

Adafruit UPDI Friend

Created by Liz Clark



https://learn.adafruit.com/adafruit-updi-friend

Last updated on 2024-04-16 01:51:04 PM EDT

Table of Contents

Overview	3
Pinouts	5
 Power Pins UPDI Pin JST SH Port Voltage Selector Switch 	
• LEDs	6
 Building a UPDI Programmer Wiring with the UPDI Friend Install megaTinyCore Program the ATtiny Blink Test 	
megaTinyCore Docs	13
Downloads	13
• Files	

• Schematic and Fab Print

Overview



UPDI stands for <u>Unified Program and Debug Interface</u> (https://adafru.it/19eD), but this board is so smol and cute that we will call it the **Unusually Playful Device Interfacer** and pat its head when it does a good job. It's designed to make programming modern ATtiny chips very easy because it has 3V or 5V power and logic select, power and transmit indicator LEDs, and a quick cable for poking into a breadboard.



We have been working a lot with <u>ATtiny816</u>, <u>ATtiny817</u> and <u>ATtiny1616</u> chips (https:// adafru.it/19eE) lately for our seesaw boards. And we're often needing to program them with a CP2102-based breakout (http://adafru.it/5335) with a 1K resistor soldered between the RX and TX pins (https://adafru.it/18ED). But we were hankering for a nicer programmer!



This UPDI Friend makes programming such chips very easy:

- Select between 3V or 5V power and logic the 3V regulator can source up to 500mA so it can run even big projects.
- CH340E USB Serial converter chip with cross-platform drivers.
- 1K Loop-back Resistor between RX and TX.
- USB Type C for data and power connection to any computer.
- JST SH cable included for quick plugging into a breadboard you can get another JST SH 3-pin cable with sockets (http://adafru.it/5765) or with pin header (http://adafru.it/5755) here.
- 0.1" spaced breakout holes for custom connections.
- Green power OK LED
- Red serial activity LED
- Inspired by this open-source hardware design (https://adafru.it/19eF) from <u>Stefan</u> Wagner (https://adafru.it/19eG)!

We use Arduino IDE with the <u>megaTinyCore board support package</u> (https://adafru.it/ VdM) installed, simply select "Serial UPDI" as the programmer type. We use 230Kbps but 56Kbps is also good.

Pinouts



Power Pins

- **PWR** this is the power output pin from the USB-C port. It can output 3V or 5V depending on the position of the voltage selector switch.
- GND common ground for power and logic.

UPDI Pin

• UPDI - This is the logic output from the UPDI Friend. Connect it to the UPDI data pin on the board you are programming. Its logic level is determined by the position of the voltage selector switch (3V or 5V).

JST SH Port

The JST SH port is located above the UPDI, PWR and GND pin. It lets you connect to those same pins without any soldering using a 3 pin JST SH cable. A <u>male pin</u> <u>connector cable</u> (http://adafru.it/5755) is included with the UPDI Friend. Optionally, a socket pin connector (http://adafru.it/5765) version is available.

The red wire is connected to **PWR**, the black wire is connected to **GND** and the white wire is connected to **UPDI**.

Voltage Selector Switch

In the middle of the board is the voltage selector switch. You can use this switch to change the power and logic level to **3V** or **5V**. The 3V regulator can source up to 500mA.

LEDs

- **Power LED** Below the voltage switch is the power LED, labeled **PWR**. It is the green LED.
- Serial activity LED To the right of the voltage switch is the serial activity LED, labeled TX. It is the red LED. It will light up anytime data is transferred to the UPDI pin.

Using the UPDI Friend

We don't provide any support for custom builds of seesaw - but we think this information is cool and useful for the Maker community!

You can reprogram these ATtiny breakouts to run your own firmware. However, the boards do not come with a bootloader. If you want to do development on seesaw (e.g. changing the configuration) you need a separate UPDI programming setup! You can build this setup with a 1K resistor and a USB to Serial cable or with the UPDI Friend board.



Adafruit UPDI Friend - USB Serial UPDI Programmer UPDI stands for https://www.adafruit.com/product/5879



USB to TTL Serial Cable - Debug / Console Cable for Raspberry Pi

The cable is easiest way ever to connect to your microcontroller/Raspberry Pi/WiFi router serial console port. Inside the big USB plug is a USB<->Serial conversion chip and at...

https://www.adafruit.com/product/954



Through-Hole Resistors - 1.0K ohm 5% 1/4W - Pack of 25

ΩMG! You're not going to be able to resist these handy resistor packs! Well, axially, they do all of the resisting for you!This is a 25 Pack of...

https://www.adafruit.com/product/4294

Building a UPDI Programmer



fritzing

USB to Serial cable power to ATtiny VIN (red wire) USB to Serial cable ground to ATtiny G (black wire) USB to Serial cable RX to 1K resistor (white wire) USB to Serial cable TX to 1K resistor (green wire) 1K resistor to ATtiny UPDI pin (white wire)

Wiring with the UPDI Friend



UPDI Friend PWR to ATtiny VIN (red wire) UPDI Friend GND to ATtiny G (black wire) UPDI Friend UPDI to ATtiny UPDI pin (white wire)

Install megaTinyCore

You can compile code for the ATtiny using the <u>megaTinyCore</u> (https://adafru.it/VdM) board support package in the Arduino IDE. There are detailed <u>installation</u> <u>instructions</u> (https://adafru.it/19e4) in the megaTinyCore GitHub repository. The following steps outline how to install it using the Boards Manager.

In the Arduino IDE, go to **Preferences** and add the megaTinyCore boards manager URL to the **Additional Boards Manager URLs list**:



Go to Tools - Board - Boards Manager...

File Edit Sketch	Tools	s Help				
		Auto Format	Ctrl+T			
		Archive Sketch				
sketch_feb27a		Fix Encoding & Reload				
<pre>void setup()</pre>		Manage Libraries	Ctrl+Shift+I			
// put your		Serial Monitor	Ctrl+Shift+M			
}		Serial Plotter	Ctrl+Shift+L			
<pre>void loop() {</pre>		WiFi101 / WiFiNINA Firmware Updater				
// put your		Board: "Arduino Uno"	:		Boards Manager	
}		Port	:		Arduino Yún	
		Get Board Info		•	Arduino Uno	
		Programmer: "AVRISP mkll"	3		Arduino Duemilanove or Diecimila	
		Burn Bootloader			Arduino Nano	
					Arduino Mega or Mega 2560	
					Arduino Mega ADK	
					Arduino Leonardo	
					Arduino Leonardo ETH	

In the **Boards Manager**, search for **megaTinyCore**. Click **Install** to install the board support package.

Boards f	anager	
megaTin	Core	
by Spen	e Konde	
Boards in	cluded in this package:	
Full Ardu	no support for the tinyAVR 0-series, 1-series, and the new	v 2-series!
24-pin p	rts: ATtiny3227/3217/1627/1617/1607/827/817/807/427	
20-pin p	rts: ATtiny3226/3216/1626/1616/1606/826/816/806/426	/416/406
14-pin p	rts: ATtiny3224/1624/1614/1604/824/814/804/424/414/	404/214/204
8-pin pa	s: ATtiny412/402/212/202	
Microchip	Boards: Curiosity Nano 3217/1627/1607 and Xplained Pro	o (3217/817), Mini (817) Nano (416). Direct USB uploads
may not	work on linux, but you can export hex and	
upload t now that but noth mEDBG,	rough the mass storage projection., 2.6.5 corrects a numl all sizes of all families are shipping, adds new bootloader ng particularly big., Supported UPDI programmers: SerialL DBG, SNAP, Atmel-ICE and PICkit4 - or use one of those	ber of small, mostly minor issues. Reorders the list of chips entry conditions and a menu to tweak optimization settings, JPDI (serial adapter w/diode or resistor), jtag2updi, nEDBG, to load
the Optil your app	oot serial bootloader (included) for serial programming. V ication and requirements.	Vhich programing method makes more sense depends on
The full we recon	ocumentation is not included with board manager installa mend viewing it through github at the link above	tions (it is hard to find and the images bloat the download);
or if it m More Inf	ist be read withouht an internet connection by downaloding	g the manual installation package.
		2.6.5 🗸 Install

Program the ATtiny

After the megaTinyCore finishes installing, go to **Tools - Board - megaTinyCore** and **select the board list** that includes your ATtiny.



Then, select the chip of your ATtiny.



Next, **set clock** if needed. The default 20 MHz clock option is only valid if powering with 5V. If powering with 3.3V, select 10 MHz for the clock.

Iools Help	
Auto Format	Ctrl+T
Archive Sketch	
Fix Encoding & Reload	
Manage Libraries	Ctrl+Shift+I
Serial Monitor	Ctrl+Shift+M
Serial Plotter	○ 20 MHz internal
WiFi101 / WiFiNINA Firmware Updater	O 16 MHz internal
ESP Exception Decoder	● 10 MHz internal
ESP32 Sketch Data Upload	🔿 8 MHz internal
Board: "ATtiny3226/3216/1626/1616/1606/826/816/806/426/416"	○ 5 MHz internal
Chip: "ATtiny3226"	○ 4 MHz internal
Clock: "10 MHz internal"	🔾 2 MHz internal
millis()/micros() Timer: "Enabled (default timer)"	🔿 1 MHz internal
Startup Time: "8ms"	🔾 20 MHz internal - tuned
BOD Voltage Level (burn bootloader req'd): "1.8V (5 MHz or less)"	🔾 16 MHz internal - tuned
BOD Mode when Active/Sleeping (burn bootloader req'd): "Disabled/Disabled"	🔾 12 MHz internal - tuned
Save EEPROM (burn bootloader req'd): "EEPROM retained"	🔾 10 MHz internal - tuned
UPDI/Reset Pin Function (burn bootloader req'd): "UPDI (no reset pin)"	🔘 8 MHz internal - tuned
printf(): "Default (doesn't print floats, 1.4k flash use)"	○ 6 MHz internal - tuned
Wire (Wire.h/I2C) Library mode: "Master or Slave (saves flash and RAM)"	○ 5 MHz internal - tuned
WDT timeout (burn bootloader req'd. Cannot be disabled at runtime if set here): "Disabled (r	€ ○ 4 MHz internal - tuned
WDT window (if desired. burn bootloader req'd. Cannot be disabled at runtime if set here): "N	○ 1 MHz internal - tuned
PWM pins (advanced, see core documentation): "PB0-2, PA3-5, 1-series: PC0/1 (default)"	○ 20 MHz external clock
attachInterrupt Mode: "On all pins, with new implementation."	◯ 16 MHz external clock
Port	◯ 12 MHz external clock
Get Board Info	○ 10 MHz external clock
Programmer: "SerialUPDI - SLOW: 57600 baud"	○ 8 MHz external clock
Burn Bootloader	○ 24 MHz internal - tuned, overclocked

Under Programmer, select SerialUPDI - SLOW: 57600 baud.

rogrammer: "SerialUPDI - SLOW: 57600 baud"	SerialUPDI - 230400 baud
im Bootloader	SerialUPDI - 230400 baud w/write delay, 2:7V+ (Linux/MacOS: other adapters, Winc maybe some adapters)
	SerialUPDI - SLOW: 57600 baud
	jtag2updi
	SerialUPDI - FAST: 4.5v+ 460000 baud (CH340 - and maybe some others)
	SerialUPDI - FAST: 4.5v+ 460800 baud w/write delay (other non-CH340 adapters, often slower than 230k, esp. on W
	SerialUPDI - TURBD: 4.5v+ 921600 baud (CH340 (and maybe some others) only)
	SerialUPDI - TURBO: 4.5v+ 921600 baud w/write delay, (adapters that support it and are not CH340, many fall back
	Curiosity Nano (nEDBG, debug chip: ATSAMD21E18)
	Xplained Pro (EDBG, debug chip: AT32UC3A4256)
	Xplained Mini (mEDBG, debug chip: ATmega32u4)
	Atmel-ICE
	PICkib4 (UPDI mode)
	MPLAB SINAP (UPDI mode)

Other options can be left at their defaults.

Finally, go to **Preferences** and **check ON Show verbose output during upload**. This will give you the upload details and progress in the monitor at the bottom of the Arduino IDE, which is very useful for debugging.

Preferences			×
Settings Network			
Sketchbook location:			
C:\Users\izillah\Documents\A	rduino		Browse
Editor language:	System Default v (requires restart of Arduino)		
Editor font size:	12		
Interface scale:	Automatic 100 * % (requires restart of Arduino)		
Theme:	Theme: Default theme v (requires restart of Arduino)		
Show verbose output during:	compilation v upload		
Compiler warnings:	None 🗸		
Display line numbers	Enable Code Folding		
Verify code after upload	Use external editor		
Check for updates on sta	rtup Save when verifying or uploading		
Use accessibility features			
Additional Boards Manager UR	Ls: http://drazzy.com/package_drazzy.com_index.json		
More preferences can be edite	ed directly in the file		
C: \Users\lizillah\AppData\Loca	Il\Arduino 15\preferences.bct		
(edit only when Arduino is not	running)		
		OK	Cancel

Now you can compile code with the megaTinyCore to upload to the ATtiny with the UPDI programmer by going to **Sketch** > **Upload Using Programmer**.



Blink Test

This simple example blinks the ATTiny816 onboard red indicator LED on pin 10.

```
void setup() {
   pinMode(10, OUTPUT);
}
void loop() {
   digitalWrite(10, HIGH);
   delay(1000);
   digitalWrite(10, LOW);
   delay(1000);
}
```

If using this with the ATTiny817, you'll want to update the pin to 5, as shown below.

```
void setup() {
   pinMode(5, OUTPUT);
}
void loop() {
   digitalWrite(5, HIGH);
   delay(1000);
   digitalWrite(5, LOW);
   delay(1000);
}
```

Upload the sketch using the UPDI programmer. You should see this output in the monitor after the upload has completed successfully:

Done uploading.
Action took 0.04s Writing from hex file
pymcuprog.programmer - INFO - Write
<pre>pymcuprog.programmer - INFO - Writing 520 bytes of data to flash [] 9/9</pre>
pymcuprog.programmer - INFO - Write complete. Action took 0.45s
Verifying
pymcuprog.programmer - INFO - Reading 520 bytes from flash [========] 2/2
pymcuprog.programmer - INFO - Verifying Marify guagessful - Data in flash matches data in specified her file
Action took 1.87s
pymcuprog.serialupdi.application - INFO - Leaving NVM programming mode
pymcuprog.serialupdi.application - INFO - Apply reset
pymcuprog.serialupdi.application - INFO - Release reset
pymcuprog.serialupdi.link - INFO - STCS 00 to 0x08
pymcuprog.serialupdi.link - INFO - STCS OC to 0x03 pymcuprog.serialupdi.physical - INFO - Closing port 'COM6'

The red LED on the ATtiny should be blinking!

megaTinyCore Docs

megaTinyCore Docs (https://adafru.it/VdM)

Downloads

Files

- Window and Mac driver download (https://adafru.it/19eH)
- CH340E Datasheet (https://adafru.it/19el)
- UPDI Interface Docs (https://adafru.it/19eJ)
- EagleCAD PCB files on GitHub (https://adafru.it/19eK)
- Fritzing object in the Adafruit Fritzing Library (https://adafru.it/19eL)

Schematic and Fab Print



