Fubarino™ Mini Reference Manual

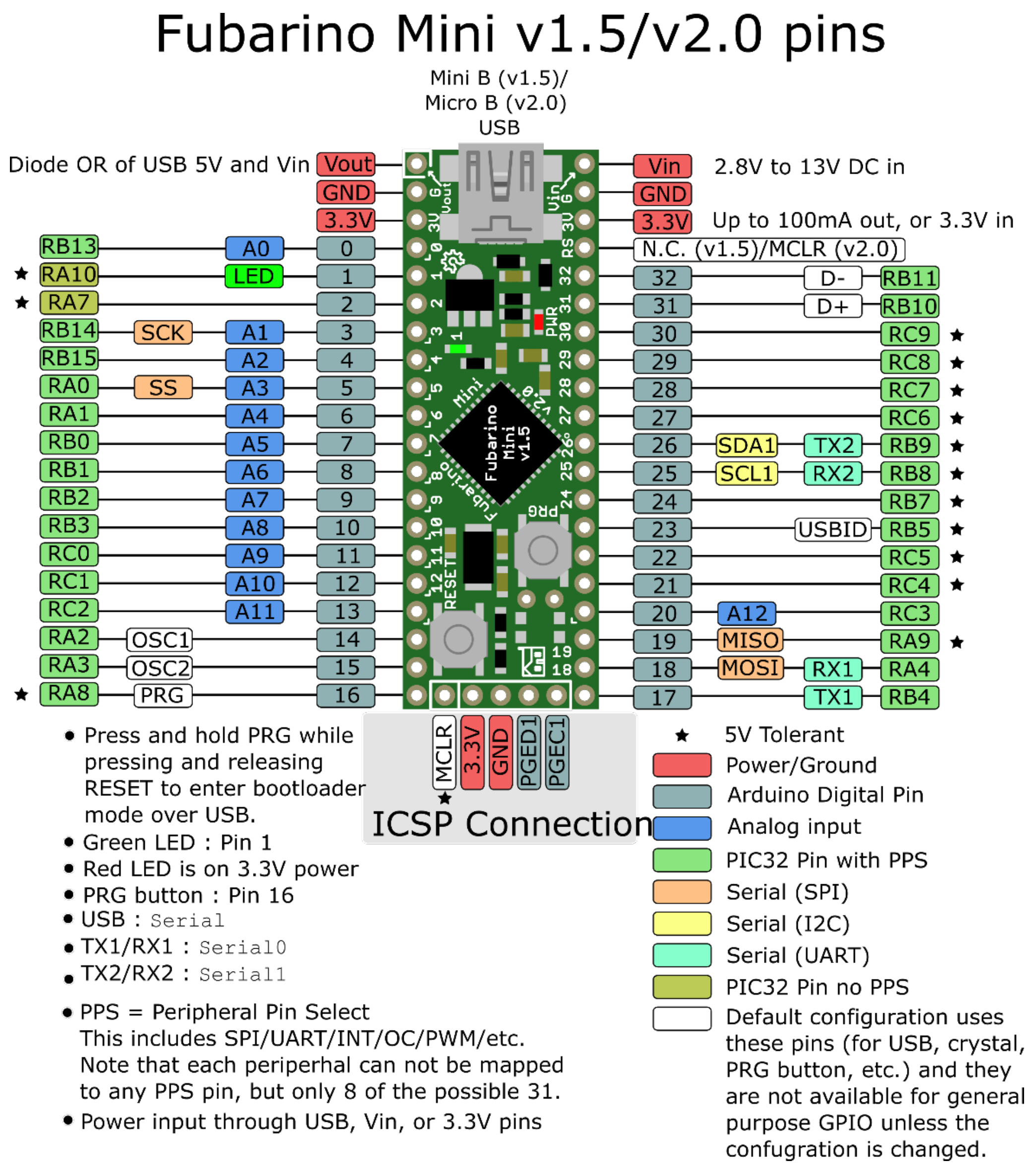
Last Revision: March 7th, 2021

Introduction:

The Fubarino™ Mini is a small microcontroller board that uses a powerful PIC32 microcontroller. It is chipKIT^{TM}/MPIDE compatible and can run the same sketches that run on an Arduino^{TM}. It is designed to have all I/O pins on either side of the board in a traditional DIP pattern so that it can easily be plugged into a breadboard. It has a USB connector for power, programming, and a connection to a PC.

Features:

- PIC32MX250F128D-50I/ML microcontroller, which includes 128KB Flash and 32K RAM (FB Mini v1.5)
- PIC32MX250F256D-50I/ML microcontroller, which includes 256KB Flash and 64K RAM (FB Mini v2.5)
- Supported as development target from within MPIDE
- Max 33 I/O pins (normally 27)
- Pads for 32 KHz crystal
- CPU runs at 48MHz
- USB connector for power, programming, and connection to PC (serial, mass storage, etc.)
- Two buttons: RESET for resetting the board, and PRG for getting into bootloader mode and user application use
- USB bootloader pre-programmed at the factory – no other hardware needed to program board\
- Separate ICSP connector – for hardware programming/debugging with PICKit3 or other ICSP programmer (not needed for use with MPIDE environment)



- Press and hold PRG while pressing and releasing RESET to enter bootloader mode over USB.
- Green LED : Pin 1
- Red LED is on 3.3V power
- PRG button : Pin 16
- USB : `Serial`
- TX1/RX1 : `Serial0`
- TX2/RX2 : `Serial1`
- PPS = Peripheral Pin Select This includes SPI/UART/INT/OC/PWM/etc. Note that each periperhal can not be mapped to any PPS pin, but only 8 of the possible 31.
- Power input through USB, Vin, or 3.3V pins

★ 5V Tolerant
🟥 Power/Ground
▨ Arduino Digital Pin
🟦 Analog input
🟩 PIC32 Pin with PPS
🟧 Serial (SPI)
🟨 Serial (I2C)
▨ Serial (UART)
▨ PIC32 Pin no PPS
□ Default configuration uses these pins (for USB, crystal, PRG button, etc.) and they are not available for general purpose GPIO unless the confugration is changed.

Figure 1 : Fubarino™ Mini Pin Map

Entering Bootloader Mode

To enter bootloader mode (also called programming mode) simply press the PRG button while pressing and releasing the RESET button. As long as the PRG button is held down when the RESET button is released, the bootloader will start waiting for a new program to be downloaded over USB. It will wait indefinitely.

The bootloader on the Fubarino™ Mini implements the avrdude STK500 v2 protocol, just like the other chipKIT™/MPIDE boards. For Windows users, you may need to install the drivers\Stk500v2.ini file so that the Fubarino™ Mini is recognized by the PC. Mac and Linux users do not have to install any drivers.

Programming in MPIDE

To program the Fubarino Mini board from within MPIDE, simply download the latest MPIDE version, http://bit.ly/getmpide, unzip and run it, and select “Fubarino Mini” from the Tools->Boards->Fubarino menu. Then, put the Fubarino Mini into bootloader mode (see above), and then select the proper serial port in the Tools->Serial Port menu.

After you enter your sketch and click the Upload button, MPIDE will compile your sketch and upload it to the Fubarino Mini. After the upload is complete, the Fubarino Mini will automatically reset and immediately begin running your sketch code. This includes sending information to the USB serial, UART1, and UART2.

Serial communications

The Fubarino Mini board has two hardware serial ports: UART1 on pins 17 (TX) and 18 (RX), and UART2 on pins 26 (TX) and 25 (RX). It also has a USB serial port that operates as the default serial port. See the Code Examples wiki page on the Fubarino Mini website for example sketch code on how to uses the three serial ports. Note that UART1 and UART2 can be configured (using PPS) to appear on multiple different I/O pins – the pins listed below are simply the defaults that are applied at boot.

- USB serial init: Serial.begin()
- UART1 pins 17 (TX), 18 (RX): Serial0.begin()
- UART2 pins 26 (TX), 25 (RX): Serial1.begin()

Power

The Fubarino Mini can be powered in a number of different ways.

1. USB: When 5V is present on the USB connector (from a PC or a powered hub for example), the Fubarino Mini will use this power source. This power source has a reverse protection diode connection to the 3.3V regulator. If both USB and Vin are powered, whichever is higher will end up providing the power to the regulator.
2. Vin pin: You can also place 2.8V to 13.2V on the Vin pin to power it from an external power source. This power source has a reverse protection diode connection to the 3.3V regulator. If both USB and Vin are powered, whichever is higher in voltage will power the device.
3. 3.3V pin(s): You can power the Fubarino Mini by connecting a 3.3V source to either 3.3V pin. You must be careful not to exceed 3.6V on these pins or the PIC32 will be destroyed.

Fubarino Mini Versions

Fubarino Mini versions 1.0 to 1.3 were not produced in any number and were not widely distributed.

Fubarino Mini version 1.4 made its way into several users’ hands and is functionally identical to version 1.5 except that it uses a 40MHz rated PIC32 part and only runs at 40MHz. Thus, the bootloader is slightly different, and the “Fubarino Mini (dev)” board should be chosen in the MPIDE Boards->Fubarino menu.

Fubarino Mini v1.5 was the first version Microchip built and uses a 50MHz rated PIC32 part. Its internal clock speed is set to 48MHz, and thus uses a slightly different bootloader than the v1.4 and earlier boards. Make sure to select "Fubarino Mini" from the MPIDE Boards->Fubarino menu for version 1.5 boards.

Fubarino Mini v2.0 was the last version built by Microchip. It has the following changes from v1.5:

- VBUS pin on PIC32 connected to VBUS net (before D1) rather than 5V net (after D1) to prevent excessive voltage on Vin from damaging VBUS pin on PIC32.
- D1 and D2 changed to handle more current (1.5A).
- 3.3V LDO changed. Higher current (800mA) and higher max input voltage (20V).
- MCU changed to PIC32MX270F256D-I/ML. Doubled the Flash and RAM from v1.5.
- USB connector changed from mini-B to micro-B.
- LED current limiting resistors changed to dim the overly-bright LEDs of v1.5.

Peripheral Pin Select

The PIC32MX250 part used on Fubarino Mini has a Peripheral Pin Select function for almost all of its I/O pins. When writing sketches for the Fubarino Mini, you must remember to connect an internal peripheral (like SPI or UART) to a particular set of I/O pins using the PPS functions (`ppsInputSelect()` and `ppsOutputSelect()`) before trying to use the peripheral if that peripheral is not connected to the pin you want by default. See the example code on the Fubarino Mini Github site for more detailed information.

Pin Reference

This table provides a cross reference of all functions and labels for each pin of the Fubarino Mini. Note that the 'Fubarino Pin' is the pin number of the whole board (as if the whole thing were one large DIP chip), starting with pin 1 = Vout, and correspond to the J1 and J2 pin numbers in the schematic. The 'Arduino Pin' is the pin number you use in your code and is what is listed on the silk screen of the board. (Note that the Mini is too small for all pins to have numbers printed next to each pin – so use the diagram above to see the pin number for every pin on the board.) Make sure to never put more than 3.3V into any pin that is not 5V tolerant, as that will damage the PIC32 CPU.

The Arduino pins 14, 15, 16, 23, 31 and 32 are taken up with existing functions in the default configuration and are not available for GPIO use unless the configuration is changed.

Fubarino Pin	Arduino Pin	Analog	PIC32 Pin	Silk Name	Alt functions	5V tolerant?
J1-1	-	-	-	Vout	-	-
J1-2	-	-	-	G	GND	-
J1-3	-	-	-	3V	-	-

J1-4	Pin 0	Yes (A0)	RB13	0	AN11/RPB13/CTPLS/PMRD/INT2	No
J1-5	Pin 1	-	RA10	1	PGED/TMS/PMA10 (LED)	Yes
J1-6	Pin 2	-	RA7	2	PGEC/TCK/CTED8/PMA7	Yes
J1-7	Pin 3	Yes (A1)	RB14	3	CVREF/AN10/C3INB/RPB14/VBUSON/SCK1/CTED5/INT1	No
J1-8	Pin 4	Yes (A2)	RB15	4	AN9/C3INA/RPB15/SCK2/CTED6/PMCS1/INT4	No
J1-9	Pin 5	Yes (A3)	RA0	5	PGED3/VREF+/CVREF+/AN0/C3INC/RPA0/CTED1/PMD7	No
J1-10	Pin 6	Yes (A4)	RA1	6	PGEC3/VREF-/CVREF-/AN1/RPA1/CTED2/PMD6/INT3	No
J1-11	Pin 7	Yes (A5)	RB0	7	PGED1/AN2/C1IND/C2INB/C3IND/RPB0/PMD0	No
J1-12	Pin 8	Yes (A6)	RB1	8	PGEC1/AN3/C1INC/C2INA/RPB1/CTED12/PMD1	No
J1-13	Pin 9	Yes (A7)	RB2	9	AN4/C1INB/C2IND/RPB2/SDA2/CTED13/PMD2	No
J1-14	Pin 10	Yes (A8)	RB3	10	AN5/C1INA/C2INC/RTCC/RPB3/SCL2/PMWR/CNB3	No
J1-15	Pin 11	Yes (A9)	RC0	11	AN6/RPC0	No
J1-16	Pin 12	Yes (A10)	RC1	12	AN7/RPC1	No
J2-17	Pin 13	Yes (A11)	RC2	13	AN8/RPC2/PMA2	No
J2-18	Pin 14	-	RA2	14	OSC1/CLKI/PRA2 (used - crystal)	No
J2-19	Pin 15	-	RA3	15	OSC2/CLKO/RPA3 (used - crystal)	No
J2-20	Pin 16	-	RA8	16	TDO/RPA8/PMA8 (PRG button)	Yes
J2-1	Pin 17	-	RB4	17	SOSCI/PRB4	No
J2-2	Pin 18	-	RA4	18	SOSCO/RPA4/T1CK/CTED9	No
J2-3	Pin 19	-	RA9	19	TDI/RPA9/PMA9	Yes
J2-4	Pin 20	Yes (A12)	RC3	20	AN12/RPC4	No
J2-5	Pin 21	-	RC4	21	RPC4/PMA4	Yes
J2-6	Pin 22	-	RC5	22	RPC5/PMA3	Yes
J2-7	Pin 23	-	RB5	23	PRB5/USBID (used - USB)	Yes
J2-8	Pin 24	-	RB7	24	RPB7/CTED3/PMD5/INT0	Yes
J2-9	Pin 25	-	RB8	25	RPB8/SCL1/CTED10/PMD4	Yes
J2-10	Pin 26	-	RB9	26	RPB9/SDA1/CTED4/PMD3	Yes
J2-11	Pin 27	-	RC6	27	RPC6/PMA1	Yes
J2-12	Pin 28	-	RC7	28	RPC7/PMA0	Yes
J2-13	Pin 29	-	RC8	29	RPC8/PMA5	Yes
J2-14	Pin 30	-	RC9	30	RPC9/CTED7/PMA6	Yes
J2-15	Pin 31	-	RB10	31	PGED2/RPB10/D+/CTED11 (used - USB)	No
J2-16	Pin 32	-	RB11	32	PGEC2/RPB11/D- (used - USB)	No
J2-17 (v1.5)	-	-	-	G	Do not use (reserved for future use)	-
J2-17			MCLR	RS	Master Clear (reset)	-

(v2.0)						
J2-18	-	-	-	3V	3.3V	-
J2-19	-	-	-	G	GND	-
J2-20	-	-	-	Vin	Vin	-

Schematic

For the v1.5 schematic, please see the Fubarino_Mini_v15_sch.pdf file on the Fubarino Mini website. https://github.com/fubarino/fubarino.github.com/blob/master/mini/v15/FubarinoMini_v15_sch.pdf?raw=true

For the v2.0 schematic, please see the Fubarino_Mini_v20_sch.pdf file on the Fubarino Mini website. https://github.com/fubarino/fubarino.github.com/blob/master/mini/v20/FubarinoMini_v20_schematic.pdf?raw=true

Default Peripherals

There are several hardware peripherals that, by default, are mapped to certain pins.

PWM

The Arduino `analogWrite()` function turns on hardware PWM on certain pins. By default, the pins of the Fubarino Mini that can output hardware PWM with the `analogWrite()` function are 0, 4, 7, 8 and 9.

Interrupts

The Arduino `attachInterrupt()` function is used to connect a software function to one of five hardware interrupts.

Interrupt Pins	
Digital Pin	**Function**
24 (PIN_INT0)	INT0
3 (PIN_INT1)	INT1
0 (PIN_INT2)	INT2
4 (PIN_INT3)	INT3
5 (PIN_INT4)	INT4

The `attachInterrupt()` function takes three arguments. The first is an interrupt number, the second is the function to call and the third is the edge to interrupt on (RISING or FALLING). You can use the `digitalPinToInterrupt()` function to look up which interrupt number a given pin has been assigned. For example, if you wanted to have the `foo()` function called on the rising edge of PIN_INT3, write:

```
attachInterrupt(digitalPinToInterrupt(PIN_INT3), foo, RISING);
```

While only INT0 is on a fixed pin, and INT1 through INT4 can be changed using PPS, the chipKIT core function `attachInterrupt()` will always re-set INT1 through INT4 to the above digital pins each time you call it.

SPI

While the Fubarino Mini has 2 SPI ports, the standard Arduino SPI library <SPI.h> only allows the use of one. On the Fubarino Mini, the SPI port used for this library is SPI1, which uses the following pins:

SPI1 – Default SPI Port for SPI.h and DSPI.h		
Digital Pin	**Function**	**Changeable?**
3	SCK (clock)	No
18	MOSI (SDO)	Yes - via PPS
19	MISO (SDI)	Yes – via PPS
5	SS (slave select/chip select)	Yes – any GPIO pin

The DSPI library (<DSPI.h>) allows the use of both SPI ports – both SPI1 and SPI2. These are the pins used for SPI2:

SPI2 – (DSPI only)		
Digital Pin	**Function**	**Changeable?**
4	SCK (clock)	No
29	MOSI (SDO)	Yes - via PPS
28	MISO (SDI)	Yes – via PPS
30	SS (slave select/chip select)	Yes – any GPIO pin

More Information

The best places to get support for Fubarino Mini are

- Main chipKIT website (http://chipkit.net/)
- chipKIT forum (http://www.chipkit.org/forum/index.php)

License

The Fubarino series of boards are released by Schmalz Haus LLC and FubarLabs under a Solderpad Hardware License v0.51. For more information, see: http://solderpad.org/licenses/SHL-0.51/

Developers

The Fubarino series of boards were developed by Rick Anderson of FubarLabs and Brian Schmalz of Schmalz Haus LLC. They are registered chipKIT^{TM} boards and are designed to be programmed with the MPIDE system.