Table of Contents

1 Introduction

1.1 Bambino Family of Controllers

1.2 Bambino 200 Overview

The Micromint Bambino 200 is a .NET Gadgeteer mainboard powered by NXP's LPC4330, a dual core Cortex M microcontroller capable of operating at 204-MHz. The Bambino 200 allows NetMF and mbed developers to take advantage of the LPC4330 in their application using APIs and tools they are familiar with. It has a built in power module and comes in a small 4" x 2.3" (101.6 x 58.4 mm) form factor. The enhanced Micromint Bambino 200E is available for applications requiring networking and additional I/O, including the same base features but with enhanced functionality.The hardware block diagram and feature summary is included below:

1.3 Bambino 200 Base Features

- 204 MHz Dual Core 32-bit ARM® Cortex?-M4/M0
- 264k SRAM
- 4M SPIFI Flash
- USB Device Port
- 2 Push Buttons
- 2 LEDs
- 5 Gadgeteer Sockets

1.4 Bambino 200E Features

- Same Base Features
- 8M SPIFI Flash
- 10 Gadgeteer Sockets
- 100 Mbps Ethernet
- microSD Socket
- Xbee Socket

1.5 Software and Support

NEXT: Getting Started

PREVIOUS: Table of Contents

2 Getting Started

2.1 Getting Started with Gadgeteer Applications

2.1.1 Install System Files

The Micromint Bambino supports NetMF and Gadgeteer application development. These are the steps to prepare your development system and board.

1. Install Microsoft Visual Studio 2012 Express for Windows Desktop or Microsoft Visual Studio 2010 Express (C# or VB). These are available from Microsoft at no cost. The Professional, Premium and Ultimate Editions of Visual Studio 2012 or 2010 are also supported.

2. If you use Visual Studio 2012, install the .Net Micro Framework SDK 4.3. If you use Visual Studio 2012, install the .Net Micro Framework SDK 4.2.

3. Install the .NET Gadgeteer Core.

4. Install the Bambino NetMF SDK.

SDK Installer

5. Install the Bambino Gadgeteer SDK.

6. Update the firmware on your board to the current level.

2.1.2 Run Test Application

NEXT: Getting Started - mbed Applications

PREVIOUS: Introduction

3 Getting Started mbed

3.1 Getting Started with mbed Applications

4 Updating Firmware

4.1 Updating Firmware

The Bambino 200 firmwarte can be updated via the primary USB port using the standalone NXP DFU Flash Programmer. This tool is a simple alternative to update firmware in boards deplayed in the field. It also allows NetMF developers to update the TinyCLR firmware to a current version. Other applications can install their corresponding firmware using this flash programmer. To update firmware via DFU, please follow these steps:

1. Install the DFU Flash Programmer to a directory in your hard disk. Currently the NXP DFU programmer is only available for Windows.

2. Place the board in USB boot mode by shorting the two contacts labelled "Boot JP1" as shown below while pressing and releasing the RESET button. Metal tweezers work great for shorting the two contacts. If you will be doing frequent updates, you may consider soldering a 2-pin header in and use a jumper to enter the USB boot mode.

Boot Jumper

3. An entry "LPC USB" should appear in the Windows Device Manager. If your device is not recognized, please check that you have the USB Drivers installed.

LPC USB device

4. Run lpc_dfutil.exe in the dfusec folder. You should see "HIGH SPEED USB" in the status bar indicating it was able to connect to the board.

LPC DFU Flash Programmer

5. Use the following parameters and press START. You may need to use full paths for the algorithm (*.hdr) and firmware (*.bin) files. If you want to maintain deployed assemblies when updating TinyCLR, reduce the size to 0x070000 so the rest of the flash is not updated.

```
Algorithm: .\Programming_algorithms\iram_dfu_util_spiflash.bin.hdr
File: <Path and name of firmware to be copied to flash>
Address: 0x14000000
Size:    0x00400000
Param:   0x00000000
Device erase: Region
Operation after: Reset
No checkboxes neede
```

You can test the procedure using the file listed below.

.\Prebuilt_examples\bambino200\blinky_spifi_bambino_200.bin

5. After the flash is complete, exit the utility, remove the boot jumper, and reset your board. You need to press RESET twice due to an issue documented in the LPC43X0 Errata for boards booting from SPIFI flash.

6. Three USB Drivers are provided for the Bambino 200. Selecting the top directory allows Windows to select the appropriate one based on the vendor and product ID (VID/PID).

Driver	Directory	Purpose
WinUSB	WinUSB\BambinoUSB.inf	Default NetMF driver
CDC	VCom\LpcVcom.inf	Alternate NetMF driver
DFU	WinUSB\LpcDevice.inf	Flash programmer

NEXT: Rebuild NetMF Firmware

PREVIOUS: Getting Started - mbed Applications

5 Rebuild Firmware

5.1 Command Line Build

All NetMF firmware used in the Bambino 200 is open source and can be downloaded from the Micromint repository. To recompile it please follow these steps:

1. Install a compatible ARM toolchain. Currently the Keil MDK is supported. Our primary firmware development tool is Keil MDK 4.71. The GCC Code Red toolchain will be supported by the end of August.

2. Install the NetMF Porting Kit. This includes full source code for the CLR. The default installation directory is **C:\MicroFrameworkPK_v4_2**.

3. Install the LPC43XX and Bambino 200 source files from the Micromint repository.

4. Compile the TinyBooter. These are typical commands to change to the source directory, setup compiler environment and build the bootloader.

```
cd \MicroFrameworkPK_v4_2\Solutions\Bambino200\TinyBooter
call ..\..\..\setenv_mdk.cmd 4.71 C:\Keil\ARM
msbuild TinyBooter.proj /t:rebuild /p:flavor=release;memory=flash >msbuild.log
```

If your build is successful, a TinyBooter binary will be generated. If not, check the msbuild.log file.

```
dir ..\..\..\BuildOutput\THUMB2\MDK4.71\le\FLASH\release\Bambino200\bin\*.bin
Volume in drive C is OS
Volume Serial Number is F051-6A1D

Directory of C:\MicroFrameworkPK_v4_2\BuildOutput\THUMB2\MDK4.71\le\FLASH\release\Bambino200\bin

07/05/2013  10:54 AM            36,180 Tinybooter.bin
```

5. Compile the TinyCLR. These are typical commands to change to the source directory, setup compiler environment and build the CLR.

```
cd \MicroFrameworkPK_v4_2\Solutions\Bambino200\TinyCLR
call ..\..\..\setenv_mdk.cmd 4.71 C:\Keil\ARM
msbuild TinyCLR.proj /t:rebuild /p:flavor=release;memory=flash >msbuild.log
```

If your build is successful, the TinyCLR config and firmware hex files will be generated. If not, check the msbuild.log file.

```
dir ..\..\..\BuildOutput\THUMB2\MDK4.71\le\FLASH\release\Bambino200\bin\tinyclr.hex
Volume in drive C is OS
Volume Serial Number is F051-6A1D

Directory of C:\MicroFrameworkPK_v4_2\BuildOutput\THUMB2\MDK4.71\le\FLASH\release\Bambino200\bin\tinyclr.hex

07/05/2013  11:26 AM             6,968 ER_CONFIG
07/05/2013  11:26 AM         1,343,816 ER_FLASH
```

5.2 IDE Build

When implementing new or modified NetMF functionality, a large percentage of the development time is spent debugging and troubleshooting code changes. To improve prodctivity, our implementation includes project files to build and debug TinyBooter and TinyCLR with the Keil MDK. These allow you to access a full featured ARM development environment and debugger. A JTAG is required for debugging.

TinyCLR project for Keil MDK

For production builds, the command line procedure based on 'msbuild' should be used since it is integrated with other NetMF tools. For example, changing CLR features with the Solution Wizard changes the 'msbuild' projects and implements any stubs or templates required. These changes need to be implemented manually in the project file.

NEXT: Rebuild mbed Firmware

PREVIOUS: Updating Firmware

6 Rebuild mbed Firmware

6.1 Command Line Build

The source code for the mbed library is open source and can be downloaded from the mbed mainline repository or the Micromint repository. To recompile it please follow these steps:

1. Install a compatible ARM toolchain. The mbed port for LPC4330 targets currently supports the Keil MDK, GCC ARM and GCC LPCXpresso toolchains. The Keil MDK is the main ARM toolchain at Micromint.

2. Install Python 2.7. The mbed build scripts are written in Python.

3. Install the mbed Library Sources. This includes full source code for all supported multiple microcontrollers and boards.

4. Edit workspace_tools/private_settings.py to reflect your toolchain directories.

```
from os.path import join

# ARM
armcc = "keil"
ARM_PATH = "C:/Keil/ARM"
ARM_BIN = join(ARM_PATH, "ARMCC", "bin")
ARM_INC = join(ARM_PATH, "ARMCC", "include")
ARM_LIB = join(ARM_PATH, "ARMCC", "lib")

# GCC ARM
GCC_ARM_PATH = "C:/Program Files (x86)/GNU Tools ARM Embedded/4.8 2014q2/bin"

# GCC CodeRed
GCC_CR_PATH = "C:/nxp/LPCXpresso_7.5.0_254/lpcxpresso/tools/bin"
```

5. Compile the base mbed libraries. These are typical commands to change to the source directory, setup compiler environment and build the bootloader. Note that the Python directory in the PATH may be different in your development system.

```
cd mbed-master
SET PATH=C:\Windows\system32;C:\Windows;C:\Python27
python workspace_tools/build.py -v -r -m LPC4330_M4 -t ARM > build.log
python workspace_tools/build.py -v -r -m LPC4330_M4 -t GCC_ARM >> build.log
```

If your build is successful, the mbed libraries and headers for the LPC4330 will be available in the ..\build directory. If not, check the build.log file. For more documentation on these procedures please visit the mbed tools site.

6.2 IDE Build

When implementing new or modified mbed functionality, a large percentage of the development time is spent debugging and troubleshooting code changes. To improve prodctivity, our implementation includes project files to build and debug the mbed library with the Keil MDK. These allow you to access a full featured ARM development environment and debugger. A JTAG is required for debugging.

mbed project for Keil MDK

For production builds, the mbed build system should be used.

NEXT: Hardware

PREVIOUS: Rebuild NetMF Firmware

7 Hardware

The following image shows where some of the hardware components are located.

Bambino 200 Hardware

7.1 Microcontroller

The Bambino 200 includes a NXP LPC4330 microcontroller. These dual core 32-bit ARM Cortex-M4/M0 RISC microcontroller are capable of 204-MHz operation with a Thumb2 instruction set for smaller object code. It uses a Harvard architecture with separate local instruction and data buses as well as a separate peripherals bus. Please see NXP?s LPC4330 Microcontroller's User Manual for more information and register definitions.

7.1.1 LPC4330 key features

- Cortex-M4 Processor core
  - Built-in Memory Protection Unit (MPU) supporting eight regions.
  - Running at frequencies of up to 204 MHz.
  - Built-in Nested Vectored Interrupt Controller (NVIC).
  - Hardware floating-point unit.
  - Non-maskable Interrupt (NMI) input.
  - JTAG and Serial Wire Debug (SWD), serial trace, eight breakpoints, and four watch points.
  - Enhanced Trace Module (ETM) and Enhanced Trace Buffer (ETB) support.
  - System tick timer.
- Cortex-M0 Processor core
  - Running at frequencies of up to 204 MHz.
  - JTAG and built-in NVIC.
- On-chip memory
  - 264 kB SRAM for code and data use.
  - Multiple SRAM blocks with separate bus access. Two SRAM blocks can be powered down individually.
  - 64 kB ROM containing boot code and on-chip software drivers.
  - 128 bit general-purpose One-Time Programmable (OTP) memory.
- Clock generation unit
  - Crystal oscillator with an operating range of 1 MHz to 25 MHz.
  - 12 MHz Internal RC (IRC) oscillator trimmed to 1 % accuracy over temperature and voltage.
  - Ultra-low power Real-Time Clock (RTC) crystal oscillator.

- Three PLLs allow CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. The second PLL is dedicated to the High-speed USB, the third PLL can be used as audio PLL.
- Clock output.

- Configurable digital peripherals
  - Serial GPIO (SGPIO) interface.
  - State Configurable Timer (SCT) subsystem on AHB.
  - Global Input Multiplexer Array (GIMA) allows to cross-connect multiple inputs and outputs to event driven peripherals like the timers, SCT, and ADC0/1.
- Serial interfaces
  - Quad SPI Flash Interface (SPIFI) with 1-, 2-, or 4-bit data at rates of up to 52 MB per second.
  - 10/100T Ethernet MAC with RMII and MII interfaces and DMA support for high throughput at low CPU load. Support for IEEE 1588 time stamping/advanced time stamping (IEEE 1588-2008 v2).
  - One High-speed USB 2.0 Host/Device/OTG interface with DMA support and on-chip high-speed PHY (USB0).
  - One High-speed USB 2.0 Host/Device interface with DMA support, on-chip full-speed PHY and ULPI interface to external high-speed PHY (USB1).
  - USB interface electrical test software included in ROM USB stack.
  - Four 550 UARTs with DMA support: one UART with full modem interface; one UART with IrDA interface; three USARTs support UART synchronous mode and a smart card interface conforming to ISO7816 specification.
  - Up to two C_CAN 2.0B controllers with one channel each.
  - Two SSP controllers with DMA, FIFO and multi-protocol support.
  - One SPI controller.
  - One Fast-mode Plus I2C-bus interface with monitor mode and with open-drain I/O pins conforming to the full I2C-bus specification. Supports data rates of up to 1 Mbit/s.
  - One standard I2C-bus interface with monitor mode and with standard I/O pins.
  - Two I2S interfaces, each with DMA support and with one input and one output.
- Digital peripherals
  - External Memory Controller (EMC) supporting external SRAM, ROM, NOR flash, and SDRAM devices.
  - Secure Digital Input Output (SD/MMC) card interface.
  - Eight-channel General-Purpose DMA controller can access all memories on the AHB and all DMA-capable AHB slaves.
  - General-Purpose Input/Output (GPIO) pins with configurable pull-up/pull-down resistors.
  - GPIO registers are located on the AHB for fast access. GPIO ports have DMA support.
  - Up to eight GPIO pins can be selected from all GPIO pins as edge and level sensitive interrupt sources.
  - Two GPIO group interrupt modules enable an interrupt based on a programmable pattern of input states of a group of GPIO pins.
  - Four general-purpose timer/counters with capture and match capabilities.
  - One motor control Pulse Width Modulator (PWM) for three-phase motor control.
  - One Quadrature Encoder Interface (QEI).
  - Repetitive Interrupt timer (RI timer).
  - Windowed watchdog timer (WWDT).
  - Ultra-low power Real-Time Clock (RTC) on separate power domain with 256 bytes of battery powered backup registers.
  - Alarm timer; can be battery powered.
- Analog peripherals
  - One 10-bit DAC with DMA support and a data conversion rate of 400 kSamples/s.
  - Two 10-bit ADCs with DMA support and a data conversion rate of 400 kSamples/s. Up to eight input channels per ADC.
- Unique ID for each device.
- Power
  - Single 3.3 V (2.2 V to 3.6 V) power supply with on-chip internal voltage regulator for the core supply and the RTC power domain.
  - RTC power domain can be powered separately by a 3 V battery supply.
  - Four reduced power modes: Sleep, Deep-sleep, Power-down, and Deep power-down.
  - Processor wake-up from Sleep mode via wake-up interrupts from various peripherals.
  - Wake-up from Deep-sleep, Power-down, and Deep power-down modes via external interrupts and interrupts generated by battery powered blocks in the RTC power domain.
  - Brownout detect with four separate thresholds for interrupt and forced reset.
  - Power-On Reset (POR).

7.1.2 LPC4330 Block Diagram

LPC4330 Block Diagram

7.1.3 LPC4330 Memory Map

LPC4330 Memory Map

7.2 Serial Flash Memory

The Bambino 200 uses Quad SPI Flash for it's program and non-volatile data storage. The quad SPI flash has a maximum clock rate of 80 MHz. The Bambino 200 uses a 4M flash and the Bambino 200E uses an 8M flash. Both memories have 4KB sectors.

7.3 USB Power Mux

Texas Instruments TPS2115 auto-switching power mux is used to select between USB0 and USB1. It provides a seamless transition between USB1 and USB0 on the Bambino 200. The TPS2115 includes thermal protection and reverse-conduction blocking.

7.4 Ethernet PHY (Bambino 200E)

The Bambino 200E includes a Micrel KSZ8081 10 Base-T/100 Base-TX Physical Layer Transceiver (PHY). The PHY has a RMII interface to transmit and receive data to the LPC4330's Media Access Controller. It has auto-negotiation to automatically select the highest link-up speed (10/100 Mbps) and duplex (half/full). For further information please see KSZ8081 Data Sheet.

NEXT: User Interfaces, Connectors, and Jumpers

PREVIOUS: Rebuild mbed Firmware

8 User Interfaces, Connectors, and Jumpers

The following image shows where the connectors, headers, and jumpers are located on the Bambino 200.

Bambino 200 User Interfaces, Connectors, and Jumpers

8.1 Power Supply

The Bambino 200 is powered from the USB device port on J4. The Bambino 200E can be powered from the USB device port (J4) or the power jack (J1). A FET is used to automatically select power from J1 should power be applied both the USB device port and J1.

J1 comes standard with a 2.1 mm inner diameter and 5.5 mm outer diameter, positive center tapped female power supply jack. The minimum voltage that can be applied to J1 is 7 VDC and the maximum is 15 VDC. J1 can be changed to a 2 position screw terminal by desoldering the power jack and soldering in a screw terminal. A diode (D2) will protect the Bambino 200E should polarity of the power supply be reversed on the J1 connector. The protection diode is limited to a maximum of 1 amperes through it.

Figure x.x: Power supply connector configurations

8.2 Onboard Peripherals

8.2.1 USB Device

The Bambino 200 comes equipped with a USB Device Port. The Device port is compliant with the USB V2.0 high-speed device specification. It's connector is a micro USB Type AB. The Bambino 200 is powered through the USB device port.

USB Device	
Connector Pin#	**MCU Pin Name**
1	USB0_VBUS (+5.0V)
2	USB0_DM
3	USB0_DP
4	USB0_ID
5	Ground

8.2.2 Boot Jumper

The boot jumper is used to put the Bambino 200 into Device Firmware Upgrade (DFU) mode. This is accomplished by shorting the two pins before power is applied or by shorting the pins and pressing the reset button. For further information please see the Getting Started Section of this manual.

8.2.3 User Buttons and LEDs

The Bambino 200 comes standard with a user push button, a reset push button, and two user LEDs. The user push button is connected to GPIO0[7] with a 22k-ohm pull-up resistor connected to it. User LED1 (yellow) can be illuminated by clearing GPIO3[7] of the LPC4330. User LED2 (green) can be illuminated by clearing GPIO5[5].

User Buttons and LEDs							
BAM200 Peripheral	**MCU Pin Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
LED1	P6_11	GPIO3[7]	0	T2_MAT3	5		
LED2	P2_5	GPIO5[5]	4	T3_MAT2	6	USB0_IND0	7
BTN1	P2_7	GPIO0[7]	0				

8.2.4 Serial Flash Memory

The Bambino 200 uses serial flash for program and nonvolitile data storage. It uses the LPC43030's Quad SPI Flash Iinterface (SPIFI). The SPIFI interface has data rates up to 52 MB per second. The Bambino 200 comes standard with 4M of flash and the Bambino 200E comes standard with 8M of flash.

Serial Flash Memory		
MCU Pin Name	**Peripheral**	**SCU Func**
P3_3	SPIFI_SCK	3
P3_4	SPIFI_SIO3	3
P3_5	SPIFI_SIO2	3
P3_6	SPIFI_MISO	3
P3_7	SPIFI_MOSI	3
P3_8	SPIFI_CS	3

8.2.5 10/100 Ethernet (200E Only)

The Bambino 200E is equipped with a fully-integrated 10/100 Mbps Ethernet port. The Media Access Control (MAC) is implemented in the LPC4330 and the Physical (PHY) layer is implemented with Micrel?s KSZ8031. J3 is the RJ-45 connector and it has integrated magnetics and LEDs completes the Ethernet sub-system. Please see the KSZ8031 data sheet for further information on the PHY and the LPC4330 User's Manual for the MAC.

Ethernet		
MCU Pin Name	**Peripheral**	**SCU Func**
P0_0	ENET_RXD1	2
P0_1	ENET_TX_EN	6
P1_15	ENET_RXD0	2
P1_16	ENET_RX_DV	7
P1_17	ENET_MDIO	3
P1_18	ENET_TXD0	2
P1_19	ENET_REF_CLK	0
P1_20	ENET_TXD1	3
P2_0	ENET_MDC	7

8.2.6 MICRO SD (200E Only)

The microSD socket (J2) enables micro-secure-digital memory cards to be plugged into the Bambino 200E microcontroller board. The microSD card allows the user the ability of a standard removable media for transferring data to and from the Bambino 200E. The LPC4330 interfaces to the microSD card through the Secure Digital Input Output card interface.

Micro SD Card		
MCU Pin Name	**Peripheral**	**SCU Func**
CLK2	SD_CLK	4
P1_6	SD_CMD	7
P1_9	SD_DAT0	7
P1_10	SD_DAT1	7
P1_11	SD_DAT2	7
P1_12	SD_DAT3	7
P1_13	SD_CD	7

8.2.7 XBEE (200E Only)

The XBEE socket adds wireless support to the Bambino 200E. Digi International has several different versions of XBEE modules with different wireless protocols in the same physical footprint. Zigbee and WiFi are a couple of protocols supported by Digi International's XBEE modules. Please see Digi International's website for further details. The XBEE signals are shared with socket 5. Socket 5 should not be used if the XBEE module is being used.

XBEE		
MCU Pin Name	**Peripheral**	**SCU Func**
P5_6*	U1_TXD	4
P1_14*	U1_RXD	1
P5_2*	U1_RTS	4
P5_4*	U1_CTS	4

P5_1*	GPIO2[10]	0
*Shared between XBEE and Socket 5		

8.3 Sockets

8.3.1 Socket 1 OSX

Socket 1's type for Gadgeteer is O (Analog Output), S (SPI Interface), and X (3 GPIO). The Analog out function is a 10-bit DAC and it is also shared with Socket 3. The SPI function is the LPC4330's SSP0 port.

Socket 1: O, S, X							
		Gadgeteer Type					
	MCU Pin	**O**		**S**		**X**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	
2		5.0V		5.0V		5.0V	
3	P4_5	GPIO2[5]	0	GPIO2[5]	0	GPIO2[5]	0
4	P5_0	GPIO2[9]	0	GPIO2[9]	0	GPIO2[9]	0
5	P4_4	DAC	A	GPIO2[4]	0	GPIO2[4]	0
6	P1_0			SSP0_SSEL	5		
7	P1_2			SSP0_MOSI	5		
8	P1_1			SSP0_MISO	5		
9	P3_0			SSP0_SCK	4		
10		Ground		Ground		Ground	

8.3.2 Socket 2 IPSUY

Socket 2's type for Gadgeteer is I (I2C), P (PWM), S (SPI), U (UART), and Y (7 GPIO). The I2C or SPI function are generated by the Serial General Purpose Input Output (SGPIO) peripherial on the LPC4330. The UART function is the LPC4330's U0 port.

Socket 2: I, P, S, U, Y											
		Gadgeteer Type									
	MCU Pin	**I**		**P**		**S**		**U**		**Y**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V		3.3V		3.3V	
2		5.0V		5.0V		5.0V		5.0V		5.0V	
3	P1_7	GPIO1[0]	0	GPIO1[0]	0	GPIO1[0]	0	GPIO1[0]	0	GPIO1[0]	0
4	P6_4					GPIO3[3]	0	U0_TXD	2	GPIO3[3]	0
5	P6_5					GPIO3[4]	0	U0_RXD	2	GPIO3[4]	0

6	P4_6	GPIO2[6]	0	GPIO2[6]	0	SGPIO12	7	GPIO2[6]	0	GPIO2[6]	0
7	P4_8			SGPIO13	7	SGPIO13	7			GPIO5[12]	5
8	P4_9	SGPIO14	7	SGPIO14	7	SGPIO14	7			GPIO5[13]	4
9	P4_10	SGPIO15	7	SGPIO15	7	SGPIO15	7			GPIO5[14]	4
10		Ground		Ground		Ground		Ground		Ground	

8.3.3 Socket 3 AOP

Socket 3 type for Gadgeteer is A (Analog In), O (Analog Out), and P (PWM). The analog input function are the LPC4330's ADC0 and ADC1 port. The Analog out is a 10-bit DAC and it is also shared with Socket 1. The PWM function uses the LPC4330's motor control PWM.

Socket 3: A, O*, P							
		Gadgeteer Type					
	MCU Pin	**A**		**P**		**O***	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	
2		5.0V		5.0V		5.0V	
3	P4_3	GPIO2[3]/ADC0_0	0/A	GPIO2[3]	0	GPIO2[3]	0
4	P4_1	GPIO2[1]/ADC0_1	0/A	GPIO2[1]	0		
5*	ADC0_0*	ADC1_0*	A*	DAC*	A*		
5	ADC0_2	ADC1_2	A				
6	P4_2	GPIO2[2]	0			GPIO2[2]	0
7	P4_0					MCOA0	1
8	P5_5					MCOA1	1
9	P5_7					MCOA2	1
10		Ground		Ground		Ground	
* Denotes REV A of the Printed Circuit Board							

8.3.4 Socket 4 AIY

Socket 4's type for Gadgeteer is A (Analog In), I (I2C), and Y (7 GPIO). The analog input function are the LPC4330's ADC0 and ADC1 port. The I2C function is the LPC4030's I2C1 port.

Socket 4: A, I, Y							
		Gadgeteer Type					
	MCU Pin	**A**		**I**		**Y**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	
2		5.0V		5.0V		5.0V	

3	P7_4	GPIO3[12]/ADC0_4	0/A	GPIO3[12]	0	GPIO3[12]	0
4	P7_5	GPIO3[13]/ADC0_3	0/A			GPIO3[13]	0
5	P7_7	ADC1_6	A			GPIO3[15]	0
6	P7_6	GPIO3[14]	0	GPIO3[14]	0	GPIO3[14]	0
7	P7_2					GPIO3[10]	0
8	P2_3			I2C1_SDA	1	GPIO5[3]	4
9	P2_4			I2C1_SCL	1	GPIO5[4]	4
10		Ground		Ground		Ground	

8.3.5 Socket 5 KUX

Socket 5's type for Gadgeteer is K (UART+ Handshaking), U (UART), and X (3 GPIO). The UART and handshaking signal functions are the LPC4330's U1 port.

Socket 5: K, U, X							
		Gadgeteer Type					
	MCU Pin	**K**		**U**		**X**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	
2		5.0V		5.0V		5.0V	
3	P5_1*	GPIO2[10]	0	GPIO2[10]	0	GPIO2[10]	0
4	P5_6*	U1_TXD	4	U1_TXD	4	GPIO2[15]	0
5	P1_14*	U1_RXD	1	U1_RXD	1	GPIO1[7]	0
6	P5_2*	U1_RTS	4	GPIO2[11]	0		
7	P5_4*	U1_CTS	4				
8							
9							
10		Ground		Ground		Ground	
*Shared between XBEE and Socket 5							

8.3.6 Socket 6 ISY (200E Only)

Socket 6's type for Gadgeteer is I (I2C), S (SPI), and Y (7 GPIO). The I2C or SPI functions are generated by the Serial General Purpose Input Output (SGPIO) peripherial on the LPC4330.

Socket 6: I, S, Y							
		Gadgeteer Type					
	MCU Pin	**I**		**S**		**Y**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	

2		5.0V		5.0V		5.0V	
3	P6_1	GPIO3[0]	0	GPIO3[0]	0	GPIO3[0]	0
4	P1_8			GPIO1[1]	0	GPIO1[1]	0
5	P5_3			GPIO2[12]	0	GPIO2[12]	0
6	P9_6	GPIO4[11]	0	SGPIO8	6	GPIO4[11]	0
7	P6_6			SGPIO5	2	GPIO0[5]	0
8	P6_7	SGPIO6	2	SGPIO6	2	GPIO5[15]	4
9	P6_8	SGPIO7	2	SGPIO7	2	GPIO5[16]	4
10		Ground		Ground		Ground	

8.3.7 Socket 7 UY (200E Only)

Socket 7's type for Gadgeteer U (UART) and Y (7 GPIO). The UART function is generated by the Serial General Purpose Input Output (SGPIO) peripherial on the LPC4330.

Socket 7: U, Y					
		Gadgeteer Type			
	MCU Pin	**U**		**Y**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V	
2		5.0V		5.0V	
3	P2_1	GPIO5[1]	4	GPIO5[1]	4
4	P9_5	SGPIO3	0	GPIO5[18]	4
5	P6_3	SGPIO4	6	GPIO3[22]	0
6	P6_2	GPIO3[1]	0	GPIO3[1]	0
7	P2_2			GPIO5[2]	4
8	P6_10			GPIO3[6]	0
9	P7_3			GPIO3[11]	0
10		Ground		Ground	

8.3.8 Socket 8 DHI (200E Only)

Socket 8's type for Gadgeteer is D (USB Device), H (USB Host), and I (I2C). The USB Host and Device function are the LPC4330's USB1 port. The USB1 port is a full speed USB 2.0 port. The I2C function is the LPC4330's I2C1 port.

Socket 8: D, H, I							
		Gadgeteer Type					
	MCU Pin	**D**		**H**		**I**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	

2		5.0V		5.0V		5.0V	
3	P2_8	GPIO5[7]	4	GPIO5[7]	4	GPIO5[7]	4
4	USB1_DM	USB1_DM	0	USB1_DM	0		
5	USB1_DP	USB1_DP	0	USB1_DP	0		
6	P2_6	GPIO5[6]	4			GPIO5[6]	4
7	P6_9	GPIO3[5]	0				
8	I2C0_SDA					I2C0_SDA	0
9	I2C0_SCL					I2C0_SCL	0
10		Ground		Ground		Ground	

8.3.9 Socket 9 Y (200E Only)

Socket 9's type for Gadgeteer is Y (7 GPIO).

Socket 9: Y			
		Gadgeteer Type	
	MCU Pin	**Y**	
Pin #	**Name**	**Peripheral**	**SCU Func**
1		3.3V	
2		5.0V	
3	P2_9	GPIO1[10]	0
4	P3_2	GPIO5[9]	0
5	P3_1	GPIO5[8]	0
6	P2_12	GPIO1[12]	0
7	P2_13	GPIO1[13]	0
8	P7_1	GPIO3[9]	0
9	P7_0	GPIO3[8]	0
10		Ground	

8.3.10 Socket 10 SUX (200E Only)

Socket 10's type for Gadgeteer is S (SPI), U (UART), and X (3 GPIO). The UART function is the LPC4330's U2 port. The SPI function is the LPC4330's SSP1 port.

Socket 10: S, U, X							
		Gadgeteer Type					
	MCU Pin	**S**		**U**		**X**	
Pin #	**Name**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**	**Peripheral**	**SCU Func**
1		3.3V		3.3V		3.3V	

2		5.0V		5.0V		5.0V	
3	P6_12	GPIO2[8]	0	GPIO2[8]	0	GPIO2[8]	0
4	P2_10	GPIO0[14]	0	U2_TXD	2	GPIO0[14]	0
5	P2_11	GPIO1[11]	0	U2_RXD	2	GPIO1[11]	0
6	P1_5	SSP1_SSEL	5	GPIO1[8]	0		
7	P1_4	SSP1_MOSI	5				
8	P1_3	SSP1_MISO	5				
9	PF_4	SSP1_SCK	0				
10		Ground		Ground		Ground	

8.4 Field Installable Options

The coin battery holder on the bottom of the Bambino 200 is not populated at production time. The Cortex M JTAG is also not populated when the board is built. Both parts may be purchased from DigiKey and Mouser.

Field Installable Options Parts List				
Option	**Manufacturer**	**Part #**	**Digikey Part#**	**Mouser Part #**
Coin Battery Holder	Keystone	3002	3002K-ND	534-3002
Coin Battery	Panasonic	CR2032	P189-ND	658-CR2032
Cortex M JTAG	Sullins Connector Solutions	GRPB052VWQS-RC	S9012E-05-ND	

8.4.1 Coin Battery

The Bambino 200's microcontroller has a built in real-time clock calendar that can be battery backed by supplying 2.2 VDC to 3.6 VDC to the VBAT pin on the LPC4330. A battery holder can be added to the bottom of the board to power the VBAT pin with a coin cell battery. The battery holder is manufactured by Keystone and it's part number is 3002. The battery holder accepts CR2032 series coin cells. Power is only drawn from the battery when the power is off to the Bambino 200.

8.4.2 Cortex M JTAG

A JTAG port (J5) can be added for software download and debugging. The JTAG port allows users to set break points and to single step through their program. For detailed information on the operation of the JTAG port and TAP controller, please refer to IEEE Standard 1149.1-Test Access Port and Boundary-Scan Architecture.

Cortex M JTAG	
Connector Pin#	**Pin Name**
1	VCC (+3.3V)
2	TMS/SWDIO
3	Ground
4	TCK/SWDCLK
5	Ground
6	TDO/SWO
7	No Connect

8	TDI
9	Ground
10	RESET

NEXT: Mechanical and Electrical Characteristics

PREVIOUS: Hardware

9 Mechanical and Electrical Characteristics

9.1 Absolute Minimum and Maximum Ratings

Characteristic	Minimum	Maximum	Unit
Voltage on J1	7.0	15.0	VDC
Combined current available on +3.3V and +5.0V pin on Gadgeteer socket when powered through J1		500	mA
Combined current available on +3.3V and +5.0V pin on Gadgeteer socket when powered through USB		250	mA
Voltage on VBAT (Coin Cell Battery Holder)	0.0	3.3	VDC
Voltage on ADC	0.0	3.3	VDC
Voltage on Digital Input	0.0	5.0	VDC
Operating Temperature	0	70	ºC
Storage Temperature	-50	125	ºC

The Bambino 200 is currently available for commercial temperature ranges. Contact the Micromint sales department if you require support for industrial temperature ranges.

9.2 Mechanical Dimensions

Below is the physical dimensions for the Bambino 200. The mounting holes will accept a #4 size screw.

DIM	Inches	Millimeters	DIM	Inches	Millimeters	DIM	Inches	Millimeters
A	4.0	101.6	**E**	0.54	13.72	**I**	1.2	30.48
B	0.88	22.35	**F**	0.88	20.35	**J**	0.29	7.37
C	0.55	13.97	**G**	0.14	3.56	**K**	0.79	20.1
D	2.3	58.42	**H**	0.64	16.256	**L**	0.48	12.19

Bambino 200 Mechanical Dimensions

NEXT: References

PREVIOUS: User Interfaces, Connectors, and Jumpers

10 References

This section outlines material that may be useful for further reading.

10.1 Documents

LPC43XX User Manual
http://www.nxp.com/documents/user_manual/UM10503.pdf
This user manual provides reference information for the NXP LPC43XX microcontrollers. All MCU registers are documented in the user manual.
.NET Gadgeteer: A Platform for Custom Devices
http://research.microsoft.com/pubs/163162/Gadgeteer%20Pervasive%202012%20Proof.pdf
Describes the .NET Gadgeteer system in detail, explaining a number of key design decisions and reporting use by new users and experts alike.

.NET Gadgeteer Module Builder's Guide version 1.10
http://gadgeteer.codeplex.com/downloads/get/665907
Explains mainboard socket types and peripheral buses used by Gadgeteer-compatible modules, assemblies that are written in managed code to provide interfaces to applications, and mechanical design requirements for components.
mbed Texbook Overview
http://mbed.org/cookbook/Textbook
Overview of the textbook "Fast and Effective Embedded Systems Design: Applying the ARM mbed" by Rob Toulson and Tim Wilmshurst.

mbed Texbook Course Notes
http://mbed.org/cookbook/Course-Notes
Course notes for textbook referenced above.

10.2 Books

Microsoft .NET Gadgeteer: Electronics Projects for Hobbyists and Inventors

ISBN: 0071797955 Publisher: McGraw-Hill/TAB Electronics; (November, 2012)

Practical projects with Microsoft .NET Gadgeteer. Learn how to choose components, write Gadgeteer applications, connect your creations to the Web, and troubleshoot.

Expert .NET Micro Framework

ISBN: 9781430223870 Publisher: Apress; (September, 2012)

Learn how use the Microsoft .NET Micro Framework to create effective embedded applications.

Fast and Effective Embedded Systems Design: Applying the ARM mbed

ISBN: 0080977685 Publisher: Newnes; (August, 2012)

Introduction to embedded systems design, using the ARM mbed and C programming language as development tools.

The Definitive Guide to ARM® Cortex-M3 and Cortex-M4 Processors, Third Edition

by Joseph Yiu

ISBN: 0124080820 Publisher: Newnes (November, 2013)

Overview of the processor and instruction set architecture of the ARM® Cortex®-M3 and Cortex®-M4 processors. Several code examples using IAR, Keil, gcc and CooCox CoIDE.

The Designer's Guide to the Cortex-M Processor Family: A Tutorial Approach

by Trevor Martin

ISBN: 0080982964 Publisher: Newnes (May, 2009)

Tutorial-based book giving the key concepts required to develop programs in C with a Cortex M- based processor.

ARM System Developer's Guide: Designing and Optimizing System Software

ISBN: 1558608745 Publisher: Morgan Kaufmann; (March, 2004)

In-depth overview of the ARM architecture with examples that outline impact of programming practices on performance, power and cost.

10.3 Useful Web Links

Micromint Web Site
http://www.micromint.com/
Product information and software updates for the Electrum SBCs.

NXP?s Web Site
http://www.nxp.com/
Manuals, Erratas, and application notes for NXP MCUs
.NET Micro Framework Platform SDK
http://msdn.microsoft.com/en-us/library/ee436350.aspx
Online documentation for the Micro Framework SDK

.NET Gadgeteer Core Reference
http://netmf.com/gadgeteer/docs/GadgeteerCore/2.42.600_NETMF4.2/
Online documentation for the Gadgeteer API
mbed Community
http://mbed.org/
mbed community site with documentation, code examples and forums

PREVIOUS: Mechanical and Electrical Characteristics