ATMEGA328P Breadboard Kit

CHIP: *ATMEGA328P-PU*	**Parts**: *18 (+1 optional programmer)*

Kit Description:

This ATMEGA328 Developer Kit & Datasheet has everything you need to build your very own MCU on a breadboard/protoboard. This kit contains one ATMEGA328P, capacitors, resistors, led's, LDO voltage regulator, ESD protections, memory, a programmable timer, switches & optionally an rp2040 for seamless UART programming. This kit's purpose is not only for education however, it is truly a fantastic applied microcontroller. Featuring an 8 bit 8mhz (up to 20 depending on bootloader & external crystal) AVR processor. This ATMEGA is tested, reliable and a foundation for any hobbyist/engineer. If you choose to include the RP2040 as the programmer, you get built-in USB-C > UART programming and a serial monitor for all your debugging needs. You may also opt for the classic USB to serial adapter, featuring chips such as the CH340.

This ATMEGA is pre-burned with a bootloader and typically operates on a 8mhz internal oscillator. If you wish, reburning a bootloader is easy and unlocks more possibilities.

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

PIN	NAME	DOES THIS	PIN	NAME	DOES THIS
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
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

Programming:

There are 3 common ways to program the ATMEGA

1. The typical method of programming the ATMEGA328P uses a USB to serial/uart adapter, such as the CH340 found on Amazon. Needing only 4 connections, RX, TX, GND and VDD, it is a dead simple way to program the ATMEGA and access the serial monitor.
2. A secondary popular method to program the ATMEGA is using another arduino MCU, such as the arduino uno. Attached [Here] is code for ArduinoISP, which after uploaded to the ATMEGA, allows the user to upload minicore boards via SPI (uploaded via programmer). Be sure to program minicore with correct settings (8mhz internal by default). For reading serial monitor, simply upload [This] code to the arduino uno after programming it, and connecting selected UART lines.
3. A third method of programming is using the RP2040 optionally included in the kit. Working on a basic pass through UART program found [Here], you may again use minicore, and the bootloader settings (internal 8mhz oscillator), and upload regularly. You must power the ATMEGA via 3.3v if using RP2040. Be sure to pull reset low right as programming starts. You may use a pushbutton to GND or a GPIO, be sure to also include 10k pull up to VDD on reset. Right as programming starts in arduinoIDE, send a short quick pulse to reset the ATMEGA.

Parts in kit:

COMPONENT	# Included	JOB
ATMEGA328P	1	Microcontroller

COMPONENT	# Included	JOB
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	VDD ESD Protection
Ferrite Bead	1	Analog VDD Filtering
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
RP2040 Programmer (Optional)	1	UART Programming with USB-C

Example Layout:

1. Power Management

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. If using a 6v power supply (batteries), be sure to use the LDO to regulate it down to 3.3v. If using the RP2040 programmer, you should use 3.3v as your power supply. Optionally, you can choose to use a polyfuse in series, and the diode to ground as ESD protection.

2. Programming

Connect PC6 to a tactile switch, which is connected to ground on the other side, also provide PC6 a 10k pull up to VDD. Connect programmer UART pins to ATMega UART pins, and ensure common grounds.

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

That's it! The basic example layout is surprisingly simple. A photo is attached below, in the photos section as a reference.

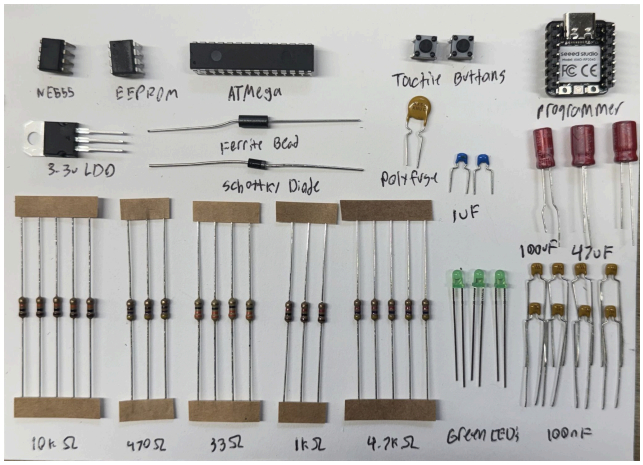
Example Code (MCU + Button):

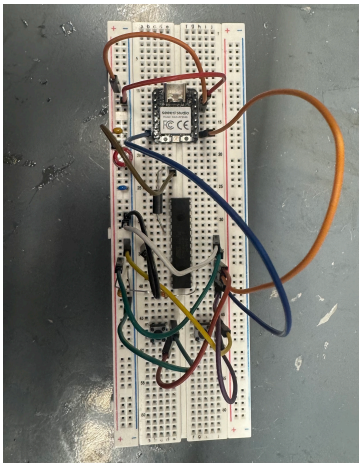
```
/*
 * ATMEGA328 Dev Kit -- Example Sketch: Button Press Counter
 * ---------------------------------------------------------
 * Press the tactile button. Each press prints the press number
 * and the elapsed time (in ms) since the board booted.
 *
 * WIRING:
 *   Tactile Button -> D2, other side 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
- Data logging
- Custom Arduino-alternative projects
- Home automation

PHOTOS:





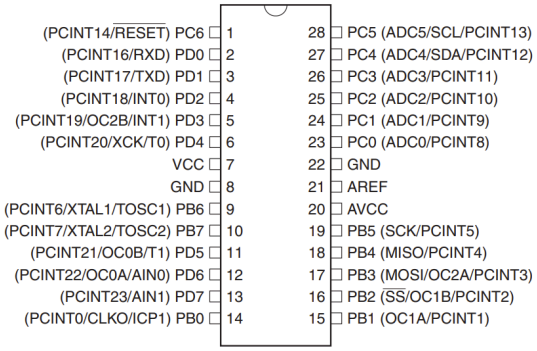
Datasheets & Pinout Images:

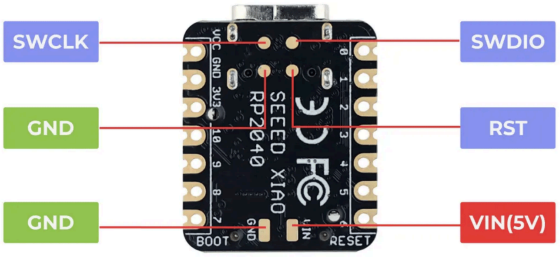
[ATMEGA328P Datasheet]

[NE555 Datasheet]

[25AA320A Datasheet]

[RP2040 Datasheet]





On the Left: ATMEGA328P

On the Right: Xiao RP2040

Notes and Debugging:

- By default the ATMEGA operates on 8mhz internal & 3.3v. This can be changed depending on the programming method.
- ATMEGA comes preburned with bootloader
- If uploading via Arduino UNO, an ATMEGA program can only be flashed when the UNO has the arduinoISP loaded.
- BAUD Rate typically set at 9600
- Be sure to connect VDD & AVCC for all ATMEGA use
- Install correct MiniCore library found [Here]
- Be persistent, working with bare microcontroller chips is NOT easy, but it WILL work. The ATMEGA is incredibly resilient and you are too. Direct any questions to Meridian Electronics, were happy to help