

Zigduino r1

Description

The Zigduino is an Arduino-compatible microcontroller platform that integrates an 802.15.4 radio on the board. The radio can be configured to support any 802.15.4-based protocol, including ZigBee, Route Under MAC/6LoWPAN, and RF4CE. It uses a reverse polarity SMA connector (RP-SMA) for an external antenna. This allows the user to use nearly any existing 2.4 GHz antenna with it. The Zigduino runs on 3.3V, but all I/O pins are 5V compatible.

Summary

Microcontroller	Atmega128RFA1
Operating voltage	3.3V
Input voltage (recommended)	7-18V
Input voltage (maximum)	6-30V (transients to -20V and +60V)
Digital I/O pins	14 + 3 auxiliary
PWM output pins	6
Analog input pins	6 (0-1.8V)
I/O protection	±30V transient -2.5V to +5.8V continuous
DC current per I/O pin	20 mA
DC current for 5V pin	250 mA
DC current for 3.3V pin	200 mA
Flash memory	128 KB of which 2 KB is used by BL
SRAM	16 KB
EEPROM	4 KB
Clock speed	16 MHz
RF transmit power	+3.5 dBm
Receiver sensitivity	-100 dB
Antenna gain	2 dBi
Current draw	30 mA (transmitting, USB, no I/O connections) 15 mA (transmitting, no USB, no I/O connections) 6 mA (radio off, no USB, no I/O connections) 250 µA (sleep)

Compatibility

- Compatible with any shield that supports 3.3V logic
- Compatible with existing Arduino libraries that do not use hard-coded pin definitions
- Compatible with Arduino IDE with updated compiler, avr-gcc-4.3.3 or later.

Software

- [Ported bootloader and Arduino core libraries](#), on [Github](#).
- [IEEE 802.15.4 MAC](#), from [Atmel](#)
- [BitCloud](#), Atmel's ZigBee PRO implementation

Power

The Zigduino can be powered through the USB connection or with an external power supply. The power source with the highest voltage is selected automatically.

External power can be supplied via a wall wart or a battery. It can be connected with a 2.1mm center-positive plug inserted into the power jack. Alternately, external power can be connected through the GND and VIN pins of the POWER header.

The board will operate correctly on an input voltage between 6V and 30V. It will survive transients as large as -20V or +60V. However, higher supply voltages may cause excessive heat dissipation at higher current draws. The input voltage regulator has integral overtemperature protection, so you can't permanently damage the board this way. However, the board may not work correctly under these circumstances.

The power pins are as follows:

- **VIN** -- The input voltage to the Arduino board when it is running from external power, i.e. not USB bus power.
- **5V** -- The regulated 5V used to power 5V components on the board and external 5V shields. It comes either from the USB or from the VIN via the 5V regulator. Maximum current draw is 250 mA.
- **3V3** -- The regulated 3.3V supply that powers the microcontroller. It is derived from the 5V bus via a second regulator. Maximum current draw is 200 mA.
- **GND** -- Ground pins.

Memory

The ATmega128RFA1 has 128 KB of flash memory, of which 2 KB is occupied by the bootloader. It also has 16 KB of SRAM (the most of any Arduino-compatible board) and 4 KB of EEPROM, which can be accessed through the [EEPROM library](#).

Input and Output

Each of the 14 digital pins of the Zigduino can be used as an input or output, using `pinMode`, `digitalWrite`, and `digitalRead`. Each pin operates at 3.3V and can source or sink 10 mA. Each also has an internal pullup, which is disabled by default. Each pin is protected against $\pm 30V$ spikes and can tolerate continuous 5V input.

The six analog input pins, labeled A0 - A5, are likewise protected against $\pm 30V$ spikes and can tolerate continuous 5V input. Each provides 10 bits of resolution and measures 0 - 1.8V. It is possible to change to a lower top voltage through use of the AREF pin and the `analogReference` function.

A key design goal of the Zigduino is maintaining compatibility with existing shields to the greatest extent possible. The ATmega128RFA1's peripherals are arranged slightly differently than the corresponding peripherals on the ATmega328 used in the stock Arduino. Therefore, in order to provide the desired shield compatibility, there are three solder jumpers provided on the back of the board. They function as follows:

- Digital pin 11 can be set as either SPI MOSI or a PWM output. Neither option is selected as shipped. SPI MOSI is also available on the SPI connector at all times along with SCK and MISO.
- Analog pin 4 can be set as either A4 or I2C SDA. Neither option is selected as shipped. Both I2C pins are available on the I2C connector.
- Analog pin 5 can be set as either A4 or I2C SCL. Neither option is selected as shipped. Both I2C pins are available on the I2C connector.

The following additional special functions are available:

- **Serial: 0 (RX) and 1 (TX)** -- Used to transmit and receive TTL serial data. These pins are connected to the corresponding pins on the FTDI USB interface chip.
- **PWM: 3, 5, 6, 9, 10, and 11** -- Provides 8-bit PWM output with the `analogWrite` function. Pin 11 must be selected for PWM operation with the solder jumper on the back of the board.
- **SPI: 11 (MOSI), 12 (MISO), 13 (SCK)** -- These pins support SPI communications using the SPI library. Pin 11 must be selected for SPI operation with the solder jumper on the back, or SPI must be accessed with the SPI connector.
- **LED: 13** -- This is the built-in LED on digital pin 13. When the pin is high, the LED is on.

- **External Interrupts: 2, 3, 6, and 7** -- These pins can be configured to trigger and interrupt on a low value, high value, or an edge. See the attachInterrupt function for details. The two I2C pins can also be used as interrupts.
- **I2C: A4 (SDA) and A5 (SCL)** -- These pins support I2C communications using the Wire library. They must be selected for I2C operation with the jumpers on the back or I2C must be accessed through the I2C connector. They can also be configured as interrupts.

Documentation

- [Zigduino schematic \(pdf\)](#)
- [Zigduino schematic \(Eagle\)](#)
- [Zigduino layout \(pdf\)](#)
- [Zigduino layout \(Eagle\)](#)
- [Zigduino manual \(draft, pdf\)](#)

Firmware

- [Zigduino Core \(github\)](#)
- [BitCloud board configuration file](#)