1300 Henley Court
Pullman, WA 99163
509.334.6306
www.digilentinc.com

chipKIT™ WiFire™ Board Reference Manual

Revised May 12, 2014
This manual applies to the chipKIT WiFire rev. B

Pre-Production Limited Quantity Release

The first manufacturing build of the chipKIT WiFire board was produced using the PIC32MZ2048ECG100 – A4 engineering silicon provided by Microchip®. The use of engineering silicon enables early adopters to get a head start on their designs. The engineering silicon was only available in limited quantities, thus limiting the initial number of chipKIT WiFire boards built. The functionality of the engineering silicon on the chipKIT WiFire board will be substantially the same as production silicon for the vast majority of designs. Most projects built on a pre-production chipKIT WiFire board will work without modification on a production board. However, as with any engineering silicon, there is the potential for some minor variation. For more detailed information on the behavior of the engineering silicon used on the board, please refer to the errata that can be downloaded from www.digilentinc.com/wifire or to get the most up to date errata from http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC32MZ2048ECH144#documentation.

Overview

The chipKIT WiFire is based on the popular Arduino™ open-source hardware prototyping platform and adds the performance of the Microchip PIC32MZ microcontroller. The WiFire has a WiFi MRF24 and SD card on the board, both with dedicated SPI signals. The WiFire board takes advantage of the powerful PIC32MZ2048ECG microcontroller. This microcontroller features a 32-bit MIPS processor core running at 200 MHz, 2MB of flash program memory, and 512K of SRAM data memory. The WiFire can be programmed using the Multi-Platform Integrated Development Environment (MPIDE), an environment based on the original Arduino IDE, modified to support PIC32. It contains everything needed to start developing embedded applications. The WiFire features a USB serial port interface for connection to the MPIDE and can be powered via USB or by an external power supply. In addition, the WiFire is fully compatible with the advanced Microchip MPLAB®X IDE and works with all MPLAB X compatible in-system programmer/debuggers, such as the Microchip PICkit™3 or the Digilent® chipKIT PGM. The WiFire is easy to use and suitable for both beginners and advanced users experimenting with electronics and embedded control systems.

Note: The Microchip PIC32MZ processor is a new generation microcontroller incorporating a very complex, aggressively ambitious analog to digital converter (ADC) design. The ADC uses a high speed pipelined architecture that does not lend itself well to multichannel applications or low speed sampling applications. As a result, the ADC does not meet Microchip's original design specification. Consequently, the ADC performance is not as good as on the previous Microchip PIC32MX processors. For detailed information on the PIC32MZ ADC performance and limitations, please refer to the errata that can be downloaded from www.digilentinc.com/wifire or get the most up to date errata from http://www.microchip.com/wwwproducts/Devices.aspx?product=PIC32MZ2048ECH144#documentation.

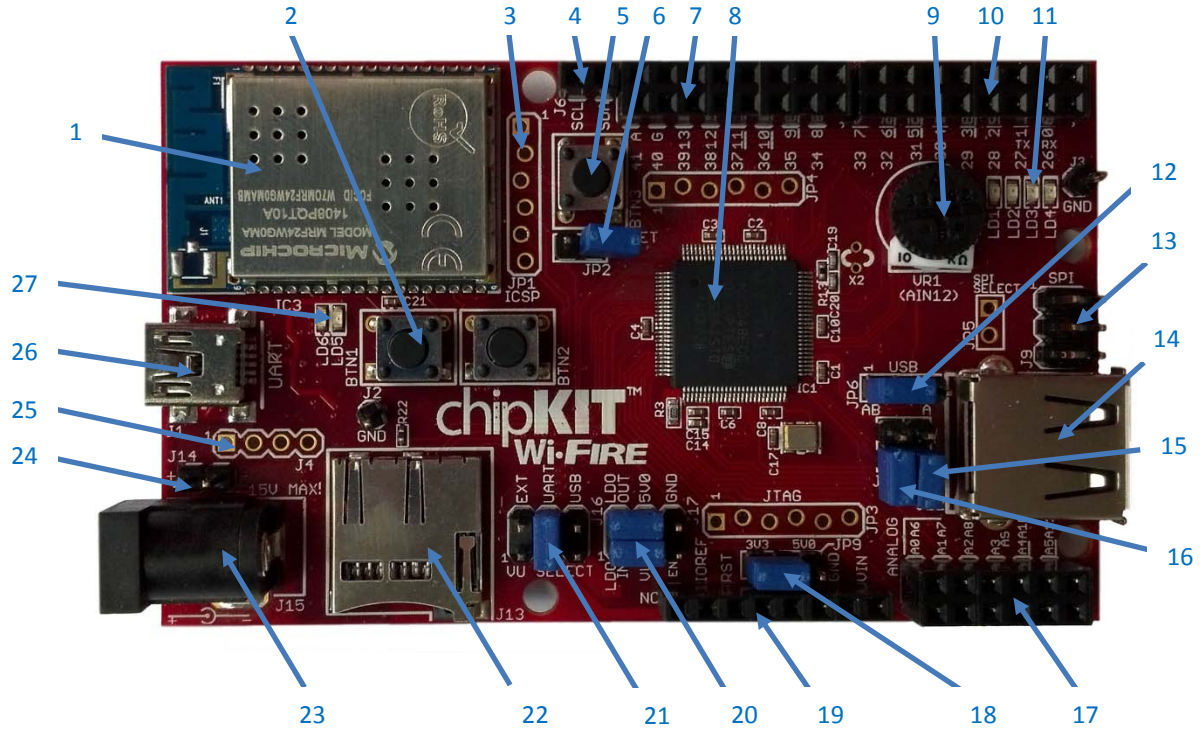
The analogRead() function in MPIDE was updated to mitigate the complexities and to improve the accuracy of the PIC32MZ ADC by careful selection of sampling times and oversampling of the input. These optimizations are only available when calling analogRead(); the raw ADC performance will be obtained if the peripheral is referenced directly.

The chipKIT WiFire board.

- Microchip PIC32MZ2048ECG microcontroller (200 MHz 32-bit MIPS, 2MB Flash, 512K SRAM)
- Microchip MRF24WG0MA WiFi module
- Micro SD card connector
- USB 2.0 Hi-Speed OTG controller with A and micro-AB connectors
- 50 MHz SPI
- 43 available I/O pins
- four user LEDs
- PC connection uses a USB A > mini B cable (not included)
- 12 analog inputs
- 3.3V operating voltage
- 200Mhz operating frequency
- 7V to 15V input voltage (recommended)
- 30V input voltage (maximum)
- 0V to 3.3V analog input voltage range
- High efficiency, switching 3.3V power supply providing low power operation

1 chipKIT WiFire Hardware Overview

The WiFire has the following hardware features:



Call Out	Component Description	Call Out	Component Description
1	IC3- Microchip MRF24WG0MA WiFi Module	15	JP8- Hos USB Bus Power Enable
2	User Buttons	16	JP7- USB Overcurrent Detect
3	JP1- Microchip Debug Tool Connector	17	J8- Analog and Digital Signal Connector
4	J6- I^2C Signals	18	JP9- 3.3v / 5.0v Shield Voltage Select
5	BTN3- Reset	19	J5- Shield Power Connector
6	JP2- Reset Disable	20	J17- 5.0V Regulator Configuration
7	J7- Digital Signal Connector	21	J16- Power Select Jumper
8	PIC32 Microcontroller	22	J13- Micro SD Connector
9	Potentiometer	23	J15- External Power Connector
10	J10- Digital Signal Connector	24	J14- External Power Connector
11	User LEDs	25	J4- USB- UART Handshaking Signals
12	JP6- USB Host or OTG Select	26	USB connector for USB Serial Converter
13	J9- SPI Connector	27	Serial Communication LEDs
14	J12- USB Connectors		

Table 1. chipKIT hardware description.

2 MPIDE and USB Serial Communications

The WiFire board is designed to be used with the MPIDE; the MPIDE development platform was created by modifying the Arduino IDE and is backwards-compatible with the Arduino IDE. Links for where to obtain the MPIDE installation files, as well as instructions for installing MPIDE, can be found at www.chipkit.net/started.

The MPIDE uses a serial communications port to communicate with a boot loader running on the WiFire board. The serial port on the WiFire board is implemented using an FTDI FT232RQ USB serial converter. Before attempting to use the MPIDE to communicate with the WiFire, the appropriate USB device driver must be installed.

The WiFire board uses a standard mini-USB connector. Generally, a USB A to mini-B cable is used for connection to a USB port on the PC.

When the MPIDE needs to communicate with the WiFire board, the board is reset and starts running the boot loader. The MPIDE then establishes communications with the boot loader and uploads the program to the board.

When the MPIDE opens the serial communications connection on the PC, the DTR pin on the FT232RQ chip is driven low. This pin is coupled through a capacitor to the MCLR pin on the PIC32 microcontroller. Driving the MCLR line low resets the microcontroller, which restarts the execution with the boot loader.

This automatic reset action (when the serial communications connection is opened) can be disabled. To disable this operation, there is a jumper labeled JP2, which can be disconnected. JP2 is normally shorted, but if the shorting block is removed, the automatic reset operation will be disabled.

Two red LEDs (LD5 and LD6) will blink when data is being sent or received between the WiFire and the PC over the serial connection.

The header connector J4 provides access to the other serial handshaking signals provided by the FT232RQ. Connector J4 is not loaded at the factory and can be installed by the user to access these signals.

3 Power Supply

The WiFire is designed to be powered via USB (J1), from an external power supply (J14 or J15), or from the USB OTG receptacle (J11). Jumper block J16 is used to select which power supply is used. The power supply voltage selected by J16 is applied to the unregulated power bus, VU.

In order to operate the WiFire as a USB device powered from the USB serial interface (connector J1), place a shorting block in the UART position of jumper block J16. To operate the WiFire from an external power supply, attach the power supply to either J14 or J15 and place a shorting block in the EXT position of J16. Be sure to observe correct polarity when connecting a power supply to J14, as a reversed connection could damage the board. To operate the WiFire as a USB powered device from the USB OTG connector (J11), place a shorting block on the USB position of J16. This will normally only be done when running a sketch on the board that programs it to operate as a USB device. The power supply section in the WiFire provides two voltage regulators, a 3.3V regulator and a 5V regulator. All systems on the WiFire board itself operate at 3.3V and are powered by the 3.3V regulator. The 5V regulator is used to provide power for external circuits, such as shields, that require 5V for operation and to supply USB 5.0V when the WiFire is used as a USB Host. The 5V regulator can be completely disabled if it is not needed for a given application.

When a shield is used, connector J5 provides power to the shield. Connector J5 pin 8 provides VIN as applied by the external power source J14 or J15. If no power is provided to J14 or J15, VIN will not be powered. For most shields, pin 5 on connector J5 would provide 5.0V to the shield; however, the WiFire is not 5v tolerant and it would be very easy for a shield to destroy an input if 5.0V were applied to the PIC32MZ. For this reason, JP9 was added to control the voltage supplied to the shield's 5V source. By default, JP9 is loaded to supply only 3.3V on the 5.0V pin so that the shield does not get 5V and thus cannot inadvertently apply 5.0V to any input to the WiFire. If the shield requires 5.0V to operate, the shield will not work when 3.3V is applied; JP9 must be selected to provide 5.0V for the shield to work. However, extreme caution should be used when selecting 5.0V on JP9 to ensure that the shield will observe IOREF and not supply 5.0V to any input to the WiFire; as this will damage the input to the PIC32MZ on the WiFire.

The WiFire board is designed for low power operation and efficient use of battery power; a switching mode voltage regulator is used for the 3.3V power supply. This switching mode regulator is made up of a Microchip MCP16301 and associated circuitry, which can operate on input voltages from 4V to 30V with up to 96% efficiency, and is rated for 600mA total current output. The MCP16301 has internal short circuit protection and thermal protection. The 3.3V regulator takes its input from the unregulated power bus, VU, and produces its output on the VCC3V3 power bus. The VCC3V3 bus provides power to all on-board systems and is available at the shield power connector (J5) to provide 3.3V power to external circuitry, such as shields.

The 5V regulator section provides a low dropout linear regulator. The 5.0 regulator is provided for powering external circuitry that needs a 5V power supply, such as providing for USB 5.0V when the WiFire is used as a USB host, or to provide 5.0V to the shield on J5 with JP9 selected to 5.0v. This voltage regulator uses an On Semiconductor NCP1117LP. The NCP1117LP is rated for an output current of 1A. The dropout voltage of the NCP1117LP is a maximum of 1.4V at 1A output current. The maximum input voltage of the NCP1117LP is 18V. The recommended maximum operating voltage is 15V. However, if the 5.0V regulator is completely disable by removing all jumpers on J17, the external input voltage applied to J14 or J15 may be as high as the 30V as limited by the switching mode 3.3V regulator.

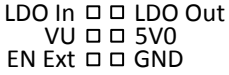
The input voltage to the 5V regulator is taken from the VU bus, and the output is placed on the VCC5V0 power bus. There is a reverse polarity protection diode in the external power supply circuit. Considering the diode drop plus the forward drop across the regulator, the minimum input voltage to the regulator should be 7V to produce a reliable 5V output.

For input voltages above 9V, the regulator will get extremely hot when drawing high currents. The NCP1117LP has output short circuit protection as well as internal thermal protection and will shut down automatically to prevent damage.

The 5V regulator selection on JP17 provides four 5V power configurations:

1) 5V regulator completely disabled and no 5V power available;
2) 5V regulator bypassed and 5V provided from an external 5V power supply, such as USB;
3) on-board 5V regulator used to provide 5V power;
4) External 5V regulator used to regulate VU and provide 5V power.

Jumper block J17 is used to select these various options and the following diagrams describe the use of J16. This diagram shows the arrangement of the signals on J17:



Signals	Description
LDO In	The input to the on-board linear regulator.
LDO Out	The output of the on-board regulator.
VU	The unregulated input voltage selected by the jumper setting jumper block J16.
5V0	The connection to the VCC5V0 power bus on the WiFire board.
EN Ext	Signal provided to enable an external voltage regulator, if one is being used. This would allow the sketch running on the WiFire to turn on/off the external voltage regulator. When used with an external voltage regulator, this allows the board to go into an extremely low power operating mode. This signal is connected to Port D, bit 13 (RD13) on the PIC32 microcontroller. This is accessible using digital pin 50.
GND	Connection to the digital ground bus on the WiFire board.

Table 2. Description of signals on J17.

To completely disable operation of the on-board linear regulator, remove all shorting blocks from J17. To use the on-board 5V regulator, use the provided shorting blocks to connect VU to LDO In, and to connect LDO Out to 5V0, as follows:

Note: In this case, when J16 is in the EXT position, and J17 is jumpered to regulate the external input, do not apply more than 18V; this can destroy the 5.0V regulator.

To bypass the on-board 5V regulator when powering the board from an externally regulator 5V power supply, such as USB, Use one of the provided shorting blocks to connect VU to 5V0, as follows:

An external 5V regulator can be used. This would be desirable, for example, when operating from batteries. An external switching mode 5V regulator could be used to provide higher power efficiency than the on-board linear regulator. In this case, use wires as appropriate to connect VU to the unregulated input of the external regulator. Connect the regulated 5V output to 5V0. Connect GND to the ground connection of the external regulator. Optionally, connect EN Ext to the enable input control of the external regulator, if available. This allows the external regulator to be turned off for low power operation. Digital pin 50 is then used to turn on/off the external regulator.

The PIC32MZ microcontroller is rated to use a maximum of 60mA of current when operating at 200 MHz. The MRF24WG0MA WiFi module typically consumes a maximum of 237mA when transmitting. This allows approximately 303mA of current to power the remaining 3.3V circuitry on the WiFire board and external circuitry powered from the VCC3V3 bus. No circuitry on the WiFire board is powered from the VCC5V0 power bus, leaving all current available from the 5V regulator to power external circuitry and the USB 5.0V power bus when the WiFire is used as a USB Host.

The POWER connector (J5) is used to power shields connected to the WiFire board. Pin 1 is unconnected, the following pins are provided on this connector:

- **IOREF** (pin 2): This pin is tied to the VCC3V3 bus.

- **RST** (pin 3): This connects to the MCLR pin on the PIC32 microcontroller and can be used to reset the PIC32.
- **3V3** (pin 4): This routes the 3.3V power bus to shields.
- **5V0** (pin 5): This routes 3.3V or 5.0V power to shields depending on the position of JP9.
- **GND** (pin 6, 7): This provides a common ground connection between the WiFire and the shields. This common ground is also accessible on connectors J2 and J3.
- **VIN** (pin 8): This connects to the voltage provided at the external power supply connectors (J14 and J15). This can be used to provide unregulated input power to the shield. It can also be used to power the WiFire board from the shield instead of from the external power connector. If no power is supplied at J14 or J15 or from the shield, VIN will not have any power on it.

4 5V Compatibility

The PIC32 microcontroller operates at 3.3V. The original Arduino boards operate at 5V, as do many Arduino shields.

There are two issues to consider when dealing with 5V compatibility for 3.3V logic. The first is protection of 3.3V inputs from damage caused by 5V signals. The second is whether the 3.3V output is high enough to be recognized as a logic high value by a 5V input.

The digital I/O pins on the PIC32 microcontroller are 5V tolerant, whereas the analog capable I/O pins are not 5V tolerant. There are 48 analog capable I/O pins on the PIC32MZ, and this applies to most GPIO pins on the processor. Historically, clamp diodes and current limiting resistors have been used to protect the analog capable I/O from being damaged; but because of the large number of analog capable I/Os, and because clamp diodes and resistors will limit the maximum speed at which these I/Os will operate, it was decided that the WiFire would not be 5V tolerant. Instead, JP9 was added to allow for the 5V0 bus to the shield to be selectable between 3.3V or 5.0V. If 5.0V is selected, great care must be used to ensure that no input to the PIC32MZ exceeds 3.6V; as that will damage the PIC32MZ.

The minimum high-voltage output of the PIC32 microcontroller is rated at 2.4V when sourcing 12mA of current. When driving a high impedance input (typical of CMOS logic) the output high voltage will be close to 3.3V. Some 5V devices will recognize this voltage as a logic high input, and some won't. Many 5V logic devices will work reliably with 3.3V inputs.

5 Input/Output Connections

The WiFire board provides 43 of the I/O pins from the PIC32 microcontroller at pins on the input/output connectors J6, J7, J8, J9, and J10.

The PIC32 microcontroller can source or sink a maximum of 15mA on all digital I/O pins; however, some pins can source or sink 25mA or even 33mA; check with the PIC32MZ datasheet for more information. To keep the output voltage within the specified output voltage range (V_{OL} 0.4V, V_{OH} 2.4V) the pin current must be restricted to +/-10mA on the 15mA pins, or for the higher current pins check the PIC32MZ datasheet for the maximum currents. The maximum current that can be sourced or sunk across all I/O pins simultaneously is +/-150mA. The maximum voltage that can be applied to any I/O pin is 3.6V. For more detailed specifications, refer to the PIC32MZ datasheet available from www.microchip.com.

Connectors J7 and J10 are 2x8 female pin header connectors that provide digital I/O signals. The outer row of pins (closer to the board edge) corresponds to the I/O connector pins on an Arduino Uno or Duemilanove board. The inner row of pins provides access to the extra I/O signals provided by the PIC32 microcontroller.

Connector J8 is a 2x6 female pin header connector that provides access to the analog input pins on the microcontroller. The outer row of pins corresponds to the six analog pins on an Arduino Uno or Duemilanove. The inner row of pins is for the additional I/O signals provided by the PIC32 microcontroller. The analog pins on J8 can also be used as digital I/O pins.

The chipKIT/Arduino system uses logical pin numbers to identify digital I/O pins on the connectors. The logical pin numbers for the I/O pins on the WiFire are 0-42. These pin numbers are labeled in the silk screen on the board. Additional pins 43-70 allow access to the on board components such as the uSD, MRF24 WiFi radio, User LEDs / BTNs, and POT.

Pin numbers 0-13 are the outer row of pins on J10 and J7, from right to left. Pin numbers 14-19 are the outer row of pins on J8, from left to right. Pins 20-25 are the inner row of pins on J8, from left to right. Pin numbers 26-41 are the inner row of pins on J10 and J7, from right to left. Pin 42 is the pin labeled A on J7. This pin is normally the reference voltage for the microcontroller's A/D converter, but can also be used as a digital I/O pin.

In addition to the connector pin, Pin 13 also connects to the user LED LD1. Pin 43, 44, and 45 connect to user LEDs LD2, LD3, and LD4. Pins 43-45 do not attach to any connector. Pins 46 and 47 connect to Buttons BTN1 and BTN2 and do not attach to any connector.

The analog inputs on connector J8 are assigned pin numbers. The outer row of pins on J8 is analog inputs A0-A5. The inner row of pins is A6-A11. These pins are also assigned digital pin numbers; A0-A5 are digital pins 14-19, and A6-A11 are 20-25.

6 802.11b/g Interface

The 802.11b/g compatible WiFi interface on the WiFire is provided by a Microchip MRF24WG0MA WiFi module. This module provides the radio transceiver, antenna, and 802.11 compatible network firmware.

The MRF24WG0MA firmware provides the 802.11 network protocol software support. The DEIPcK and DEWFcK libraries provide the TCP/IP network protocol support that works with the 802.11 protocol support provided by the WiFi module.

The primary communications interface with the MRF24WG0MA WiFi module is a 4 wire SPI bus. This SPI bus uses SPI4 in the PIC32 microcontroller, and this SPI controller is dedicated to use for communications with the WiFi module

The WiFi module supports SPI clock speeds up to 25MHz. In addition to the SPI interface, the interface to the WiFi module also includes a reset signal, an interrupt signal and a hibernate signal. The active low RESET signal is used to reset the WiFi module. The external interrupt signal, INT, is used by the module to signal to the host microcontroller that it needs servicing by the microcontroller software. The INT signal on the WiFi module is connected to external interrupt INT4 on the PIC32 microcontroller and is not routed to any connector. The active low HIBERNATE signal is used to power the WiFi module down and puts it into a low power state.

The interface signals to the WiFi module are controlled by the network libraries and are not normally accessed by the user sketch. Refer to the schematic for the WiFire board for details on these connections.

More detailed information about the operation of the MRF24WG0MA can be obtained from the manufacturer data sheet available from www.microchip.com.

7 Network Library Software

The WiFi module on the WiFire is intended for use with the Digilent Embedded chipKIT network libraries, DEIPcK and DEWFcK. The DEIPcK library provides TCP/UDP/IP protocol support for all chipKIT compatible network interfaces supported by Digilent products, including the WiFire. The DEWFcK library provides the additional library support required for connecting to and operating with the Microchip MRF24WG0MA wireless network modules. Caution should be used in understanding that the DEIPcK library is different than the DNETcK network libraries. DEIPcK is the Digilent Embedded Open Source IP stack that supports both the MX and MZ processor lines, while the DNETcK IP stack is built on top of the Microchip MLA proprietary stack and only supports the MX processor line, and will not work with the WiFire.

The DEWFcK library supports the MRF24WG0MA WiFi module as loaded on the WiFire. The correct header file must be used to specify the network hardware being used by the sketch. When writing a network sketch on the WiFire, use the following hardware library:

```
#include <MRF24G.h>
```

The Digilent Embedded chipKIT network libraries are available for download from the Digilent website: www.digilentinc.com/wifire.

There are reference examples demonstrating the use of these libraries in the library download.

8 USB Interface

The PIC32MZ microcontroller on the WiFire contains a USB 2.0 Compliant, Hi/Full-Speed Device and On-The-Go (OTG) controller. This controller provides the following features:

- USB Hi or full speed host and device support.
- Low speed host support.
- USB OTG support.
- Endpoint buffering anywhere in system RAM.
- Integrated DMA to access system RAM and Flash memory.

Connector J12 is a standard USB type A receptacle. This connector will be used when the WiFire has been programmed to operate as a USB embedded host. The USB device is connected either directly to the WiFire, or via cable to this connector.

Connector J11, on the bottom of the board, is the Device/OTG connector. This is a standard USB micro-AB connector. Connect a cable with a micro-A plug (optionally available from Digilent) from this connector to an available USB port on a PC or USB hub for device operation.

The USB specification allows for two types of devices with regard to how they are powered: self-powered devices and bus powered devices. A self-powered device is one that is powered from a separate power supply and does not draw power from the USB bus. A bus powered device is one that draws power from the USB bus and does not have a separate power supply.

The WiFire can be operated as a self-powered device or as a bus powered device from either the USB serial connector (J1) or the USB OTG/device connector (J11).

For operation as a self-powered device, place a shorting block on the EXT position of J16 and connect a suitable external power supply to either J14 or J15.

To operate the WiFire as a bus powered device powered from the USB serial connector (J1), place a shorting block in the UART position of J16. To operate as a bus powered device powered from the OTG/device connector (J11), place a shorting block in the USB position of J16.

Note that there are two completely independent USB interfaces on the WiFire board, and it is possible for the WiFire to appear as two different USB devices at the same time. These two devices can be connected to two different USB ports on the same host, or to USB ports on two different hosts. If the WiFire board is connected to two different USB hosts simultaneously, there will be a common ground connection between these two hosts through the WiFire board. In this case, it is possible for ground current to flow through the WiFire board, possibly damaging one or the other USB host if they do not share a common earth ground connection.

When the WiFire is operating as a bus powered device using USB connector J1, it will appear as a self-powered device from the perspective of a USB host connected to J11. Similarly, when operating as a bus powered device from connector J11, it will appear as a self-powered device from the perspective of connector J1.

A USB host is expected to be able to provide bus power to USB devices connected to it. Therefore, when operating as a USB host, the WiFire should normally be externally powered. Connect a power supply to the external power connector, J17. It is possible to operate the WiFire as a USB host powered from USB connector J1; however, in this case the host USB port will be providing power for the WiFire as well as the USB device connected to the WiFire. In this case, ensure that the total load does not exceed the 500mA maximum load that a USB device is allowed to present to the host.

The USB host provides regulated 5V power to the connected USB device. The internal 5V LDO regulator can be used to provide the USB power when operating from an external power supply. Place shorting blocks on jumper block J17, as described above in the power supply section.

If the external power supply being used is a regulated 5V supply, place a shorting block between pins VU and 5V0 on connector J17, as described above in the power supply section to bypass the on board 5.0V regulator.

The power supply used must be able to supply enough current to power both the WiFire and the attached USB device, since the WiFire provides power to the attached USB device when operating as a host. The USB 2.0 specification requires that the host provide at least 100mA to the device.

Jumper JP6 is used to provide the required USB host capacitance to the host connector being used. Place the shorting block in the “A” position when using the standard USB type A (host) Connector (J12). Place the shorting block in the “AB” position for use with the USB micro-AB (OTG) connector (J11).

With JP8 shorted, chipKIT pin 25 drives the enable input of a TPS2051B Current-Limited Power Distribution Switch to supply 5V USB power to the host connector. This switch has over-current detection capability and provides an over-current fault indication by pulling the signal USBOC low. The over-current output pin can be monitored via the chipKIT pin 8 (RA14/INT3) when JP7 is shorted. Details about the operation of the TPS2051B can be obtained from the datasheet available at www.ti.com.

When using the WiFire outside the MPIDE environment, the Microchip Harmony Library provides USB stack code that can be used with the board. There are reference designs available on the Microchip website demonstrating

both device and host operation of PIC32 microcontrollers. These reference designs can be modified for developing USB firmware for the WiFire.

9 SD Card Interface

The micro-SD card connector provides the ability to access data stored on micro-SD sized flash memory cards using the SD card library provided as part of the MPIDE software system.

The SD card is accessed using an SPI interface on PIC32 microcontroller pins dedicated to this purpose. The MPIDE SD library uses a "bit-banged" software SPI implementation to talk to SD card. However, software can be written to access the SD card using SPI3.

On the WiFire board, SPI3 and I/O pins used to communicate with the SD card are dedicated to that function and are not shared with other uses.

10 Peripheral I/O Functions

The PIC32 microcontroller on the WiFire board provides a number of peripheral functions. The provided peripherals are explained in the following sections.

10.1 UART Ports

UART 4: Asynchronous serial port. Pin 0 (RX), Pin 1 (TX). This is accessed using the runtime object: Serial. These pins are connected to I/O connector J10 and are also connected to the FT232RQ USB serial converter. It is possible to use these pins to connect to an external serial device when not using the FT232RQ USB serial interface. This uses UART4 (U4RX, U4TX) on the PIC32 microcontroller.

UART 1: Asynchronous serial port. Pin 39 (RX), Pin 40 (TX). This is accessed using the runtime object: Serial1. This uses UART1 (U1RX, U1TX) on the PIC32 microcontroller.

10.3 SPI

Synchronous serial port. Pin 10 (SS), Pin 11 (MOSI), Pin 12 (MISO), Pin 13 (SCK). This can be accessed using the SPI standard library. It can also be accessed using the DSPI0 object from the DSPI standard library. This uses SPI2 (SS2, SDI2, SDO2, SCK2) on the PIC32 microcontroller. These signals also appear on connector J7. Be aware that pin 13 (SCK) is shared with USER LED1, and that both LED1 and the SPI port cannot be used concurrently.

SPI1: Synchronous serial port. This is an additional SPI interface on the PIC32 microcontroller that can be assessed using the DSPI1 object from the DSPI standard library. SS1 is accessed via digital pin number 7. SDO1 is accessed via digital pin 35. SDI1 is accessed via digital pin 36. SCK1 is connected to digital pin 5.

10.4 I^2C

Synchronous serial interface. The PIC32 microcontroller shares analog pins A4 and A5 with the two I2C signals, SDA and SCL. This uses I2C4 (SDA4, SCL4) on the PIC32 microcontroller. Both SDA4 and SCL4 are accessible on connector J6.

Note: The I2C bus uses open collector drivers to allow multiple devices to drive the bus signals. This means that external pull-up resistors must be provided to supply the logic high state for the signals.

10.5 PWM

Pulse width modulated output; Pins 3 (OC1), 5 (OC2), 6 (OC3), 9 (OC4), 10 (OC9), and 11 (OC7). These can be accessed using the analogWrite() runtime function.

10.6 External Interrupts

Pin 3 (INT0), Pin 2 (INT1), Pin 7 (INT2), Pin 8 (INT3), Pin 59 (INT4). Note that the pin numbers for INT0 and INT4 are different than on some other chipKIT boards. INT4 is dedicated for use with the MRF24WG0MA WiFi module and is not brought out to a connector pin.

10.7 User LEDs

Pin 13 (LD1), Pin 43 (LD2), Pin 44 (LD3), Pin 45 (LD4). Pin 13 is shared between a connector pin and the LED. Pin 43, 44, and 45 only go to the LED and are not brought out to any connector pin. Driving the pin HIGH turns the LED on, driving it LOW turns it off.

10.8 User Push Buttons

There are two push button switches, which are labeled BTN1 (pin 46) and BTN2 (pin 47). The digitalRead() function will return LOW if the button is pressed and HIGH when the button is pressed.

10.9 A/D Converter Reference

Labeled A, the left-most outer pin on connector J7. This is used to provide an external voltage reference to determine the input voltage range of the analog pins. The maximum voltage that can be applied to this pin is 3.3V. This pin can also be used as digital pin 42.

10.10 Potentiometer

A potentiometer (pot) is provided on the board to be used as an analog signal source or analog control input. The pot is a 10KΩ trimmer pot connected between the VCC3V3 supply and ground. The wiper of the pot is connected to analog input A12 or chipKIT pin 48. The pot is read using the analogRead() function.

10.11 VU Voltage Monitor

The supply voltage as provided by J16 can be monitored on analog input A13 or digital pin 49. The voltage presented to the analog input is $1/11^{th}$ of the actual VU voltage. This allows for a supply voltage between 2.2V to 30V to be monitored and still fall within the range of 0 to 3.3V on the analog input. By doing an analogRead(49), the supply voltage can be monitored.

10.12 RTCC

Real time clock calendar. The PIC32 microcontroller contains an RTCC circuit that can be used to maintain time and date information. The operation of the RTCC requires a 32.768 KHz frequency source. Crystal X2 (not loaded), just above and to the right of the PIC32 microcontroller IC, is provided for you to solder a 32 KHz watch crystal. The Citizen CFS206-32.768KDZF-UB crystal can be used in this location.

UPDATE: At this time, the PIC32MZ processor does not support crystals as a source for the secondary clock and an oscillator must be used. The unloaded circuit as provided may not be useable for an RTCC source.

10.13 RESET

The PIC32 microcontroller is reset by bringing its MCLR pin low. The MCLR pin is connected to the RST pin, as presented on J5.

As previously described, reset of the PIC32 microcontroller can be initiated by the USB serial converter. The USB serial converter brings the DTR pin low to reset the microcontroller. Jumper JP2 can be used to enable/disable the ability for the USB serial converter to initiate a reset.

The RST is connected to pin 3 of connector J5. This allows circuitry on a shield to reset the microcontroller, or to ensure that the circuitry on the shield is reset at the same time as the microcontroller.

Connector J9 provides access to the SPI bus. Pin 5 provides access to the SPI Slave Select signal (SS).

On Arduino boards, the corresponding connector is also used as an in-system programming connector as well as providing access to some of the SPI signals. On Arduino boards, pin 5 of this connector is connected to the reset net.

Some Arduino shields, most notably the Ethernet shield, connect pin 5 to the reset net on pin 3 of connector J5. This causes the processor to be reset each time an attempt is made to access the SPI port. Jumper JP5 can be used to break the connection between J9 pin 5 and reset when using Arduino shields that make this connection. JP5 has a cuttable trace on the top of the board that can be cut to break the connection between SPI SS and reset. JP5 is not loaded at the factory. To restore the connection, solder a two pin header at the JP5 position and install a shorting block.

A reset button is located to the right of the MRF24WG0MA WiFi module. Pressing this button resets the PIC32 microcontroller.

11 Microchip Development Tool Compatibility

In addition to being used with the MPIDE, the WiFire board can be used as a more traditional microcontroller development board using Microchip Development Tools.

Unloaded connector JP1 on the right side of the MRF24WG0MA WiFi module is used to connect to a Microchip development tool, such as the PICkit™3. The holes for JP1 are staggered so that a standard 100-mil spaced 6-pin header can fit to the board without the need to solder it in place. Any Microchip development tool that supports the PIC32MZ microcontroller family, and that can be connected via the same 6-pin ICSP interface as the PICkit™3, can be used.

Typically, a standard male connector and a 6-pin cable is used with JP1 so that a PICkit™3 can be attached to the WiFire board.

The Digilent chipKIT PGM can also be used in place of a PICkit3 to program the WiFire with the Microchip Development tools. The chipKIT PGM has a smaller form factor and does not need a 6-pin cable to connect to JP1.

The Microchip MPLAB X IDE can be used to program and debug code running on the WiFire board. The MPLAB X IDE can be downloaded from the Microchip website. Please note that Microchip's MPLAB®V8 and earlier IDEs cannot be used with the WiFire, as those versions of MPLAB IDE do not support the MZ processor.

Using the Microchip development tools to program the WiFire board will cause the boot loader to be erased. To use the board with the MPIDE again, it is necessary to program the boot loader back onto the board. The boot loader HEX file can be found at www.digilentinc.com. To reprogram the bootloader, use the Microchip IPE which comes with the MPLAB X tool set. The bootloader cannot be easily reprogrammed directly with the MPLAB X IDE.

12 Pinout Tables

The following tables show the relationship between the chipKIT digital pin numbers, the connector pin numbers, and the microcontroller pin numbers.

In the following tables, columns labeled chipKIT pin # refer to the digital pin number. This is the value that is passed to the pinMode(), digitalRead(), digitalWrite() and other functions which refer to the pin.

12.1 Pinout Table by chipKIT Pin Number

ChipKIT Pin #	MCU Pin	Port Bit	PIC32 Signal Name	Function
0	57	RF02	EBIRDY3/RPF2/SDA3/RF2	GPIO, U4RX
1	58	RF08	EBIRDY2/RPF8/SCL3/RF8	GPIO, U4TX
2	18	RE08	AN25/AERXD0/RPE8/RE8	GPIO, IC1, INT1
3	71	RD00	EMDIO/AEMDIO/RPD0/RTCC/INT0/RD0	PWM 1, INT0, OC1
4	60	RA03	EBIRDY1/SDA2/RA3	GPIO
5	76	RD01	RPD1/SCK1/RD1	PWM 2, OC2
6	77	RD02	EBID14/ETXEN/RPD2/PMD14/RD2	PWM 3, OC3
7	19	RE09	AN26/AERXD1/RPE9/RE9	GPIO, IC2, INT2
8	66	RA14	AETXCLK/RPA14/SCL1/RA14	GPIO, IC3, INT3
9	78	RD03	EBID15/ETXCLK/RPD3/PMD15/RD3	PWM 4, OC4
10	16	RG09	EBIA2/AN11/C2INC/ERXCLK/EREFCLK/AERXCLK/AEREF CLK/RPG9/PMA2/RG9	SPI_SS2, PWM 5, OC9, IC6
11	70	RD11	EMDC/AEMDC/RPD11/RD11	SPI_SDO2/SDI2 PWM 6, OC7
12	85	RF00	EBID11/ETXD1/RPF0/PMD11/RF0	SPI_SDI2/SDO2, T5CK(+)
13	10	RG06	AN14/C1IND/ECOL/RPG6/SCK2/RG6	SPI_SCK2, USER LED1
14	20	RB05	AN45/C1INA/RPB5/RB5	AIN0, GPIO
15	33	RB09	EBIA7/AN49/RPB9/PMA7/RB9	AIN1, GPIO
16	7	RC02	EBIA12/AN21/RPC2/PMA12/RC2	AIN2, GPIO
17	44	RB15	EBIA0/AN10/ERXD3/AETXD2/RPB15/OCFB/PMA0/RB 15	AIN3, GPIO
18	11	RG07	EBIA4/AN13/C1INC/ECRS/RPG7/SDA4/PMA4/RG7	AIN4, SDA
19	12	RG08	EBIA3/AN12/C2IND/ERXDV/ECRSDV/AERXDV/AECRSD V/RPG8/SCL4/PMA3/RG8	AIN5, SCL
20	22	RB03	AN3/C2INA/RPB3/RB3	AIN6, GPIO

21	23	RB02	AN2/C2INB/RPB2/RB2	AIN7, GPIO
22	21	RB04	AN4/C1INB/RB4	AIN8, GPIO
23	24	RB01	PGEC1/AN1/RPB1/RB1	AIN9, GPIO
24	32	RB08	EBIA10/AN48/RPB8/PMA10/RB8	AIN10, GPIO
25	25	RB00	PGED1/AN0/RPB0/RB0	AIN11, GPIO, P32_VBUSON
26	91	RE00	EBID0/PMD0/RE0	GPIO
27	94	RE01	EBID1/PMD1/RE1	GPIO
28	98	RE02	EBID2/PMD2/RE2	GPIO
29	99	RE03	EBID3/RPE3/PMD3/RE3	GPIO
30	100	RE04	EBID4/AN18/PMD4/RE4	GPIO
31	3	RE05	EBID5/AN17/RPE5/PMD5/RE5	GPIO
32	4	RE06	EBID6/AN16/PMD6/RE6	GPIO
33	5	RE07	EBID7/AN15/PMD7/RE7	GPIO
34	82	RD05	SQICS1/RPD5/RD5	GPIO, T4CK
35	6	RC01	EBIA6/AN22/RPC1/PMA6/RC1	GPIO, T2CK, IC7
36	86	RF01	EBID10/ETXD0/RPF1/PMD10/RF1	GPIO, T6CK
37	59	RA02	EBICS0/SCL2/RA2	GPIO
38	79	RD12	EBID12/ETXD2/RPD12/PMD12/RD12	GPIO, T3Ck
39	47	RD14	AN32/AETXD0/RPD14/RD14	GPIO, U1RX
40	48	RD15	AN33/AETXD1/RPD15/SCK6/RD15	GPIO, U1TX
41	28	RA09	VREF-/CVREF-/AN27/AERXD2/RA9	GPIO, VREF-
42	29	RA10	VREF+/CVREF+/AN28/AERXD3/RA10	VREF+
43	81	RD04	SQICS0/RPD4/RD4	USER_LED2
44	35	RB11	AN6/ERXERR/AETXERR/RB11	USER_LED3
45	1	RG15	AN23/AERXERR/RG15	USER_LED4
46	2	RA05	EBIA5/AN34/PMA5/RA5	BTN1
47	61	RA04	EBIA14/PMCS1/PMA14/RA4	BTN2
48	42	RB13	AN8/ERXD1/AECOL/RB13	AIN12/POT
49	41	RB12	EBIA11/AN7/ERXD0/AECRS/PMA11/RB12	AIN13/POWER SUPPLY MONITOR
50	80	RD13	EBID13/ETXD3/PMD13/RD13	5V POWER ENABLE
51	43	RB14	EBIA1/AN9/ERXD2/AETXD3/RPB14/SCK3/PMA1/RB14	SD_SCK3
52	8	RC03	EBIWE/AN20/RPC3/PMWR/RC3	SD_SS3
53	34	RB10	EBIA13/CVREFOUT/AN5/RPB10/PMA13/RB10	SD_SDI3
54	9	RC04	EBIOE/AN19/RPC4/PMRD/RC4	SD_SDO3
55	69	RD10	RPD10/SCK4/RD10	MRF24_SCK4

56	68	RD09	EBIA15/RPD9/PMCS2/PMA15/RD9	MRF24_SS4
57	65	RF05	EBIA8/RPF5/SCL5/PMA8/RF5	MRF24_SDI4
58	88	RG00	EBID8/RPG0/PMD8/RG0	MRF24 SDO4
59	67	RA15	AETXEN/RPA15/SDA1/RA15	MRF24_INT4
60	87	RG01	EBID9/ETXERR/RPG1/PMD9/RG1	MRF24_HIBERNATE
61	64	RF04	EBIA9/RPF4/SDA5/PMA9/RF4	MRF24_RESET
62	38	RA01	TCK/EBIA19/AN29/RA1	TCK
63	17	RA00	TMS/EBIA16/AN24/RA0	TMS
64	40	RF12	TDO/EBIA17/AN31/RPF12/RF12	TDO
65	39	RF13	TDI/EBIA18/AN30/RPF13/SCK5/RF13	TDI
66	89	RA06	TRCLK/SQICLK/RA6	TRCLK
67	97	RG13	TRD0/SQID0/RG13	TRD0
68	96	RG12	TRD1/SQID1/RG12	TRD1
69	95	RG14	TRD2/SQID2/RG14	TRD2
70	90	RA07	TRD3/SQID3/RA7	TRD3
N/A	13		VSS	POWER
N/A	14		VDD	POWER
N/A	15		MCLR	MCLR, ICSP
N/A	26	RB06	PGEC2/AN46/RPB6/RB6	ICSP
N/A	27	RB07	PGED2/AN47/RPB7/RB7	ICSP
N/A	30		AVDD	POWER
N/A	31		AVSS	POWER
N/A	36		VSS	POWER
N/A	37		VDD	POWER
N/A	45		VSS	POWER
N/A	46		VDD	POWER
N/A	49	RC12	OSCI/CLKI/RC12	XTAL
N/A	50	RC15	OSCO/CLKO/RC15	XTAL
N/A	51		VBUS	POWER
N/A	52		VUSB3V3	POWER
N/A	53		VSS	POWER
N/A	54		D-	PIC32_USBD-
N/A	55		D+	PIC32_USBD+
N/A	56	RF03	USBID/RPF3/RF3	PIC32_USBID
N/A	62		VDD	POWER
N/A	63		VSS	POWER

N/A	72	RC13	SOSCI/RPC13/RC13	SOSC XTAL
N/A	73	RC14	SOSCO/RPC14/T1CK/RC14	SOSC XTAL
N/A	74		VDD	POWER
N/A	75		VSS	POWER
N/A	83		VDD	POWER
N/A	84		VSS	POWER
N/A	92		VSS	POWER
N/A	93		VDD	POWER

12.2 Pinout Table by MCU Pin and Port Bit Numbers

Port Bit	ChipKIT Pin #	MCU Pin	PIC32 Signal Name	Function
RA00	63	17	TMS/EBIA16/AN24/RA0	TMS
RA01	62	38	TCK/EBIA19/AN29/RA1	TCK
RA02	37	59	EBICS0/SCL2/RA2	GPIO
RA03	4	60	EBIRDY1/SDA2/RA3	GPIO
RA04	47	61	EBIA14/PMCS1/PMA14/RA4	BTN2
RA05	46	2	EBIA5/AN34/PMA5/RA5	BTN1
RA06	66	89	TRCLK/SQICLK/RA6	TRCLK
RA07	70	90	TRD3/SQID3/RA7	TRD3
RA09	41	28	VREF-/CVREF-/AN27/AERXD2/RA9	GPIO, VREF-
RA10	42	29	VREF+/CVREF+/AN28/AERXD3/RA10	VREF+
RA14	8	66	AETXCLK/RPA14/SCL1/RA14	GPIO, IC3, INT3
RA15	59	67	AETXEN/RPA15/SDA1/RA15	MRF24_INT4
RB00	25	25	PGED1/AN0/RPB0/RB0	AIN11, GPIO, P32_VBUSON
RB01	23	24	PGEC1/AN1/RPB1/RB1	AIN9, GPIO
RB02	21	23	AN2/C2INB/RPB2/RB2	AIN7, GPIO
RB03	20	22	AN3/C2INA/RPB3/RB3	AIN6, GPIO
RB04	22	21	AN4/C1INB/RB4	AIN8, GPIO
RB05	14	20	AN45/C1INA/RPB5/RB5	AIN0, GPIO
RB06	N/A	26	PGEC2/AN46/RPB6/RB6	ICSP
RB07	N/A	27	PGED2/AN47/RPB7/RB7	ICSP
RB08	24	32	EBIA10/AN48/RPB8/PMA10/RB8	AIN10, GPIO
RB09	15	33	EBIA7/AN49/RPB9/PMA7/RB9	AIN1, GPIO
RB10	53	34	EBIA13/CVREFOUT/AN5/RPB10/PMA13/RB10	SD_SDI3

RB11	44	35	AN6/ERXERR/AETXERR/RB11	USER_LED3
RB12	49	41	EBIA11/AN7/ERXD0/AECRS/PMA11/RB12	AIN13/POWER SUPPLY MONITOR
RB13	48	42	AN8/ERXD1/AECOL/RB13	AIN12/POT
RB14	51	43	EBIA1/AN9/ERXD2/AETXD3/RPB14/SCK3/PMA 1/RB14	SD_SCK3
RB15	17	44	EBIA0/AN10/ERXD3/AETXD2/RPB15/OCFB/PM A0/RB15	AIN3, GPIO
RC01	35	6	EBIA6/AN22/RPC1/PMA6/RC1	GPIO, T2CK, IC7
RC02	16	7	EBIA12/AN21/RPC2/PMA12/RC2	AIN2, GPIO
RC03	52	8	EBIWE/AN20/RPC3/PMWR/RC3	SD_SS3
RC04	54	9	EBIOE/AN19/RPC4/PMRD/RC4	SD_SDO3
RC12	N/A	49	OSCI/CLKI/RC12	XTAL
RC13	N/A	72	SOSCI/RPC13/RC13	SOSC XTAL
RC14	N/A	73	SOSCO/RPC14/T1CK/RC14	SOSC XTAL
RC15	N/A	50	OSCO/CLKO/RC15	XTAL
RD00	3	71	EMDIO/AEMDIO/RPD0/RTCC/INT0/RD0	PWM 1, INT0, OC1
RD01	5	76	RPD1/SCK1/RD1	PWM 2, OC2
RD02	6	77	EBID14/ETXEN/RPD2/PMD14/RD2	PWM 3, OC3
RD03	9	78	EBID15/ETXCLK/RPD3/PMD15/RD3	PWM 4, OC4
RD04	43	81	SQICS0/RPD4/RD4	USER_LED2
RD05	34	82	SQICS1/RPD5/RD5	GPIO, T4CK
RD09	56	68	EBIA15/RPD9/PMCS2/PMA15/RD9	MRF24_SS4
RD10	55	69	RPD10/SCK4/RD10	MRF24_SCK4
RD11	11	70	EMDC/AEMDC/RPD11/RD11	SPI_SDO2/SDI2 PWM 6, OC7
RD12	38	79	EBID12/ETXD2/RPD12/PMD12/RD12	GPIO, T3Ck
RD13	50	80	EBID13/ETXD3/PMD13/RD13	5V POWER ENABLE
RD14	39	47	AN32/AETXD0/RPD14/RD14	GPIO, U1RX
RD15	40	48	AN33/AETXD1/RPD15/SCK6/RD15	GPIO, U1TX
RE00	26	91	EBID0/PMD0/RE0	GPIO
RE01	27	94	EBID1/PMD1/RE1	GPIO
RE02	28	98	EBID2/PMD2/RE2	GPIO
RE03	29	99	EBID3/RPE3/PMD3/RE3	GPIO
RE04	30	100	EBID4/AN18/PMD4/RE4	GPIO
RE05	31	3	EBID5/AN17/RPE5/PMD5/RE5	GPIO
RE06	32	4	EBID6/AN16/PMD6/RE6	GPIO
RE07	33	5	EBID7/AN15/PMD7/RE7	GPIO

RE08	2	18	AN25/AERXD0/RPE8/RE8	GPIO, IC1, INT1
RE09	7	19	AN26/AERXD1/RPE9/RE9	GPIO, IC2, INT2
RF00	12	85	EBID11/ETXD1/RPF0/PMD11/RF0	SPI_SDI2/SDO2, T5CK(+)
RF01	36	86	EBID10/ETXD0/RPF1/PMD10/RF1	GPIO, T6CK
RF02	0	57	EBIRDY3/RPF2/SDA3/RF2	GPIO, U4RX
RF03	N/A	56	USBID/RPF3/RF3	PIC32_USBID
RF04	61	64	EBIA9/RPF4/SDA5/PMA9/RF4	MRF24_RESET
RF05	57	65	EBIA8/RPF5/SCL5/PMA8/RF5	MRF24_SDI4
RF08	1	58	EBIRDY2/RPF8/SCL3/RF8	GPIO, U4TX
RF12	64	40	TDO/EBIA17/AN31/RPF12/RF12	TDO
RF13	65	39	TDI/EBIA18/AN30/RPF13/SCK5/RF13	TDI
RG00	58	88	EBID8/RPG0/PMD8/RG0	MRF24 SDO4
RG01	60	87	EBID9/ETXERR/RPG1/PMD9/RG1	MRF24_HIBERNATE
RG06	13	10	AN14/C1IND/ECOL/RPG6/SCK2/RG6	SPI_SCK2, USER LED1
RG07	18	11	EBIA4/AN13/C1INC/ECRS/RPG7/SDA4/PMA4/RG7	AIN4, SDA
RG08	19	12	EBIA3/AN12/C2IND/ERXDV/ECRSDV/AERXDV/AECRSDV/RPG8/SCL4/PMA3/RG8	AIN5, SCL
RG09	10	16	EBIA2/AN11/C2INC/ERXCLK/EREFCLK/AERXCLK/AEREFCLK/RPG9/PMA2/RG9	SPI_SS2, PWM 5, OC9, IC6
RG12	68	96	TRD1/SQID1/RG12	TRD1
RG13	67	97	TRD0/SQID0/RG13	TRD0
RG14	69	95	TRD2/SQID2/RG14	TRD2
RG15	45	1	AN23/AERXERR/RG15	USER_LED4
	N/A	13	VSS	POWER
	N/A	14	VDD	POWER
	N/A	15	MCLR	MCLR, ICSP
	N/A	30	AVDD	POWER
	N/A	31	AVSS	POWER
	N/A	36	VSS	POWER
	N/A	37	VDD	POWER
	N/A	45	VSS	POWER
	N/A	46	VDD	POWER
	N/A	51	VBUS	POWER
	N/A	52	VUSB3V3	POWER
	N/A	53	VSS	POWER
	N/A	54	D-	PIC32_USBD-
	N/A	55	D+	PIC32_USBD+

	N/A	62	VDD	POWER
	N/A	63	VSS	POWER
	N/A	74	VDD	POWER
	N/A	75	VSS	POWER
	N/A	83	VDD	POWER
	N/A	84	VSS	POWER
	N/A	92	VSS	POWER
	N/A	93	VDD	POWER

12.3 Pinout Table by PIC32 Microcontroller Pin

MCU Pin	Port Bit	ChipKIT Pin #	PIC32 Signal Name	Function
1	RG15	45	AN23/AERXERR/RG15	USER_LED4
2	RA05	46	EBIA5/AN34/PMA5/RA5	BTN1
3	RE05	31	EBID5/AN17/RPE5/PMD5/RE5	GPIO
4	RE06	32	EBID6/AN16/PMD6/RE6	GPIO
5	RE07	33	EBID7/AN15/PMD7/RE7	GPIO
6	RC01	35	EBIA6/AN22/RPC1/PMA6/RC1	GPIO, T2CK, IC7
7	RC02	16	EBIA12/AN21/RPC2/PMA12/RC2	AIN2, GPIO
8	RC03	52	EBIWE/AN20/RPC3/PMWR/RC3	SD_SS3
9	RC04	54	EBIOE/AN19/RPC4/PMRD/RC4	SD_SDO3
10	RG06	13	AN14/C1IND/ECOL/RPG6/SCK2/RG6	SPI_SCK2, USER LED1
11	RG07	18	EBIA4/AN13/C1INC/ECRS/RPG7/SDA4/PMA4/RG7	AIN4, SDA
12	RG08	19	EBIA3/AN12/C2IND/ERXDV/ECRSDV/AERXDV/AECRSDV/RPG8/SCL4/PMA3/RG8	AIN5, SCL
13		N/A	VSS	POWER
14		N/A	VDD	POWER
15		N/A	MCLR	MCLR, ICSP
16	RG09	10	EBIA2/AN11/C2INC/ERXCLK/EREFCLK/AERXCLK/AEREFCLK/RPG9/PMA2/RG9	SPI_SS2, PWM 5, OC9, IC6
17	RA00	63	TMS/EBIA16/AN24/RA0	TMS
18	RE08	2	AN25/AERXD0/RPE8/RE8	GPIO, IC1, INT1
19	RE09	7	AN26/AERXD1/RPE9/RE9	GPIO, IC2, INT2
20	RB05	14	AN45/C1INA/RPB5/RB5	AIN0, GPIO
21	RB04	22	AN4/C1INB/RB4	AIN8, GPIO
22	RB03	20	AN3/C2INA/RPB3/RB3	AIN6, GPIO
23	RB02	21	AN2/C2INB/RPB2/RB2	AIN7, GPIO

24	RB01	23	PGEC1/AN1/RPB1/RB1	AIN9, GPIO
25	RB00	25	PGED1/AN0/RPB0/RB0	AIN11, GPIO, P32_VBUSON
26	RB06	N/A	PGEC2/AN46/RPB6/RB6	ICSP
27	RB07	N/A	PGED2/AN47/RPB7/RB7	ICSP
28	RA09	41	VREF-/CVREF-/AN27/AERXD2/RA9	GPIO, VREF-
29	RA10	42	VREF+/CVREF+/AN28/AERXD3/RA10	VREF+
30		N/A	AVDD	POWER
31		N/A	AVSS	POWER
32	RB08	24	EBIA10/AN48/RPB8/PMA10/RB8	AIN10, GPIO
33	RB09	15	EBIA7/AN49/RPB9/PMA7/RB9	AIN1, GPIO
34	RB10	53	EBIA13/CVREFOUT/AN5/RPB10/PMA13/RB10	SD_SDI3
35	RB11	44	AN6/ERXERR/AETXERR/RB11	USER_LED3
36		N/A	VSS	POWER
37		N/A	VDD	POWER
38	RA01	62	TCK/EBIA19/AN29/RA1	TCK
39	RF13	65	TDI/EBIA18/AN30/RPF13/SCK5/RF13	TDI
40	RF12	64	TDO/EBIA17/AN31/RPF12/RF12	TDO
41	RB12	49	EBIA11/AN7/ERXD0/AECRS/PMA11/RB12	AIN13/POWER SUPPLY MONITOR
42	RB13	48	AN8/ERXD1/AECOL/RB13	AIN12/POT
43	RB14	51	EBIA1/AN9/ERXD2/AETXD3/RPB14/SCK3/PMA1/RB14	SD_SCK3
44	RB15	17	EBIA0/AN10/ERXD3/AETXD2/RPB15/OCFB/PMA0/RB15	AIN3, GPIO
45		N/A	VSS	POWER
46		N/A	VDD	POWER
47	RD14	39	AN32/AETXD0/RPD14/RD14	GPIO, U1RX
48	RD15	40	AN33/AETXD1/RPD15/SCK6/RD15	GPIO, U1TX
49	RC12	N/A	OSCI/CLKI/RC12	XTAL
50		N/A	OSCO/CLKO/RC15	XTAL
51		N/A	VBUS	POWER
52		N/A	VUSB3V3	POWER
53		N/A	VSS	POWER
54		N/A	D-	PIC32_USBD-
55		N/A	D+	PIC32_USBD+
56	RF03	N/A	USBID/RPF3/RF3	PIC32_USBID
57	RF02	0	EBIRDY3/RPF2/SDA3/RF2	GPIO, U4RX

58	RF08	1	EBIRDY2/RPF8/SCL3/RF8	GPIO, U4TX
59	RA02	37	EBICS0/SCL2/RA2	GPIO
60	RA03	4	EBIRDY1/SDA2/RA3	GPIO
61	RA04	47	EBIA14/PMCS1/PMA14/RA4	BTN2
62		N/A	VDD	POWER
63		N/A	VSS	POWER
64	RF04	61	EBIA9/RPF4/SDA5/PMA9/RF4	MRF24_RESET
65	RF05	57	EBIA8/RPF5/SCL5/PMA8/RF5	MRF24_SDI4
66	RA14	8	AETXCLK/RPA14/SCL1/RA14	GPIO, IC3, INT3
67	RA15	59	AETXEN/RPA15/SDA1/RA15	MRF24_INT4
68	RD09	56	EBIA15/RPD9/PMCS2/PMA15/RD9	MRF24_SS4
69	RD10	55	RPD10/SCK4/RD10	MRF24_SCK4
70	RD11	11	EMDC/AEMDC/RPD11/RD11	SPI_SDO2/SDI2 PWM 6, OC7
71	RD00	3	EMDIO/AEMDIO/RPD0/RTCC/INT0/RD0	PWM 1, INT0, OC1
72	RC13	N/A	SOSCI/RPC13/RC13	SOSC XTAL
73	RC14	N/A	SOSCO/RPC14/T1CK/RC14	SOSC XTAL
74		N/A	VDD	POWER
75		N/A	VSS	POWER
76	RD01	5	RPD1/SCK1/RD1	PWM 2, OC2
77	RD02	6	EBID14/ETXEN/RPD2/PMD14/RD2	PWM 3, OC3
78	RD03	9	EBID15/ETXCLK/RPD3/PMD15/RD3	PWM 4, OC4
79	RD12	38	EBID12/ETXD2/RPD12/PMD12/RD12	GPIO, T3Ck
80	RD13	50	EBID13/ETXD3/PMD13/RD13	5V POWER ENABLE
81	RD04	43	SQICS0/RPD4/RD4	USER_LED2
82	RD05	34	SQICS1/RPD5/RD5	GPIO, T4CK
83		N/A	VDD	POWER
84		N/A	VSS	POWER
85	RF00	12	EBID11/ETXD1/RPF0/PMD11/RF0	SPI_SDI2/SDO2, T5CK(+)
86	RF01	36	EBID10/ETXD0/RPF1/PMD10/RF1	GPIO, T6CK
87	RG01	60	EBID9/ETXERR/RPG1/PMD9/RG1	MRF24_HIBERNATE
88	RG00	58	EBID8/RPG0/PMD8/RG0	MRF24 SDO4
89	RA06	66	TRCLK/SQICLK/RA6	TRCLK
90	RA07	70	TRD3/SQID3/RA7	TRD3
91	RE00	26	EBID0/PMD0/RE0	GPIO
92		N/A	VSS	POWER

93		N/A	VDD	POWER
94	RE01	27	EBID1/PMD1/RE1	GPIO
95	RG14	69	TRD2/SQID2/RG14	TRD2
96	RG12	68	TRD1/SQID1/RG12	TRD1
97	RG13	67	TRD0/SQID0/RG13	TRD0
98	RE02	28	EBID2/PMD2/RE2	GPIO
99	RE03	29	EBID3/RPE3/PMD3/RE3	GPIO
100	RE04	30	EBID4/AN18/PMD4/RE4	GPIO

Declaration of Conformity

In accordance with EN ISO/IEC 17050-1:2010

Manufacturers Name:	Digilent, Inc.
Manufacturers Address:	1300 NE Henley Court
	Pullman, WA 99163
	U.S.A.

Application of Council Directives:

EMC	2004/108/EC

Standards:

EMC	EN55022:2010
	EN55024:2010

Product Name:	chipKIT WiFire
Product Model Number:	Digilent P/N 210-302
Digilent Product Category:	Small Form Factor Microcontroller Boards

We, the undersigned, hereby declare that the equipment specified above conforms to the above Directives and Standards.

Location: _Pullman, WA_____ Signature: *Clint Cole*_____

Date: _May 12, 2014___ Full Name (print): Clint Cole

Title: President_____