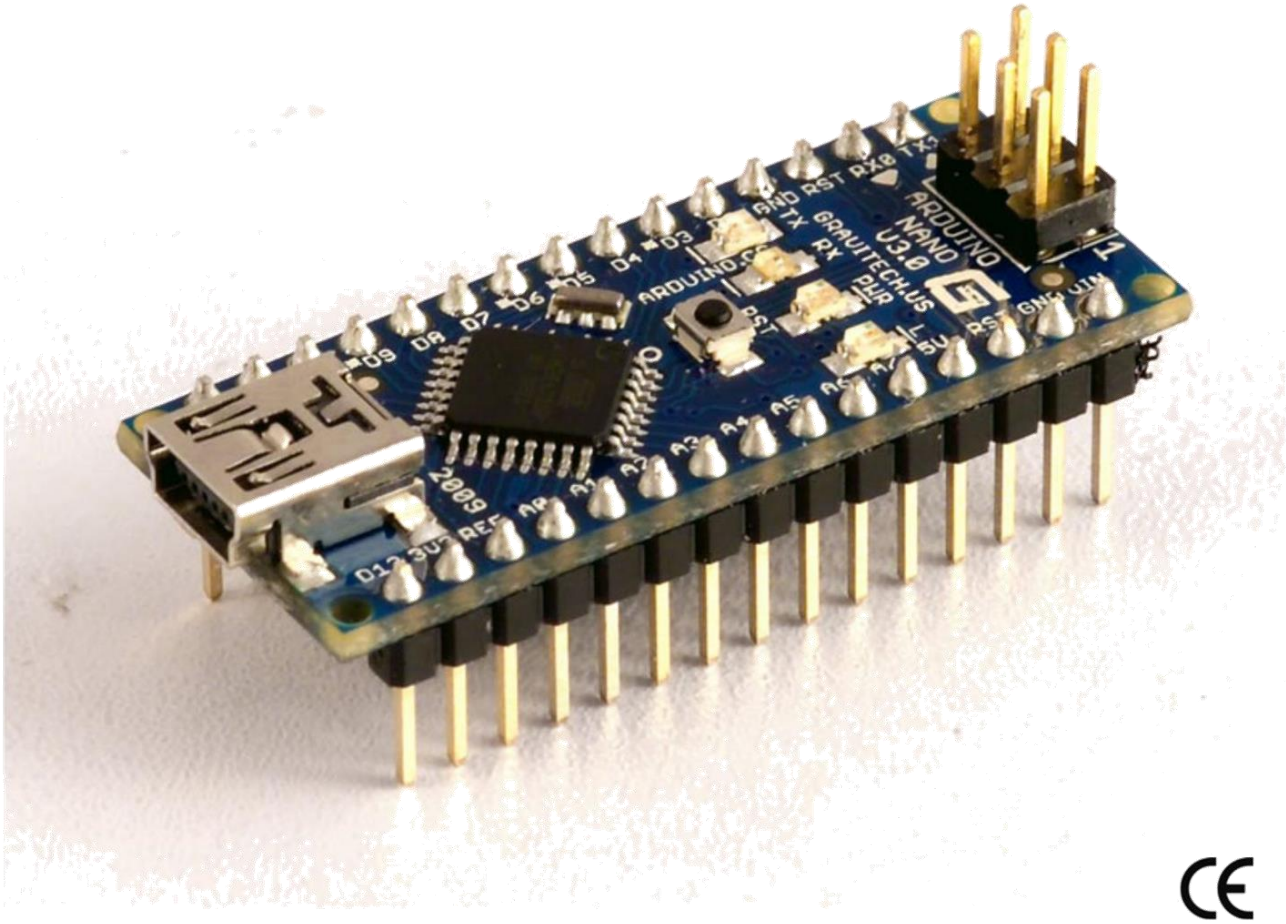
Arduino Nano



CE

Product Overview

The Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino Nano 3.0) or ATmega168 (Arduino Nano 2.x). It has more or less the same functionality of the Arduino Duemilanove, but in a different package. It lacks only a DC power jack, and works with a Mini-B USB cable instead of a standard one. The Nano was designed and is being produced by Gravitech.

Index

Technical Specifications — Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials — Page 6

Terms & Conditions — Page 7

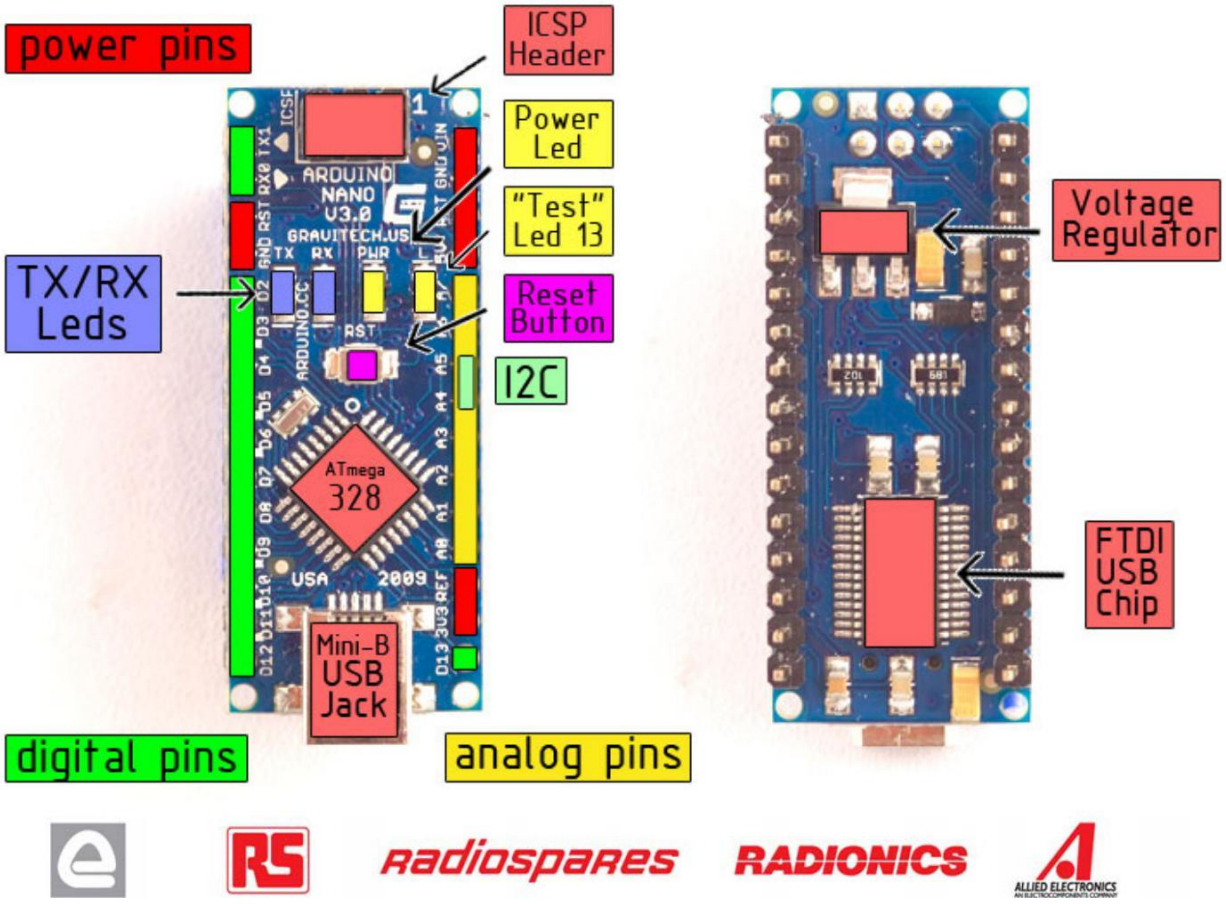
Arduino Nano 3.0 (ATmega328): schematic, Eagle files.

Arduino Nano 2.3 (ATmega168): manual (pdf), Eagle files. *Note:* since the free version of Eagle does not handle more than 2 layers, and this version of the Nano is 4 layers, it is published here unrouted, so users can open and use it in the free version of Eagle.

Summary

Microcontroller	Atmel ATmega168 or ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	16 KB (ATmega168) or 32 KB (ATmega328) of which 2 KB used by bootloader
SRAM	1 KB (ATmega168) or 2 KB (ATmega328)
EEPROM	512 bytes (ATmega168) or 1 KB (ATmega328)
Clock Speed	16 MHz
Dimensions	0.73" x 1.70"

the board



Power

The Arduino Nano can be powered via the Mini -B USB connection, 6-20V unregulated external power supply (pin 30), or 5V regulated external power supply (pin 27). The power source is automatically selected to the highest voltage source.

The FTDI FT232RL chip on the Nano is only powered if the board is being powered over USB. As a result, when running on external (non-USB) power, the 3.3V output (which is supplied by the FTDI chip) is not available and the RX and TX LEDs will flicker if digital pins 0 or 1 are high.

Memory

The ATmega168 has 16 KB of flash memory for storing code (of which 2 KB is used for the bootloader); the ATmega328 has 32 KB, (also with 2 KB used for the bootloader). The ATmega168 has 1 KB of SRAM and 512 bytes of EEPROM (which can be read and written with the EEPROM library); the ATmega328 has 2 KB of SRAM and 1 KB of EEPROM.

Input and Output

Each of the 14 digital pins on the Nano can be used as an input or output, using pinMode(), digitalWrite(), and 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. These pins are connected to the corresponding pins of the FTDI USB-to-TTL Serial chip.

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() function for details.

PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.

LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Nano has 8 analog inputs, 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 analogReference() function. Additionally, some pins have specialized functionality:

I^2C: 4 (SDA) and 5 (SCL). Support I^2C (TWI) communication using the Wire library (documentation on the Wiring website).

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.

See also the mapping between Arduino pins and ATmega168 ports.

Communication

The Arduino Nano has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega168 and ATmega328 provide UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An FTDI FT232RL on the board channels this serial communication over USB and the FTDI drivers (included with the Arduino software) provide a virtual com port to software on the computer. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the FTDI chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Nano's digital pins.

The ATmega168 and ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. To use the SPI communication, please see the ATmega168 or ATmega328 datasheet.

Programming

The Arduino Nano can be programmed with the Arduino software (download). Select "Arduino Diecimila, Duemilanove, or Nano w/ ATmega168" or "Arduino Duemilanove or Nano w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega168 or ATmega328 on the Arduino Nano comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Arduino Nano is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the FT232RL is connected to the reset line of the ATmega168 or ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Nano is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Nano. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board

receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this dat

How to use Arduino

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross -platoform program. You'll have to follow different instructions for your personal OS. Check on the Arduino site for the latest instructions. *http://arduino.cc/en/Guide/HomePage*

Linux Install

Windows Install

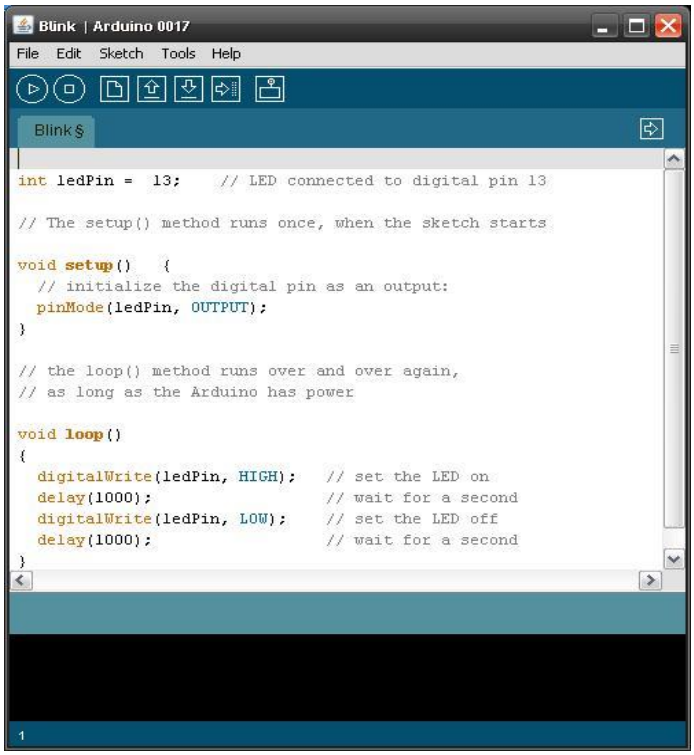
Mac Install

Once you have downloaded/unzipped the arduino IDE, you'll need to install the FTDI Drivers to let your PC talk to the board. First **Plug the Arduino to your PC via USB cable.**

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink



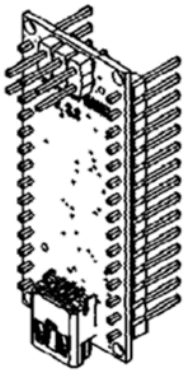
Once you have your skecth you'll see something very close to the screenshot on the right.

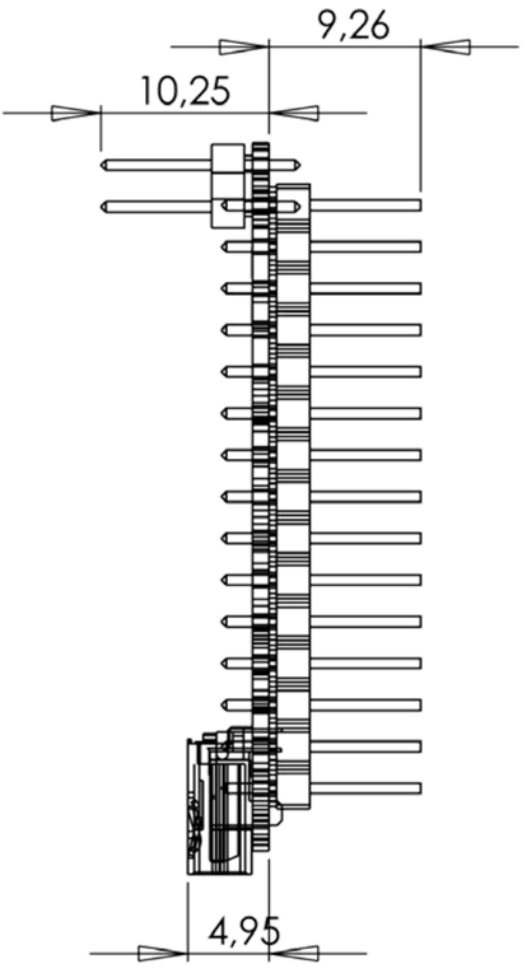
In **Tools>Board** select Arduino NANO and with the AtMEGA you're using (probably 328)

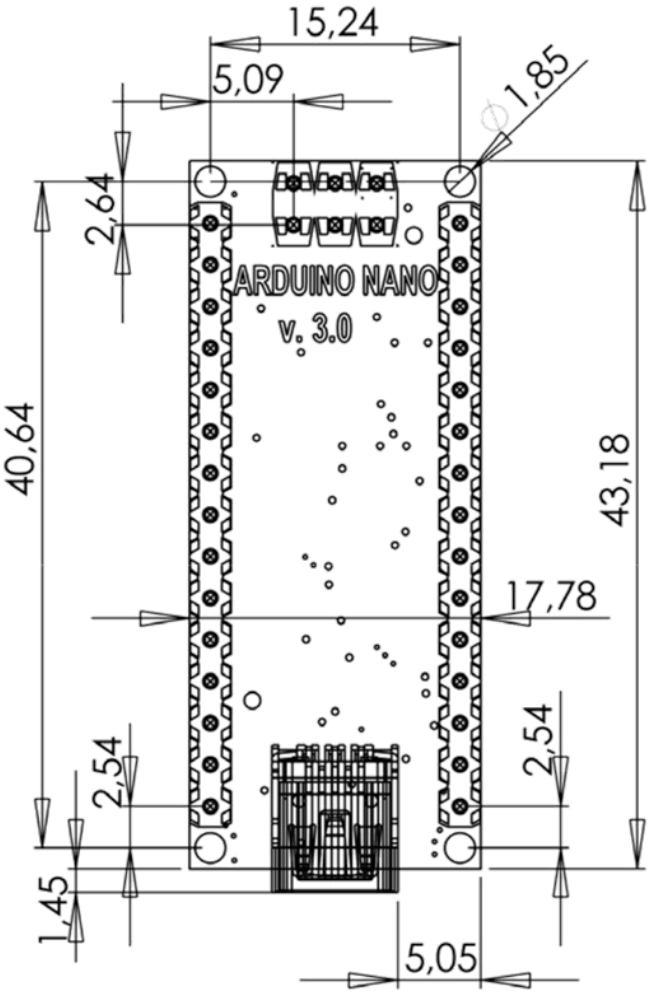
Now you have to go
to **Tools>SerialPort**

and select the right serial port,
the one arduino is attached to.

Dimensioned Drawing







Terms & Conditions

1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and

systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer

acknowledges and agrees that any such use of Arduino products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related

requirements concerning its products and any use of Arduino products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

Quick Reference **NEMA size 17 1.8° 2-phase stepper motor**

Notes and Warnings

Installation, configuration and maintenance must be carried out by qualified tech-nicians only. You must have detailed information to be able to carry out this work.

- Unexpected dangers may be encountered when working with this product!
- Incorrect use may destroy this product and connected components!

For more information, go to www.imshome.com

Specifications

1.5 Amp motors		**Single length**	**Double length**	**Triple length**
Part number		**M-1713-1.5 •** *(1)*	**M-1715-1.5 •** *(1)*	**M-1719-1.5 •** *(1)*
Holding torque	oz-in	32	60	75
	N-cm	23	42	53
Detent torque	oz-in	1.7	2.1	3.5
	N-cm	1.2	1.5	2.5
Rotor inertia	oz-in-sec^2	0.000538	0.0008037	0.0011562
	kg-cm^2	0.038	0.057	0.082
Weight	oz	7.4	8.1	12.7
	grams	210	230	360
Phase current	amps	1.5	1.5	1.5
Phase resistance	ohms	1.3	2.1	2.0
Phase inductance	mH	2.1	5.0	3.85

(1) Indicate S for single-shaft or D for double-shaft. Example M-1713-1.5S

Wiring and Connections

Signals and wire colors	
Phase A	Red
Phase /A	Blue
Phase B	Green
Phase /B	Black

Mechanical Specifications

Dimensions in inches (mm)

0.94 ± 0.02

(23.88 ± 0.51)

Optional L_{MAX}

rear

0.590 (14.86)

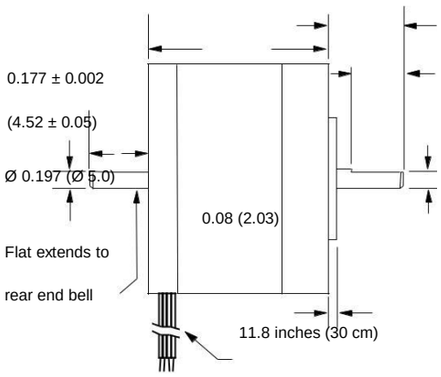
shaft

0.55

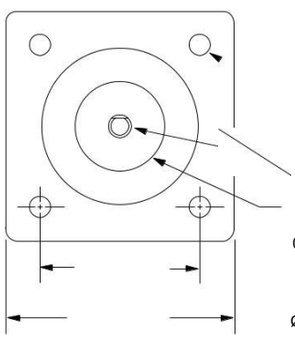
(14)

0.177 ± 0.002

(4.52 ± 0.05)



FRONT VIEW



4X Ø M3xP0.5

0.177 (4.5) deep min

Ø 0.197 +0/-0.001

(Ø 4.99 +0/-0.012)

Ø 0.866 +0/-0.002

(Ø 22.0 +0/-0.052)

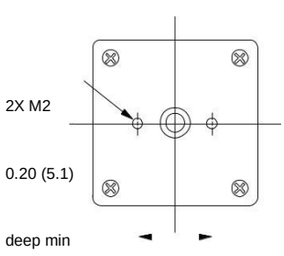
⋯1.22

(⋯30.99)

⋯1.67

(⋯42.3)

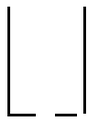
REAR VIEW (Reduced)



2X M2

0.20 (5.1)

deep min



Motor stack length inches (mm)	Single	Double	Triple
LMAX	1.34 (34.0)	1.57 (40)	1.89 (48)

Part Numbers

Example:	**M - 1 7 1 3 - 1 . 5 S**
Stepper motor frame size	**M - 1 7** 1 3 - 1 . 5 S
M - 17 = NEMA 17 (1.7" / 42 mm)	
Motor length	M - 1 7 **1 3** 1 . 5 S
13 - = single stack	
15 - = double stack	
19 - = triple stack	
Phase current	M - 1 7 1 3 - **1 . 5** S
1.5 = 1.5 Amps	
Shaft	M - 1 7 1 3 - 1 . 5 **S**
S = single, front shaft only	
D = double, front and rear shafts	
Optional optical encoder *(1)*	M - 1 7 1 3 - 1 . 5 **E S 1 0 0**
ES = Single-end	
ED = Differential	
Line count	
100, 200, 250, 400, 500 or 1000 *(2)*	

(1) An encoder replaces the shaft designator in the part number.

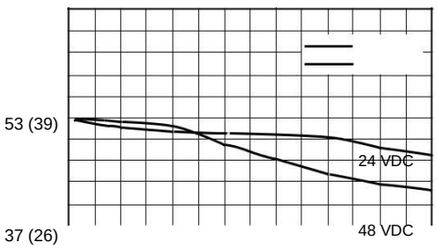
(2) All encoders have an index mark, except the 1000 line count version.

Torque-speed performance

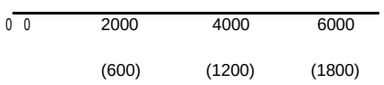
Measured at 1.5 Amps RMS

M-1713-1.5

Torque in oz-in (N-cm)



18(13)

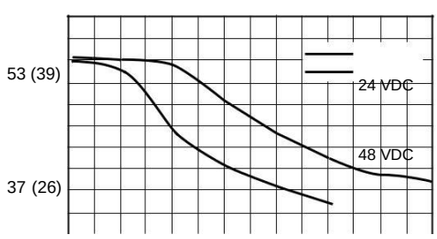


(600) (1200) (1800)

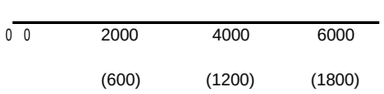
Speed in full steps per second (rpm)

M-1715-1.5

Torque in oz-in (N-cm)



18(13)

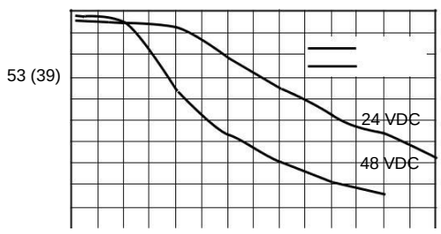


(600) (1200) (1800)

Speed in full steps per second (rpm)

M-1719-1.5

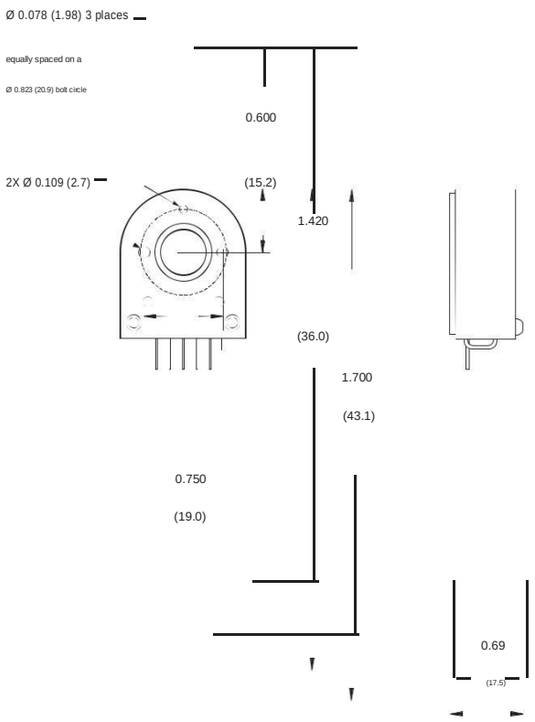
Torque in oz-in (N-cm)



2. (26)

Optical Encoder Option

Dimensions in inches (mm)

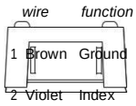


Connectivity single-end encoder differential encoder

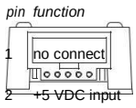
2 4 6 8 10

1 3 5 7 9

1 2 3 4 5



wire	function
1 Brown	Ground
2 Violet	Index



pin	function	pin	function
1	no connect	6	Channel A+
2	+5 VDC input	7	Channel B –

3	Blue	Channel A	3	Ground	
4	Orange	+5 VDC input	4	no connect	9 Index –
5	Yellow	Channel B	5	Channel A –	10 Index +

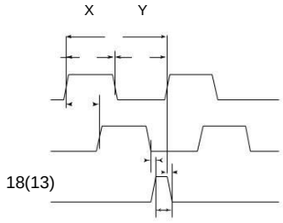
optional interface cable available: ES-CABLE-2

interface cable included

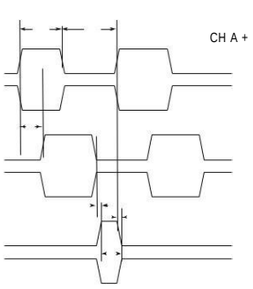
Timing single-end encoder differential encoder

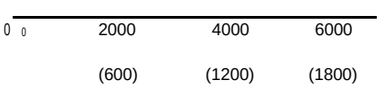
C X Y

X Y



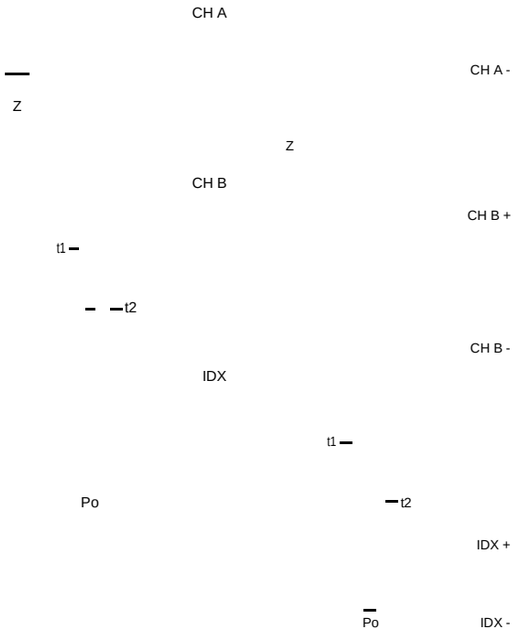
18(13)





(600) (1200) (1800)

Speed in full steps per second (rpm)



Parameter	Symbol	Min	Typ	Max	Units
Cycle error			3	5.5	ºe
Symmetry		130	180	230	ºe
Quadrature		40	90	140	ºe
Index pulse width	Po	60	90	120	ºe
Index rise (after Ch A or B rise)	t1	-300	100	250	ns
Index fall (after Ch A or B fall)	t2	70	150	1000	ns

C One cycle: 360 electrical degrees (ºe).

X/Y Symmetry: the measure of the relationship between X and Y, nominally 180ºe.

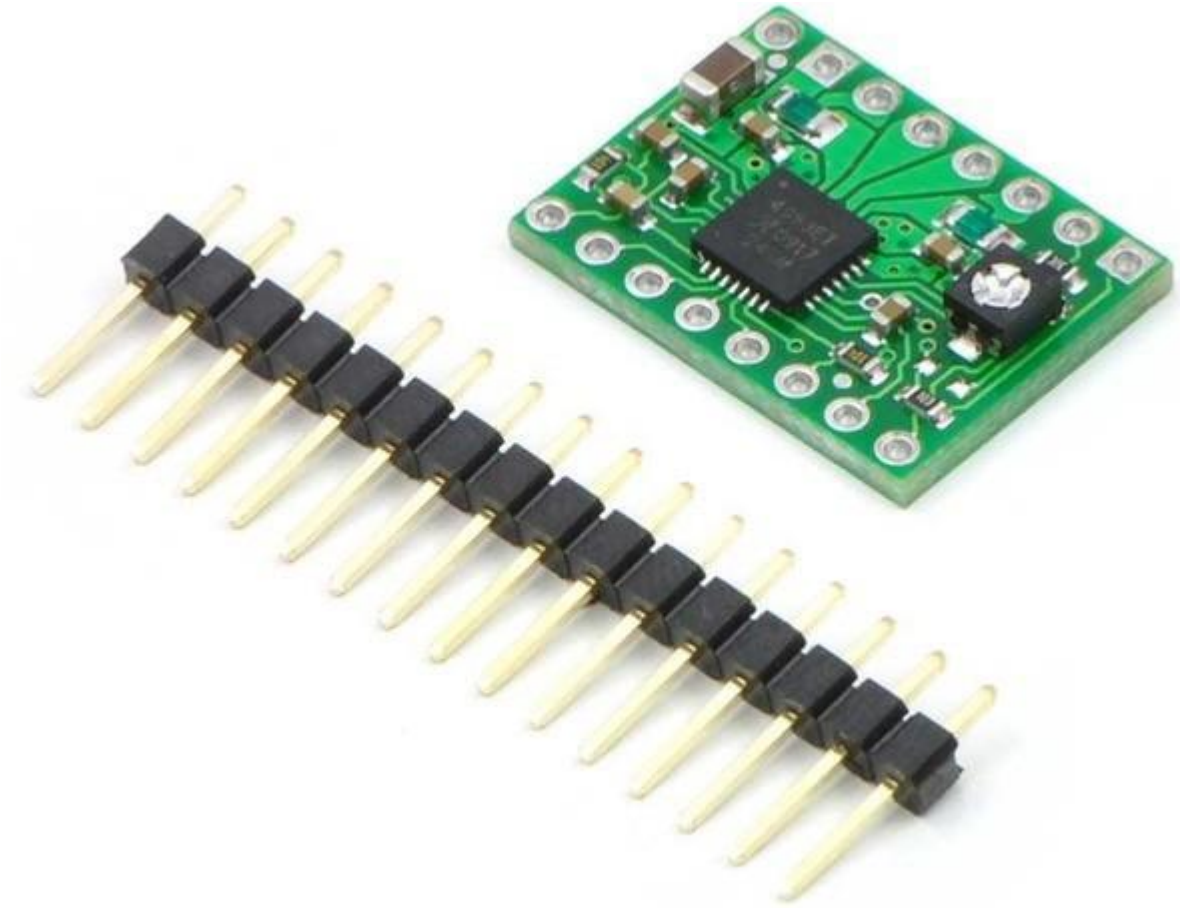
Z Quadrature: the phase lead or lag between channels A and B, nominally 90ºe.

Po Index pulse width, nominally 90 ºe.

NOTE: Rotation is as viewed from the cover side of the encoder.

RB-Pol-176

Pololu 8-35V 2A Single Bipolar Stepper Motor Driver A4988



Stepper Motor Driver Carrier

The A4988 stepper motor driver carrier is a breakout board for Allegro's easy-to-use A4988 microstepping bipolar stepper motor driver and is a drop-in replacement for the A4983 stepper motor driver carrier. The driver features adjustable current limiting, overcurrent protection, and five different microstep resolutions. It operates from 8 – 35 V and can deliver up to 2 A per coil.

Note: This board is a drop-in replacement for the original A4983 stepper motor driver carrier. The newer A4988 offers overcurrent protection and has an internal 100k pull-down on the MS1 microstep selection pin, but it is otherwise virtually identical to the A4983.

Description

Overview

This product is a carrier board or breakout board for Allegro's A4988 DMOS Microstepping Driver with Translator and Overcurrent Protection; we therefore recommend careful reading of the A4988 datasheet (380k pdf) before using this product. This stepper motor driver lets you control one bipolar stepper motor at up to 2 A output current per coil (see the Power Dissipation Considerations section below for more information). Here are some of the driver's key features:

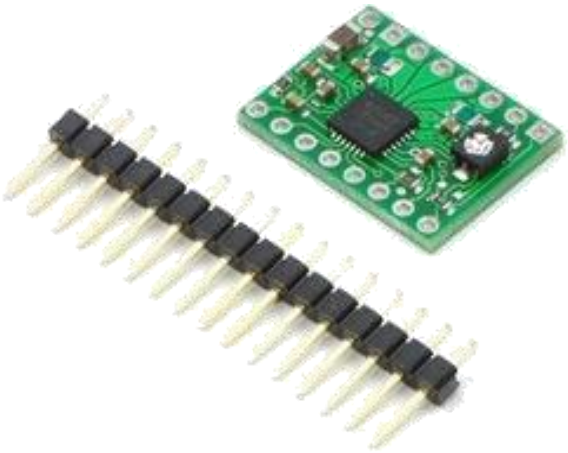
- Simple step and direction control interface
- Five different step resolutions: full-step, half-step, quarter-step, eighth-step, and sixteenth-step
- Adjustable current control lets you set the maximum current output with a potentiometer, which lets you use voltages above your stepper motor's rated voltage to achieve higher step rates
- Intelligent chopping control that automatically selects the correct current decay mode (fast decay or slow decay)
- Over-temperature thermal shutdown, under-voltage lockout, and crossover-current protection
- Short-to-ground and shorted-load protection (this feature is not available on the A4983)

Like nearly all our other carrier boards, this product ships with all surface-mount components —including the A4988 driver IC—installed as shown in the product picture.

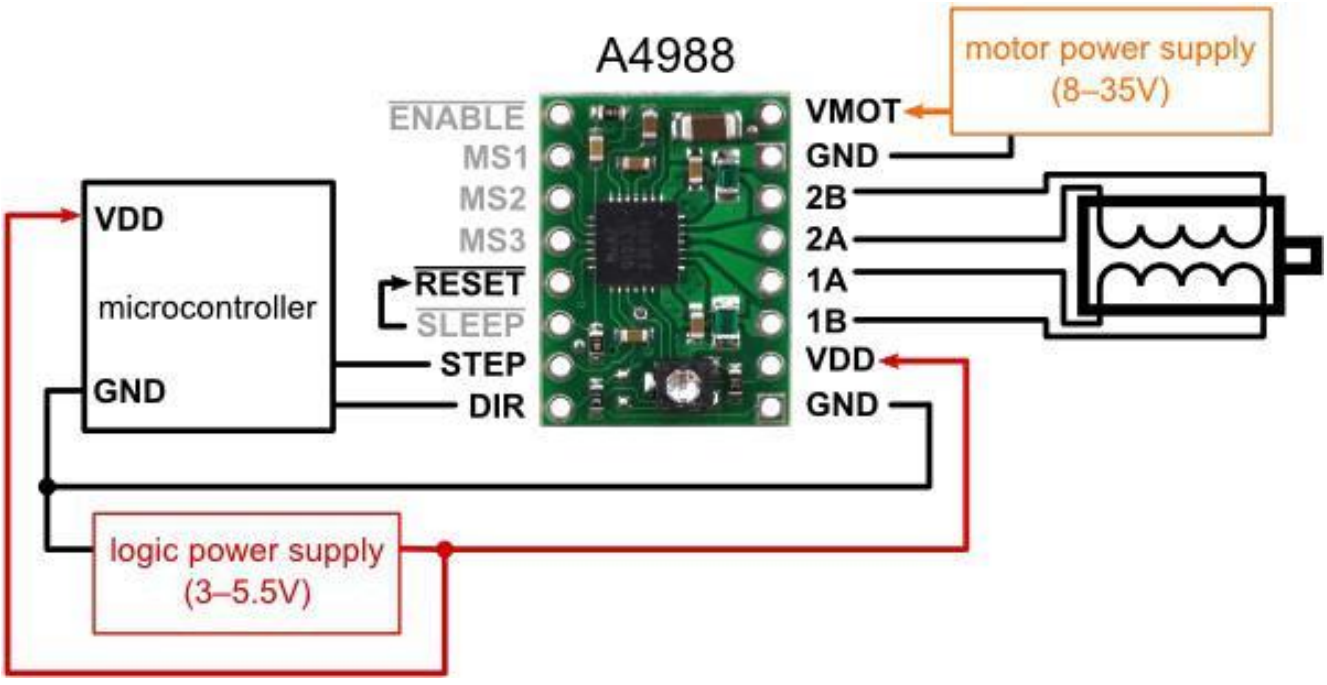
We also sell a larger version of the A4988 carrier that has reverse power protection on the main power input and built-in 5 V and 3.3 V voltage regulators that eliminate the need for separate logic and motor supplies.

Included hardware

The A4988 stepper motor driver carrier comes with one 1×16-pin breakaway 0.1" male header. The headers can be soldered in for use with solderless breadboards or 0.1" female connectors. You can also solder your motor leads and other connections directly to the board.



Using the driver



Minimal wiring diagram for connecting a microcontroller to an A4988 stepper motor driver carrier (full-step mode).

Power connections

The driver requires a logic supply voltage (3 – 5.5 V) to be connected across the VDD and GND pins and a motor supply voltage of (8 – 35 V) to be connected across VMOT and GND. These supplies should have appropriate decoupling capacitors close to the board, and they should be capable of delivering the expected currents (peaks up to 4 A for the motor supply).

Motor connections

Four, six, and eight-wire stepper motors can be driven by the A4988 if they are properly connected; a FAQ answer explains the proper wirings in detail.

Warning: Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver. (More generally, rewiring anything while it is powered is asking for trouble.)

Warning: Connecting or disconnecting a stepper motor while the driver is powered can destroy the driver. (More generally, rewiring anything while it is powered is asking for trouble.)

Step (and microstep) size

Stepper motors typically have a step size specification (e.g. 1.8° or 200 steps per revolution), which applies to full steps. A microstepping driver such as the A4988 allows higher resolutions by allowing intermediate step locations, which are achieved by energizing the coils with intermediate current levels. For instance, driving a motor in quarter-step mode will give the 200-step-per-revolution motor 800 microsteps per revolution by using four different current levels.

The resolution (step size) selector inputs (MS1, MS2, MS3) enable selection from the five step resolutions according to the table below. MS1 and MS3 have internal 100kΩ pull-down resistors and MS2 has an internal 50kΩ pull-down resistor, so leaving these three microstep selection pins disconnected results in full-step mode. For the microstep modes to function correctly, the current limit must be set low enough (see below) so that current limiting gets engaged. Otherwise, the intermediate current levels will not be correctly maintained, and the motor will effectively operate in a full-step mode.

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

Control inputs

Each pulse to the STEP input corresponds to one microstep of the stepper motor in the direction selected by the DIR pin. Note that the STEP and DIR pins are not pulled to any particular voltage internally, so you should not leave either of these pins floating in your application. If you just want rotation in a single direction, you can tie DIR directly to VCC or GND. The chip has three different inputs for controlling its many power states: RST, SLP, and EN. For details about these power states, see the datasheet. Please note that the RST pin is floating; if you are not using the pin, you can connect it to the adjacent SLP pin on the PCB.

Current limiting

To achieve high step rates, the motor supply is typically much higher than would be permissible without active current limiting. For instance, a typical stepper motor might have a maximum current rating of 1 A with a 5Ω coil resistance, which would indicate a maximum motor supply of 5 V. Using such a motor with 12 V would allow higher step rates, but the current must actively be limited to under 1 A to prevent damage to the motor.

The A4988 supports such active current limiting, and the trimmer potentiometer on the board can be used to set the current limit. One way to set the current limit is to put the driver into full-step mode and to measure the current running through a single motor coil without clocking the STEP input. The measured current will be 0.7 times the current limit (since both coils are always on and limited to 70% in full-step mode). Please note that the current limit is dependent on the Vdd voltage.

Another way to set the current limit is to measure the voltage on the “ref” pin and to calculate the resulting current limit (the current sense resistors are 0.05Ω). The ref pin voltage is accessible on a via that is circled on the bottom silkscreen of the circuit board. See the A4988 datasheet for more information.

Power dissipation considerations

The A4988 driver IC has a maximum current rating of 2 A per coil, but the actual current you can deliver depends on how well you can keep the IC cool. The carrier's printed circuit board is designed to draw heat out of the IC, but to supply more than approximately 1 A per coil, a heat sink or other cooling method is required.

This product can get hot enough to burn you long before the chip overheats. Take care when handling this product and other components connected to it.

Please note that measuring the current draw at the power supply does not necessarily provide an accurate measure of the coil current. Since the input voltage to the driver can be significantly higher than the coil voltage, the measured current on the power supply can be quite a bit lower than the coil current (the driver and coil basically act like a switching step-down power supply). Also, if the supply voltage is very high compared to what the motor needs to achieve the set current, the duty cycle will be very low, which also leads to significant differences between average and RMS currents.

Schematic diagram

Schematic diagram of the md09b A4988 stepper motor driver carrier.