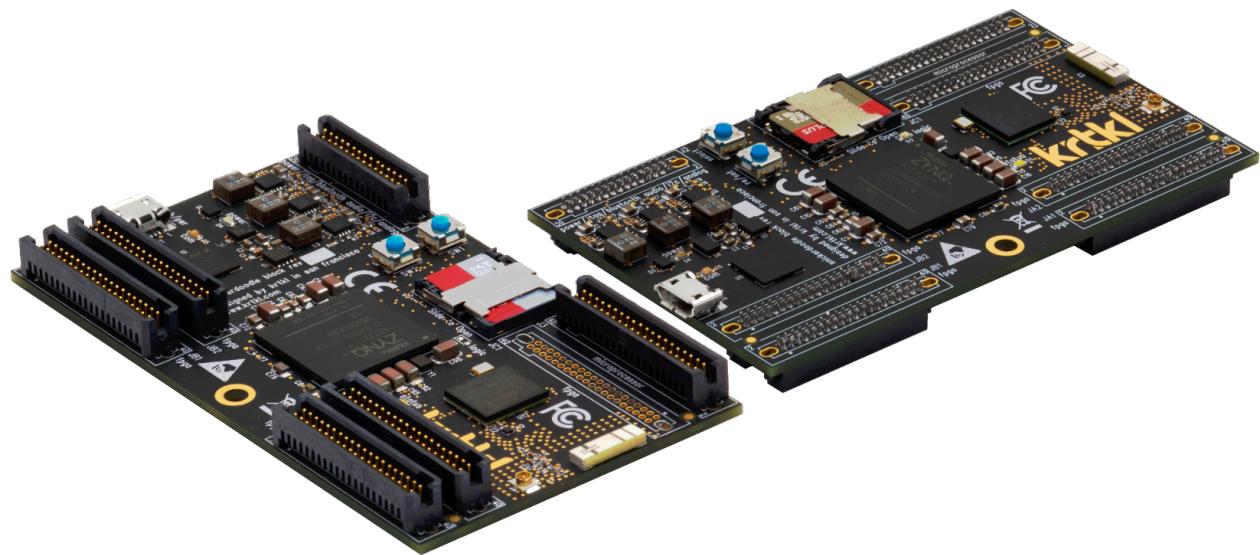


snickerdoodle



SNICKERDOODLE USER GUIDE

snickerdoodle User Guide
Copyright © 2016 krtkl inc.

krtkl inc.
350 TOWNSEND STREET
SUITE 301A
SAN FRANCISCO, CA 94107
UNITED STATES

The information contained in this document is made available under a Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license please visit <http://creativecommons.org/licenses/by-sa/4.0>

Contents

GETTING STARTED	1
1 Connecting snickerdoodle	2
2 Programming snickerdoodle	4
HARDWARE ARCHITECTURE	5
3 Overview	6
<i>Connectors</i>	7
4 Power Supplies	9
5 Zynq AP SoC Architecture	10
6 Zynq Configuration	11
<i>Boot Process</i>	11
<i>microSD Boot Mode</i>	12
<i>QSPI Boot Mode</i>	12
<i>JTAG Boot Mode</i>	13
7 DDR Memory	14
8 USB UART Bridge (Serial Port)	15
9 microSD Slot	16
10 Processor Subsystem Connectivity	17
11 FPGA Connectors	18
12 Wireless	20
APPENDICES	22
A snickerdoodle High-Level Block Diagram	23

List of Figures

1.1	<i>krtkl connect Login and Device Selection</i>	2
1.2	<i>krtkl connect Settings and Wi-Fi Configuration</i>	3
2.1	<i>krtkl connect Project Load</i>	4
3.1	<i>snickerdoodle Connector Types and Pin Numbering</i>	7
3.2	<i>snickerdoodle Connector Designation</i>	8
9.1	<i>Loaded microSD Card Cage</i>	16
10.1	<i>Processor Subsystem Connector</i>	17
10.2	<i>Processor Subsystem Multiplexed Input/Output (MIO) on J3</i>	17
11.1	<i>FPGA Connectors</i>	19
12.1	<i>Wireless Radio Antenna UFL Connector</i>	20
A.1	<i>snickerdoodle High-Level Block Diagram</i>	23
B.1	<i>FPGA Connector JA1</i>	24
B.2	<i>FPGA Connector JA2</i>	25
B.3	<i>FPGA Connector JB1</i>	25
B.4	<i>FPGA Connector JB2</i>	26
B.5	<i>FPGA Connector JC1</i>	26

List of Tables

3.1	<i>snickerdoodle Features and Upgrade Options</i>	6
3.2	<i>snickerdoodle Connector Descriptions</i>	8
5.1	<i>Zynq Z-7010 and Z-7020 Common Features</i>	10
5.2	<i>snickerdoodle Zynq SoC Features</i>	10
6.1	<i>Boot Mode Configuration Pins</i>	12
6.2	<i>MIO Boot Mode Configuration Pins</i>	12
6.3	<i>Connector J2</i>	13
7.1	<i>snickerdoodle Memory Specifications</i>	14
11.1	<i>FPGA Connector Specifications</i>	18
12.1	<i>snickerdoodle Wireless Module Features</i>	20

GETTING STARTED

NOTE: iOS app obsoleted
Please perform initial connection using snickerdoodle hosted WiFi access point.

SSID is snickerdoodleAP-XXXXXX
WPA2 password is “snickerdoodle”

default login is “snickerdoodle”
default password is “snickerdoodle”

please see snickerdoodle-wireless.pdf for snickerdoodle WiFi station configuration instructions

Connecting snickerdoodle

To connect your snickerdoodle to the internet, you will need to connect it to an internet connected access point (AP) using the on-board WiFi. To do this, you must provide your snickerdoodle with the name and credentials of the access point you would like to connect to, using the krtkl connect app.

The connection between the app and your snickerdoodle is done using Bluetooth low-energy (BLE). Using BLE, your mobile device will detect nearby snickerdoodles and display them by their serial number.

After pairing your snickerdoodle with your device, you can use the krtkl connect app to specify the name and password of the network you’d like to connect to. You can also change some configuration settings, such as giving your snickerdoodle a personal name or alias, from the settings menu.



Figure 1.1: krtkl connect Login and Device Selection

After navigating to the Wi-Fi settings menu, you will be able to select or enter the

NOTE: iOS app obsoleted
Please perform initial connection using snickerdoodle hosted WiFi access point.

work that you
initialization will be
internet to ensure
doodle is able to

SSID is snickerdoodleAP-XXXXXX
WPA2 password is “snickerdoodle”

default login is “snickerdoodle”
default password is “snickerdoodle”

Getting snickerdoodle to work that you
initialization will be
internet to ensure
doodle is able to

please see snickerdoodle-wireless.pdf for snickerdoodle WiFi station configuration instructions

STEP 1 Select the device you'd like to configure from the list in the krtkl connect app

STEP 2 Navigate to the Wi-Fi menu from within the device settings and select or enter the network you'd like snickerdoodle to connect

STEP 3 Enter the network's security information (security type and password) and select 'Connect'.

As shown in [Figure 1.2](#), snickerdoodle will suggest a list of networks that it detects and the krtkl connect app will display those networks or you may manually enter the network information. krtkl connect will encrypt and communicate the network information to the snickerdoodle and it will begin connecting. From this menu, you also have the ability to edit the list of trusted networks.

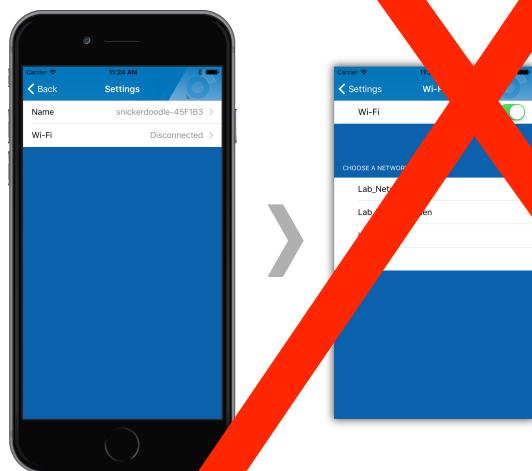


Figure 1.2: krtkl connect Settings and Wi-Fi Configuration

NOTE: iOS app obsoleted
Please perform initial connection using snickerdoodle hosted WiFi access point.

SSID is snickerdoodleAP-XXXXXX
WPA2 password is “snickerdoodle”

default login is “snickerdoodle”
default password is “snickerdoodle”

please see snickerdoodle-wireless.pdf for snickerdoodle WiFi station configuration instructions

Programming a snickerdoodle

After successfully connecting to snickerdoodle, you can download existing built projects using the krtkl connect app. Projects will be listed and selectable from the projects menu and will load using the connected Wi-Fi access point. Figure 2.1 shows an example list of loaded projects that can be installed onto snickerdoodle.

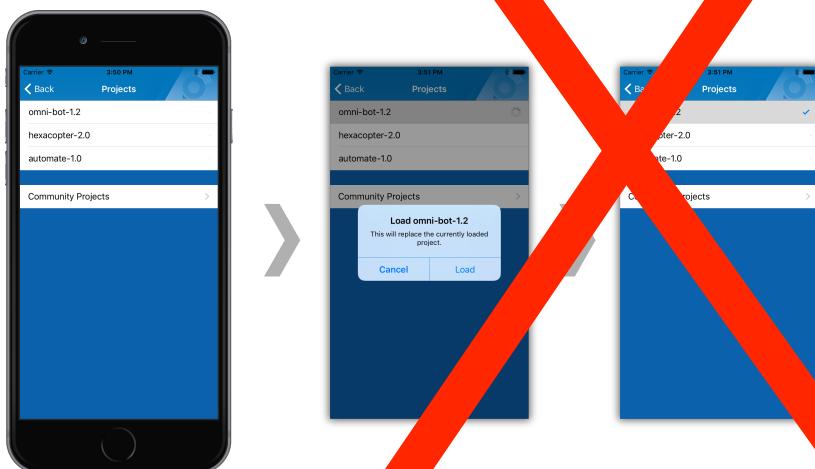


Figure 2.1: krtkl connect Project Load

From the ‘Projects’ menu you can also select and load a variety of public community projects that can get snickerdoodle running without creating a custom project. This is especially useful for testing functionality and configuring snickerdoodle for a third-party application.

If WiFi is unavailable, projects can be loaded using the internet connection on your mobile device and transmitted to snickerdoodle over BLE. The throughput of BLE project transfer is much lower than WiFi (about 250kb/s for BLE and 80Mb/s for WiFi).

HARDWARE ARCHITECTURE

3

Overview

	snickerdoodle	upgrade options
Chipset	Xilinx® Zynq® -7010	Xilinx Zynq-7020
CPU	32-Bit dual-core ARM® Cortex™-A9 w/640kB cache and dual 128-bit NEON™ coprocessors	
Performance	3335 DMIPS/2668 MFLOPS @ 667 MHz	4330 DMIPS/3464 MFLOPS @ 866 MHz
Flash	16 MB XIP NOR and up to 200 GB SDIO NAND via captive microSD card cage	
DRAM	512 MB @ 25.6 Gbps	1 GB @ 25.6 Gbps
SRAM	256 kB @ 28.4 Gbps	256 kB @ 36.9 Gbps
CPU to FPGA/Bandwidth	896-bit @ 150 MHz AXI3/134.4 Gbps	896-bit @ 200 MHz AXI3/179.2 Gbps
FPGA Programmable Logic	430k gates/17600 LUT-6	1.3M gates/53200 LUT-6
32-bit Performance	143150 MIPS @ 350 MHz	587575 MIPS @ 475 MHz
Distributed RAM	270kB/3354 Gbps	630kB/10275 Gbps
DSP Units/Performance	80 @ 464 MHz/74240 MMACs	220 @ 628 MHz/276320 MMACs
Total User GPIO	142	167
FPGA GPIO/performance	16 ADC/100 reconfigurable/46.2 Gbps	16 ADC/125 reconfigurable/75.7 Gbps
Fixed GPIO	28 GPIO, 4 I2S audio, 2 I2C, 5 1-Wire, 1 ADC, 2 DAC	
Wireless	150Mbps SISO 2.4GHz 802.11b/g/n 3Mbps dual-mode Bluetooth® 4.1 Classic+EDR/BLE, dual-band antenna, switched U.FL ports	150Mbps MIMO 2.4GHz/5GHz 802.11a/b/g/n
Serial Interfaces	2 gigabit ethernet, 2 CAN, 2 I2C, SPI, UART, USB 2.0 high-speed, microUSB console/JTAG	
Analog Interfaces	dual 1MSPS 12-bit ADCs w/16 channel multiplexer, dual 1MSPS 12-bit DACs	
Other Peripherals	5 LEDs, 2 pushbuttons, secure cryptographic key/certificate storage	
Software Support	krtkl iOS app, Linux, FreeRTOS, ROS, ArduPilot, Wiring, bare-metal C/C++, Vivado	
Power/Dimensions	+5V via microUSB or 3.7V-17V via power pins/3.5" x 2.0" (88.9mm x 50.8mm)	

snickerdoodle is a compact, modular, highly-integrated embedded development platform that combines powerful processing capability with wireless connectivity and flexible configurability. snickerdoodle utilizes a Xilinx Zynq-7000 series All-Programmable SoC which combines a dual-core ARM Cortex-A9 processor with a Xilinx 7-Series FPGA. The ARM core provides a well known architecture for application development and is an ideal target for an embedded Linux system.

Table 3.1: snickerdoodle Features and Upgrade Options

The tightly-integrated FPGA fabric allows for connection with a wide variety of peripheral devices as well as the realization of custom logic for high-speed application-specific processing. snickerdoodle comes with either a Zynq Z-7010 or a more powerful Zynq Z-7020 available as an optional upgrade.

Connectors

The USB power input and serial console are both accessed using the USB micro-B receptacle, designated J1. Power input is described in more detail in [Chapter 4](#). The most notable connectors on snickerdoodle are the 0.050 in. double row headers. There are two possible connector orientations:



J2 is a Samtec TFM-115-01-F-D-A/SFM-115-L1-F-D-A. J3, JA1, JA2, JB1, JB2 and JC1 are Samtec TFM-120-01-F-D-A/SFM-120-L1-F-D-A

TFM Top-facing shrouded male terminal header

SFM Bottom-facing female socket strip

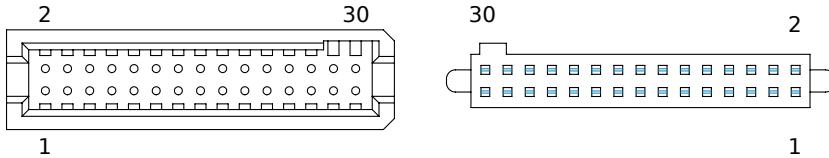


Figure 3.1: snickerdoodle Connector Types and Pin Numbering (TFM left, SFM right)

CAUTION The pin numbering appears as a mirror image between the TFM and SFM connectors as they are intended to be mounted on opposite sides of the board. This pin numbering is reflected in all tables in this manual.

Figure 3.2 shows a male pin configuration mounted on the top-side of the board for easy access with wire-to-board connectors. Female connectors mounted on the bottom side of the board can be used for modular connection to baseboards. Table 3.2 shows the connectors and their descriptions.

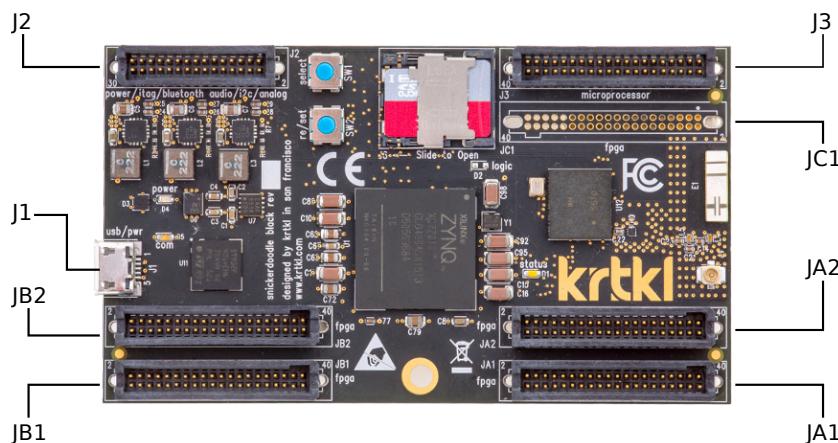


Figure 3.2: snickerdoodle Connector Designation

Connector	Description
J1	USB and Power Input
J2	Power, JTAG, Bluetooth Audio, I ² C, Analog
J3	Microprocessor subsystem
JA1	FPGA
JA2	FPGA
JB1	FPGA
JB2	FPGA
JC1	FPGA

Table 3.2: snickerdoodle Connector Descriptions

4

Power Supplies



WARNING Plugging snickerdoodle into a power source that does not conform to the electrical requirements can damage the snickerdoodle.

snickerdoodle can be powered directly through the USB micro-B connector (J1) or through power input pins located on J2. When plugging snickerdoodle into USB power, the power source should provide adequate electrical current for the application. When connected through the power input pins on J2 (pins 29 and 30), the input source should supply between +3.7V and +17V.

Each FPGA connector has a +3.3V reference tied to pin 1 and the PS connector has a +1.8V signal reference tied to pin 1. snickerdoodle can be configured to drive a variety of electronic circuits. Driving signals originating from snickerdoodle should be connected to high impedance inputs and not used as a power source. The signals coming from snickerdoodle can be used to control a variety of electronic circuits including motor controllers and charge circuits.

High performance applications should be connected through J2 with a maximum of 3.2A per power supply pin. A +12V supply is recommended for very high performance applications. The +1.8V source on J2 will supply a total of 0.5A to connected devices/accessories. The +3.3V outputs on the FGPA connectors (JA1, JA2, JB1, JB1 and JC1) are able to supply a total of 1A to connected devices/accessories.

5

Zynq AP SoC Architecture

The Zynq AP SoC is divided into two distinct subsystems: The Processing System (PS), and the Programmable Logic (PL). Figure 3 shows an overview of the Zynq AP SoC architecture, with the PS colored light green and the PL in yellow. Note that the PCIe Gen2 controller and Multigigabit transceivers are not available on the Zynq7010 device.

Common specifications between the Z-7010 and the Z-7020:

Processor Core	Dual ARM Cortex-A9 MPCore with CoreSight
L1 Cache	32 KB Instruction, 32 KB Data per processor
L2 Cache	512 KB
On-Chip Memory	256 KB
External Memory Support	2x Quad-SPI, NAND, NOR
DMA Channels	8 (4 dedicated to Programmable logic)
Peripherals	2x UART, 2x CAN 2.0B, 2x I ² C, 2x SPI, 4x 32b GPIO
Security	RSA Authentication, AES and SHA 256-bit Decryption and Authentication for Secure Boot

Table 5.1: Zynq Z-7010 and Z-7020 Common Features

	Z-7010-1	Z-7020-3
Maximum Frequency	667 MHz	866 MHz
Programmable Logic Cells	28K Logic Cells	85K Logic Cells

Table 5.2: snickerdoodle Zynq SoC Features

6

Zynq Configuration

The hardware architecture of the Zynq allows full control of the FPGA fabric through the processing subsystem. The processor acts as a master to the programmable logic so that the state of the SoC and the boot process is not dependent on the FPGA configuration. In fact, the FPGA bitstream does not need to be defined prior to booting the device. This makes the boot process more similar to a microcontroller than an FPGA. A boot image is loaded and executed upon startup which loads a First Stage Bootloader (FSBL), an optional bitstream for configuring the programmable logic and, finally, a user-defined processing subsystem application or operating system.

Boot Process

The boot process is typically defined by a primary boot loader such as U-Boot which is responsible for initiating each stage of the boot process. The stages of the boot process are as follows:

STEP 1 Upon startup or reset, one of the processing cores executes a set of code from read-only memory (ROM) which initiates the boot process. The ROM is responsible for initiating processing of the first stage boot loader (FSBL) from a designated set of non-volatile memory (e.g., microSD card, flash). The FSBL should be included in a Zynq Boot Image which contains the data for the remaining boot process stages.

STEP 2 Once the boot process is handed off to the FSBL, it configures the processing system and loads a bitstream to configure the programmable logic (if a bitstream exists within the boot image). The FSBL then loads any user-defined application/system into memory and prepares to finish the boot process.

STEP 3 The user application/system is loaded during the final stage of the boot process. During this stage a secondary boot loader can be used to load a operating system (most commonly Linux) or a user defined application can be directly loaded by the FSBL. For a more thorough explanation of the boot process, refer to Chapter 6 of the Zynq Technical Reference Manual.

MIO Bank 500							
2	3	4	5	6	7	8	
BOOT_MODE							
Device		pll	V				

Table 6.1: Boot Mode Configuration Pins

snickerdoodle can be booted using three difference sources. [Table 6.1](#) shows the boot mode configuration pins that can be used to specify the boot source. The specific configuration pins for determining the three available boot sources are shown in [Table 6.2](#) (MIO [4:5]).

MIO 4	MIO 5
JTAG	0
QSPI	0
SD Card	1

Table 6.2: MIO Boot Mode Configuration Pins

The boot mode can be configured wirelessly, through the krtkl app.

microSD Boot Mode

The most common way to boot snickerdoodle is from a microSD card installed in J6 (described in more detail in [Chapter 9](#)). The microSD card slot provides a removable, upgradable, high-volume source of non-volatile storage from which a range of projects can be loaded. A microSD card can easily be loaded with a boot image that provides a small user application or a general purpose operating system such as Linux. A FAT32 formatted microSD card can be created and loaded with a bootable image by following a simple process:

1. Format the microSD card with a FAT32 file system using a host computer.
2. Copy the Zynq Boot Image to the microSD card.
3. Eject the microSD card from your computer and insert it into connector J6.
4. Power on snickerdoodle using a conforming power supply connected to J1 or J2.

QSPI Boot Mode

snickerdoodle has a 16 Mbyte quad-SPI NOR flash (Micron N25Q128A11ESE40F). The flash memory can be used to provide non-volatile storage of data and pro-

gram code. The flash can be used as general storage for initialization of the processing subsystem as well as configuration of the FPGA by storing bitstream data that has been wrapped in a bootable image as described in [Chapter 6](#).

JTAG Boot Mode

A JTAG connection can be made through J2 where the signals have been made available on pins 17-26, as shown in [Table 6.3](#). These connections can be accessed directly from J2 or broken out when snickerdoodle is mounted on a baseboard. The JTAG connection can be used to boot snickerdoodle when connected to a host computer.

VP_0	1	2	VN_0
DAC_OUT1/ADC_IN4	3	4	DAC_OUT2/ADC_IN5
GND	5	6	GND
SMD_I2C_SCL7	7	8	SMB_I2C_SDA7
GND	9	10	GND
BT_AUD_CLK	11	12	BT_AUD_FSYNC
BT_AUD_OUT	13	14	BT_AUD_IN
MCU_SWCLK/PA13	15	16	MCU_SWDIO/PA14
MCU_nRST	17	18	MCU_VDDIO/VCCO_0 (+3.3V)
Zynq nRST (PS_SRST_B_501)	19	20	GND
TCK_0	21	22	TDI_0
TMS_0	23	24	TDO_0
MCU_VDDIO2/VCCO_0 (+3.3V)	25	26	PGND
GND	27	28	GND
VIN (+3.7-17V)	29	30	VIN (+3.7-17V)

Table 6.3: Connector J2

Because of the common JTAG interface the PL system can be configured using a JTAG connection without interfering with the processor.

7

DDR Memory

snickerdoodle comes with either 512 Megabytes or 1 Gigabyte of mobile low-power DDR2 SDRAM from Micron. [Table 7.1](#) shows the [Table 3.1](#) shows the specifications of the two memory configuration options.

Density	4 Gbit (512 MByte)	8 Gbit (1 GByte)
Clock Frequency	400 MHz (800 Mbps)	400 MHz (800 Mbps)
Organization	32 bits	32 bits

Table 7.1: snickerdoodle Memory Specifications

Both the Micron EDB4432BBPA-1D-F-D (512 MB) and the Micron EDB8132B4PB-8D-F-D (1 GB) LPDDR2 memory modules used a 32-bit wide interface. The processing subsystem is responsible for managing the the memory modules. As indicated by [Table 7.1](#) , memory speeds up to 533 MHz are supported by the on-board Zynq 7-Series SoC.

8

USB UART Bridge (Serial Port)

The power input connector (J1) also carries USB high-speed (2.0) signals. The USB connection establishes a serial console that can be used to communicate to snickerdoodle. The serial console can be set up to interact with any user-defined application/system; most commonly to display Linux console data.

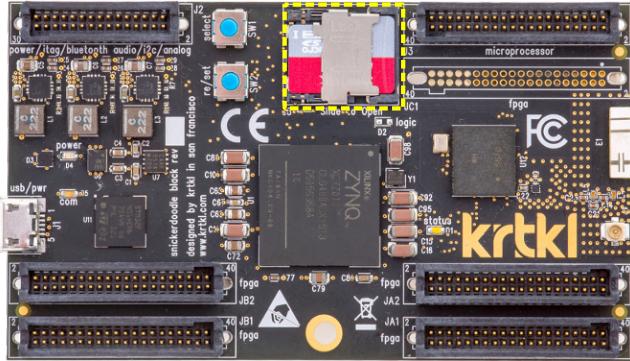
9

microSD Slot

CAUTION Make sure the microSD card cage is securely latched before powering your snickerdoodle

The microSD card cage (J6) on snickerdoodle provides a compact, locking connector for non-volatile storage of up to 200 GB. The microSD interface uses an SDIO interface that is connected to MIO Bank 501 pins 40-45 (SDIO 0). The microSD slot provides a source for non-volatile storage. The Zynq processing subsystem can be booted from the microSD connection and, optionally, a logic bitstream can be defined.

Figure 9.1: Loaded microSD Card Cage

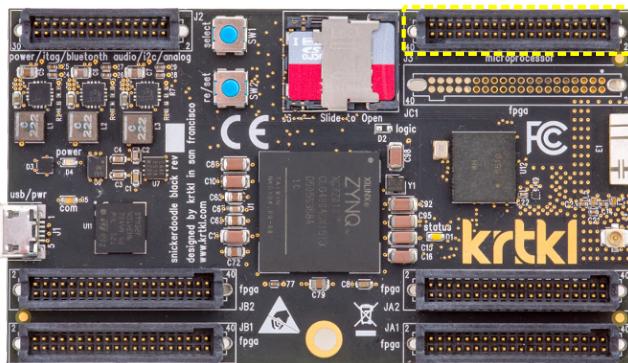


The microSD card interface supports microSD cards of various speeds with a maximum clock frequency of 50 MHz. High speed cards (Class 10) are recommended when using the microSD card as the boot device.

10

Processor Subsystem Connectivity

Connection to the processor subsystem (PS) is done through J3 and provides access to multiplexed input/output (MIO). The MIO connected J3 (MIO [16:27]) has been configured to include a gigabit ethernet port and a high speed USB 2.0 port (MIO [28:39]). The configuration of the PS and the pinout of the connector is illustrated in [Figure 10.2](#).



[Figure 10.1:](#) Processor Subsystem Connector

VCCO_MIO1_501	1	2	PS_MIO52_501
NC	3	4	PS_MIO53_501
PS_MIO16_501	5	6	PS_MIO19_501
PS_MIO17_501	7	8	PS_MIO18_501
GND	9	10	GND
PS_MIO20_501	11	12	PS_MIO23_501
PS_MIO21_501	13	14	PS_MIO22_501
GND	15	16	GND
PS_MIO24_501	17	18	PS_MIO27_501
PS_MIO25_501	19	20	PS_MIO26_501
GND	21	22	GND
PS_MIO28_501	23	24	PS_MIO31_501
PS_MIO29_501	25	26	PS_MIO30_501
GND	27	28	GND
PS_MIO32_501	29	30	PS_MIO35_501
PS_MIO33_501	31	32	PS_MIO34_501
GND	33	34	GND
PS_MIO36_501	35	36	PS_MIO39_501
PS_MIO37_501	37	38	PS_MIO38_501
SMB_I2C_SCL5	39	40	SMB_I2C_SDA5

[Figure 10.2:](#) Processor Subsystem Multiplexed Input/Output (MIO) on J3

11

FPGA Connectors



WARNING Care must be taken when connecting to the FPGA system as the pins can be reconfigured to a variety of input and output types.

The versatility and flexibility of snickerdoodle is due to the integrated FPGA fabric built into the SoC. The power of the FPGA is made available on five dedicated connectors. Electrically, the connections available on these connectors have the ability to accept and generate a variety of signals; all specified in the snickerdoodle configuration. The configuration of the FPGA pins can (and will often) include routing/multiplexing logic, logical processing and hardware acceleration to be done using the FPGA. Each connector has 25 reconfigurable I/O as well as 1 ADC input and 1 I²C config port.

As described in [Chapter 3](#), the connectors can be configured as top-facing male headers (TFM) or bottom-facing female headers (SFM). Table [Table 11.1](#) shows some important details of the connectors.

	Top-Facing	Bottom-Facing
Part Number	TFM-120-01-F-D-A	SFM-120-L1-F-D-A
Contact System	Tiger Eye™	
Description	Terminal Strip	Socket Strip
Pitch	0.050in	
Pins	40	
Rows	2	
Current Rating	2.9A	
Mating	SFM (board-to-board)	TFM (board-to-board)
Connectors	SFSD (discrete wire)	TFSD (discrete wire)

Table 11.1: FPGA Connector Specifications

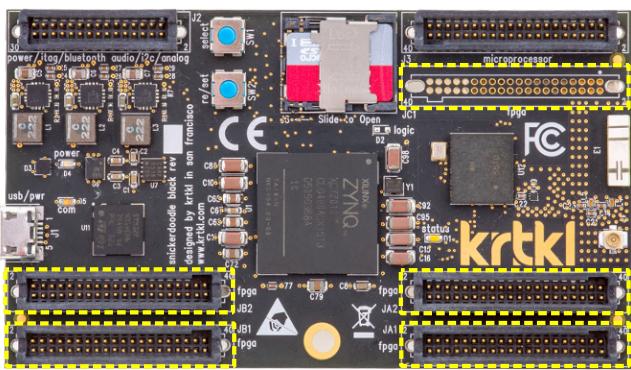


Figure 11.1: FPGA Connectors (JC1 shown unloaded)

12

Wireless

snickerdoodle provides Wi-Fi and Bluetooth/BLE connectivity using a Texas Instruments WiLink™8 certified RF transceiver module.

	WL1831	WL1837
Radio	Single-Band (2.4 GHz)	Dual-Band (2.4 GHz & 5 GHz)
Technology	SISO	MIMO

Table 12.1: snickerdoodle Wireless Module Features

Both modules support IEEE 802.11b/g/n standards, Bluetooth 4.0 and Bluetooth low energy; providing snickerdoodle with a variety of connectivity options. [Figure 12.1](#) shows the location of the UFL jack that can be used to connect an antenna to snickerdoodle.



Figure 12.1: Wireless Radio Antenna UFL Connector

APPENDICES

A

snickerdoodle High-Level Block Diagram

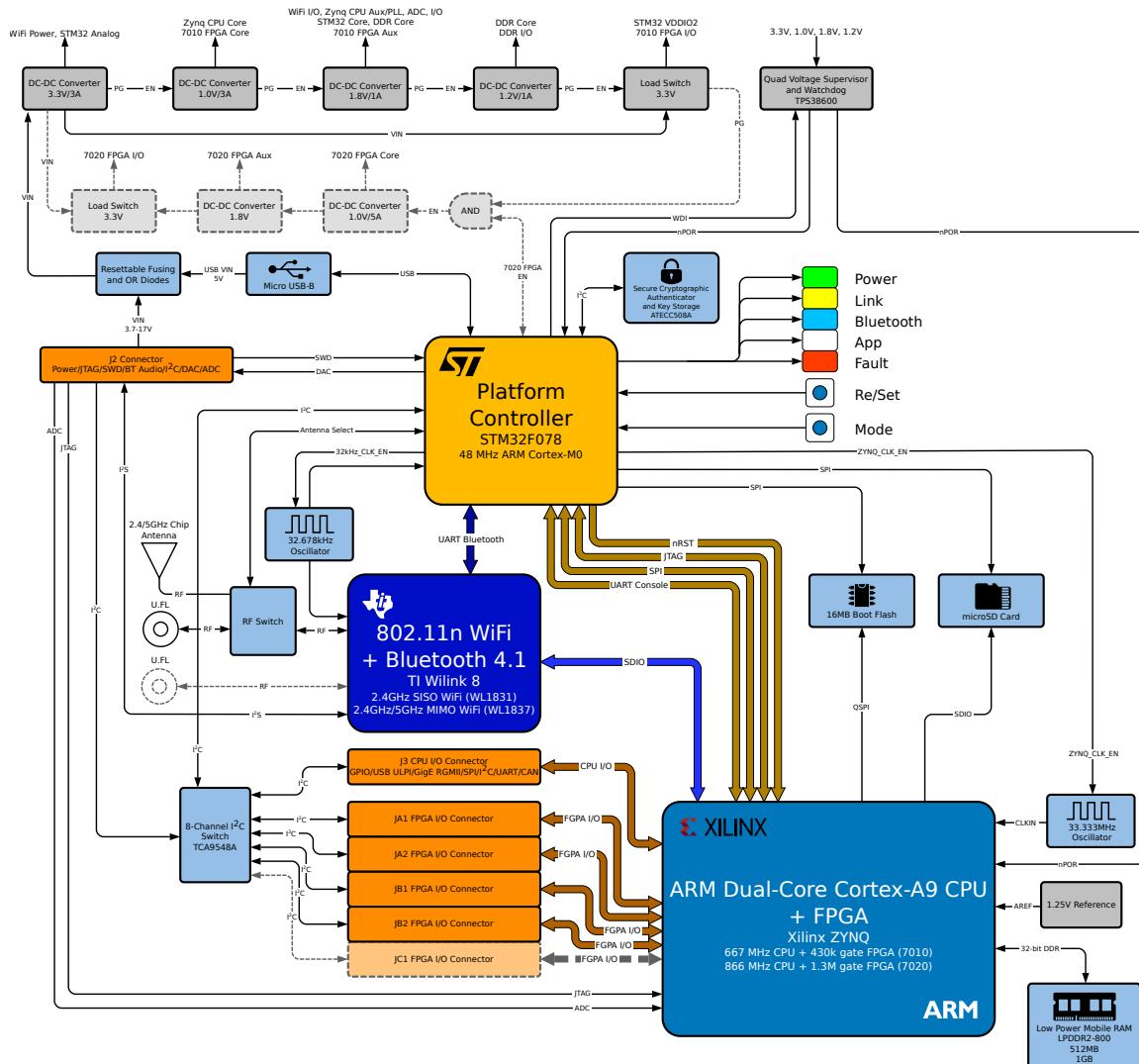


Figure A.1: snickerdoodle High-Level Block Diagram

B

FPGA Connectors

VIO_OUT (+3.3V)	1	2	JA1_SMB_nINT
VCCO_35	3	4	IO_0_35
IO_L5P_T0_AD9P_35	5	6	IO_L4N_T0_35
IO_L5N_T0_AD9N_35	7	8	IO_L4P_T0_35
GND	9	10	GND
IO_L6P_T0_35	11	12	IO_L1N_T0_AD0N_35
IO_L6N_T0_VREF_35	13	14	IO_L1P_T0_AD0P_35
GND	15	16	GND
IO_L3P_T0_DQS_AD1P_35	17	18	IO_L2N_T0_AD8N_35
IO_L3N_T0_DQS_AD1N_351	19	20	IO_L2P_T0_AD8P_35
GND	21	22	GND
IO_L15P_T2_DQS_AD12P_35	23	24	IO_L18N_T2_AD13N_35
IO_L15N_T2_DQS_AD12N_35	25	26	IO_L18P_T2_AD13P_35
GND	27	28	GND
IO_L17P_T2_AD5P_35	29	30	IO_L16N_T2_35
IO_L17N_T2_AD5N_35	31	32	IO_L16P_T2_35
GND	33	34	GND
IO_L14P_T2_AD4P_SRCC_35	35	36	IO_L13N_T2_MRCC_35
IO_L14N_T2_AD4N_SRCC_35	37	38	IO_L13P_T2_MRCC_35
SMB_I2C_SCL0	39	40	SMB_I2C_SDA0

Figure B.1: FPGA Connector JA1

VIO_OUT (+3.3V)	1	2	JA2_SMB_nINT
VCCO_35	3	4	IO_25_35
IO_L22P_T3_AD7P_35	5	6	IO_L24N_T3_AD15N_35
IO_L22N_T3_AD7N_35	7	8	IO_L24P_T3_AD15P_35
GND	9	10	GND
IO_L23P_T3_35	11	12	IO_L19N_T3_VREF_35
IO_L23N_T3_35	13	14	IO_L19P_T3_35
GND	15	16	GND
IO_L21P_T3_DQS_AD14P_35	17	18	IO_L20N_T3_AD6N_35
IO_L21N_T3_DQS_AD14N_35	19	20	IO_L20P_T3_AD6P_35
GND	21	22	GND
IO_L9P_T1_DQS_AD3P_35	23	24	IO_L10N_T1_AD11N_35
IO_L9N_T1_DQS_AD3N_35	25	26	IO_L10P_T1_AD11P_35
GND	27	28	GND
IO_L8P_T2_AD10P_35	29	30	IO_L7N_T1_AD2N_35
IO_L8N_T2_AD10N_35	31	32	IO_L7P_T1_AD2P_35
GND	33	34	GND
IO_L11P_T1_SRCC_35	35	36	IO_L12N_T1_MRCC_35
IO_L11N_T1_SRCC_35	37	38	IO_L12P_T1_MRCC_35
SMB_I2C_SCL1	39	40	SMB_I2C_SDA1

Figure B.2: FPGA Connector JA2

VIO_OUT (+3.3V)	1	2	JB1_SMB_nINT
VCCO_34	3	4	IO_25_34
IO_L1P_T0_34	5	6	IO_L2N_T0_34
IO_L1N_T0_34	7	8	IO_L2P_T0_34
GND	9	10	GND
IO_L6P_T0_34	11	12	IO_L4N_T0_34
IO_L6N_T0_VREF_34	13	14	IO_L4P_T0_34
GND	15	16	GND
IO_L3P_T0_DQS_PUDC_B_34	17	18	IO_L5N_T0_34
IO_L3N_T0_DQS_34	19	20	IO_L5P_T0_34
GND	21	22	GND
IO_L9P_T1_DQS_34	23	24	IO_L7N_T1_34
IO_L9N_T1_DQS_34	25	26	IO_L7P_T1_34
GND	27	28	GND
IO_L8P_T1_34	29	30	IO_L10N_T1_34
IO_L8N_T1_34	31	32	IO_L10P_T1_34
GND	33	34	GND
IO_L11P_T1_SRCC_34	35	36	IO_L12N_T1_MRCC_34
IO_L11N_T1_SRCC_34	37	38	IO_L12P_T1_MRCC_34
SMB_I2C_SCL2	39	40	SMB_I2C_SDA2

Figure B.3: FPGA Connector JB1

VIO_OUT (+3.3V)	1	2	JB2_SMB_nINT
VCCO_34	3	4	IO_0_34
IO_L23P_T3_34	5	6	IO_L24N_T3_34
IO_L23N_T3_34	7	8	IO_L24P_T3_34
GND	9	10	GND
IO_L20P_T3_34	11	12	IO_L19N_T3_VREF_34
IO_L20N_T3_34	13	14	IO_L19P_T3_34
GND	15	16	GND
IO_L21P_T3_DQS_34	17	18	IO_L22N_T3_34
IO_L21N_T3_DQS_34	19	20	IO_L22P_T3_34
GND	21	22	GND
IO_L15P_T2_DQS_34	23	24	IO_L18N_T2_34
IO_L15N_T2_DQS_34	25	26	IO_L18P_T2_34
GND	27	28	GND
IO_L16P_T2_34	29	30	IO_L17N_T2_34
IO_L16N_T2_34	31	32	IO_L17P_T2_34
GND	33	34	GND
IO_L14P_T2_SRCC_34	35	36	IO_L13N_T2_MRCC_34
IO_L14N_T2_SRCC_34	37	38	IO_L13P_T2_MRCC_34
SMB_I2C_SCL3	39	40	SMB_I2C_SDA3

Figure B.4: FPGA Connector JB2

VIO_OUT (+3.3V)	1	2	JC1_SMB_nINT
VCCO_13	3	4	IO_L6N_T0_VREF_13
IO_L11P_T1_SRCC_13	5	6	IO_L12N_T1_MRCC_13
IO_L11N_T1_SRCC_13	7	8	IO_L12P_T1_MRCC_13
GND	9	10	GND
IO_L19P_T3_13	11	12	IO_L20N_T3_13
IO_L19N_T3_VREF_13	13	14	IO_L20P_T3_13
GND	15	16	GND
IO_L21P_T3_DQS_13	17	18	IO_L22N_T3_13
IO_L21N_T3_DQS_13	19	20	IO_L22P_T3_13
GND	21	22	GND
IO_L15P_T2_DQS_13	23	24	IO_L18N_T2_13
IO_L15N_T2_DQS_13	25	26	IO_L18P_T2_13
GND	27	28	GND
IO_L17P_T2_13	29	30	IO_L16N_T2_13
IO_L17N_T2_13	31	32	IO_L16P_T2_13
GND	33	34	GND
IO_L14P_T2_SRCC_13	35	36	IO_L13N_T2_MRCC_13
IO_L14N_T2_SRCC_13	37	38	IO_L13P_T2_MRCC_13
SMB_I2C_SCL4	39	40	SMB_I2C_SDA4

Figure B.5: FPGA Connector JC1