chipKIT WiFire Board

Reference Manual

May 12, 2014

Production Release

The production boards of the WiFIRE are manufactured using the Microchip PIC32MZ2048EFG100 MCU. Earlier pre-production, Rev B and earlier, uses the PIC32MZ2048ECG100 MCU. The MCUs are pin for pin compatible, however the PIC32MZ2048EFG100 has substantially improved ADCs, and there is an FPU coprocessor. For the most part, code written to the pre-production WiFIRE will run unaltered on the Rev C or newer WiFIREs, with the exception of the ADCs. The chipKIT core will support either MCU, even with respect to the new ADCs, as long as the chipKIT hardware abstraction API, analogRead(), was used; no sketch source code change is required. The production PCB is identical between the Rev B and Rev C, with the exception of the silk screen to indicate Rev C.

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 PIC32MZ2048EFG microcontroller. This microcontroller features a 32-bit MIPS M5150 processor core running at 200 MHhz, 2MB of flash program memory, and 512K of RAM 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.

- Microchip® PIC32MZ2048EFG100 microcontroller (200 MHz 32-bit MIPS M5150, 2MB Flash, 512K RAM)
- 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:

2 MPIDE and USB Serial Communications

The WiFire board is designed to be used with the Multi-Platform IDE (MPIDE;), the MPIDE development platform was created by modifying the Arduino™ IDE. It and is backwards-compatible with the Arduino IDE. Links for where to obtain the MPIDE installation files, and as well as instructions for installing MPIDE, can be found at _____

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 5.0Vv 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, connectoer J5 provides power to the shield. Connectoer 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 5.0Vv to the shield; however, the WiFire is not 5v tolerant and it would be very easy for a shield to destroy an input if 5.0Vv were applied to the PIC32MZ. For this reason, JP9 was added to control the voltage supplied to the shield's 5Vv source. By default, JP9 is loaded to supply only 3.3Vv on the 5.0Vv pin so that the shield does not

get 5Vv and thus cannot inadvertently apply 5.0Vv to any input to the WiFire. If the shield requires 5.0Vv to operate, the shield will not work when 3.3Vv is applied; JP9 must be selected to provide 5.0Vv for the shield to work. However, extreme caution should be used when selecting 5.0Vv on JP9 to ensure that the shield will observe IOREF and not supply 5.0Vv 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;, as such 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. It 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 5.0Vv when the WiFire is used as a USB hHost, or to provide 5.0Vv 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 5.0Vv 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 3.3Vv 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 asand 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:

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;. tThis 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 MHhz. 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 5.0Vv 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. The, 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 3.3Vv or 5.0Vv. If 5.0Vv is selected, great care must be used to ensure that no input to the PIC32MZ exceeds 3.6Vv; 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 (VOL 0.4V, VOH 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 Data Sheetdatasheet available from _____

Connectors J7 and J10 are 2×8 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 2×6 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 _____

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 as part of the chipKIT core (MPIDE) download at I_____f you have previously installed the
Digilent Network Stack as a 3rd party library, you will need to delete the Network libraries from your 3rd party sketchbook\libraries subdirectory and use the one installed with the chipKIT core (MPIDE). Having both libraries installed will cause compile time errors.

There are reference examples demonstrating the use of these libraries as part of the examples code downloaded with the chipKIT core (MPIDE).

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 5.0Vv 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 data sheet available at

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 web site 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 goes 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 not 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 10KΩohm 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 1/11th 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 tTime cClock cCalendar. 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 KHhz 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 KHhz 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 earlier, 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. JP9 has a cutt-able trace on the top of the board that can be cut to break the connection between SPI SS and reset. JP9 is not loaded at the factory. To restore the connection, solder a two pin header at the JP9 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 web site. Please note that Microchip's MPLAB®Vv8 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 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