



Overview

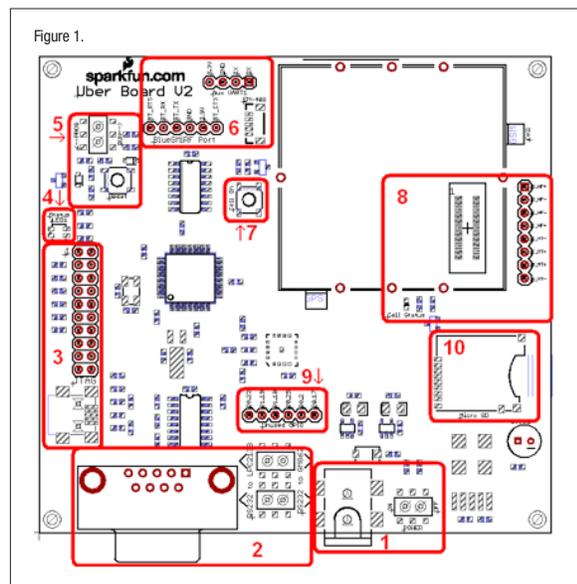
The UberBoard v2 is the second generation development board that brings together an array of our favorite gadgets, all linked together with an LPC2148 ARM7 processor. Simple-to-use sample code is provided and pre-installed on the board when you get it, and all code is open source and compiles with WinARM.

Features:

- LPC2148 ARM7 processor
- Telit GM862 Cellular port
- US Globalsat EM-408 GPS module port
- BlueSMiRF port
- Freescale MMA7260Q accelerometer
- Micro SD socket
- RS-232 and USB communications, plus optional JTAG
- Auxiliary UART port, plus used GPIO made easily accessible

Hardware Description

We've divided the board into sections to better describe the hardware, shown in figure 1.



1 Power input and switch

5V-9V, 500mA, 3mm barrel jack, center positive.

2 RS-232 serial input and select switches

With the switches in the left positions, the serial line goes directly to the LPC2148's UART0. When the switches are in the right positions TX, RX, CTS and RTS go directly to the GM862 cell module. The upshot of this is to make it possible to do firmware upgrades to the GM862, and also to make troubleshooting easier ("Is it my code, or is it the hardware?"). **Note:** When performing a firmware upgrade in the GM862 cell module, set the LPC2148 program/run switch to program. This will keep the logic lines between the processor and the cell module in a hi-Z state.

3 Mini USB connector and optional JTAG connector

The LPC2148 processor is capable of USB communication, so we've provided the interface. Open source libraries for USB and the LPC2148 are available, but are not integrated with the sample code. Also, 99% of the work we do with the LPC2148 at Sparkfun does not involve JTAG (we generally just use the serial bootloader), but we've provided the port should the user wish to employ it. Pin 1 is at the top left in section 3 of the figure.

4 Tri-color status LED

Who doesn't like blinky things? The sample firmware makes no use of the status LEDs after the initial power-up sequence. That is left up to the user.

5 LPC2148 reset button and program/run select switch

The reset button only affects the processor, no other devices. To program over the serial line with the on-board bootloader, set the select switch in the down position. To run your code, set the select switch in the up position.

6 Auxiliary UART1 port, BlueSMiRF port and EM-408 connector

UART1 on the LPC2148 is shared between several devices by means of a 2-channel multiplexer. Section 6 shows 3 of the UART1 interfaces, those being for a BlueSMiRF, a US Globalsat EM-408 GPS module and an auxiliary port for the user's own discretion. The fourth device that can be accessed is the GM862 module.

7 Manual power on/off button for the GM862 module

8 GM862 connector, status LED and audio lines

This is where the cell module plugs into the board. The status LED serves the cell module alone. And it's not fair to assume



that the user will only intend to do data transfer, so we've broken out the audio lines as well.

9 Unused GPIO

These are the lines that were left after everything else was done. Instead of deserting them, we break them out. The board and schematic show which are available.

10 Micro SD socket

Running the Sample Firmware

To start, plug your board into your power supply through the 3mm barrel jack and plug in a serial cable into the DB9 connector of your computer. Open up a terminal emulator (Hyperterminal, Teraterm, etc.) to the COM port to which you're connected and set it for 115200 baud, 8-bit data, no parity, 1 stop bit, no flow control. Make sure that the run/program switch is set to run and that the RS-232 serial input and select switches are set to the LPC2148 (to the left).

Turn the power switch on and you will see the sample firmware menu come up:

Uber Board V2 Test Code

- 1 Accelerometer/ADC
- 2 Toggle accelerometer sensitivity, currently 1.5g
- 3 Toggle Mux Channel, currently GPS
- 4 Initialize SD card
- 5 Toggle UART1 baud rate, currently 115200
- 6 Toggle EM-408 power, currently off
- 7 Send power pulse to the cell module
- 8 Talk to UART1 (press ^s to exit)

Menu Description

1 Accelerometer/ADC

Pressing "1" will start a slow sampling loop that will report X, Y and Z axes of the Freescale MMA7260Q accelerometer. This serves as a demonstration of both the accelerometer and how to access the ADC lines on the LPC2148. Values returned are 10-bit ADC values (between 0 and 1023). Pressing the space bar will cause the loop to terminate and return to the menu.

2 Toggle accelerometer sensitivity

The MMA7260Q has 4 sensitivity settings: 1.5g, 2g, 4g and 6g. Pressing "2" allows the user to toggle sequentially through these settings.

3 Toggle Mux channel

As stated earlier, there are four ports accessible to UART1 on the LPC2148 through a multiplexer, those being the EM-408 GPS module, the BlueSMiRF port, an auxiliary open port and the GM862 cell module. Pressing "3" will allow the user to select the device with which to talk.

4 Initialize SD card

Pressing "4" will initialize a micro SD card should there be one in the socket. A FAT16 library has been included in the source code, but this small demonstration is basically the "Hello World" equivalent for SD cards.

5 Toggle UART1 baud rate

Not all of these devices, or whatever else you may connect, necessarily talk at the same speed. For example, the EM-408 works at 4800 baud, while the BlueSMiRF will talk at 115200 baud. The GM862 actually auto-bauds, so you can set it to pretty much anything. Pressing "5" will allow the user to toggle between 4800, 9600, 14400, 19200, 38400, 57600 and 115200 baud.

6 Toggle EM-408 power

Pressing "6" will allow the user to shut down the GPS module when not in use.

7 Send power pulse to the cell module

The GM862 cell module is turned on and off by means of a pulse on a specific pin. Pressing "7" will automatically send a pulse to turn the device on or off. It does exactly the same thing as the push button mentioned earlier, only with the processor.

8 Talk to UART1

Pressing "8" allows direct user access to whatever is selected for UART1 at whatever selected baud rate. This link remains open until the user presses "^s" ("<control>s"), which then returns the user to the menu.

Examples

Let's say you want to talk to your EM-408 GPS module (assuming you have one of those plugged in to your board). The first thing to do is to set the UART1 baud rate to 4800 by pressing "5" until the baud rate displayed is 4800. Then press "3" until the current mux channel shown is GPS, then "6" to turn the module on. Lastly, press "8" to open the channel to UART1. You should see NMEA data streaming on your screen after that. To exit back to the main menu, press "^s" (<control>s).



Or if you want to talk to the GM862 cell module (again, assuming you have one plugged into your board), press “3” until “cell” is shown as the current mux channel. Then press “7” to send a pulse to the on/off pin on the GM862. Setting the baud rate in this case isn’t so important because the GM862 auto-bauds (automatically adjusts to whatever you send it, we typically use 38400). Lastly, press “8” to talk to the port.