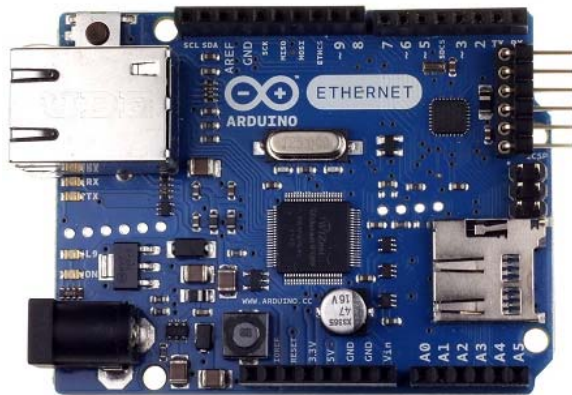
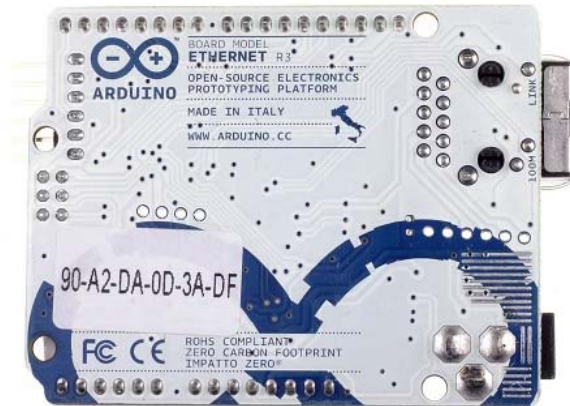


Search the Arduino Website

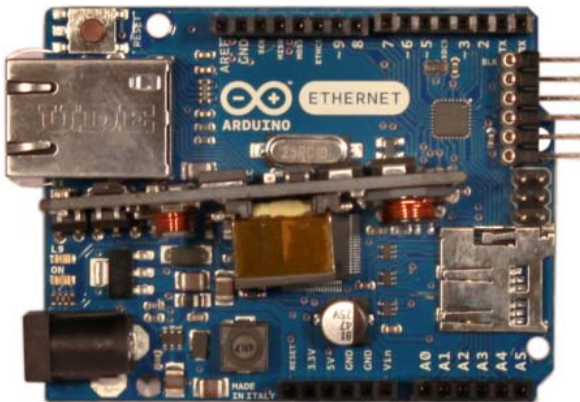
Arduino Ethernet



(<http://arduino.cc/en/uploads/Main/ArduinoEthernetFront.jpg>)
 Arduino Ethernet Rev. 3 board front view



(<http://arduino.cc/en/uploads/Main/ArduinoEthernetBack.jpg>)
 Arduino Ethernet Rev. 3 board rear view



(<http://arduino.cc/en/uploads/Main/ArduinoEthernetFrontPOE.jpg>)
 Arduino Ethernet Rev. 2 board front view with optional PoE module

w/o POE module

with POE module

Buy From
Arduino Store

Buy From
Arduino Store

Buy From
Distributors

(http://store.arduino.cc/eu/index.php?main_page=product_info&cPath=11_12&products_id=141)

(http://store.arduino.cc/ww/index.php?main_page=product_info&cPath=11_12&products_id=142)

(<http://arduino.cc/en/Main/B>)

Overview

The Arduino Ethernet is a microcontroller board based on the ATmega328 (datasheet (http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf)). It has 14 digital input/output pins, 6 analog inputs, a 16 MHz crystal oscillator, a RJ45 connection, a power jack, an ICSP header, and a reset button.

NB: Pins 10, 11, 12 and 13 are reserved for interfacing with the Ethernet module and should not be used otherwise. This reduces the number of available pins to 9, with 4 available as PWM outputs.

An optional Power over Ethernet module can be added to the board as well.

The Ethernet differs from other boards in that it does not have an onboard USB-to-serial driver chip, but has a Wiznet Ethernet interface. This is the same interface found on the Ethernet shield.

An onboard microSD card reader, which can be used to store files for serving over the network, is accessible through the SD Library. Pin 10 is reserved for the Wiznet interface, SS for the SD card is on Pin 4.

The 6-pin serial programming header is compatible with the USB Serial (<http://arduino.cc/en/Main/USBSerial>) adapter and also with the FTDI USB cables or with Sparkfun and Adafruit FTDI-style basic USB-to-serial breakout boards. It features support for automatic reset, allowing sketches to be uploaded without pressing the reset button on the board. When plugged into a USB to Serial adapter, the Arduino Ethernet is powered from the adapter.

The Revision 3 of the board introduces the standardized 1.0 pinout, that consist in:

- added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, this will the opportunity to shield that use i2c or TWI components to be compatible with all the Arduino boards;
- the IOREF that allow the shields to adapt to the voltage provided from the board. Shields that use the IOREF pin will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. Next to the IOREF pin there is a not connected pin, that is reserved for future purposes.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage Plug (recommended)	7-12V
Input Voltage Plug (limits)	6-20V
Input Voltage PoE (limits)	36-57V
Digital I/O Pins	14 (of which 4 provide PWM output)
Arduino Pins reserved:	10 to 13 used for SPI 4 used for SD card 2 W5100 interrupt (when bridged)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
W5100 TCP/IP Embedded Ethernet Controller	
Power Over Ethernet ready Magnetic Jack	
Micro SD card, with active voltage translators	

Schematic & Reference Design

EAGLE files: Arduino-ethernet-R3-reference-design.zip (<http://arduino.cc/en/uploads/Main/arduino-ethernet-R3-reference-design.zip>)

Schematic: Arduino-ethernet-R3-schematic.pdf (<http://arduino.cc/en/uploads/Main/arduino-ethernet-R3-schematic.pdf>)

Power

The board can also be powered via an external power supply, an optional Power over Ethernet (PoE) module, or by using a FTDI cable/USB Serial connector.

External power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. 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. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- 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.

The optional PoE module is designed to extract power from a conventional twisted pair Category 5 Ethernet cable:

- IEEE802.3af compliant
- Low output ripple and noise (100mVpp)
- Input voltage range 36V to 57V
- Overload and short-circuit protection
- 9V Output
- High efficiency DC/DC converter: typ 75% @ 50% load
- 1500V isolation (input to output)

NB: the Power over Ethernet module is proprietary hardware not made by Arduino, it is a third party accessory. For more information, see the datasheet (<http://arduino.cc/en/uploads/Main/PoE-datasheet.pdf>)

When using the power adapter, power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library (<http://arduino.cc/en/Reference/EEPROM>)).

Input and Output

Each of the 14 digital pins on the Ethernet board can be used as an input or output, using pinMode() (<http://arduino.cc/en/Reference/PinMode>), digitalWrite() (<http://arduino.cc/en/Reference/DigitalWrite>), and digitalRead() (<http://arduino.cc/en/Reference/DigitalRead>) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 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.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` (<http://arduino.cc/en/Reference/AttachInterrupt>) function for details.
- PWM: 3, 5, 6, 9, and 10. Provide 8-bit PWM output with the `analogWrite()` (<http://arduino.cc/en/Reference/AnalogWrite>) function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library (<http://arduino.cc/en/Reference/SPI>).
- LED: 9. There is a built-in LED connected to digital pin 9. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. On most other arduino boards, this LED is found on pin 13. It is on pin 9 on the Ethernet board because pin 13 is used as part of the SPI connection.

The Ethernet board has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the `analogReference()` (<http://arduino.cc/en/Reference/AnalogReference>) function. Additionally, some pins have specialized functionality:

- TWI: A4 (SDA) and A5 (SCL). Support TWI communication using the Wire library (<http://arduino.cc/en/Reference/Wire>).

There are a couple of other pins on the board:

- AREF. Reference voltage for the analog inputs. Used with `analogReference()` (<http://arduino.cc/en/Reference/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.

See also the mapping between Arduino pins and ATmega328 ports (<http://arduino.cc/en/Hacking/PinMapping168>).

Communication

The Arduino Ethernet has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers.

A `SoftwareSerial` library (<http://www.arduino.cc/en/Reference/SoftwareSerial>) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports TWI and SPI communication. The Arduino software includes a Wire library to simplify use of the TWI bus; see the documentation (<http://arduino.cc/en/Reference/Wire>) for details. For SPI communication, use the SPI library (<http://arduino.cc/en/Reference/SPI>).

The board also can connect to a wired network via ethernet. When connecting to a network, you will need to provide an IP address and a MAC address. The Ethernet Library (<http://arduino.cc/en/Reference/Ethernet>) is fully supported.

The onboard microSD card reader is accessible through the SD Library (<http://arduino.cc/en/Reference/SD>). When working with this library, SS is on Pin 4.

Programming

It is possible to program the Arduino Ethernet board in two ways: through the 6 pin serial programming header, or with an external ISP programmer.

The 6-pin serial programming header is compatible with FTDI USB cables and the Sparkfun and Adafruit FTDI-style basic USB-to-serial breakout boards including the Arduino USB-Serial connector. It features support for automatic reset, allowing sketches to be uploaded without pressing the reset button on the board. When plugged into a FTDI-style USB adapter, the Arduino Ethernet is powered off the adapter.

You can also program the Ethernet board with an external programmer like an AVRISP mkII or USBTinyISP. To set up your environment for burning a sketch with a programmer, follow these instructions (<http://arduino.cc/en/Hacking/Programmer>). This will delete the serial bootloader, however.

All the Ethernet example sketches work as they do with the Ethernet shield. Make sure to change the network settings for your network.

Physical Characteristics

The maximum length and width of the Ethernet PCB are 2.7 and 2.1 inches respectively, with the RJ45 connector and power jack extending beyond the former dimension. Four 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.

Setup

If you want to use a FTDI cable to download your sketches on the Arduino Ethernet, please refer to this guide: Upgrade the Arduino Ethernet bootloader to the latest version (<http://scuola.arduino.cc/en/content/upgrade-arduino-ethernet-bootloader-latest-version>)

COMMENTS



Mohamed_Batal Friday, February 15, 2013

How Can i program this board using foca ?
and what is the description of the 6 pin which connect to the FTDI usb cable



matthijskooijman Saturday, March 23, 2013

The 6pin header connections are the same as the pinout on the USB serial light adapter linked above. On the Arduino side, the pins are like this. Pin 1 is the pin 1 is the closest to the regular TX / RX pins and the pin 6 is next to the micro SD reader.

1. GND
2. Not connected
3. 5V
4. RX
5. TX
6. Reset

These signals are from the perspective of the Arduino, so RX lets the Arduino receive data from the computer, while TX lets the Arduino send data to the computer (it depends on how your serial adapter is labeled if you need to connect RX-RX and TX-TX or if you need to cross them. You might just need to try both).

The reset pin allows automatic reset and is commonly connected to the DTR pin of your serial port to auto reset when the serial port is opened (but you can also use manual reset, where you press the reset button and then quickly start the upload process within the first second or so after releasing the reset button).



mshishkin Sunday, March 31, 2013

Just got Ethernet R3. There are 2 extra pins right above RESET pin, one marked as IOREF, another unmarked. Any references on use and purpose of those pins?



sanhardik Monday, April 08, 2013

The schematic on the website for Arduino Ethernet seems to be of an older rev. I have got the Rev 3 Arduino Ethernet board and I see differences. Can you please post the latest schematic and eagle files ?



luisborlido Friday, April 26, 2013

I also have the Rev 3 Arduino Ethernet. Can you please update this page with the latest eagle files?
Thanks!

AC3000 Monday, April 29, 2013

Hi,
pictures and schematics have been updated now.

AC



spikeyhair Thursday, May 02, 2013

Hi,

What Current does the power supply have to be?

AC3000 Thursday, May 02, 2013

Hi Spikeyhair,
the current has to be at least 1A.

AC



chiwheels Saturday, May 11, 2013

I am looking at using the Arduino Ethernet module as a daughter board on a PCB I will design for a product.

I know that there is a physical drawing showing the outline of the board and mounting hole positions but is there a proper dimensioned drawing showing all the physical positions of the connector pins. I need this to create a CAD profile.

I know that you have data for the Eagle CAD system but I am not using that; instead I am using Design Spark.

ACicchi Monday, May 13, 2013

Hi Chiwheels,
you can write to support@arduino.cc in order to have those files.

Regards,
AC



torero69 Thursday, June 06, 2013

Hi, are the previous versions of the ethernet shield (without the 4 new pins) compatible with Arduino Uno R3??

Thanks.

ACicchi Friday, June 07, 2013

Hi torero69,
yes, it is compatible :)

You are welcome,
AC



molahs Saturday, June 22, 2013

Hi,
I want to connect this board to an UNO, but this board doesn't have any bottom pins. Which pins must be connected? Thank you.

ACicchi Tuesday, June 25, 2013

Hi Molahs,
this is not the Ethernet Shield... It is a standalone board that has the characteristics of the Uno and the Ethernet together. So you can use it as a Uno with all the potentialities of an Ethernet connection.

AC



FrankyP Monday, July 08, 2013

Hi there!

Does anybody of you guys maybe know how fast the ethernet board is? I want to record Videodata, they're sent in udp-packetes and the rate is about 10Mbit/sec.

Would appreciate if somebody could tell me, especially if somebody build a similar project!

ACicchi Monday, July 08, 2013

Hi FrankyP,

I'm not so sure that the Ethernet can manage that rate, but we still didn't try yet. You could post the same question on the official forum (<http://forum.arduino.cc/>) under the proper section. Maybe somebody has already encountered the same issue.

Regards,
AC



FrankyP Tuesday, July 09, 2013

Thanks ACicchi, I will do that :-)



Slav Wednesday, August 07, 2013

Hi there,

I use a FTDI cable to download my sketches on a Arduino Ethernet and I need to refresh my bootloader for the Arduino Ethernet R3. In February I used the link provided in "Setup" above to just do that. Now the link is dead(!) and has been for a while. If the link cannot be brought to live, can anyone please advise how I burn the latest bootloader for my Arduino Ethernet - using a Arduino Uno R3 as programmer (ISP).

Thank you



a_guadalupi Wednesday, August 07, 2013

Take a look here arduino.cc/en/Tutorial/ArduinoISP



Slav Wednesday, August 07, 2013

Thank you - I already looked there, but see no examples of programming an Arduino Ethernet. Pin 10-13 is reserved for SPI on that board and cannot be used. So where do I connect Pin 11, 12 and 13 from the Arduino Uno? I remember that was explained in the now-dead link...



a_guadalupi Wednesday, August 07, 2013

You can use the SPI connector
<http://arduino.cc/en/Reference/SPI>
 the white point on the board indicates the 1



Slav Wednesday, August 07, 2013

So 11, 12 and 13 goes to MOSI, MISO and SCK respectively.
 Sorry for my ignorance - I will read up on your reference link and try the programming to night. BR Steffen



Larivierec Tuesday, September 10, 2013

Would it be possible to combine this board with a mega?

My Arduino ethernet doesn't have any header pins (its not a shield) is there any way of combining the two?



Angelo9999 Wednesday, September 11, 2013

Hello Larivierec,
 To combine the two boards you can pass data between the them via the serial communication and let the arduino ethernet doing all the ethernet stuff.



Larivierec Wednesday, September 11, 2013

Thank you very much for your input ill try this as soon as i receive the mega! :)

You must be logged in to post a comment.

Like 98 Tweet 28

3

©Arduino [Copyright Notice \(/en/Main/CopyrightNotice\)](#) [Contact us \(/en/Main/ContactUs\)](#)