

ATMEGA328P Breadboard Kit

CHIP: ATMEGA328P-PU

Parts: 19

KIT DESCRIPTION:

This ATMEGA328 Breadboard kit allows you to build out your very own MCU on a breadboard/protoboard. The ATMEGA328 is the chip used on the famous Arduino Uno, and it features 32KB of Flash and 2KB SRAM. The chip runs on 1.8-5.5v and can reach a top frequency of 20MHz (at 4.5v+). Native programming is available via SPI, however the preinstalled bootloader allows uploading sketches over UART. This kit is ideal for low power applications, early prototyping, space-bound system, and most importantly: education.

Quick note: This ATMEGA is pre-burned with a bootloader and is set to run off of the 8mhz internal oscillator, if you wish to use a higher frequency external oscillator, uploading the bootloader is very simple.

Happy building!

ATMEGA328P PINOUT:

PIN	NAME	DOES THIS	PIN	NAME	DOES THIS
1	RESET (PC6)	Chip Reset (Pull low to reset)	15	PB1	General IO, PWM
2	PD0	General IO, RX	16	PB2	General IO, SPI-SS
3	PD1	General IO, TX	17	PB3	General IO, SPI-MOSI
4	PD2	General IO, INT0	18	PB4	General IO, SPI-MISO
5	PD3	General IO, INT1	19	PB5	General IO, SPI-SCK
6	PD4	General IO, Timer In	20	AVCC	Tied to VCC (typically with Ferrite Bead)
7	VCC	+3.3v (Can be +5v)	21	AREF	ADC Reference
8	GND	Ground	22	GND	Ground
9	PB6	General IO, External OSC Input	23	PC0	General IO, ADC Input
10	PB7	General IO, External OSC Output	24	PC1	General IO, ADC Input

PIN	NAME	DOES THIS	PIN	NAME	DOES THIS
11	PD5	General IO, PWM	25	PC2	General IO, ADC Input
12	PD6	General IO, PWM	26	PC3	General IO, ADC Input
13	PD7	General IO	27	PC4	General IO, SDA
14	PB0	General IO, Timer In	28	PC5	General IO, SCL

This pinout is simplified, for full pinout, see datasheets below

CHIP SPECIFICATIONS:

Core: 8 Bit AVR RISC, 28-pin DIP Package, up to 20MHz with external oscillator

Memory: 32KB Flash, 2KB SRAM, 1KB EEPROM

Power: 1.8-5.5v, 0.2mA active draw @ 1 MHz 1.8v, 0.1uA at 1.8v power-down mode

IO Peripherals: 23 GPIOs, 6 PWMs, 6 ADCs, 1 UART, 1 SPI, 1 I2C/TWI

Other: 8MHz internal RC Oscillator, 2x8 bit and 1x16 bit timers, 2 external interrupts

4MHz @ 1.8v

10MHz @ 2.7v

20MHz @ 4.5v+

PARTS IN KIT:

COMPONENT	# Included	JOB
ATMEGA328P	1	Microcontroller
3.3v 0.8A LDO	1	Voltage Regulator (3.3v)
NE555P (Timer)	1	Programmable Timer & Clock
25AA320A (EEPROM)	1	Non Volatile SPI Memory
100nF Ceramic Capacitor	8	Local Decoupling
1uF Ceramic Capacitor	2	Local Decoupling
47uF Electrolytic Capacitor	1	Bulk Decoupling
100uF Electrolytic Capacitor	2	Bulk Decoupling
0.5A Polyfuse	1	VDD Overcurrent Protection
1A Schottky Diode	1	Low Forward Voltage Diode
Ferrite Bead	1	Analog VDD Filtering

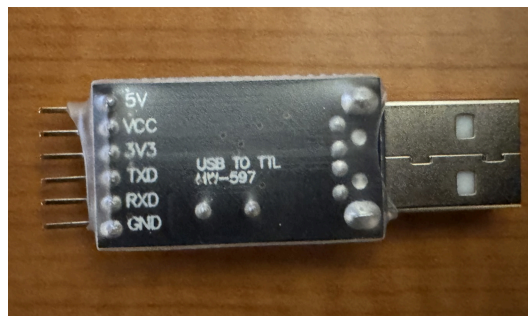
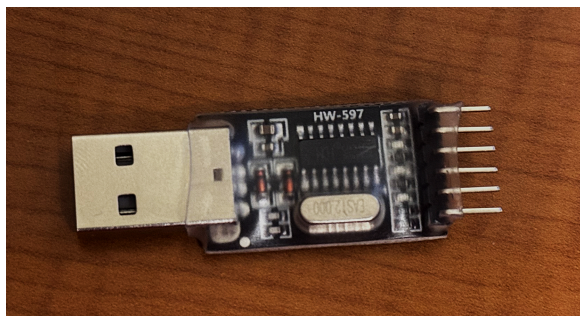
COMPONENT	# Included	JOB
33 Ohm Resistor	5	Series resistors for high speeds
470 Ohm Resistor	3	LED resistors
1k Ohm Resistor	3	Pull up/down
4.7k Ohm Resistor	5	Pull up/down
10k Ohm Resistor	5	Pull up/down
Green LED	3	Power good, general use, IO's
Tactile Switch	2	Reset/General Purpose
USB to UART Programmer	1	USB-UART programming

PROGRAMMING:

There are 2 common ways to program the ATMEGA328P

Quick note: the ATMEGA328P comes preburned with a OptiBoot bootloader, set to use the 8MHz internal oscillator by default. We recommend using a modest baud rate, like 38400-57600.

1. One method is using the USB to UART programmer. The module has 5v, 3.3v, VCC, GND, RX and TX. First, select your logic level voltage (same as the voltage your MCU is running off of) by connecting VCC to the desired power pin. Next, ensure a common ground and connect serial lines (USB RX to MCU TX, and vice versa)
Click upload on Arduino IDE and right when uploading displays on the IDE, pull the Reset pin to GND temporarily (eg using the push button and 10k pull up) (it will take a few tries, the window is short).
2. Another method is using another Arduino UNO (Make sure the ATMEGA328P is powered at 5v to avoid conflicting logic levels). Open ArduinoISP example and upload to UNO. Wire SPI pins (the Arduino UNO is master) and MCU Reset pin to UNO pin 10. Be sure to power and ground both boards. Switch the board to ATMEGA328P (MiniCore @ 8MHz internal), set programmer arduino as ISP and upload using programmer.
(warning, this erases the ATMEGA328Ps bootloader, burning a new one is simple)



EXAMPLE LAYOUT:**1. Set up**

Place the MCU on the center of the breadboard, giving each pin its own row. Connect VDD to your power rail, and AVCC through a ferrite bead to your power rail. Be sure to also connect all ground pins to your ground rail. On both VDD and AVCC pins, place 100nF capacitors from power to the ground rail. On the VDD rail, place a 47uF electrolytic capacitor, being mindful of polarity, along with a 1uF ceramic capacitor. You can choose which voltage to run your ATMEGA328P off of, depending on specs you want. The USB to UART programmer gives both 3.3v and 5v options, the kit also comes with a 3.3v LDO. Optionally, you can choose to use a polyfuse in series, and configure the Schottky diode for reverse polarity & overcurrent protections.

2. Programming

Use either another Arduino UNO or the USB-UART programmer, ensure common grounds & power both. Remember, 8MHz internal oscillator is default used, and the ATMEGA has a OptiBoot bootloader by default. See above for more detailed programming instructions.

3. Application

To use the code featured in the example code section of this datasheet, connect 1 side of a tactile button to GND and the other side to PD2

EXAMPLE CODE (MCU + Button):

```
/*
 * ATMEGA328 Dev Kit -- Example Sketch: Button Press Counter
 * -----
 * Press the tactile button. Each press prints the press number
 * and the elapsed time (in ms) since the board booted.
 *
 * WIRING:
 * Tactile Button -> D2, other side to GND (uses internal pull-up)
 */

const uint8_t BUTTON_PIN = 2;

int pressCount = 0;
bool lastState = HIGH;

void setup() {
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  bool currentState = digitalRead(BUTTON_PIN);

  if (lastState == HIGH && currentState == LOW) {
```

```

    pressCount++;
    Serial.print("Press #");
    Serial.print(pressCount);
    Serial.print(" - Elapsed time: ");
    Serial.print(millis());
    Serial.println(" ms");
    delay(50); // basic debounce
  }

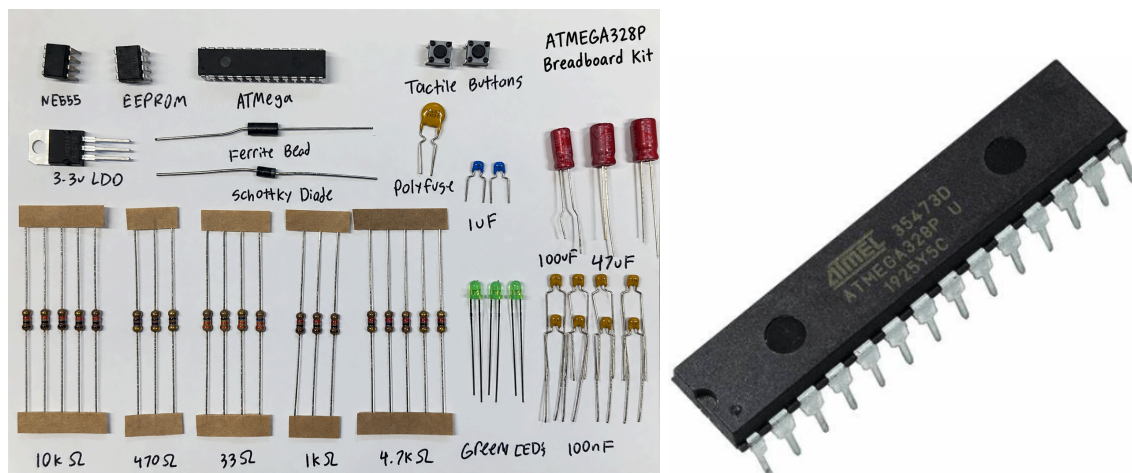
  lastState = currentState;
}

```

APPLICATIONS:

- Embedded systems prototyping & education
- IoT sensor nodes
- Robotics & motor control
- Lower-power applications
- Custom Arduino-alternative projects
- Space sensitive system

PHOTOS:



DATASHEETS & PINOUTS:

[\[ATMEGA328P Datasheet\]](#)

[\[NE555 Datasheet\]](#)

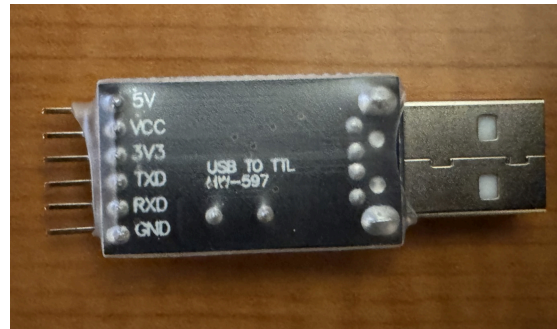
[\[25AA320A Datasheet\]](#)

[\[RXEF050 Datasheet\]](#)

[\[1N5819 Datasheet\]](#)

[\[LD1117V33C Datasheet\]](#)

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)



On the left: Official ATMEGA328P Pinout (More details available on datasheet)

On the right: USB to UART programmer, with access to 5v, GND and your Serial Pins

NOTES & DEBUGGING:

- The ATMEGA328P comes preburned with bootloader
- By default, the ATMEGA is running off its internal 8MHz RC Oscillator
- If uploading via Arduino UNO, an ATMEGA program can only be flashed when the UNO has the ArduinoISP.
- BAUD Rate set at 9600 in example code
- Be sure to connect VDD & AVCC for all ATMEGA use (even if ADC is not being used)
- Install correct MiniCore library found [\[Here\]](#) (If applicable)

Feel free to send any questions to sales@meridianelectronic.com

Check MeridianElectronic.com for datasheets, 3D printing, solder stencils & more!