ARM Core Benefits

The Zero has a 32-bit ARM core that can outperform typical 8-bit microcontroller boards. The most significant differences are:

- 32-bit core that allows operations on 4 byte wide data within a single CPU clock.
- CPU Clock at 48MHz
- 12 channels DMA controller that can relieve the CPU from doing memory intensive tasks
- 32 bit Real Time Counter (RTC) with clock/calendar function
- 32 bit CRC generator
- Two-channel Inter IC Sound (I2S) interface
- Peripheral Touch Controller (PTC)

Atmel Embedded Debugger

The Atmel Embedded Debugger (EDBG) implements a JTAG interface in order to program the on-board SAMD21 and is also connected to hardware serial of the microcontroller. This means that the 'Serial' class responds to the programming port of the board. The Arduino Zero has been designed in collaboration with ATMEL, and the on-board EDBG can be used through ATMEL Studio to get full access to the microcontroller memories to help debug your code.

Power

The Arduino Zero can be powered via the USB connector or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (such as a wall-wart) or battery, and can be connected using a 2.1mm center-positive plug connected to the board's power jack, or directly to the GND and VIN pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or if supplying voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board if it is not sufficiently regulated. We don't advise it.
- 3.3V. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 800 mA. This regulator also provides power to the SAMD21 microcontroller.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The SAMD21 has 256 KB Flash Memory. It also has 32 KB of SRAM and up to 16 KB of EEPROM by emulation.

Input and Output

Each of the 20 general purpose I/O pins on the Zero can be used for digital input or digital output using pinMode(), digitalWrite(), and digitalRead() functions. Pins that can be used for PWM output are: 3, 4, 5, 6, 8, 9, 10, 11, 12, 13 using analogWrite() function. All pins operate at 3.3 volts. Each pin can source or sink a maximum of 7 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms.

In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the Serial1 class.

- External Interrupts: available on all the pins except pin 4.
- DAC: A0. Provide a 10bit voltage output with the analogWrite() function.
- PWM: 3, 4, 5, 6, 8, 9, 10, 11, 12, 13. Provide 8-bit PWM output with the analogWrite() function.

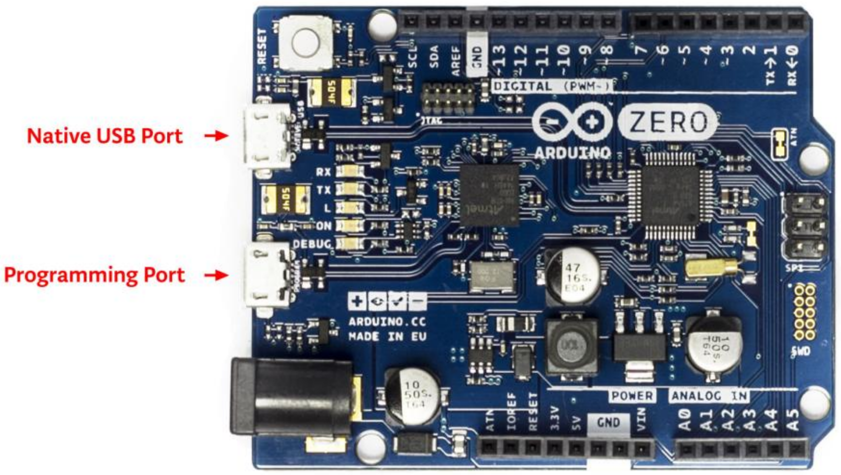
- SPI: SS, MOSI, MISO, SCK. Located on the SPI header support SPI communication using the SPI library.
- LED: 13. There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- Analog Inputs. Six of the 20 general purpose I/O pins on the Zero provide analog input. These are labeled A0 through A5, and each provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 3.3 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function.
- TWI: SDA pin and SCL pin. Support TWI communication using the Wire library

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Programming

Uploading sketches to the SAMD21 is different from the AVR microcontrollers found in other Arduino boards.



Either USB port can be used for programming the board, although using the Programming port is recommended due to the way the erasing of the chip is handled:

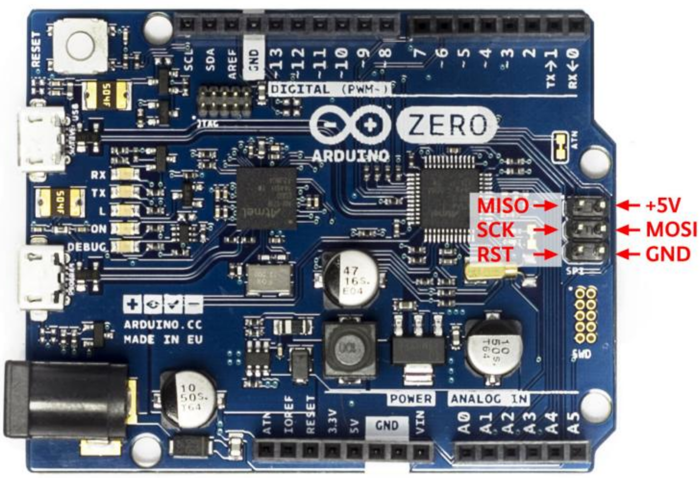
- Programming port: To use this port, select "Arduino Zero (Programming Port)" as your board in the Arduino IDE. Connect the Zero's programming port (the one closest to the DC power jack) to your computer. The programming port uses the EDBG as a USB-to-JTAG chip.

- Native port: To use this port, select "Arduino Zero (Native USB Port)" as your board in the Arduino IDE. The Native USB port is connected directly to the SAMD21. Connect the Zero's Native USB port (the one closest to the reset button) to your computer.

Unlike other Arduino boards which use avrdude for uploading, the Zero relies on bossac while the programming port uses openOCD .

ICSP Connector used for SPI communication

Here are details of the SPI pins location within the ICSP connector:



EEPROM

Part of the Flash memory may be used as a non-volatile storage with some limitations, the lifetime of the typical flash memory is about 25K write-cycles, and unlike EEPROM, and it must be erased in pages before writing. The flash memory is erased when Arduino uploads a new sketch.

Burn the Bootloader

Using the Arduino Zero Programming Port it is possible to burn the booloader used by the Native USB port. To burn the bootloader follow this procedure:

- select Tools->Programmer->Atmel EDBG

- select Tools->Board->Arduino Zero (Programming Port)
- select Tools->Burn Bootloader

USB Overcurrent Protection

The Arduino Zero has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Arduino Zero PCB are 2.7 and 2.1 inches respectively, with the USB connectors and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins. The Arduino Zero is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and "ICSP" (SPI) header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1).