

Simplifying real world interfacing using a Linux based development board.

By **Kyle Borgerson**.

Embedded microcontroller development boards are a popular, low cost way of getting to know an mcu architecture. Meanwhile, 8bit boards, such as Arduino, provide an easy way of getting to know the basics of programming and the components needed to interact with the real world.

Such development kits and evaluation boards have been available to professional developers for years, but the benefit of popular, yet simple, mcu boards to professional developers has always been in doubt. However, more capable, open source boards have started to appear and are gaining support from professional developers.

Based on a microprocessor, rather than an mcu, these newer development platforms are typically Linux based. One such example is TI's BeagleBone Black. It not only offers a platform for prototyping Linux applications, but also comprehensive IO support, making it perfect for designs that interact with the real world. The BeagleBone board format, initially released in 2011, not only squeezes in all of the capabilities of earlier BeagleBoard systems into a credit card sized package, but has also established a standard footprint of two dual row 46pin connectors for expansion modules called 'capes'. Similar to the Arduino 'shields', they support a variety of plug in boards to add more advanced I/O.

BeagleBone Black features TI's Sitara AM3359, an ARM Cortex-A8 microprocessor running at 1GHz. There is also 2Gbyte of flash and 512Mbyte of DDR3 at 400MHz. A micro D type HDMI connector, Ethernet and USB ports are included and the board is powered by a 5V dc supply. The board can also be USB powered, since it consumes no more than 250mA.

From the software perspective, the board comes preloaded with a range of software and is ready to boot. Once powered up, the board boots the Angstrom Linux distribution, after which the Gnome desktop appears. During the boot process, the four user LEDs (USR0 to 3) will flash reassuringly to indicate activity. In addition to Linux, the Black can also run Ubuntu or Android.

Most developers are likely to have an IDE that supports the TI Sitara A8, such as those from IAR Systems or Green Hills, but TI's Code Composer IDE is also a natural choice. However, a number of tools are supplied with the BeagleBone Black that allow you to become familiar with the board.

Preloaded development tools include a Python interpreter and C/C++ compiler, along with a local replica of the Cloud9 IDE preconfigured to run Node.js. Also included is the Bonescript library, based on Node.js, which provides a number of Arduino like functions for interfacing with the hardware. Readers familiar with Arduino's 'digitalWrite' function will recognise this and similar functions included within Bonescript. The beagleboard.org community resource also serves as a useful repository of example projects, helpful forums and hardware/software documentation.

These tools, and the capability to use the board's extensive GPIO, earn respect from professional embedded developers. The device has 92 pins accessible via the two dual row headers P8 and P9. These headers also form the connections to the capes. These pins can have many different possible functions, from

Borgerson: "With its open source approach, the BeagleBone Black provides a cost effective and well documented route for prototyping a commercial design."

Fig 1: The BeagleBone Black's major components

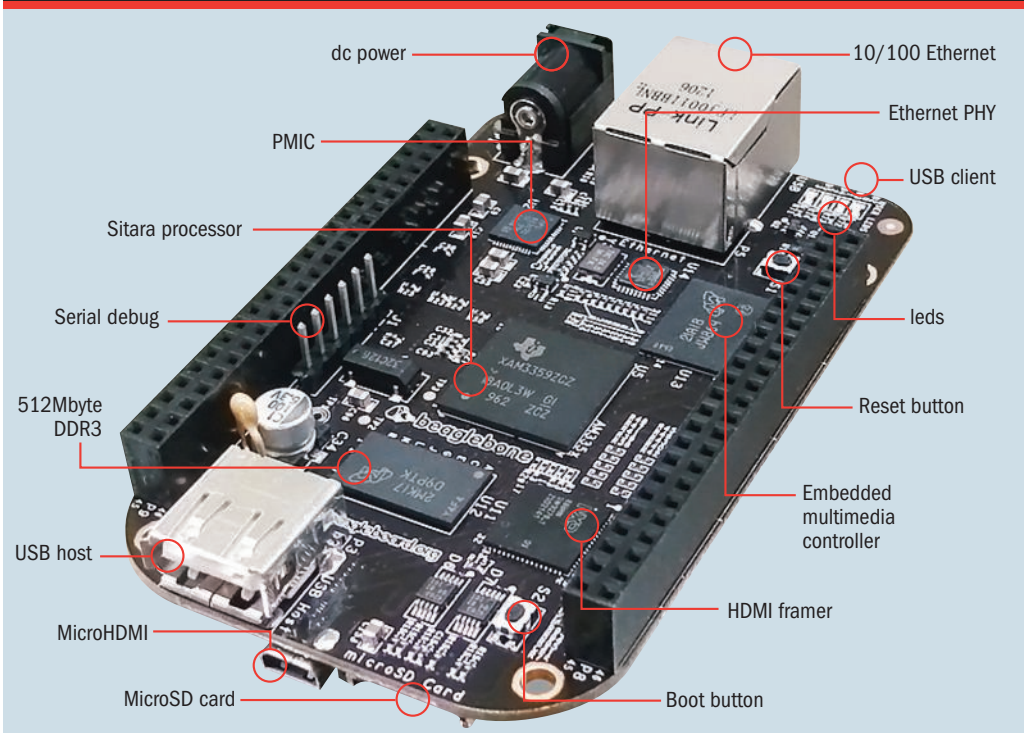

Fig 2: BeagleBone Black block diagram

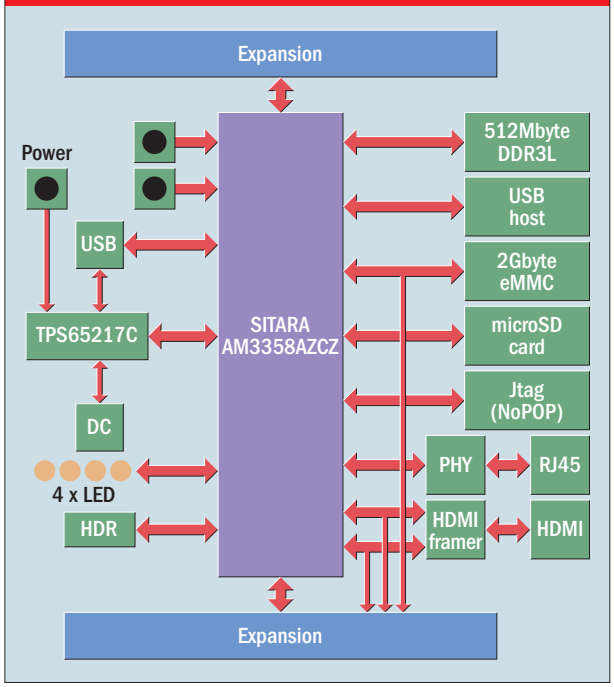

Fig 3: The 'blinked.js' code example

```
require('bonescript');
ledPin = bone.USR3;
setup = function () {
        pinMode(ledPin, OUTPUT);
};
loop = function () {
        digitalWrite(ledPin, HIGH);
        delay(1000);
        digitalWrite(ledPin, LOW);
        delay(1000);
};
```

controlling I/O and reading sensors to driving leds. Third party capes provide everything from a simple breadboard area and an lcd screen to a cape which can control underwater vehicle projects. Up to four capes can be stacked on top of each other, as long as there are no GPIO conflicts.

The easiest way of experimenting with this GPIO is to use the Cloud9 IDE. Cloud9 starts automatically at boot time and is accessed using the Black's web server. The Epiphany browser finds the IDE automatically on start, but any browser can be pointed to port 3000. The Black's web server also provides a set of pages that gives access to the Cloud9 IDE and some Bonescript code examples.

Cloud9 provides a number of simple examples written in node.js JavaScript and incorporating the Bonescript library. The 'blinked.js' code example (see fig 3) toggles the USR3 led, but this can be extended to use one of the GPIO pins by connecting an led and associated pull up resistor to the desired pin and by changing the assignment of ledPin to the relevant GPIO, for example bone.P8_3. An entry level IDE, Cloud9 provides a quick and easy way of writing short code projects and then running and debugging them.

The use of node.js JavaScript appears to be the preferred way of programming the BeagleBone Black; it certainly serves as an easier introduction for those unfamiliar with programming or higher level languages, or simply as a way of pulling together a quick prototype. More experienced programmers and those with a need for a more complex design are supported with Python and C. In the same way that Bonescript adds Arduino style digital and analogue I/O commands to node.js, a library called PyBBIO is available for Python developers.

The GPIO can also be addressed directly from Linux. This could be done directly on the board or remotely over SSH. Firstly, this requires the correct Linux signal name to be identified with a specific GPIO pin and, secondly, an in depth knowledge of working at the Linux command line level. Each GPIO pin will have a directory named with the Linux signal within the /sys/class parent when it is in use. In this way, potential signal/GPIO conflicts can be spotted when one or more capes are being used. For example, if connect P8.pin 16 is identified as GPIO46 and the gpio46 directory does not exist, the signal is available for use (see fig 4).

When driving a led connected to the pin, writing a 1 to Linux value file will turn it on and writing a 0 will turn it off. After use, don't forget to 'unexport' the directory to clear the use of the pin. These shell commands can also be incorporated into Python instructions.

Designers could, of course, use a manufacturer's evaluation kit. Such boards require a lot of set up and knowledge and depend on the availability of example code and documentation from the vendor. This is an area where BeagleBone Black is a better candidate, since TI has encouraged a more open community, with example projects and resources available, as are a range of of capes.

With its open source approach, the BeagleBone Black provides a cost effective and well documented route for prototyping a commercial design. And the open source components are not limited to the tools provided; board schematics, circuit diagrams and Gerber files are available, making the transition from concept to preproduction easier.

Fig 4: Example commands over SSH to control a GPIO pin

```
root@beagleboard:~# cd /sys/class/gpio
root@beagleboard:/sys/class/gpio# ls
export   gpio44 gpiochip0 gpiochip32 gpiochip64 gpiochip96 unexport
root@beagleboard:/sys/class/gpio# echo 46 > export
root@beagleboard:/sys/class/gpio# ls
export   gpio44 gpiochip0 gpiochip32 gpiochip64 gpiochip96 unexport
root@beagleboard:/sys/class/gpio# cd gpio46
root@beagleboard:/sys/class/gpio/gpio46# echo out > direction
root@beagleboard:/sys/class/gpio/gpio46#
root@beagleboard:/sys/class/gpio/gpio46# echo 1 > value
root@beagleboard:/sys/class/gpio/gpio46# echo 0 > value
root@beagleboard:/sys/class/gpio/gpio46#
root@beagleboard:/sys/class/gpio/gpio46# echo 46 > /sys/class/gpio/unexport

root@beagleboard:/sys/class/gpio# ls
export   gpio44 gpiochip0 gpiochip32 gpiochip64 gpiochip96 unexport
root@beagleboard:/sys/class/gpio#
```

Author profile:
Kyle Borgerson is TI product manager for Digi-Key.