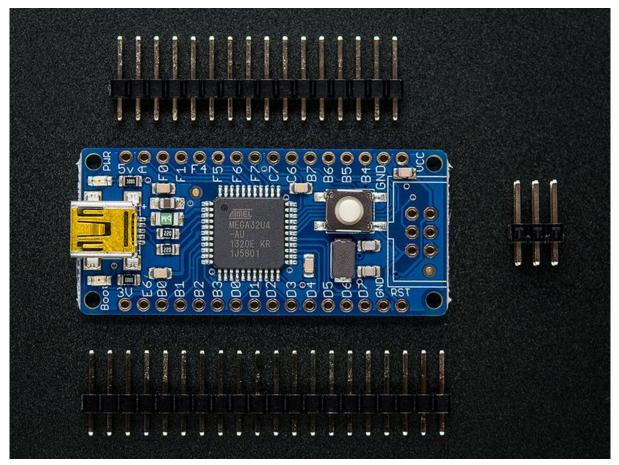


Atmega32u4 Breakout

Created by lady ada



https://learn.adafruit.com/atmega32u4-breakout

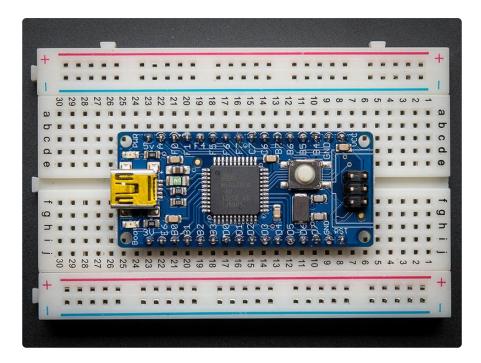
Last updated on 2022-12-01 02:01:38 PM EST

Table of Contents

Intro	3
About the Atmega32u4 Breakout board+	
• Why not use a Teensy	
Assembly	5
Design	6
Design Specifications	
Microcontroller	
• Power	
• Pinout	
USB Development	
Using with AVRDude	8
AVR109 Bootloader & AVRdude	
Arduino IDE Setup	10
Using with Arduino	12
Using it with Teensyduino	
Download	15
Download	
Schematic	

Fabrication Print

Intro



About the Atmega32u4 Breakout board+

We like the AVR 8-bit family and were excited to see Atmel upgrade the series with a USB core. Having USB built in allows the chip to act like any USB device. For example, we can program the chip to 'pretend' it's a USB joystick, or a keyboard, or a flash drive! Another nice bonus of having USB built in is that instead of having an FTDI chip or cable (like an Arduino), we can emulate the serial port directly in the chip. This costs some Flash space and RAM space but that's the trade-off.

The only bad news about this chip is that it is surface mount only (SMT), which means that it is not easy to solder the way the larger DIP chips are. For that reason, we made a breakout board. The board comes with some extras like a fuse, a 16mhz crystal, USB connector and a button to start the bootloader.

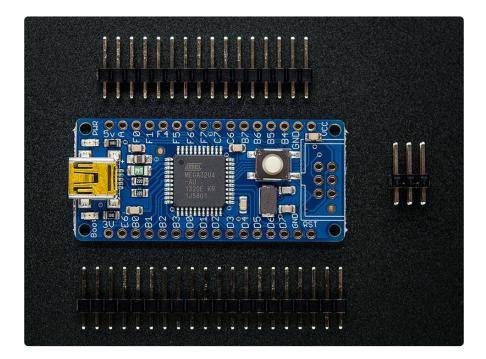
Why not use a Teensy

We also carry a similar board from PJRC called the <u>Teensy</u> (). The Teensy uses the same chip so you may be wondering, why did we design a different-but-still-basically-the-same board? We've used the Teensy in a few projects and like it a lot but there are a few details that we wanted to change. We wanted...

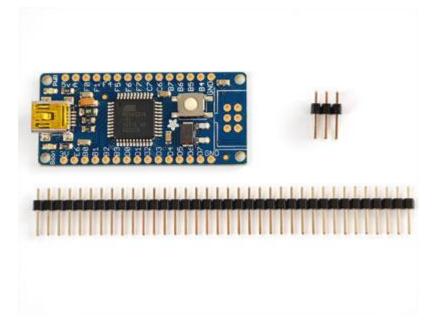
- ...a bootloader that would work with avrdude since that is our preferred software
- ...another LED, for power-good indication
- ...a 500mA fuse on the USB power-source pin
- ...mounting holes so we could attach it easily
- ...reprogramming ISP header so we could program the board directly without the bootloader
- ...a larger reset button
- ...all the pins broken out for use with a breadboard
- ... open source bootloader that works with AVRdude

That doesn't in any way mean that it is better or replaces the Teensy. Here are some reasons we will still use the teensy for many projects

- It's really really small. A third the size of this breakout
- The bootloader that is programmed in uses only 512 bytes (instead of 2K)
- It works nicely with Teensyduino and auto-resets right before programming



Assembly

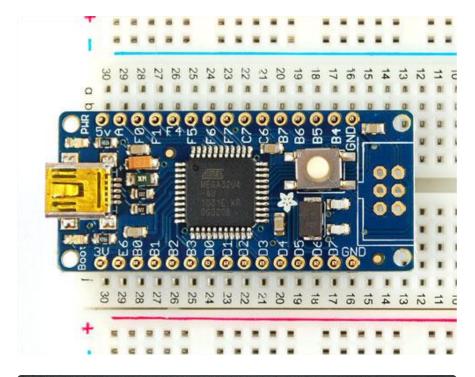


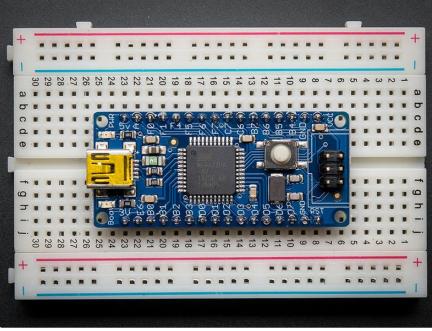
The breakout is almost completely assembled. We don't solder on the header or 6-pin ISP connector since we find it's easy to do but hard to undo, in case you want to put different headers on, or just connect up wires directly.

The board ships with a strip of header, you can simply break the header into two pieces and place them in a breadboard.

+	Number of					-	10102								1															
		-		4	10	0	2	00	00	10	11	12	13	14	15	16	17	18	36	8	24	2	2	54	25	æ	27	38	53	30
															A														٨	0
		14									14			10	1	0	2	٠	2	2		٠	2	2	2	2	3	2		
£		-	1			1			1	8		10		.0	2	4	4		*			4	x			a	4			1
0	*								1			4		4		۵	3		4		a	4				4		2	-	
+		14			4				4	4	4	4	4	4	4	4	4	4	4	4	4	4	4		a	*	*			*
																													242	
0																*														
σ																	1	1	4	1	1									
0		4				J.			14							3	Ξ	2	2	2		2	2	2	2	2	2			
۵	4	14	18	u.	14	1		18			18	×.									*									
D	-	5 1		4	5	• 9		. 8		10 -	- 11	12 =	13 -	14 #	15 =	- 91	17 =	- 81	- 61	20 -	21 -	22 .	23 =	24 -	25 -	26 -	27 -	28 -	- 62	30.
1	-					i.																								
4							1		i 1																					
t		_	_	_	-	-	_	_	-	_	-	_	_	-	_	_	_	-	_	-	-	_	_	-	_	-	_	_	-	

Then place the board on top and solder it in place.





Design

If you're using windows, you will need an inf driver file, you can download it below (in the Downloads section) ()

Design Specifications

This breakout board is designed to make it easy for you to get started with the Atmega32u4, a USB-native 8 bit AVR microcontroller. We tried to keep the design

simple, while taking care of all the details so that you can focus on your project's firmware and hardware.

Microcontroller

The main chip is an ATmega32u4, an 8 bit AVR-core processor which has the bonus of a USB core built in. This makes writing USB native programs such as serial ports, mouse/keyboard, mass-storage-controller, MIDI, etc very easy. The chip has 32K of flash and 2.5K of RAM. The microcontroller is clocked at 16 MHz with an on-board crystal.

Power

The board is powered by the USB port at 5V. There is a 500mA polyfuse to protect your computer from a shorted circuit. If you'd like to run the board off of another voltage, you can do that by cutting the VCC solder jumper bridge underneath and connecting an external voltage to pin #2 of the ISP header. Note that running the board at 3.3v @16mhz is considered overclocking. We do it for prototypes and it seems to work fine but it's out of spec!

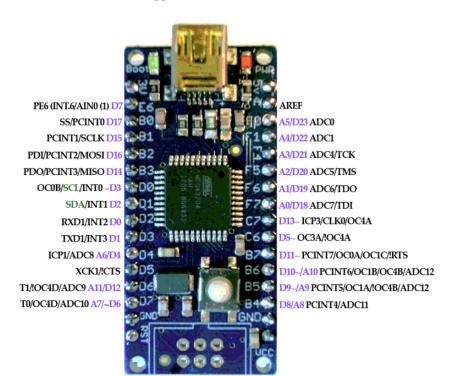
There is a 3V pin from the microcontroller's internal regulator (USB signals are at 3V) you can use this for maybe 10mA or so, it might work as a reference voltage).

Pinout

Pins are labeled on the silkscreen, but for additional information, Johngineer has created a nice pinout diagram () that is helpful for using this board.

Adafruit ATMega32u4 Breakout Board

Pins mapped to Arduino Leonardo



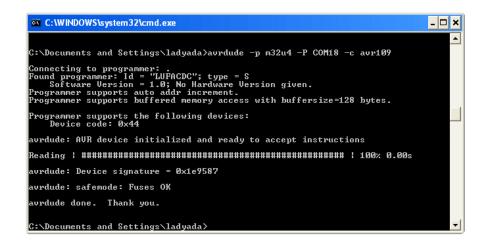
USB Development

The very nice thing about this chip is the USB core built in which makes USB-device development easy. What makes it even easier & better is <u>the full USB stack already</u> written for you by Dean Camera (). Called "LUFA" (), the package comes with tons of working examples for all sorts of USB devices and its completely open source. We use the firmware as a 'starting point' which we then expand upon. Please check it out and if you find it useful consider donating time (fixing bugs, improving documentation) or funds to Dean.

Using with AVRDude

AVR109 Bootloader & AVRdude

You can tell the bootloader is active when the red 'Boot' LED pulses/breathes The board will then show up as a Serial or COM port, and you can use avrdude to program it. The 'programmer name' is avr109 so for example, to test you should run avrdude -p m32u4 -P COM3 -c avr109 which will initialize the bootloader.



You can find the serial port COM# in the Device driver, or for mac/linux users run ls / dev/cu.usb* to list all usb serial devices

🚔 Device Manager	- 0	x
<u>File Action View H</u> elp		
🛛 🛁 ladyada301-PC		
⊵ -₁		
Disk drives		
Display adapters		
DVD/CD-ROM drives		
Human Interface Devices		
DE ATA/ATAPI controllers		
> 🔚 Imaging devices		
⊳ .∰ Jungo ⊳		
. Mice and other pointing devices		
Monitors		
Network adapters		
Portable Devices		
Ports (COM & LPT)		
Communications Port (COM1)		
Processors		
Sound, video and game controllers		
🔈 🚛 System devices		
🖒 - 🏺 Universal Serial Bus controllers		

The bootloader will time out eventually (after about 10 seconds) Because we are not using a USB/serial converter, bootloading is tremendously fast, we can program a full chip in under 2 seconds!

The bootloader takes up the last 4K of FLASH, so be aware that you will only have 28K instead of 32K. We have found that this isn't very constricting as 28K is still

plenty. If you'd like more space, you can always use the 6-pin ISP connector and an AVR programmer (which will delete the bootloader)

If you ever want to disable the bootloader you can cut the bottom 'HWB' jumper trace. This will disconnect the 'hardware bootloader' pin, you can then use the button as a plain reset button. For the first few runs of this board we set the fuses to still use the bootloader even with the HWB jumper cut, if you want to get rid of the bootloader, please set the fuses to remove the BOOTRST fuse. Sorry!

Arduino IDE Setup

The first thing you will need to do is to download the latest release of the Arduino IDE. You will need to be using version 1.8 or higher for this guide

Arduino IDE Download

After you have downloaded and installed the latest version of Arduino IDE, you will need to start the IDE and navigate to the Preferences menu. You can access it from the File menu in Windows or Linux, or the Arduino menu on OS X.

💿 Blink Arduind	Blink Arduino 1.8.2								
File Edit Sketch	Tools Help								
New	Ctrl+N	De la companya de la							
Open	Ctrl+O								
Open Recen	ıt	► ×							
Sketchbook		•							
Examples		▶ e second, then off for one second, repe							
Close	Ctrl+W	e second, then our for one second, repe							
Save	Ctrl+S	board LED you can control. On the UNO, ₌							
Save As	Ctrl+Shift+S	pin 13, on MKR1000 on pin 6. LED_BUIL1 endent of which board is used.							
Page Setup	Ctrl+Shift+P								
Print	Ctrl+P	ur board at https://www.arduino.cc/en/							
Finit	Cuttr								
Preferences	Ctrl+Comma	he public domain.							
Quit	Ctrl+Q								

A dialog will pop up just like the one shown below.

Preferences				×		
Settings Network						
Sketchbook location:						
C: \Users \halbe \OneDrive \Do	ocuments\Arduino			Browse		
Editor language:	System Default	 (requires restart of Arduino) 				
Editor font size:	12					
Interface scale: 🛛 Automatic 100 💠 % (requires restart of Arduino)						
Theme: Default theme 🗸 (requires restart of Arduino)						
Show verbose output during:	compilation 🔽 upload					
Compiler warnings:	All 🗸					
Display line numbers		Enable Code Folding				
Verify code after upload		Use external editor				
Check for updates on sta	artup	Save when verifying or uploading				
Use accessibility features	5					
Additional Boards Manager U	RLs:					
More preferences can be edit	ted directly in the file					
C: \Users \halbe \OneDrive \Do	cuments\ArduinoData\preferences.txt					
(edit only when Arduino is no	t running)					
			ОК	Cancel		

We will be adding a URL to the new Additional Boards Manager URLs option. The list of URLs is comma separated, and you will only have to add each URL once. New Adafruit boards and updates to existing boards will automatically be picked up by the Board Manager each time it is opened. The URLs point to index files that the Board Manager uses to build the list of available & installed boards.

To find the most up to date list of URLs you can add, you can visit the list of <u>third party</u> board URLs on the Arduino IDE wiki (). We will only need to add one URL to the IDE in this example, but you can add multiple URLS by separating them with commas. Copy and paste the link below into the Additional Boards Manager URLs option in the Arduino IDE preferences.

https://adafruit.github.io/arduino-board-index/ package_adafruit_index.json

Preferences			×
Settings Network			
Sketchbook location:			
C: \Users \halbe \OneDrive \Doc	uments\Arduino		Browse
Editor language:	System Default v (requires restart of Arduino)		
Editor font size:	12 Window Snip		
Interface scale:	Automatic 100 🗇 % (requires restart of Arduino)		
Theme:	Default theme \lor (requires restart of Arduino)		
Show verbose output during:	□ compilation ☑ upload		
Compiler warnings:	None 🗸		
Display line numbers	Enable Code Folding		
Verify code after upload	Use external editor		
Check for updates on star	tup Save when verifying or uploading		
Use accessibility features			
Additional Boards Manager UR	Ls: https://adafruit.github.io/arduino-board-index/package_adafruit_index.json		
More preferences can be edite	ed directly in the file		
C:\Users\halbe\OneDrive\Doc	uments\ArduinoData\preferences.txt		
(edit only when Arduino is not	running)		
		ОК	Cancel

Here's a short description of each of the Adafruit supplied packages that will be available in the Board Manager when you add the URL:

- Adafruit AVR Boards Includes support for Flora, Gemma, Feather 32u4, ItsyBitsy 32u4, Trinket, & Trinket Pro.
- Adafruit SAMD Boards Includes support for Feather M0 and M4, Metro M0 and M4, ItsyBitsy M0 and M4, Circuit Playground Express, Gemma M0 and Trinket M0
- Arduino Leonardo & Micro MIDI-USB This adds MIDI over USB support for the Flora, Feather 32u4, Micro and Leonardo using the arcore project ().

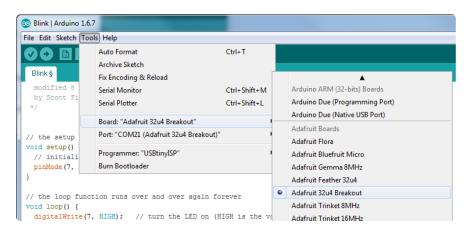
If you have multiple boards you want to support, say ESP8266 and Adafruit, have both URLs in the text box separated by a comma (,)

Once done click OK to save the new preference settings. Next we will look at installing boards with the Board Manager.

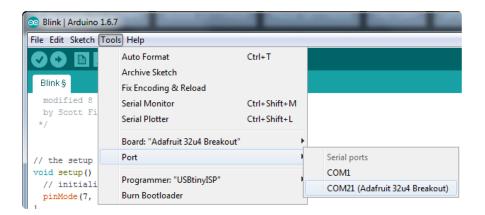
Now continue to the next step to actually install the board support package!

Using with Arduino

Once you have the Arduino IDE Set up and you've installed the board manager package, you can select the Adafruit 32u4 Breakout from the boards list



Then select the port



Create a new sketch:

```
// the setup function runs once when you press reset or power the board
void setup() {
    // initialize digital pin 13 as an output.
    pinMode(7, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
    digitalWrite(7, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(7, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

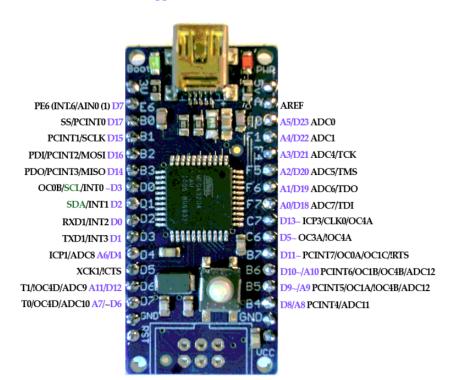
And upload it! You should see the red LED blink once a second

Unlike modern Arduino boards, this board does not auto-reset! Before uploading, you'll need to press the RESET button to get the BOOT LED pulsing

That's it, you can now use the board from the Arduino IDE, you can use this pinmapping guide to help line up which pin is which:

Adafruit ATMega32u4 Breakout Board

Pins mapped to Arduino Leonardo



Using it with Teensyduino

If you'd like to use Teensyduino type programs, you can do that too, by using the TeeOnArdu board plugin. Use the board manager, and select the TeeOnArduino package and install it

💿 Boards Manager	×
Type All 🔹 tee	
Adafruit TeeOnArdu by Adafruit Boards included in this package: TeeOnArdu (Leo on TeensyCore), Flora (TeensyCore). <u>Online helo</u> <u>More info</u>	Install
	Close

Then in the board menu, select TeeOnArdu (Leonardo on TeensyCore)

💿 TrellisMidi168	Arduino 1.6.7			Generic ESP8266 Module Adafruit HUZZAH ESP8266	
File Edit Sketch	Tools Help			NodeMCU 0.9 (ESP-12 Module)	
	Auto Format	Ctrl+T		NodeMCU 1.0 (ESP-12E Module)	
	Archive Sketch			Olimex MOD-WIFI-ESP8266(-DEV)	
TrellisMidi168	Fix Encoding & Reload			SparkFun ESP8266 Thing	
for (uin	Serial Monitor	Ctrl+Shift+M		SweetPea ESP-210	
// Ge uint8	Serial Plotter	Ctrl+Shift+L		TeeOnArdu	
uinco		N.U.	۲	TeeOnArdu (Leo on TeensyCore)	
if(tr	Board: "TeeOnArdu (Leo on TeensyCore)"		Flora (TeensyCore)	
usb	USB Type: "Serial"		1	▼	
tre	Keyboard Layout: "US English"		.		
} els usb	Port				
tre	Programmer: "USBtinyISP"				

You can then select which USB device core you want to use

💿 TrellisMidi168 Ardı	uino 1.6.7			
File Edit Sketch Tool	ls Help			
TrellisMidi168	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T		
for(uin // Ge uint8	Serial Monitor Serial Plotter	Ctrl+Shift+M Ctrl+Shift+L	n. er	
if(tr usb	Board: "TeeOnArdu (Leo on TeensyCore)" USB Type: "Serial"	1		Serial
tre } els usb	Keyboard Layout: "US English" Port	I		Keyboard + Mouse + Joystick Serial + Keyboard + Mouse + Joystick
tre }	Programmer: "USBtinyISP" Burn Bootloader	I		Disk(Internal) + Keyboard Disk(SD Card) + Keyboard MIDI
trellis.wri } prevReadTime	<pre>tteDisplay(); = t;</pre>			Raw HID Flight Sim Controls

Unlike modern Arduino boards, this board does not auto-reset! Before uploading, you'll need to press the RESET button to get the BOOT LED pulsing

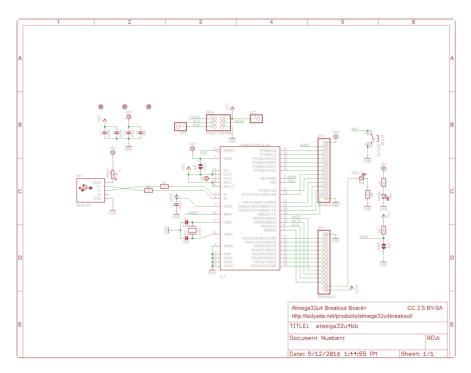
Download

Download

- EagleCAD PCB files on GitHub ()
- Fritzing object available in the Adafruit Fritzing Library ()
- The LUFA USB-stack website ()
- Our minor fork to the LUFA core () this is where our AVR109 bootloader lives (in Bootloaders/CDC)

Download Windows Driver atmega32u4cdc.inf

Schematic



Fabrication Print

Dims in inches

