Api-Do Project

Api-Mote

Base r4.0beta

River Loop Security

February 4, 2014

1 Product Description

Api-Mote is a low-power wireless module supporting IEEE 802.15.4. It allows for rapid application prototyping as well as wireless and security research. It leverages industry-standard chips including the MSP430 microprocessor and an IEEE 802.15.4 and ZigBee-ready RF transceiver. These main ICs provide a mix of ease-of-use/development with common wireless RF and security research frameworks as well as low-level hardware access (such as provided by the CC2420 compared to other models). Capability for expansion is designed into the Api-Mote Base , with two expansion headers, a battery-board/expansion header, and the option of an internal (PCB) or external (SMA-connected) antenna.

2 Key Features

- 2.4 GHz IEEE 802.15.4 Compliant and ZigBee^{TM}Ready RF Transceiver (CC2420)
- Interoperability with other IEEE 802.15.4 devices
- 16-Bit Ultra-Low-Power MCU (116kB Flash, 8KB RAM) (MSP430F2618), featuring Integrated ADC, DAC, Supply Voltage Supervisor, and DMA Controller
- FTDI USB-to-Serial IC
- Programming and data collection via USB
- Integrated onboard antenna
- Low current consumption
- Hardware link-layer encryption and authentication is supported
- Optional SMA antenna connector
- Basic GoodFET-based firmware support is already available (TinyOS support should be easy)

Contents

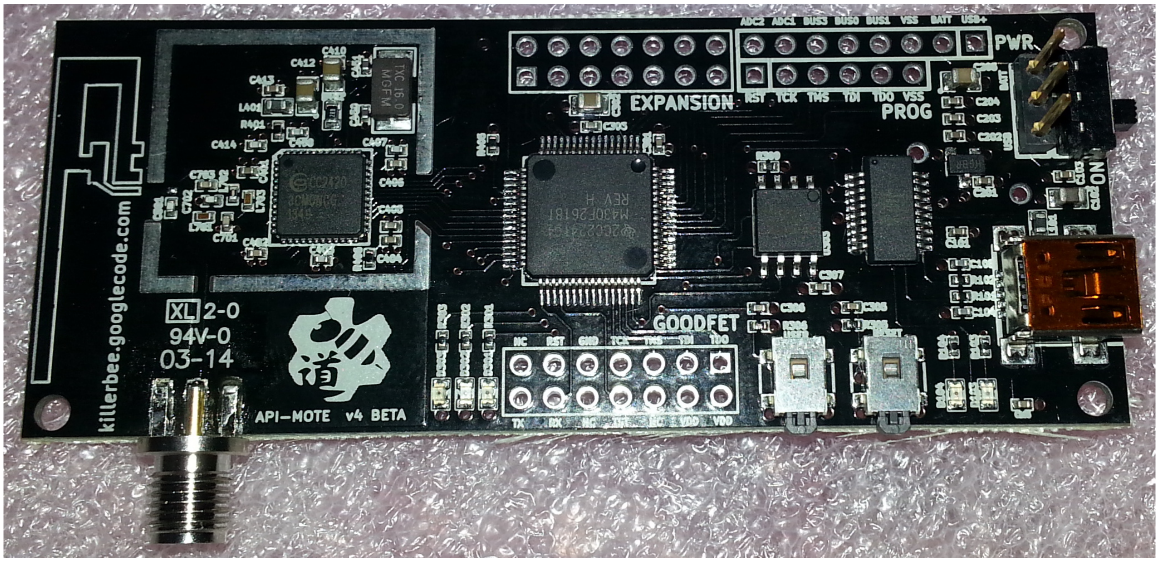

Figure 1: Api-Mote Base 4.0beta Front

3 Module Description

The Api-Mote features a number of components in the standard board population, which is depicted in Figures 1 and 2:

3.1 Block Diagram

The Api-Mote Base standard components and their logical connections are shown at a high-level in Figure 3:

4 Microprocessor

4.1 Description

The Api-Mote Base uses a TI MSP430F2618 micro-controller, part of the MSP430 family of ultralow-power micro-controllers. The architecture has five low-power modes optimized to achieve extended battery life in portable applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators. The calibrated digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 s. The MSP430F261x series has two built-in 16-bit timers, a fast 12-bit A/D converter, a comparator, dual 12-bit D/A converters, four universal serial communication interface (USCI) modules, DMA, and up to 64 I/O pins. Of these, up to 48 are available on the Api-Mote Base as I/O pins, as the rest are for dedicated functions (JTAG, Vdd/Vss, crystals, etc).

For more detailed information on the micro-controller, please refer to `http://www.ti.com/product/msp430f2618`.

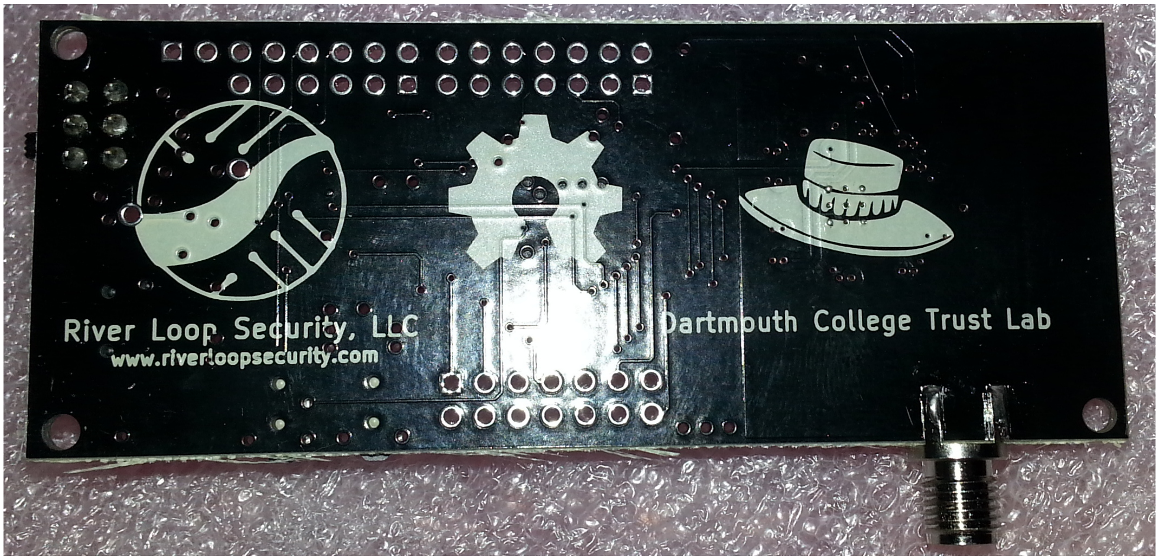

Figure 2: Api-Mote Base 4.0beta Back

4.2 Computer Communication

The Api-Mote uses an FTDI USB controller to communicate with the host computer. In order to communicate with the Api-Mote , the FTDI drivers must be installed on the host. These are often installed by default, but otherwise can be downloaded from FTDI's website at: `http://www.ftdichip.com`. The Api-Mote appears as a COM port in Windows' device manager or as a device in `/dev` in Linux, OSX, etc (see `lsusb` or `dmesg`). An application may communicate with the Api-Mote by opening the COM port or serial device assigned to the Api-Mote . Api-Mote communicates with the host computer through USART1 on the TI MSP430.

The Api-Mote uses a modern FTDI chip, and a newer Linux kernel is needed for built-in support. If you're using Linux and you see the device in `lsusb` but don't see the device appear in `/dev/ttyUSB*`, then this may be the issue. We've tested with 3.2.0-26-generic (on Ubuntu 12.04) among other kernels.

4.3 Programming

The Api-Mote revision 4.0beta can be wired either with direct access to chip programming using the FTDI DTR and RTS pins tied to the TCLK and RST pins on the MSP430, or may instead use an optional PIC micro-controller with special code to enable programming mode on the MSP430 after receiving a special command sequence. Unless otherwise specified, we discuss the Api-Mote Base with the direct programming mode. This is most closely compared to the GoodFET41's wiring for programming (but with a different UART being used).

Firmware images must be programmed and compiled for this specific architecture and hardware layout/pins. The programming is intended to be done using the patched BSL loader in the GoodFET trunk. Bootloading can be invoked as follows:

```
./goodfet.bsl --speed=38400 -p apimote4.hex
```

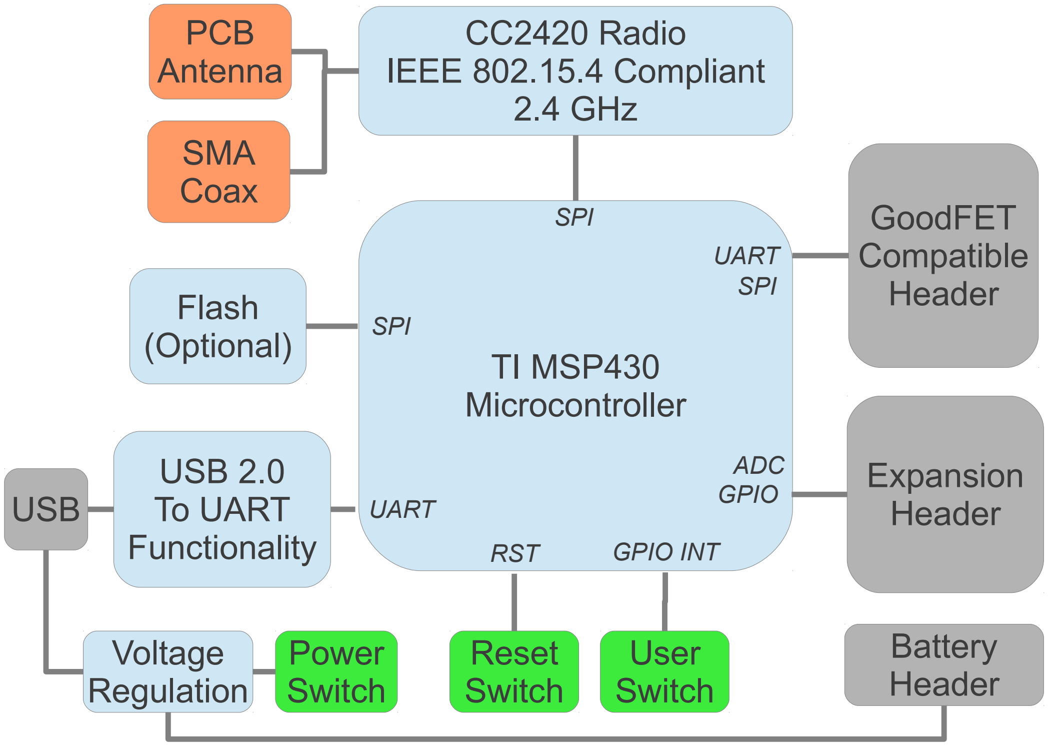

Figure 3: Api-Mote Base Block Diagram

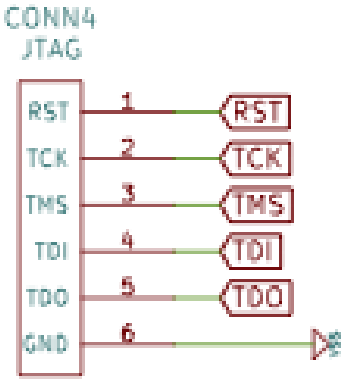

Figure 4: Programming/JTAG Header

If you need to compile modifications to the firmware, we show the compilation of the GoodFET trunk firmware for this board as an example:

```
$ cd goodfet/trunk/firmware
$ export board=apimote2
$ make clean install
$ ../client/goodfet.bsl --speed=38400 -p goodfet.hex
```

Note: The firmware for ApiMote 2, 3, and 4 is currently the same. Although currently compiled with board set to apimote2, the firmware is optimized for Api-Mote 4.0beta boards. They should work on ApiMote r2, boards, with the exception of the 3 user LEDs being controlled backwards (off turns them on, and visa versa).

4.3.1 Advanced Programming Access

The Api-Mote Base provides a programming header offering access to the MSP430's JTAG pins, as shown in Figure 4.

5 Radio

Api-Mote features the Chipcon CC2420 radio for wireless communications. The CC2420 is an IEEE 802.15.4 compliant radio providing the PHY and some MAC functions.

The CC2420 is highly configurable for many applications with the default radio settings providing IEEE 802.15.4 compliance. It also has programmable output power. More information on the CC2420 is available in Chipcon's datasheet at http://www.ti.com/product/cc2420.

The CC2420 is controlled by the TI MSP430 micro-controller through the SPI port and a series of digital I/O lines and interrupts. The radio may be shut off by the micro-controller for low power duty cycled operation.

The CC2420 provides a digital received signal strength indicator (RSSI) that may be read any time. Also, on each packet reception, the CC2420 samples the first eight chips, calculates

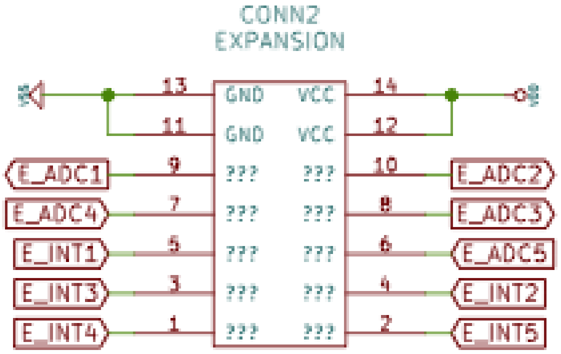

Figure 5: Expansion Header

the error rate, and produces a link quality indication (LQI) value that may optionally be included with each received packet.

6 Memory

The Api-Mote Base has support for external memory (rather than what is contained in the MCU), specifically a flash chip mounted on the board. This is available by SPI, and is specified as a 32M flash IC. Currently, support for this chip is not in the GoodFET firmware. Please feel free to add it! Note, it is connected on a separate SPI bus than the radio.

7 Expansion Connectors

The Api-Mote Base has three expansion connectors. These are detailed below:

7.1 Expansion Header

The Api-Mote Base provides a 7-by-2 header meant to enable additional external devices such as sensors, LCD displays, user I/O, and peripherals can be controlled. The pinout of this header is shown in Figure 5.

7.2 GoodFET Header

The Api-Mote Base provides a 7-by-2 header that is intended to be compatible with the headers common on GoodFET boards. GoodFET firmware images which seek this compatibility should setup this header as shown in Figure 6.

7.3 Battery Header

The Api-Mote Base is designed to operate on battery power, and has a 8-by-1 header to support the addition of a custom battery pack that provides advanced power regulation,

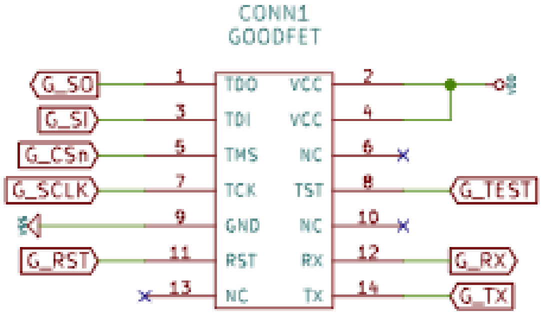

Figure 6: GoodFET Header

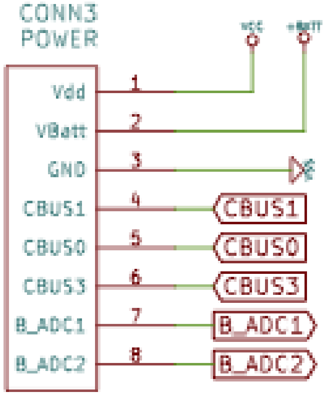

Figure 7: Battery Header

sleep, and recharge capabilities. The pinout of this header is shown in Figure 7. The Api-Mote Base USB-to-serial IC supports high current USB port detection for decreased charge time. At this time, Api-Mote Base revision 4.0beta does not yet have a complementary battery pack, but one is under development.

8 License

This product is released by River Loop Security as an open-source project. The hardware designs and firmware are freely available. However, we ask that one does not sell products based on this device without first obtaining written permission. Collaboration, however, is always welcome and we strongly encourage people to help develop the firmware, etc. Please contact us if you have questions.

9 Ordering/Contact Information

To request an Api-Mote, please visit `http://rmspeers.com/contact/pcb-requests`. For more information, please contact us at `team@riverloopsecurity.com`.

Acknowledgements

The Api-Mote Base project lead, Ryan Speers, extends many thanks to the board engineer and designer, Javier Vazquez, as well as the original Api-Mote v1 engineer. Also, much credit and thanks goes to Professor Sergey Bratus of Dartmouth College for his encouragement and funding of the test runs of production. Thank you to River Loop Security for supporting this project with time and funding. Finally, thanks goes to Travis Goodspeed, Ricky Melgares, and Michael Ossmann for their ongoing collaboration with hardware and ZigBee security research.