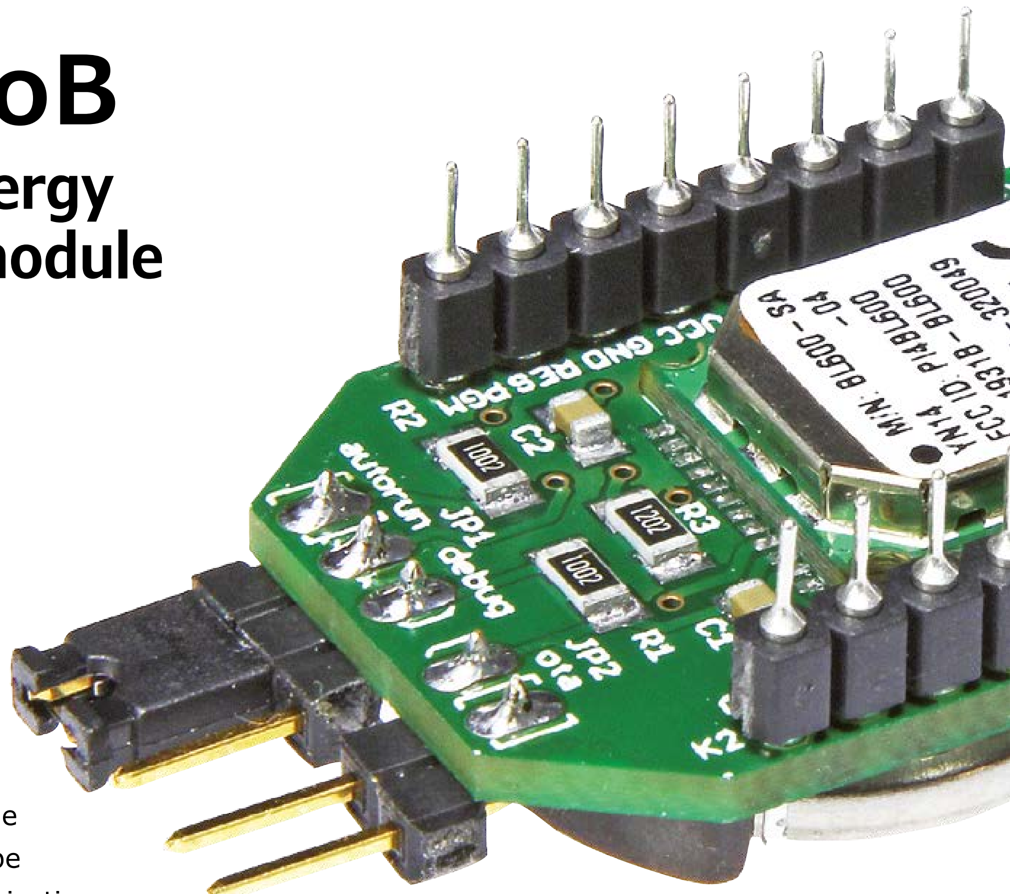


BL600-eBoB

Bluetooth Low Energy communication module

Part 1

Wireless communication on a plate

By Jennifer Aubinais (France)
elektor@aubinais.net

For the network of connected objects (or Internet of things – IoT) to be able to thrive, certain conditions need to be met — in particular, wireless communication and low power consumption by the circuits used to connect these objects. Without long battery life, we'll soon lose interest. So the breakout board for the ultra-low consumption radio communication module presented in this article is going to be an ideal accessory for exploring IoT.

In the January & February 2015 edition, I presented a wireless outdoor thermometer [1], fitted with the Laird Technologies BL600 module (**Figure 1**). In association with iOS and Android applications, this project lets you read a remotely-measured temperature on your smartphone in Bluetooth Low Energy. In order to get to know this module, I strongly recommend you to read up that article; it shows just what you can get out of it.

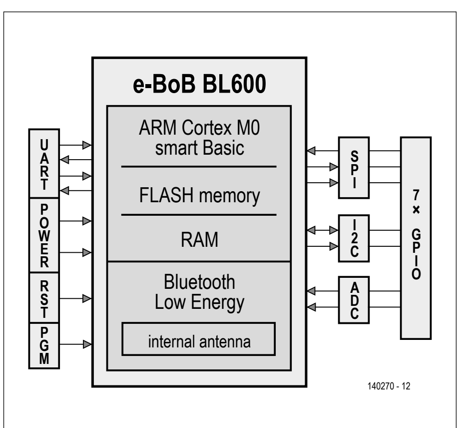


Figure 1. The BL600 Bluetooth Low Energy module is an ideal accessory for achieving wireless communication with connected objects.

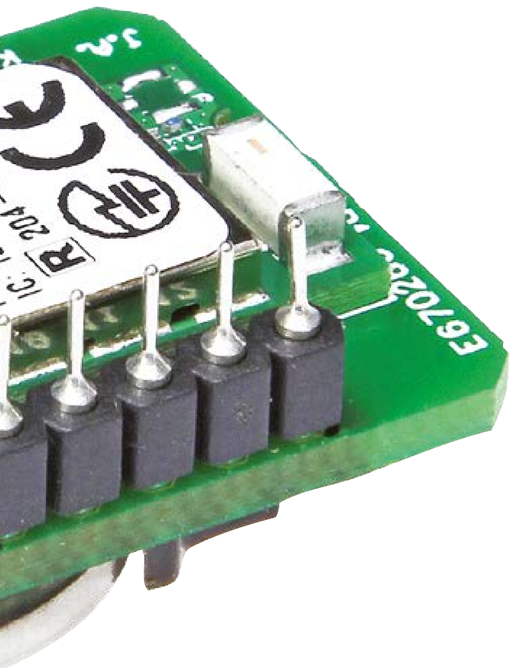
The BL600 takes wireless communication for connected objects into a new era, thanks in particular to its low consumption and high degree of miniaturization; however, the latter comes with its own problems for anyone hoping to solder this component by hand. Aware of this drawback, the manufacturer offers a simple, effective trick for positioning the module with its miniaturized connections on a PCB to within 0.1 mm. This is also described in last month's article, but to spare our readers the pitfalls of this tricky operation, the PCB for this thermometer is available *with the module already fitted* from the e-shop [2]. It's jolly handy!

On the same subject, I also recommend the video elektor.labs have posted on YouTube [3]; it shows how easy it is to establish BT communication between — in this case — a remote thermometer and a smartphone (Android in the video, but it's just as simple under iOS). The same principle can be used for countless other applications.

Breakout board

Given the universal interest of the BL600 module and of the other BLE applications envisaged by elektor.labs, we thought it would be useful to now offer you a breakout board (BoB). Despite the compact size of the Laird module (19 mm × 12.5 mm × 3 mm), this new e-BoB from Elektor will let you access the main signals on the BL600 module, even if you solder by hand.

The circuit (**Figure 2**) shows the two connectors K1 and K2 and the two jumpers JP1 and JP2 found around the edge of the board (**Figure 3**). As a choice had to be made out of all the module's pins, due to lack of space on the breakout board for connecting them all, certain of them have been left off (2, 6–8, 18–21, 24–26,



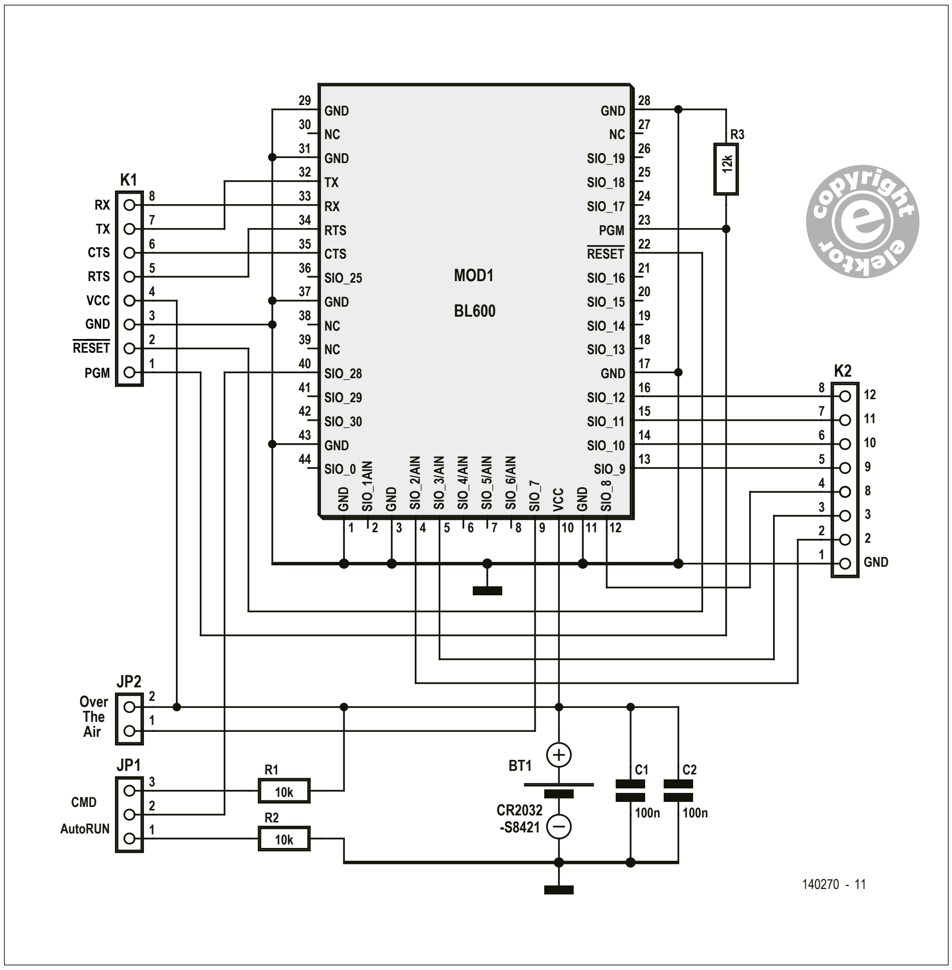


Figure 2. As is only fitting for a breakout board, the circuit of the e-BoB includes only very few components: the BL600 module itself and a few bias resistors and decoupling capacitors. Ultimately, the key elements are the 0.1″ pitch pin headers giving access to the module's main signals

36, 41, 42, 44) in favor the ADC, I²C, and SPI outputs, which are accessible (we'll come back to this in later articles devoted to the module on its BoB).

BoB connections

As is only fitting for a breakout board, the module presides between two rows of 0.1″ pitch pins and two jumpers. The MOD1 signals are grouped as follows:

- The serial port (K1) is used for loading the program into the BL600. It can also be used as a port for dialog between the module and a microcontroller. However, the BL600 has enough inputs/outputs, and its programming language SmartBASIC is powerful enough to allow the module to operate without the help of a microcontroller. Don't miss the next issues of the magazine, where you'll discover this language that certainly deserves its name!

- Power pins (3.3 V) (K1). As the BL600 module only draws 5 µA (!) in stand-by, it can be powered by a

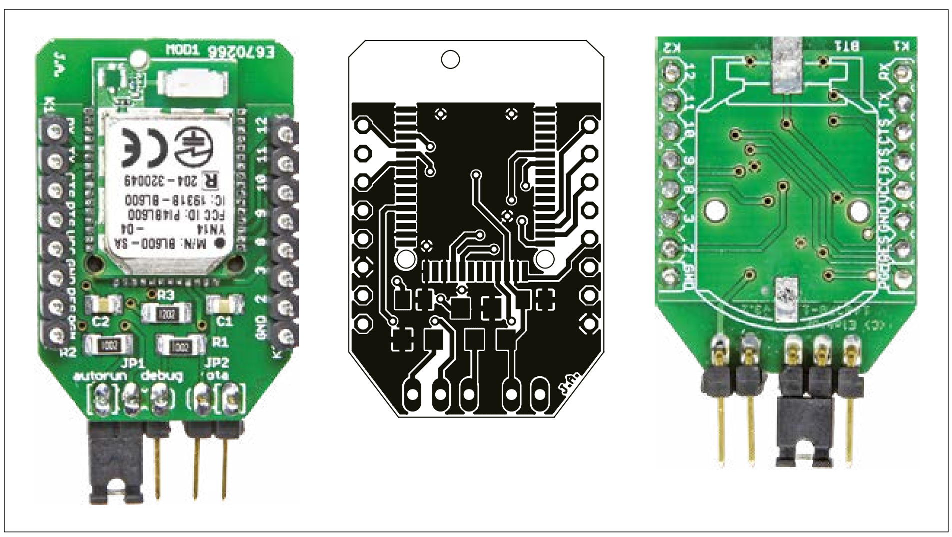
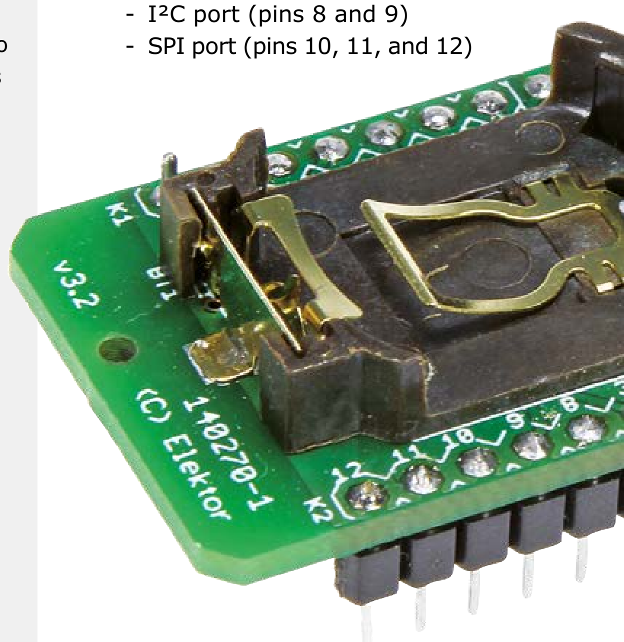


Figure 3. The module is available assembled and ready to use from the Elektor e-shop. K1 and K2 are not fitted but are supplied as loose parts. Depending on whether or not you will be using a battery holder and a button cell (optional, not supplied), you will have to fit K1 and K2 on one side or the other of the board.

CR2032 button cell (BT1), the holder for which is fitted underneath the board; pin-headers K1 and K2 will be fitted to the other side if the battery-holder is fitted.
- The RESET line must be connected to a mini push-button.
- The PGM pin (marked “Not Connected” in the Laird Technologies documentation [4]) can be used for any updates to the modules firmware (for which a J-LINK programmer is required).
- 7 logic inputs/outputs (K2), which can also be used as follows:
  - 2 × 10-bit analog inputs (pins 2 and 3)
  - I²C port (pins 8 and 9)
  - SPI port (pins 10, 11, and 12)



- Jumper JP1 `autorun / cmd` (previously referred to by the author as `autorun / debug`) lets you select between the following modes:
  - AT commands (e.g. `AT&F 1` which performs a complete initialization of the module)
  - `autorun` which, at startup or following manual initialization (RESET), allows automatic execution of a program named `$autorun$`.............
- The `ota` (Over The Air) jumper (JP2) allows an already-compiled program to be downloaded *via a radio link* using a Laird Technologies application. We’ll discuss this further below.

Apart from MOD1, this e-BoB uses just three bias resistors and two decoupling capacitors.

Bluetooth Low Energy, Bluetooth Smart

Bluetooth is a standard for bidirectional radio (UHF) communication over short distances (10 m), principally between portable devices (computers, phones, etc.) and their peripherals: keyboards, mice, earpieces, headphones, etc. In Bluetooth version 4.0, known as BLE for Bluetooth Low Energy, the current consumption is much lower than for the previous 1.0 and 2.0 standards. The prevalence of BLE in the new generations of smartphones is encouraging a proliferation of new connected objects: watches, fitness and health accessories, remote controls, toys, home automation, alarms, etc. and the application of Bluetooth communication in new areas.

Bluetooth Low Energy is not trying to compete with its predecessors on speed: its data rate is 0.3 Mbps, as against 1 Mbps for Bluetooth 1.0. Its objective is low consumption when quiescent (5 µA) and during transmission (10 mA). The economical power reserve management facility for BLE modules allows them to be powered from AAA cells and even button cells (e.g. CR2032).
There are different functional profiles for BLE, particularly for medical applications, e.g. temperature, blood pressure (BL), heart-rate (HRP), etc. To do this, all BLE devices adopt a profile of generic attributes (GATT), which simplifies programming applications around consistent notions:

- ***Client***: a device capable of transmitting GATT requests and commands and receiving responses (e.g. a smartphone)
- ***Server***: a device that receives GATT commands and requests and sends responses (temperature sensor)
- ***Peripheral***: a peripheral can signal the presence of other devices.
- ***Central***: only a central can send a connection request and establish communication.

The notion of ***Service*** combines several characteristics specific to a function; e.g. the Health Thermometer service gives the characteristics of a temperature value as well as the interval between two successive measurements. The notion of ***characteristic*** is a value exchanged between client and server, e.g. the battery voltage. The ***descriptor*** gives information about a characteristic, e.g. the measurement units (degree Celsius). We shall see that all this makes programming easier.
The other services from the BLE protocol that the BL600 module knows are: **BPM** (blood pressure), **HRM** (heart rate), **HTM** (body temperature), ***Proximity***, ***Batch*** (*Send file*), ***Serial*** (UART or VSP interface) and **OTA** (Over The Air). We are using the last two here in our application example (see “Let’s connect up the e-BoB” paragraph).
BLE uses the 2.4 GHz band and works with iPhone 4S and iOS 5, Android 4.3 and Windows Phone 8, but does not communicate with Bluetooth 2.0. It has only 37 channels (against 79 for traditional Bluetooth) and contents itself with exploring three of them (against 32), noticeably speeding up connection. Modules from certain manufacturers combine both these technologies; Bluetooth Smart Ready indicates compatibility with both modes, Bluetooth Smart with the Bluetooth Low Energy mode only (**Table 1).**

	version	
	2.0	**4.0**
Bluetooth	×	
Bluetooth SMART		×
Bluetooth SMART READY	×	×

To sum up, the advantages of BLE are:
- reduced consumption (battery life in months or even years)
- reduced size and price
- compatibility with recent smartphones
- easy programming

If you decide to build it yourself using the PCB design offered by Elektor, you'll find that the three holes around the edge of the BL600 let you temporary fit 1.6 mm screws that will wedge the module accurately for soldering (in a reflow oven). The procedure is described in my article on the wireless remote thermometer [1].

I also recommend reading this article because it gives an idea of just how easy it is to use the BL600 module, thanks to the SmartBASIC programming language offered by Laird technologies. Only certain aspects are covered there; in the following articles on the application of the BL600 on its e-BoB, we'll be coming back to take a closer look at this language. While you're waiting, if you're curious, try studying the source code for the examples given in the manufacturer's documentation.

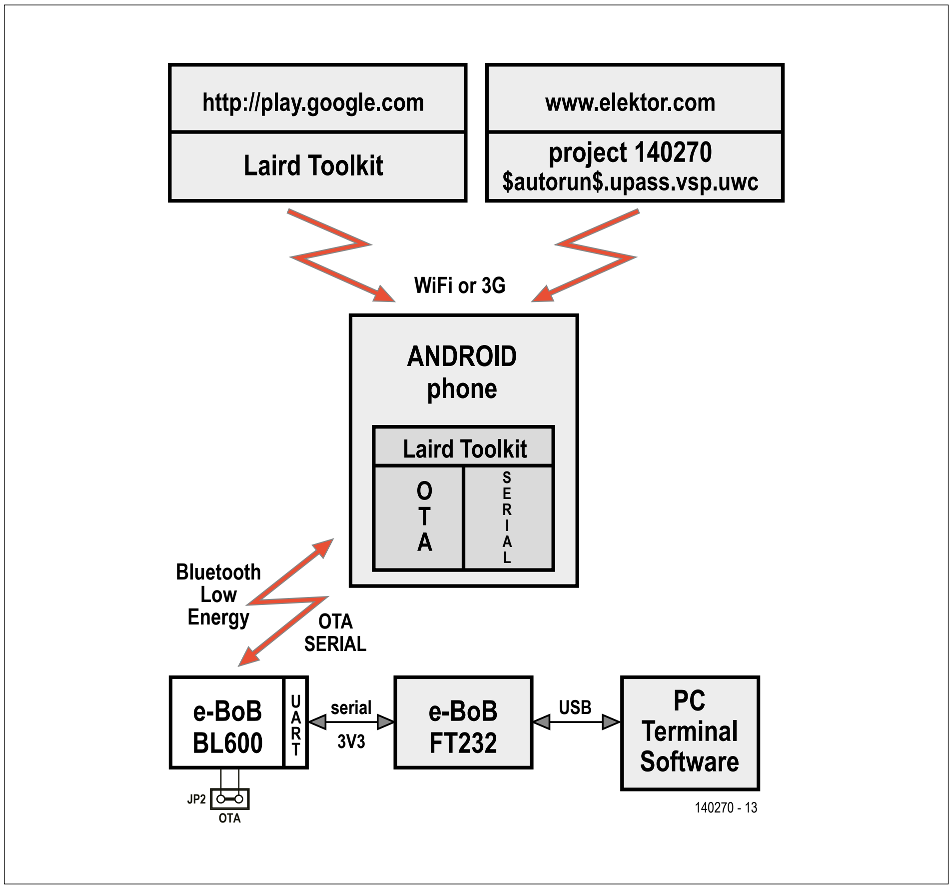
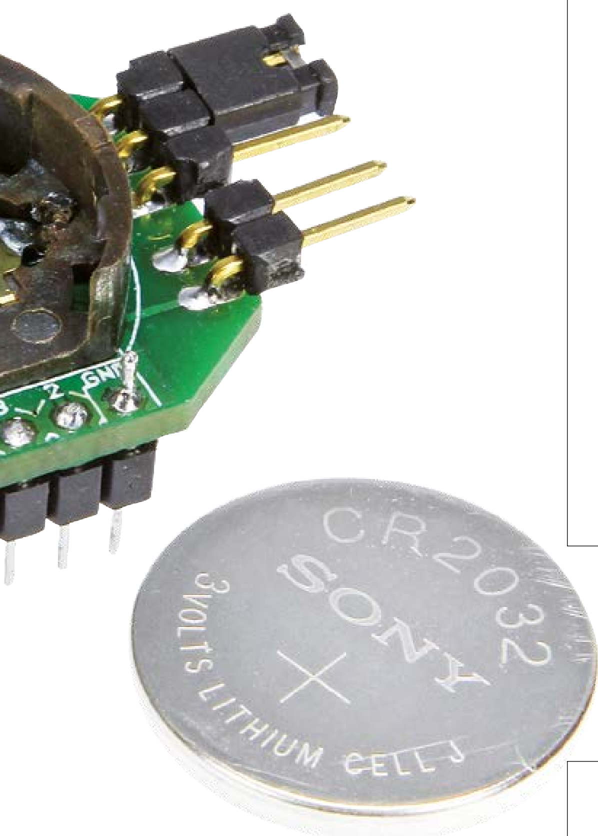


Figure 4. The application example I am suggesting consists of two steps. First of all, thanks to the OTA service, we send the program the e-BoB will be required to run at start-up to the BL600 by radio from an Android phone. Then this UART (or VSP) communication program lets you make your phone communicate with a BLE connected object — in this case, a PC.



Let's connect up the e-BoB

To conclude, I'm suggesting an initial example of communication between your Android phone and a connected object using our e-BOB. Yes, but which object? A connected watch, perhaps? It's the fashion, of course — but to sim-

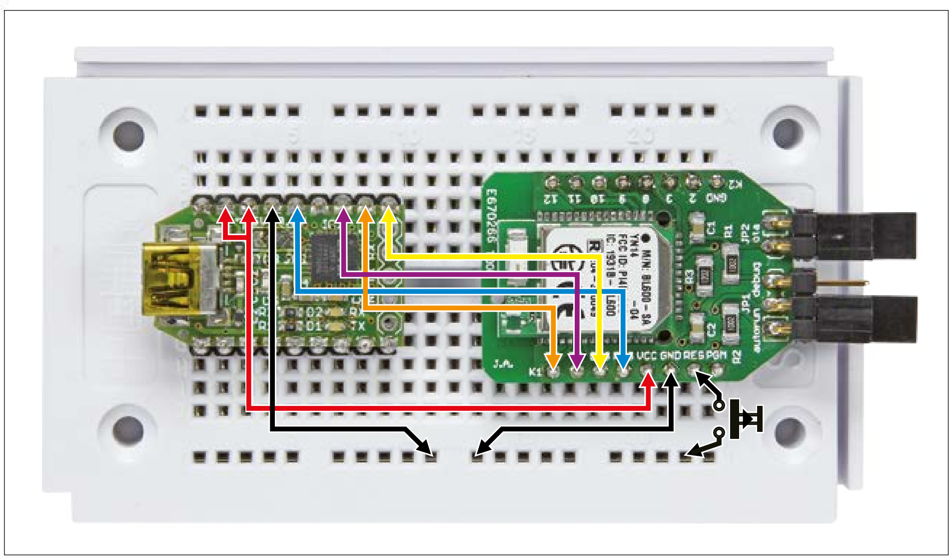


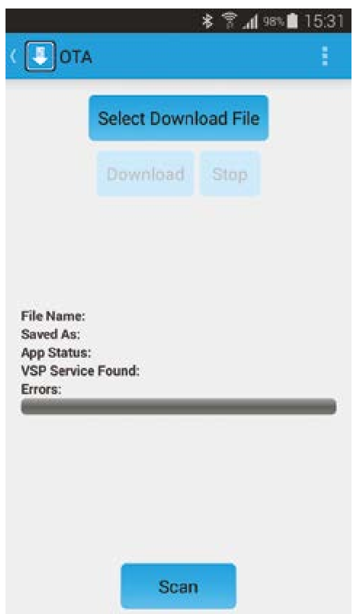
Figure 5. Wiring on the breadboard for connecting the BL600 e-BoB and the FT232 e-BoB for the application in Figure 4.

plify our first test, the connected object is going to be, quite simply… your PC! After all, isn't that the first connected object we all have at our fingertips?
The operation is performed in several steps, summarized in the illustration in **Figure 4**.
Thanks to the remarkable possibilities of Elektor's new BL600 e-BOB, we're first going to download the program for it, in wireless mode (!), and from our smartphone, thanks to the "Over The Air" service; OTA is a standard function of Bluetooth Low Energy that we're going to be taking advantage of.
Once the communication program has been downloaded from the phone to the e-BoB, we'll exchange some data between the Android phone and the PC via the UART interface.
To keep things simple, we're going to start off with an example provided by Laird technologies [5], namely their UART (or VSP) communication program `upass.vsp.sb`. Here, we're not going to be looking at either its source code or how to compile it, but will go directly to the already-compiled version, which you can find on the Elektor website [6] in the `$autorun$.upass.vsp.uwc` file, all ready to run. The fact this file is compiled and is named `$autorun$.xxx` actually offers two advantages: on the one hand, it can be transferred from the phone to the BL600 using BLE's "Over The Air" service (see **inset** about BLE); and on the other, once transferred into the e-BOB, this program will run automatically at start-up or following a manual reset.
To connect the BL600 e-Bob to my PC's USB, I've chosen the Elektor FT232 USB/serial bridge [7]. The experimental wiring will be done on a breadboard as in **Figure 5**.
All we need now is a program capable of operating the OTA and Serial services. Laird Technologies offers us this in the form of the `Laird BL600 Toolkit` application for Android smartphone. Let's get going!

1st step: download the UART program to the e-BoB

For this first step, the BOB-FT232 is used only to power our BL600 e-Bob. On the latter, fit the OTA jumper JP2 and set jumper JP1 to the autoRUN position. Download the compiled file `$autorun$.upass.vsp.uwc` to your Android phone from the Elektor site [6]. Download the `Laird BL600 Toolkit` application from the Laird site [7], run it, and choose the option OTA (Over The Air), then click on Select Download File

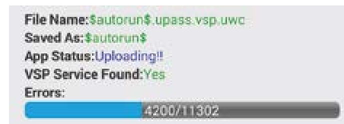
and search for the `$autorun$.upass.vsp.uwc` file on your phone. Now run Scan:



then choose the LAIRD BL600 service

If it is not displayed, reset the module (RESET). And then hit Download…

During the transfer of the file from the phone to the e-BoB, the progress bar will increment.



We're there — the BL600 is ready! All that remains is to terminate the OTA connection (Disconnect), reset the module manually, and quit the OTA application.

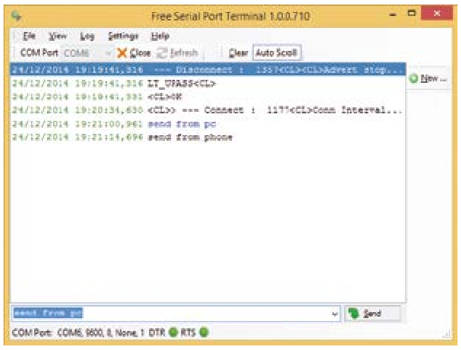
2nd step: testing the communication

I use the "Free Serial port Terminal" program, but any other terminal program will do. On the phone, you need to run the Laird BL600 Toolkit application, choose the Serial tool, then Scan. Then connect to the module.

Your Android phone is now ready to communicate with your PC via the BL600 module on its e-BoB.

Here's an example of exchanging text:

- \- from the phone to the e-BOB and PC: send from phone
- \- via the e-BoB to the PC, to the phone: send from pc



OK, there's maybe nothing very revolutionary about exchanging data between phone and PC — but isn't this simple application a convincing first demonstration of our e-BoB's capabilities? Over to you now to use it in your own projects! Elektor will be devoting other articles to the BL600 module in forthcoming issues. Stay connected!

I've posted online [8] a video of a remote-control application produced last summer using an earlier version of the e-BoB. I hope in this way I can encourage you too to think up Bluetooth Low Energy projects with Android or iOS phones. Thanks to Laird technologies who are putting online the sources for its Android and iOS programs (the Apple license is not free).

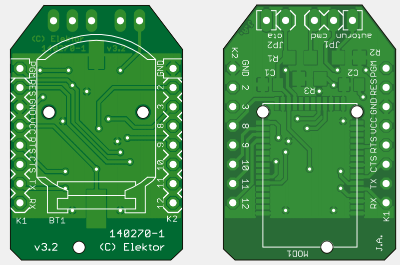
140270

Component List

Resistors:
(5% 250 mW 1206)
R1,R2 = 10kΩ
R3 = 12kΩ

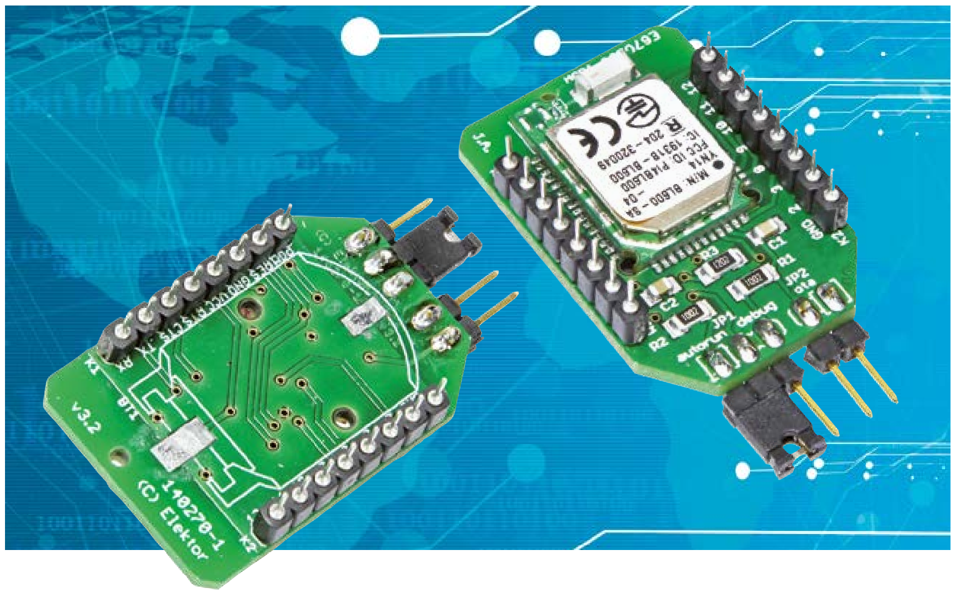
Capacitors
(25V 0805)
C1,C2 = 100nF

Miscellaneous
MOD1 = BL600-SA Bluetooth Lo-Energy module (Laird Technologies)
K1,K2 = 8-pin pinheader, 0.1" pitch
JP1 = 3-pin pinheader, 0.1" pitch
JP2 = 2-pin pinheader, 0.1" pitch
2 jumpers 0.1"
Battery holder S8421-45R (option) (2115305)
Battery type CR2032 (optional)
PCB, Elektor Store # 140270-1



BL600e-BoB, assembled, Elektor Store # 140270-91
(K1 & K2 included as loose parts)

(Farnell part numbers in parentheses)



Selection of topics to be covered in future episodes of this series on the BL600 e-BoB:

- inputs/outputs: a light-chaser
- handler, events
- the Red Green Blue program
- Low Energy, 5 µA
- the I²C port
- the SPI port
- Bluetooth communication
- explanation of the remote wireless thermometer program
- writing a program for Android
- writing a program for iOS (hmm… the Apple license is not free)

Web Links

[1] Wireless thermometer using the BL600 (Elektor January & February 2015)
www.elektor-magazine.com/140190

[2] Wireless thermometer using the BL600 in the Elektor Store
http://www.elektor.com/bluetooth-thermometer

[3] elektor.labs video on the thermometer using the BL600
http://youtu.be/WZSQZGUgJXI

[4] Laird Technologies documentation on the BL600
http://www.lairdtech.com/Products/Embedded-Wireless-Solutions/Bluetooth-Radio-Modules/BL600-Series/

[5] source code
https://laird-ews-support.desk.com/?b_id=1945#software

[6] www.elektor.com/140270

[7] USB/Serial bridge: Elektor BOB-FT232R
http://www.elektor.com/ft232r-usb-serial-bridge-bob-110553-91

[8] Another example of a BL600 application
https://www.youtube.com/watch?v=SxwaVI0Kkk8

[9] Author's website
www.aubinais.net