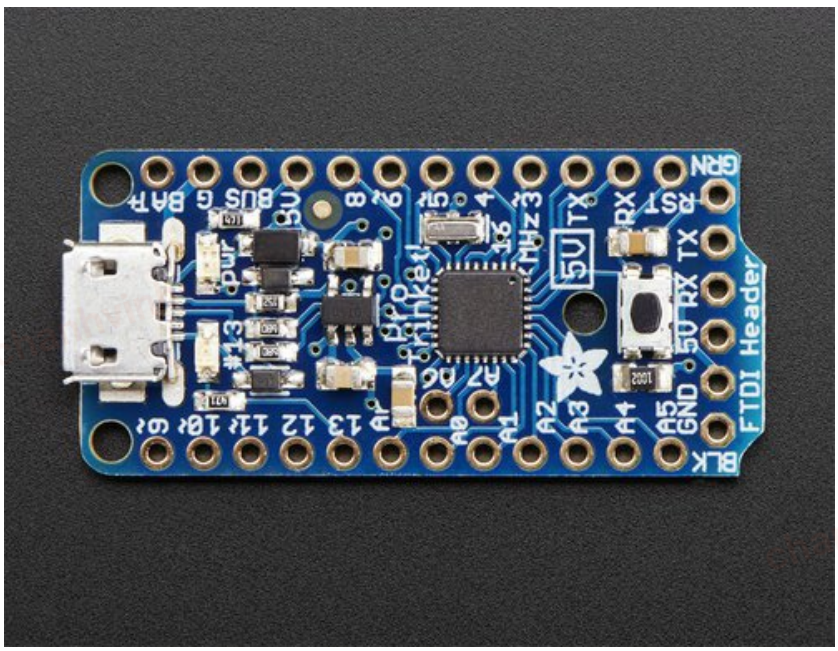
chanhvinh.com

Introducing Pro Trinket

Created by lady ada



Last updated on 2016-09-13 04:15:06 PM UTC

chanhvinh.com

Guide Contents

Guide Contents 2
Overview 4
Guided Tour 11
Pinouts 14
Power Pins 15
GPIO Pins 16
- Logic Level 16
- The Digital Only GPIO Pins 17
- The Analog/Digital GPIO Pins 18
- The Analog only pins 19

Other Pins 20
FTDI Breakout 21
Windows Driver Installation 23
Using the USB Bootloader 26
About the bootloader 26
Pro Trinket USB Drivers for Windows 26
Special Notes on using Pro Trinket with Linux 27
How to start the USB bootloader 28
Using FTDI Cables 29
Select the correct Board in IDE 31
Connect FTDI Friend or Cable 32
Setting up Arduino IDE 36
Arduino IDE Setup 36
Blink! 36
Something Went Wrong! 39
- If you get the error message avrdude: Error: Could not find USBtiny device (0x1781/0xc9f) 39

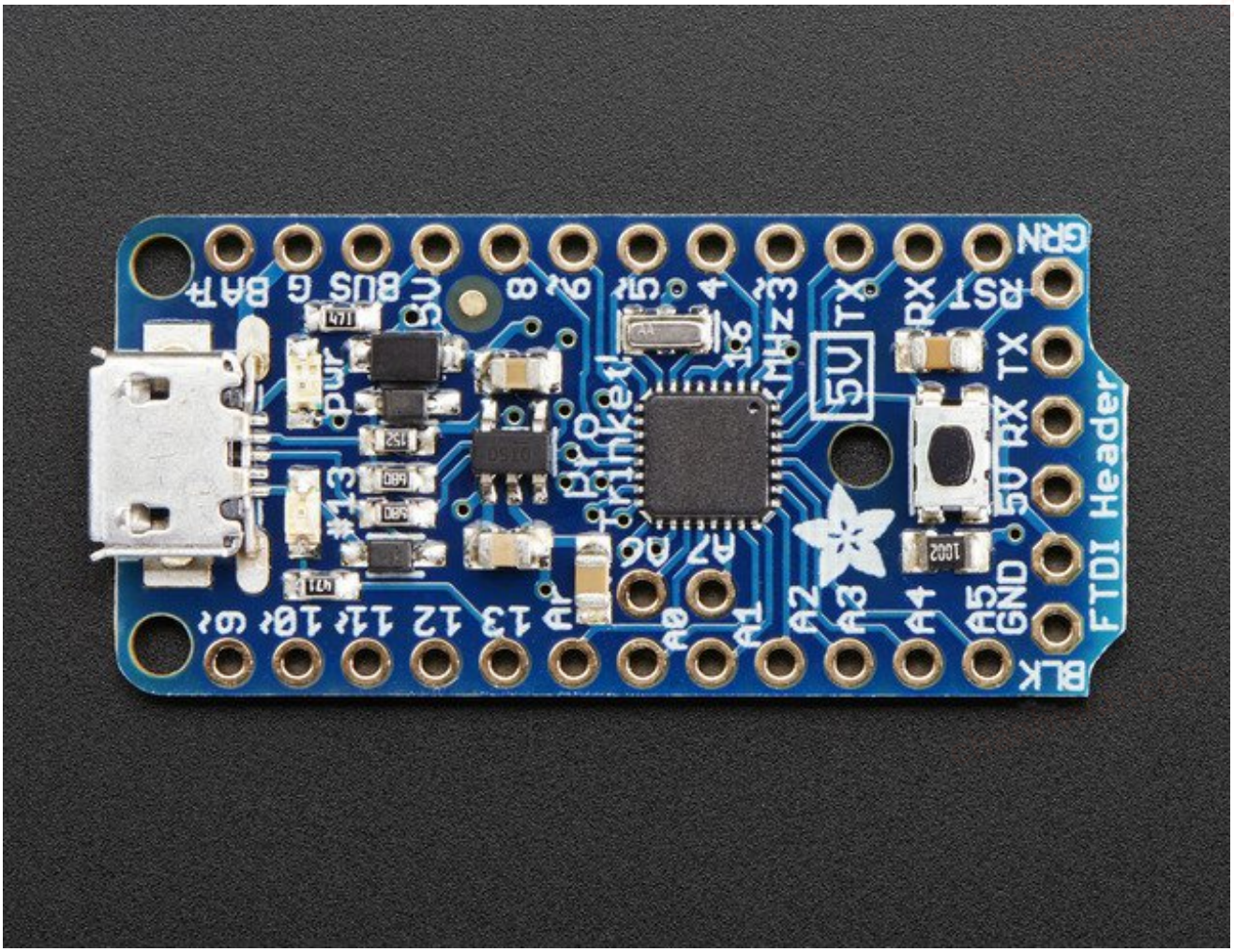
Hints for Arduino-compatibility 41
Re-programming Bootloader 43
F.A.Q. 45
Downloads 47
Datasheets & Files 47

chanhvinh.com

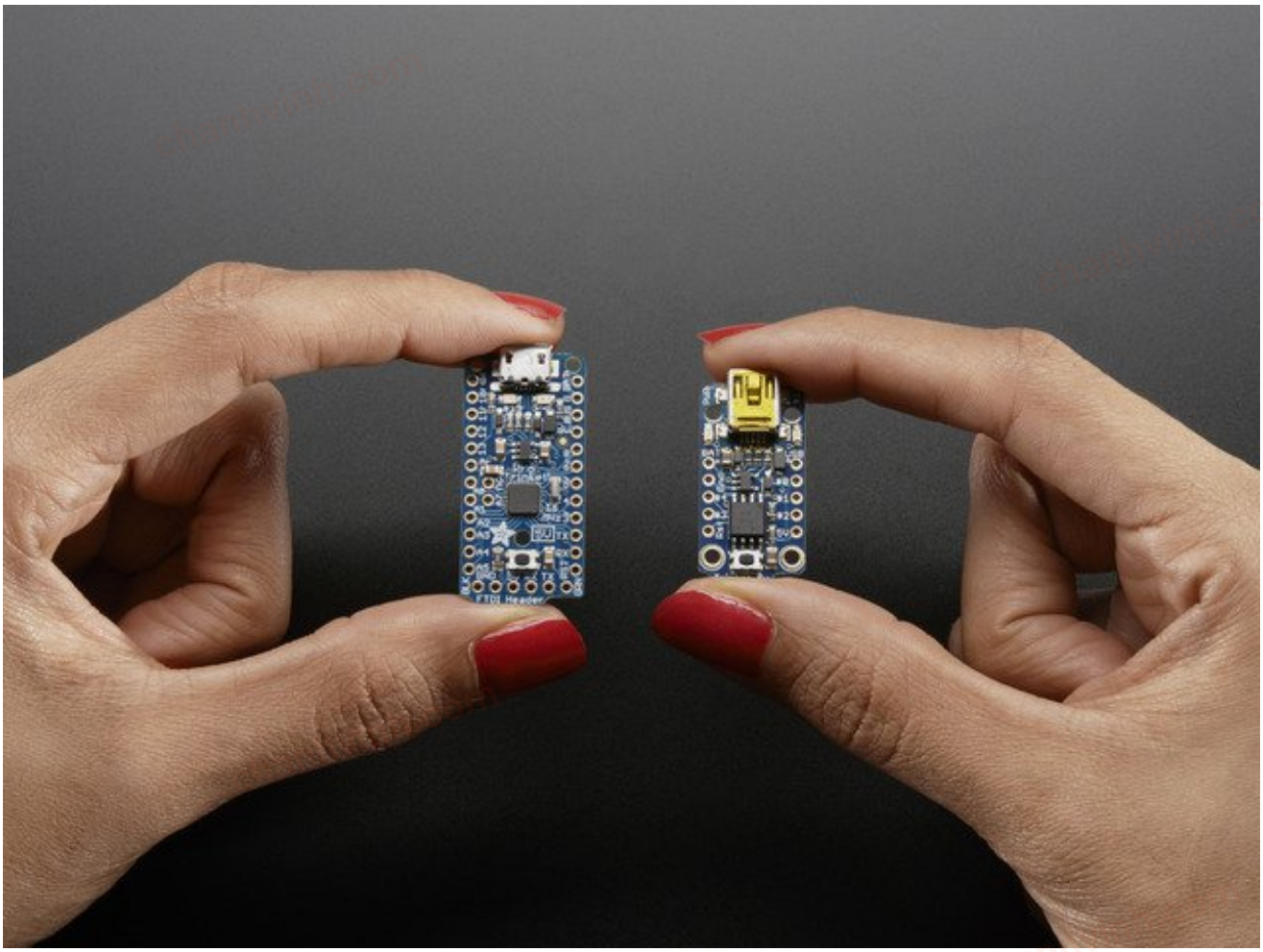
Source code 47
Schematics 47
Pro Trinket 5V 16MHz: 47
Pro Trinket 3V @ 12MHz 48

chanhvinh.com

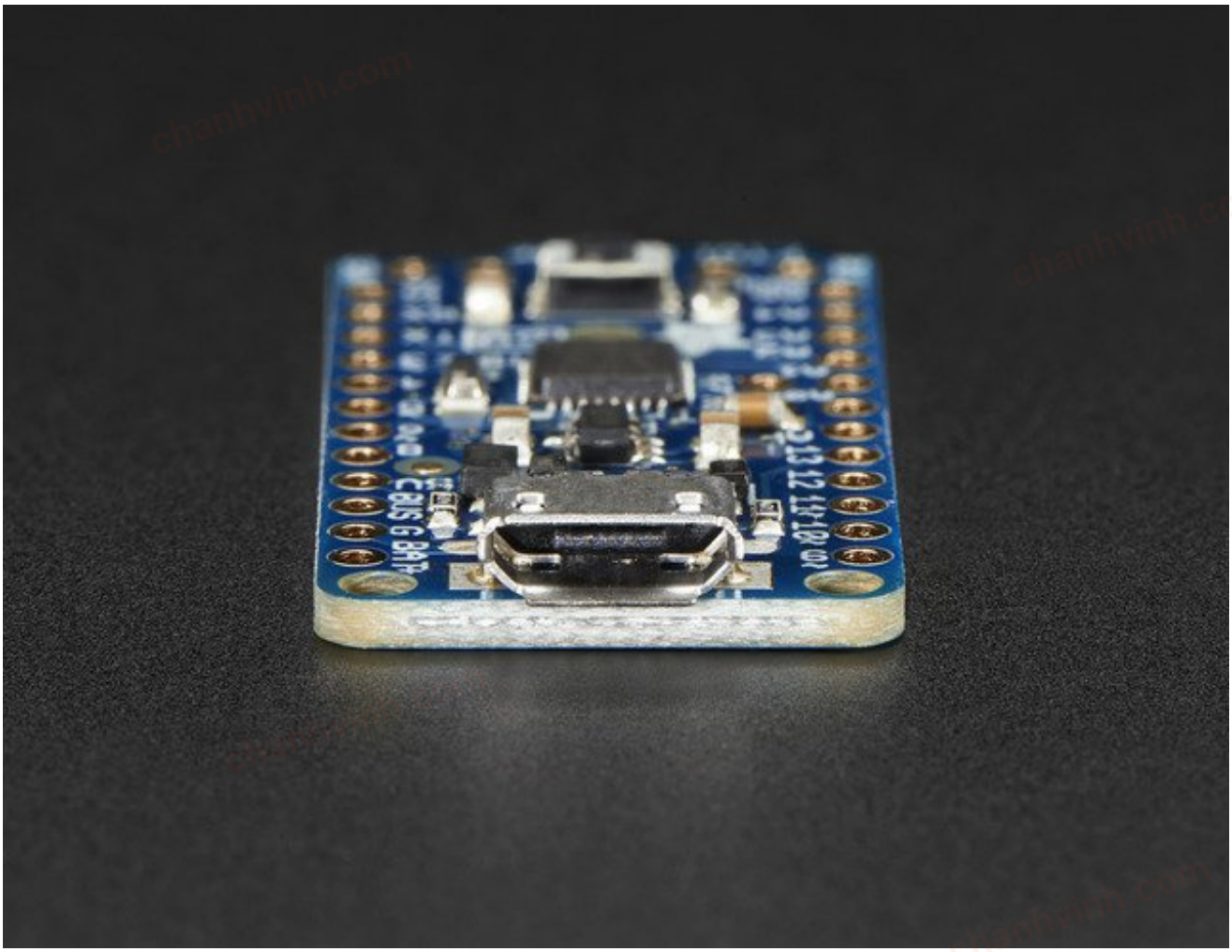
Overview



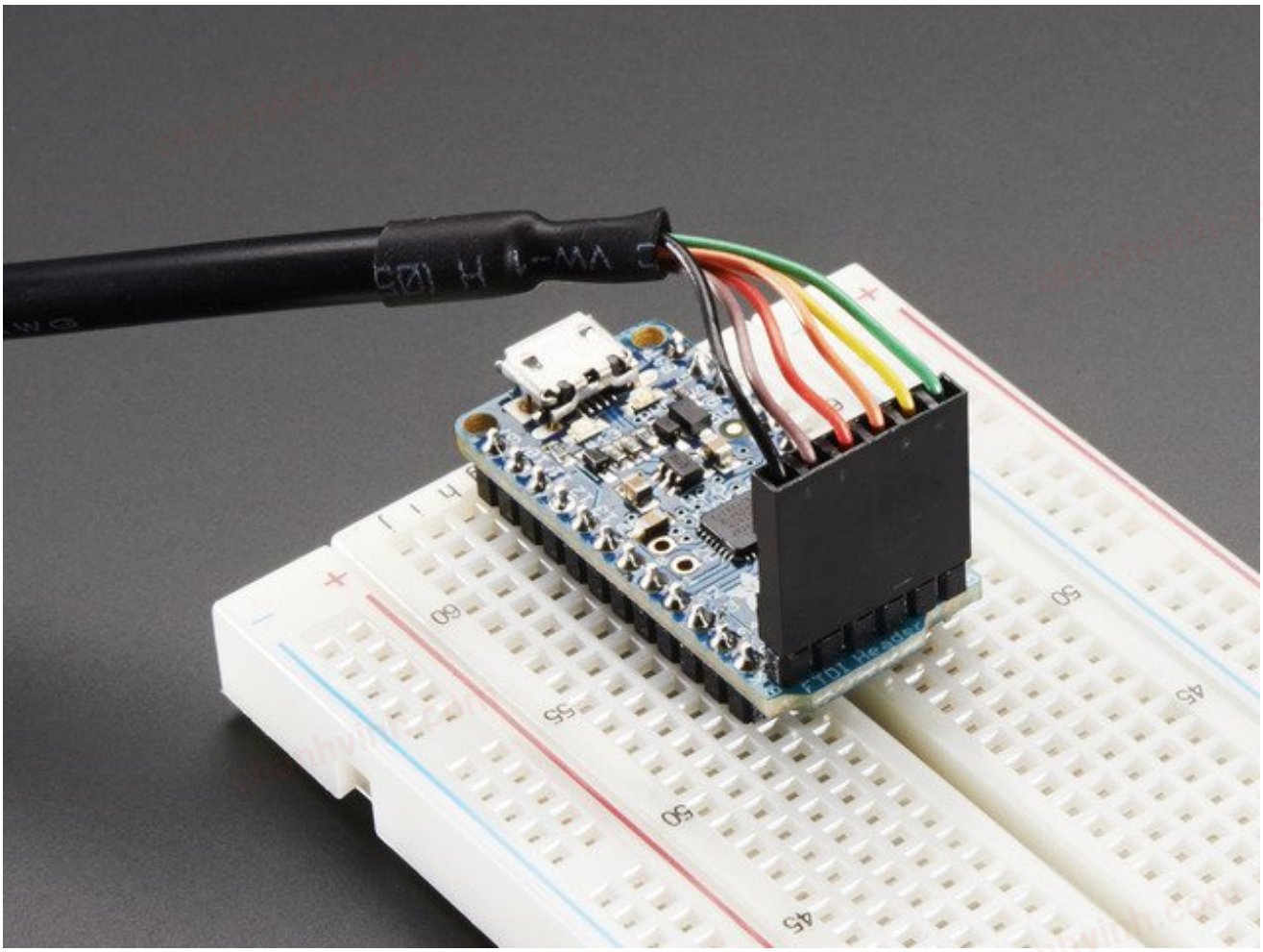
Trinket's got a big sister in town - the Pro Trinket! Pro Trinket combines everything you love about Trinket with the familiarity of the core Arduino chip, the ATmega328. It's like an Arduino Pro Mini with more pins and USB tossed in.



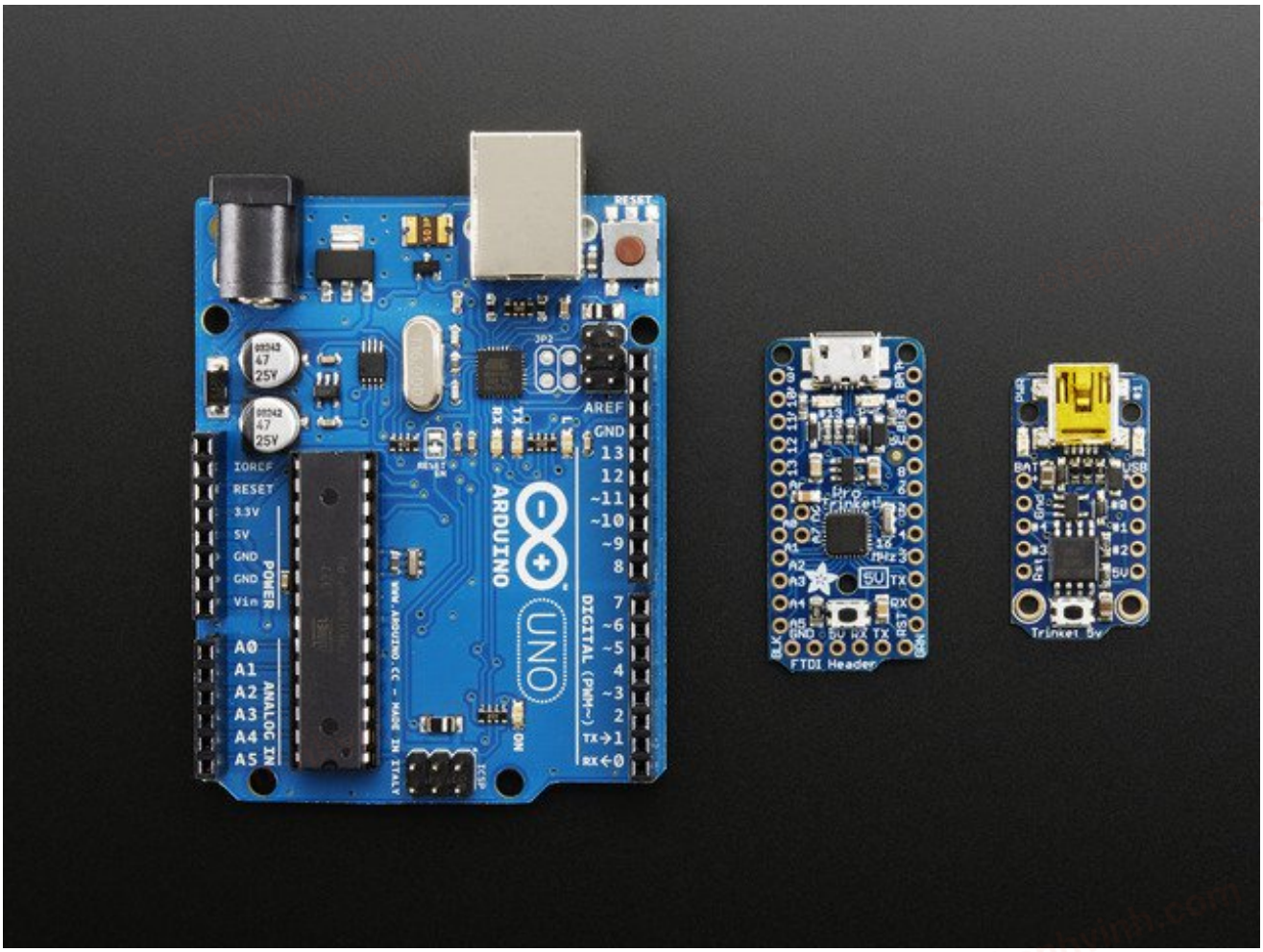
Trinket's a year old now, and while its been great to see tons of tiny projects, sometimes you just need more pins, more FLASH, and more RAM. That's why we designed Pro Trinket, with 18 GPIO, 2 extra analog inputs, 28K of flash, and 2K of RAM.



Like the Trinket, it has onboard USB bootloading support - we opted for a MicroUSB jack this time. We also added Optiboot support, so you can either program your Pro Trinket over USB or with a FTDI cable just like the Pro Mini and friends.



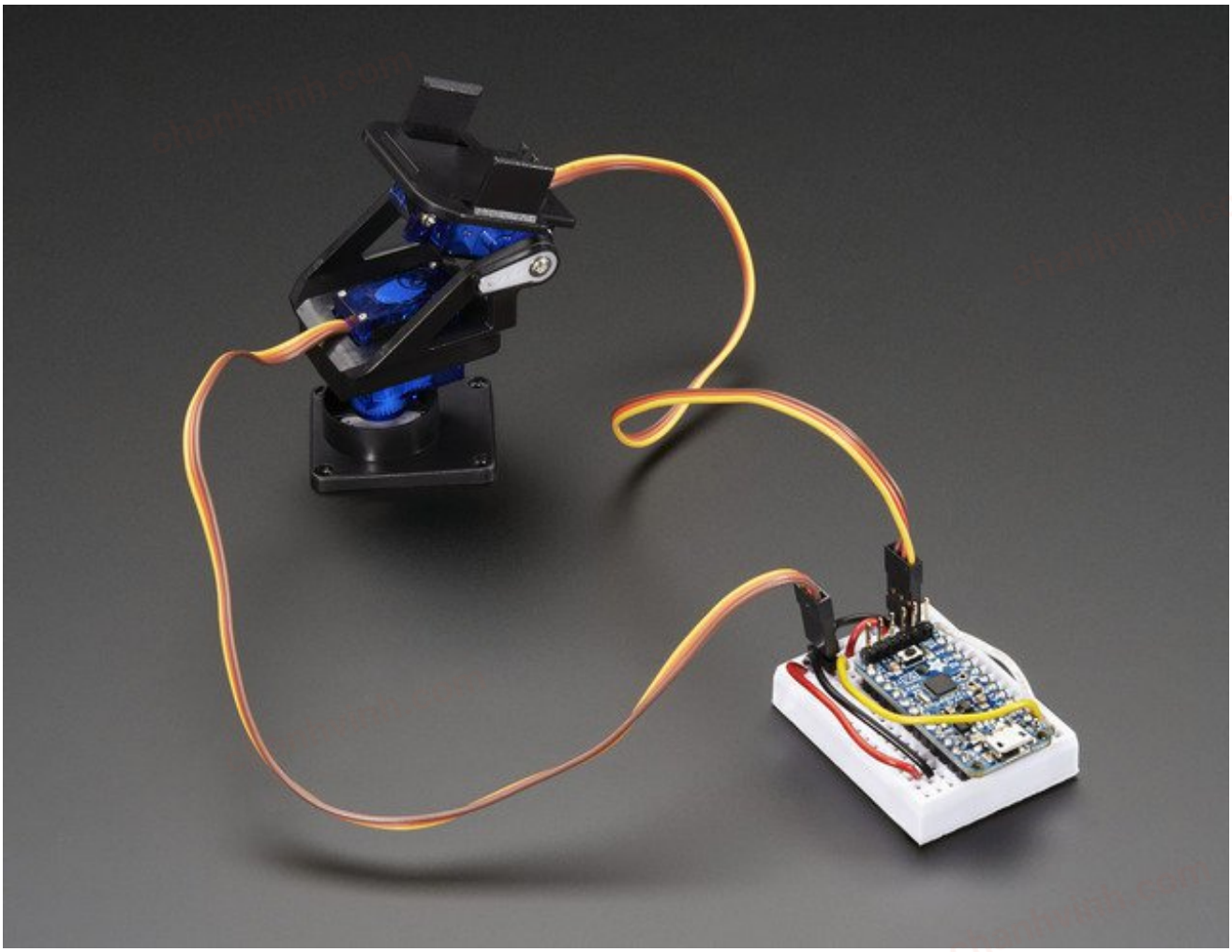
The Pro Trinket PCB measures only 1.5" x 0.7" x 0.2" (without headers) but packs the same capability as an Arduino UNO. So it's great once you've finished up a prototype on an official Arduino UNO and want to make the project smaller.



The Pro Trinket 5V uses the Atmega328P chip, which is the same core chip in the Arduino UNO/Duemilanove/Mini/etc. at the same speed and voltage. So you'll be happy to hear that not only is Pro Trinket programmable using the Arduino IDE as you already set up, but 99% of Arduino projects will work out of the box!

Here's some things you may have to consider when adapting Arduino sketches:

- Pins #2 and #7 are not available (they are exclusively for USB)
- The onboard 5V regulator can provide 150mA output, not 800mA out
- You cannot plug shields directly into the Pro Trinket
- SoftwareSerial will not work on the 3V Pro Trinket without changes.See this FAQ (http://adafru.it/man).
- There is no Serial-to-USB chip onboard. This is to keep the Pro Trinket small and inexpensive, you can use any FTDI cable to connect to the FTDI port for a Serial connection. The USB connection is for uploading new code only.
- The bootloader on the Pro Trinket use 4KB of FLASH so the maximum sketch size is 28,672 bytes. The bootloader does not affect RAM usage.
- Trinket is supported on Linux, Mac OS or Windows!



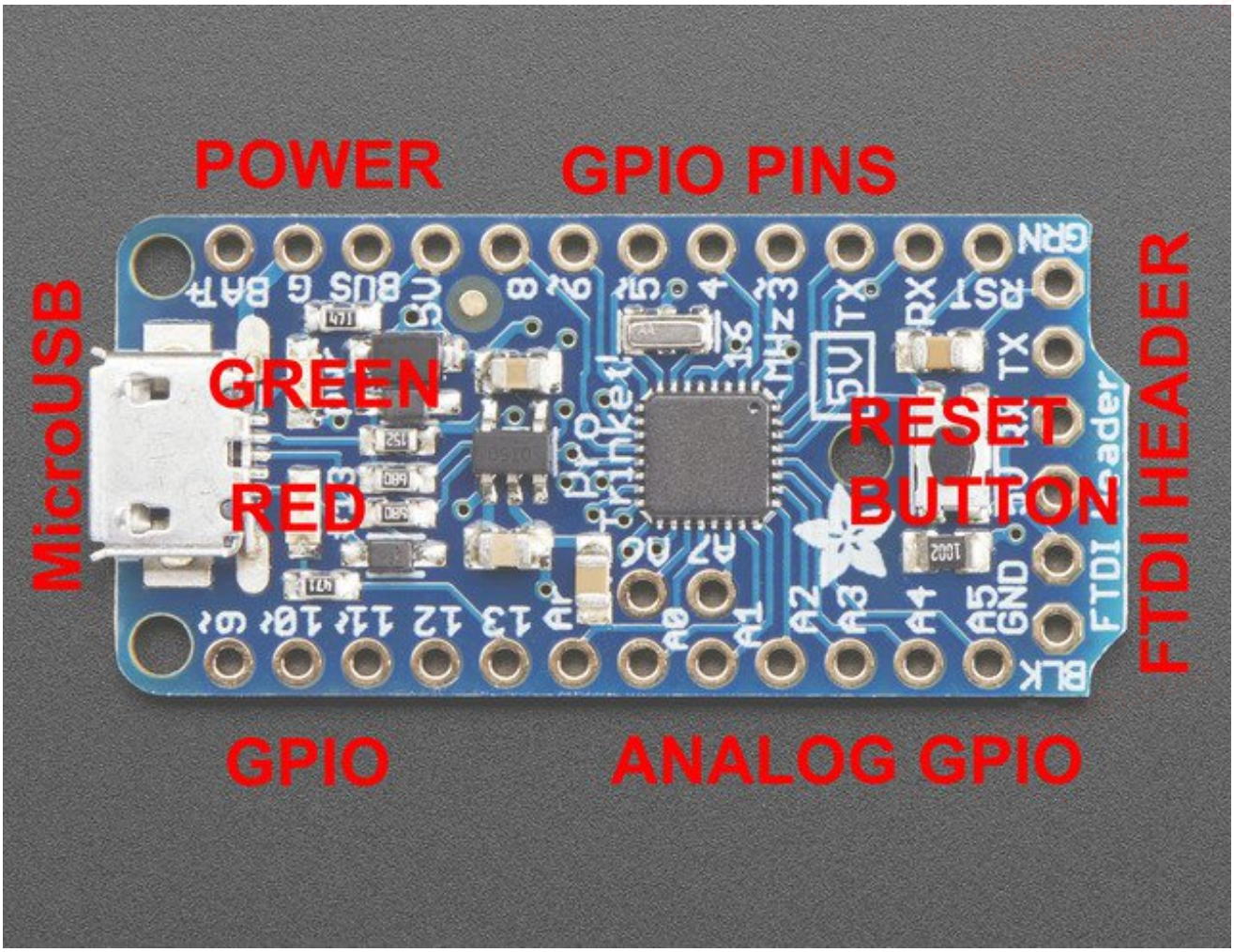
Here's some handy specifications:

- ATmega328P onboad chip in QFN package
- 16MHz clock rate, 28K FLASH available
- USB bootloader with a nice LED indicator looks just like a USBtinyISP so you can program it with AVRdude and/or the Arduino IDE (with a few simple config modifications).
- Also has headers for an FTDI port for reprogramming
- Micro-USB jack for power and/or USB uploading, you can put it in a box or tape it up and use any USB cable for when you want to reprogram.
- On-board 5.0V power regulator with 150mA output capability and ultra-low dropout. Up to 16V input, reverse-polarity protection, thermal and current-limit protection.
- Power with either USB or external output (such as a battery) - it'll automatically switch over
- On-board green power LED and red pin #13 LED
- Reset button for entering the bootloader or restarting the program.
- Works with 99% of existing Arduino sketches (anything that doesn't use more than 28K, and doesn't require pins #2 and #7)

- **Mounting holes! Yeah!**

chanhvinh.com

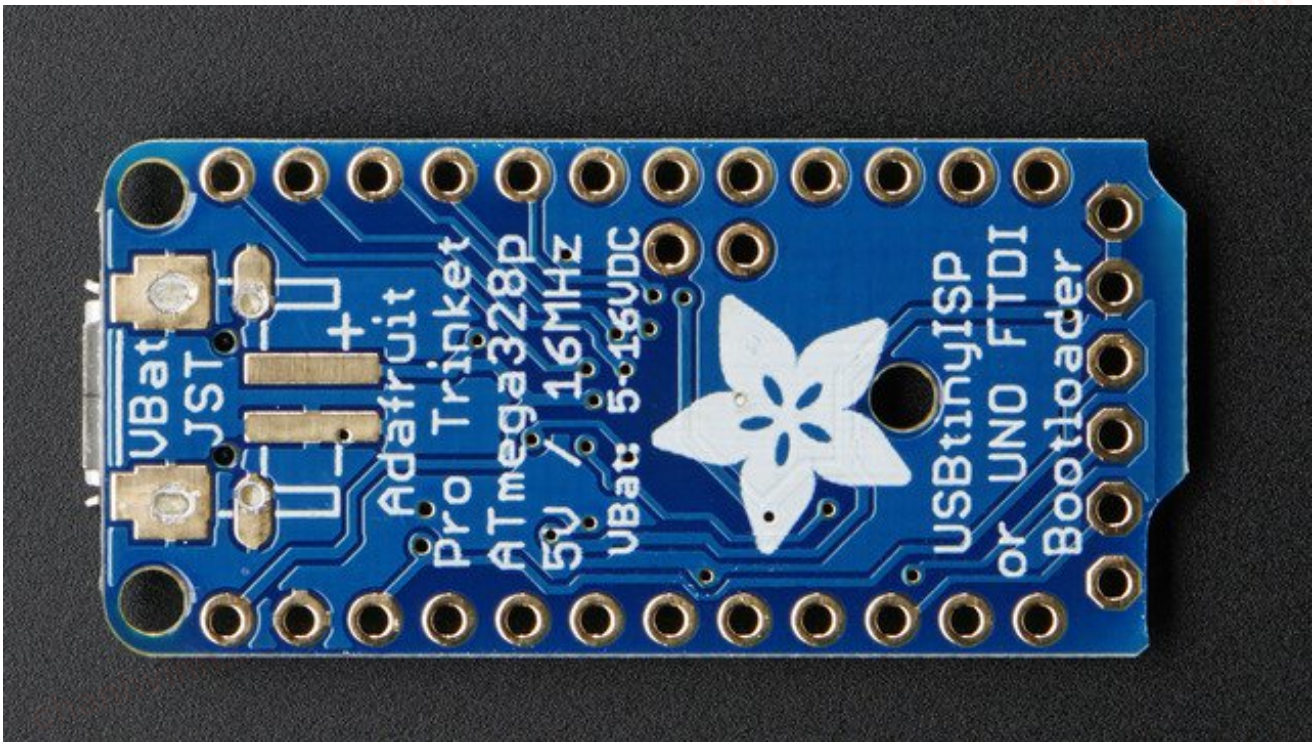
Guided Tour



Let me take you on a tour of your Trinket! Each trinket is assembled here at Adafruit and comes chock-full of good design to make it a joy to use.

- **Micro-B USB connector** - We went with the ultra-common micro-B USB connector for power and/or USB bootloading. We use a special half SMT/half thru hole connector with extra big pads so the connector is very strong. It's also a proper USB connector, so you can use any length cable. Don't forget to use a data+power USB cable if you want to bootload! Some really inexpensive USB cables are 'power only'
- **Green Power LED** - you'll know that the board is powered up when this bright LED is lit
- **Red #13 LED** - this LED does double duty. Its connected with a series resistor to the digital #13 GPIO pin. It pulses or blinks nicely when the Pro Trinket is in bootloader mode, and its also handy for when you want an indicator LED.

- **Battery + Input** - take your Trinket anywhere and power it from an external battery. This pin can take up 16V DC input, and has reverse-polarity, over-curent and thermal protections. The circuitry inside will use either the battery or USB power, safely switching from one to the other. If both are connected, it will use whichever has the higher voltage
- **USB Power Output** - You can also snag the 5V power from the USB jack in case you need 500mA+ current from your computer or portable USB power pack.
- **Breadboard friendly pins** - You can easily plug this into a little breadboard and have plenty of space for working and plugging stuff in
- **GPIO pins** - 18 GPIO pins + 2 analog-only pins, at 3 or 5V logic, check the next section for a detailed pinout guide
- **3 or 5V output** - an onboard regulator provides 3.3V or 5V output for powering LEDs, sensors, small motors, etc.
- **Reset Button** - an onboard reset button will launch the bootloader when pressed and the Trinket is plugged into a computer. If it is not connected to a computer, it's smart enough to go straight to the program.
- **External Reset Pin** - we bring out the reset pin so you can reset or restart your Trinket on the road. If the Trinket is in a box or otherwise hard to get to, you can wire up a button to this pin for an external reset button.
- **Three mounting holes** make it easy to attach with 2mm screws (M2 metric or #2 imperial) or even tiny zip-ties, string, etc.

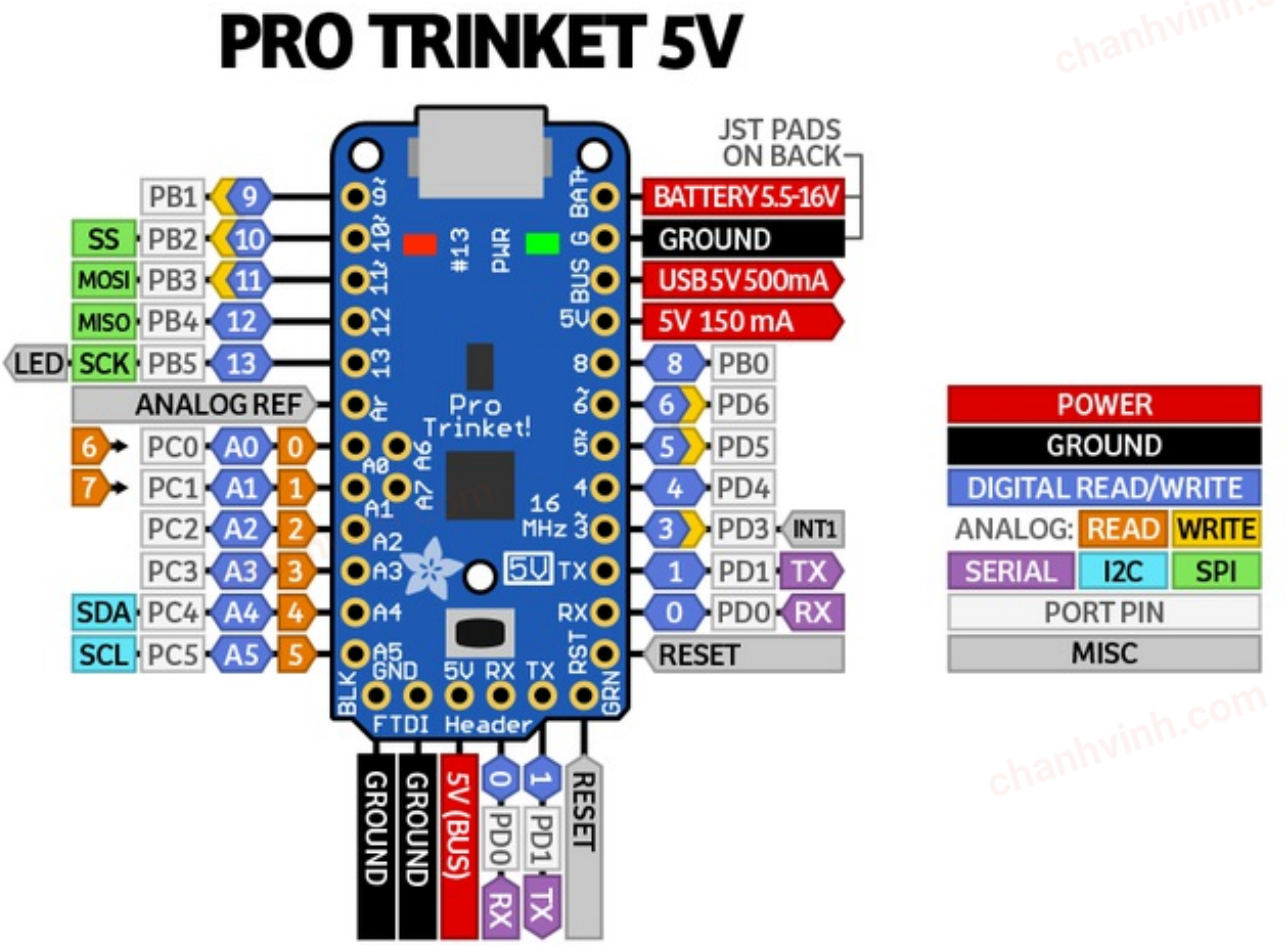


On the bottom there's a spot for a2-pin JST-PH connector (http://adafru.it/1769). this is optional and can be soldered on by hand. It will let you use a lipoly or AAA battery

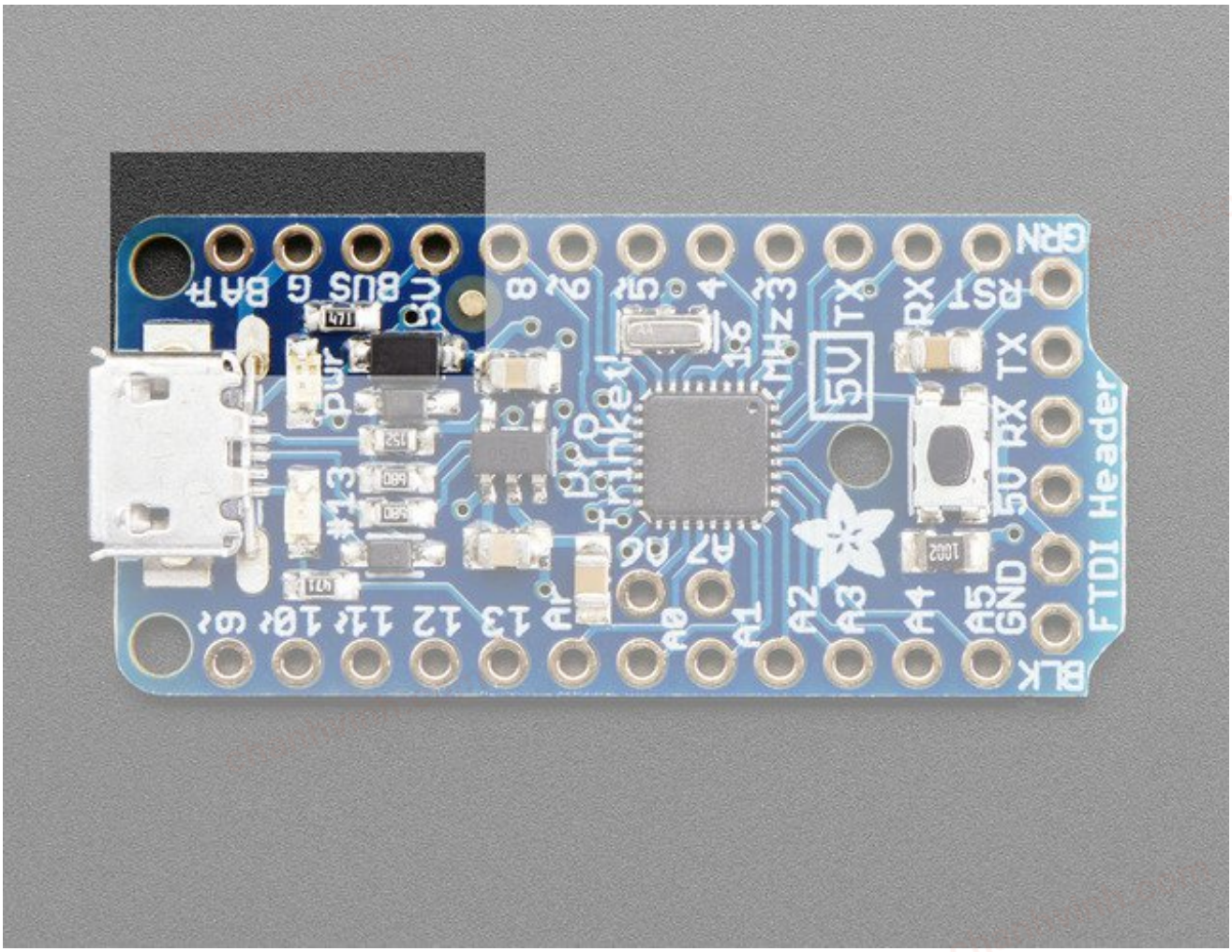
pack (http://adafru.it/727) or a JST cable (http://adafru.it/drM) for other power input. It's connected to the GND and VBAT pins

chanhvinh.com

Pinouts



There are two versions of the Pro Trinket: 3V and 5V. They are almost identical but there are slight differences in the pinouts: one has a 3V output pin in the top right, the other has a 5V output pin instead



Power Pins

We'll start with the top pins **BAT+, GND**, **USB+**, and **3V** or **5V**

- **BAT+** is the **Battery + Input** pin. If you want to power the Pro Trinket from a battery or power adapter or solar panel or any other kind of power source, connect the **+** (positive) pin here! You can connect up to 16V DC. If you have a 3V Pro Trinket, you'll want at least 3.5V input to get a good 3.3V output. If you have a 5V Pro Trinket, 5.5V or higher is suggested. This input is reverse-polarity protected.
- **GND** is the common ground pin, used for logic and power. It is connected to the USB ground and the power regulator, etc. This is the pin you'll want to use for any and all ground connections
- **BUS** is the **USB Bus + Output** pin. If you want to use the USB 5V power for something, like charging a battery, or if you need more than 150mA of current (this pin can supply 500mA+ from USB ports) or to detect when the Pro Trinket is plugged into

USB, this pin will have 5V power on it if and only if its plugged into something via the micro B connector. Don't put power into this pin, that's what **BAT+** is for - use it only as an output!

The next pin is either **3V** or **5V** depending on which Trinket Pro you have. It is the output of the low dropout regulator.

The regulator is powered by either **BUS** or **BAT** (whichever is higher!) It will automatically switch between the two if USB or battery is removed using two schottky diodes. There is a 0.2V dropout due to the diode and maybe 50-100mV minimum drop in the regulator.

The regulator can supply approximately 150mA as long as it doesnt overheat. Make sure that you keep **BAT+** low to minimize heat-loss! About 1 or 2 Volts higher than 3V or 5V is ideal.

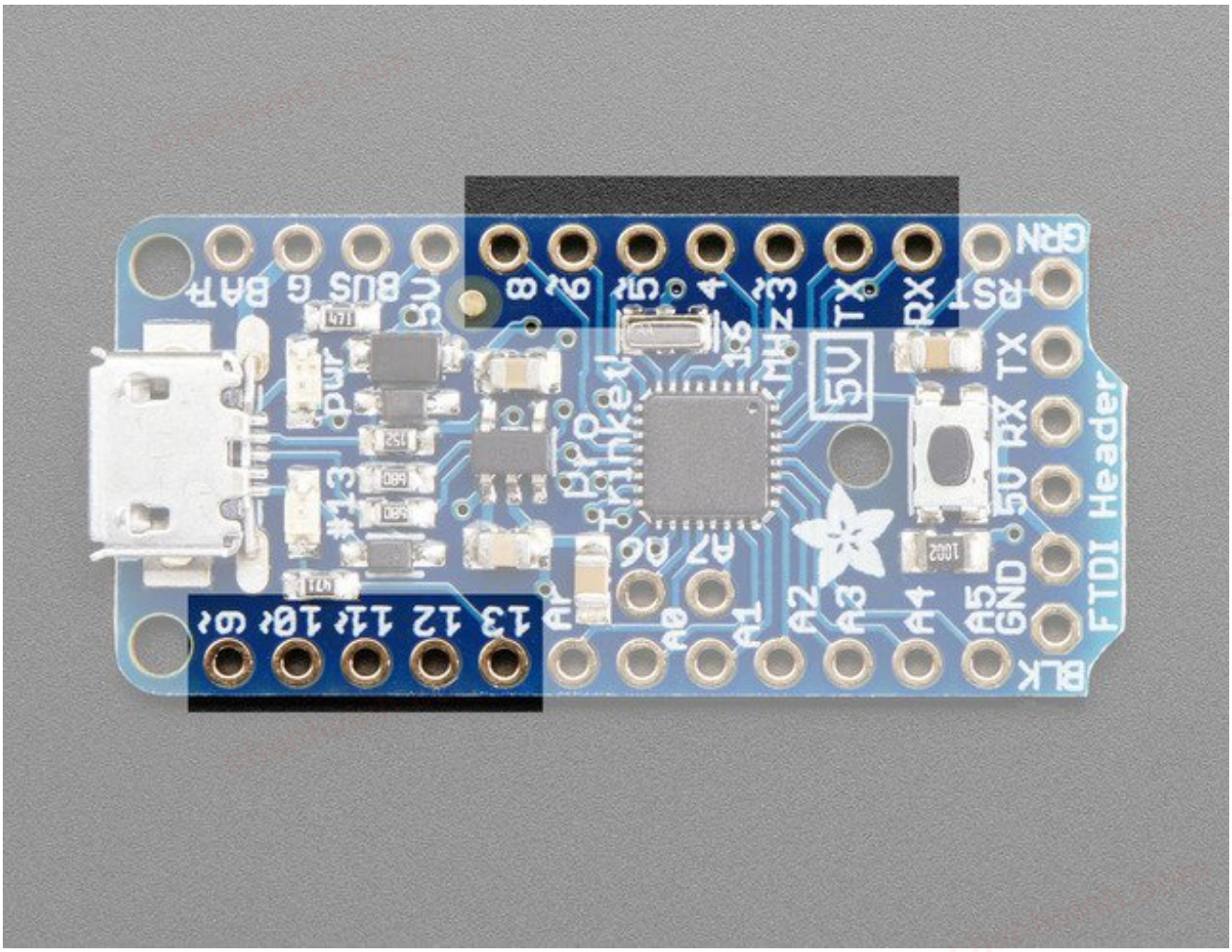
GPIO Pins

Next we will cover the 18 GPIO (General Purpose Input Ouput) pins! 18 GPIO is a big step up from the little Trinket with only 5 or so pins. This is almost as many as a 'full grown' Arduino. In fact, since the Pro Trinket uses the same core chip as an Arduino, nearly any existing project that uses special pins like hardware PWM or SPI will work 'out of the box'.

All the GPIO pins can be used as digital inputs, digital outputs, for LEDs, buttons and switches etc. They can provide up to 20mA of current. Don't connect a motor or other high-power component directly to the pins! Instead, use a transistor to power the DC motor on/off (http://adafru.it/aUD)

Logic Level

On a 3V Pro Trinket, the GPIO are 3.3V output level, and should not be used with 5V inputs. On a 5V Pro Trinket, the GPIO are 5V output level, and can be used with 3V inputs but may damage electronic devices that are 3V input only!

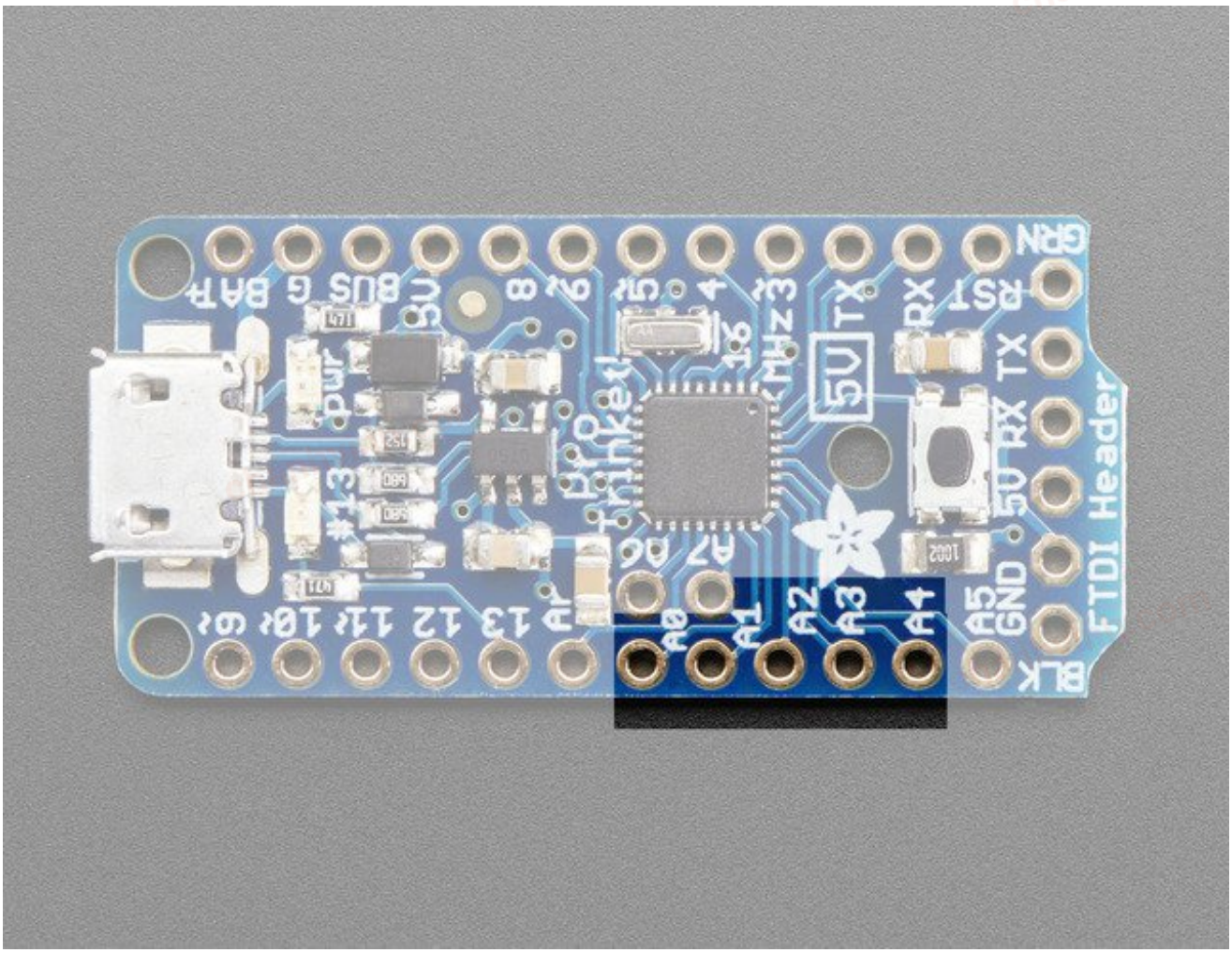


The Digital Only GPIO Pins

- **RX** - also known as Digital #0, this is the hardware serial input pin. This is used when programming with an FTDI cable but is available when using the native USB to program
- **TX** - also known as Digital #1, this is the hardware serial output pin. This is used when programming with an FTDI cable but is available when using the native USB to program
- **Digital 3** - Also known as external interrupt #1. This pin can also be used as a PWM output pin using **analogWrite()**
- **Digital 4** - Nothing particularly special about this GPIO pin
- **Digital 5** - this pin can also be used as a PWM output pin using**analogWrite()**
- **Digital 6** - this pin can also be used as a PWM output pin using**analogWrite()**
- **Digital 8** - Nothing particularly special about this GPIO pin
- **Digital 9** - this pin can also be used as a PWM output pin using**analogWrite()** It's also good for driving servos because its a high-speed PWM output
- **Digital 10** - this pin can also be used as a PWM output pin using**analogWrite()** It's

also good for driving servos because its a high-speed PWM output

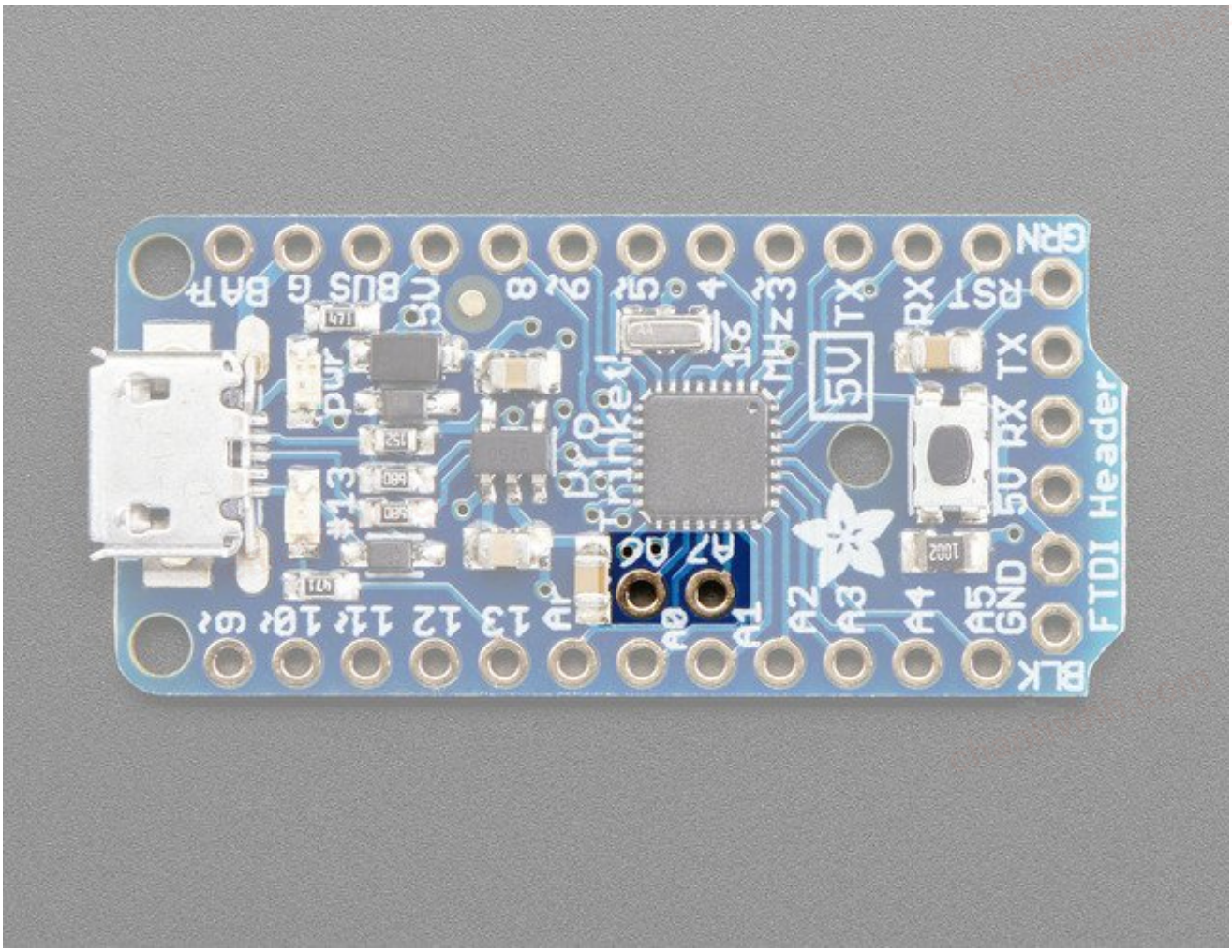
- **Digital 11** - Also known as the SPI **MOSI** pin. this pin can also be used as a PWM output pin using **analogWrite()**
- **Digital 12** - Also known as the SPI **MISO** pin
- **Digital 13** - Also known as the SPI **CLOCK** pin. This is *also* connected to the red **#13** LED!



The Analog/Digital GPIO Pins

- **Analog 0** - also known as **Digital 14**, this pin can be a digital I/O pin or an analog input pin
- **Analog 1** - also known as **Digital 15**, this pin can be a digital I/O pin or an analog input pin
- **Analog 2** - also known as **Digital 16**, this pin can be a digital I/O pin or an analog input pin
- **Analog 3** - also known as **Digital 17**, this pin can be a digital I/O pin or an analog input pin

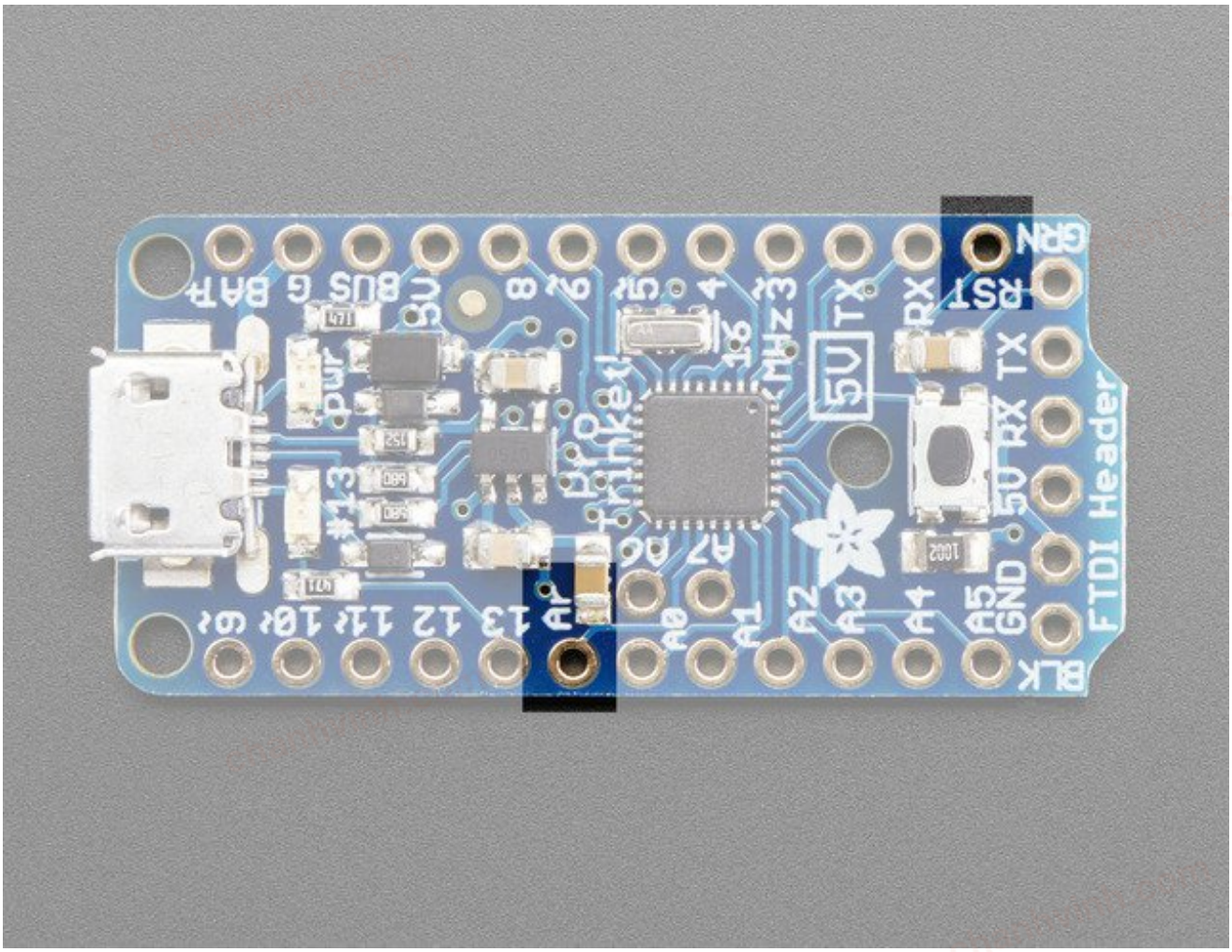
- **Analog 4** - also known as **Digital 18**, this pin can be a digital I/O pin or an analog input pin. It's also the I2C **SDA** pin
- **Analog 5** - also known as **Digital 19**, this pin can be a digital I/O pin or an analog input pin. It's also the I2C **SCL** pin



The Analog only pins

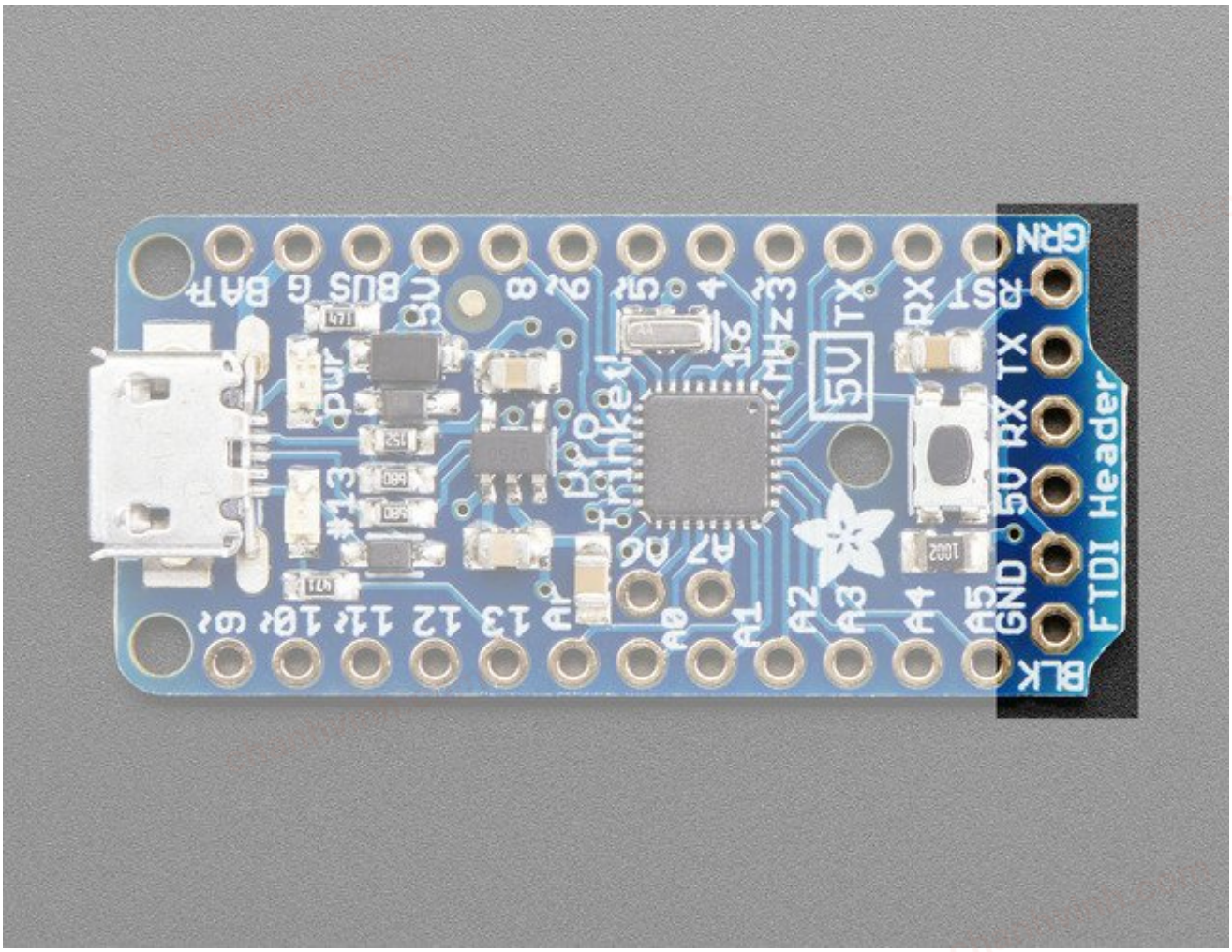
The two pins that sit sort-of inside the body of the Pro Trinket, **A6** and **A7**, are analog input only pins. They're kind of an 'extra' that you get with the Pro Trinket. These pins **cannot** be used for LEDs, buttons, servos, etc. They're only for **analogRead()** usage!

If you're used to using an Arduino Uno, you may notice that pins **#2** and **#7** are not available. That's because we use those two pins for the USB bootloader. They are not available for use and are not broken out.



Other Pins

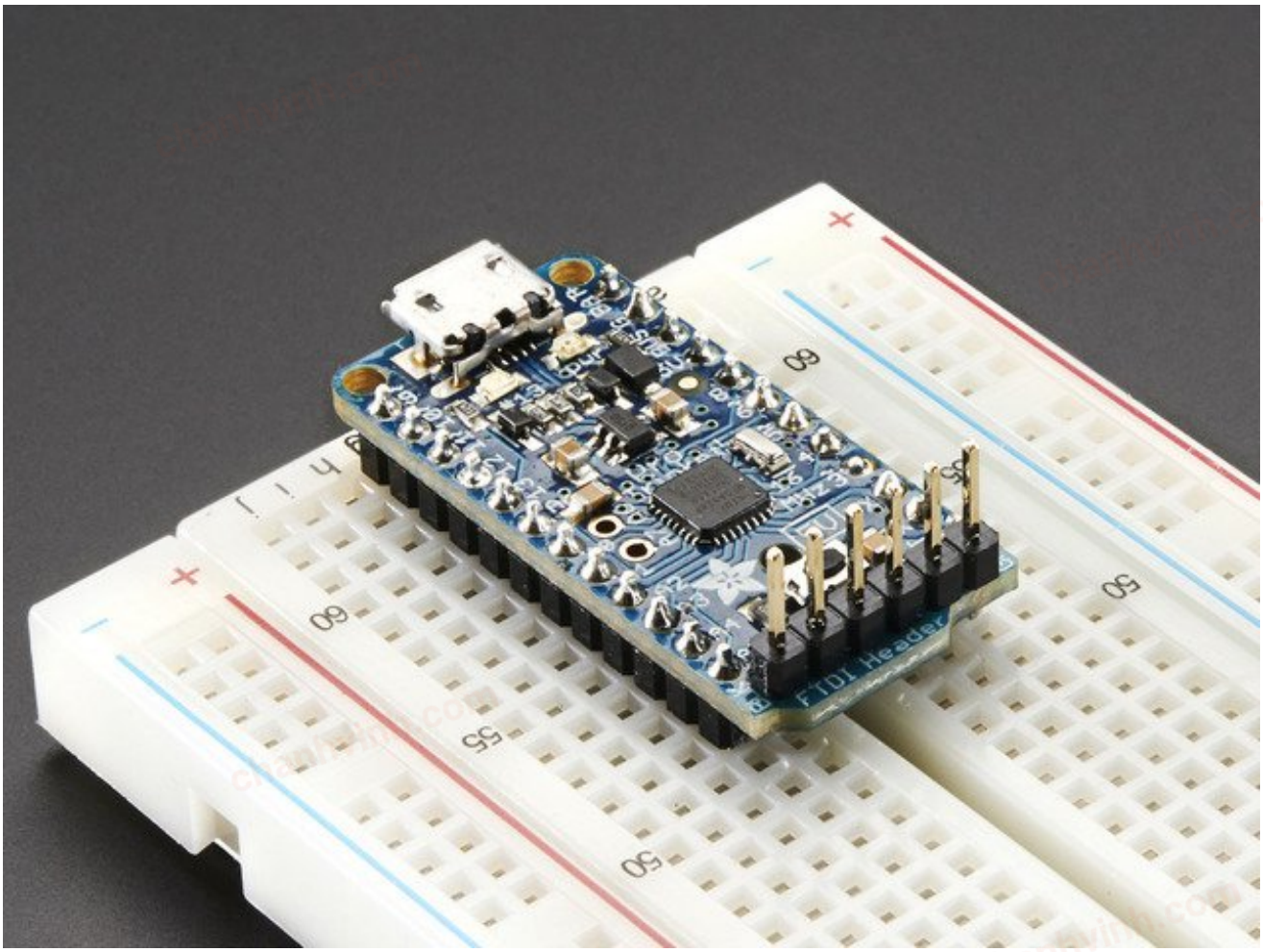
- **Aref** - this is the optional analog reference pin for the analog converter, to be used when you want the 'top' of the analog converter to be different than 3V or 5V. Connect this to your desired reference voltage (between 0 and the Pro Trinket voltage) and use **analogReference(EXTERNAL)**
- **RST** - This is the Reset pin for the Pro Trinket. Connecting this to ground momentarily resets the Trinket and also starts up the bootloader.



FTDI Breakout

At the bottom is 6 pins in a row that we call the FTDI header. You can use these pins to program the Pro Trinket with an FTDI cable or FTDI Friend. You can also use it for serial debugging.

- **GND** - (Black **BLK** wire on FTDI Cable) same as the power ground pins
- **GND** - ditto
- **5V** - same as the VBUS pin
- **RX** - same as the RX pin
- **TX** - same as the TX pin
- **RST** - (Green **GRN** wire on FTDI Cable) there's a 0.1uF capacitors in between this pin and the Reset pin, so that Arduino-compatible programmers will work correctly with the DTR/RTS-toggle technique


FTDI Header

Windows Driver Installation

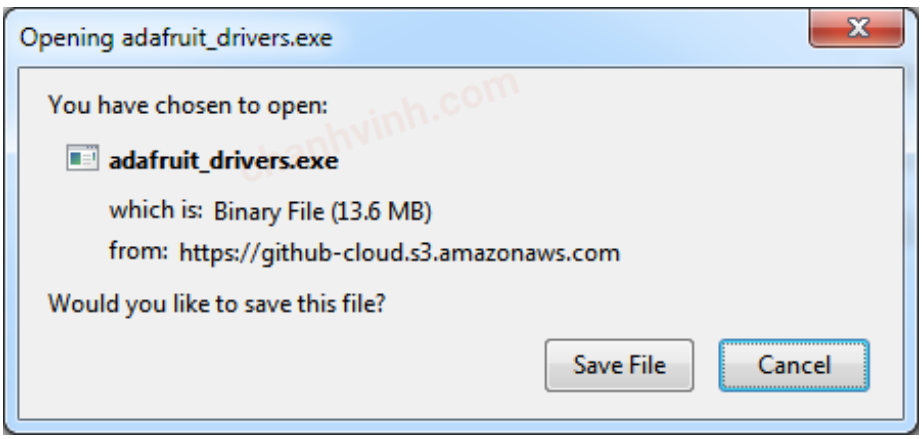
Mac and Linux do not require drivers, only Windows folks need to do this step

Before you plug in your board, you'll need to possibly install a driver!

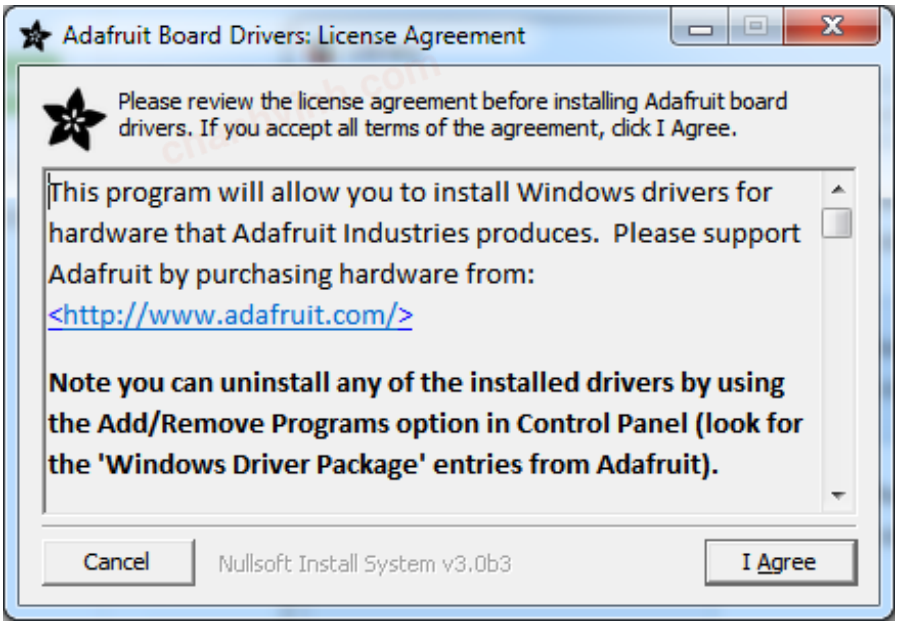
Click below to download our Driver Installer

[Download Adafruit Windows Driver Installer](http://adafru.it/mai)
http://adafru.it/mai

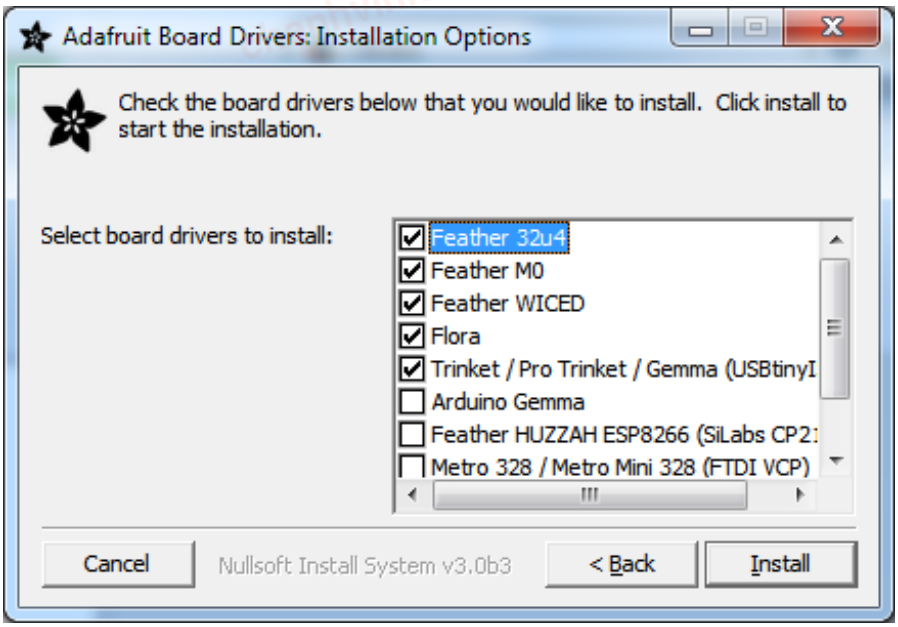
Download and run the installer



Run the installer! Since we bundle the SiLabs and FTDI drivers as well, you'll need to click through the license



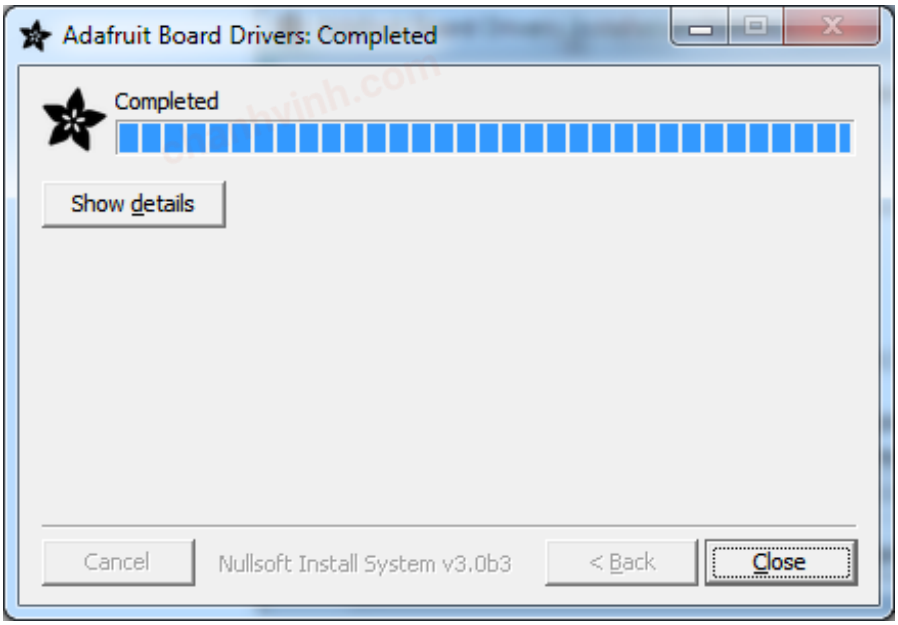
Select which drivers you want to install, we suggest selecting all of them so you don't have to do this again!



By default, we install the **Feather 32u4** , **Feather M0**, **Flora** and **Trinket** / **Pro Trinket** / **Gemma** / **USBtinyISP** drivers.

You can also, optionally, install the **Arduino Gemma** (different than the Adafruit Gemma!), Huzzah and Metro drivers

Click **Install** to do the installin'



chanhvinh.com

Using the USB Bootloader

About the bootloader

A *bootloader* is a tiny piece of software residing on the microcontroller that that helps load your own code into the remaining space.

One of the challenges with the Pro Trinket is that we wanted to have a built-in USB bootloader, but the ATmega328 doesn't have built-in USB hardware!

So instead, Frank (our awesome engineer with mad USB chops)(http://adafru.it/cDX) created a USB bootloader that combines the elegance of V-USB with the well-supported and tested nature of the USBtinyISP. This bootloader looks just like a USBtinyISP - and since it uses the unique Adafruit VID/PID we own and that we added to avrdude so long ago, it works with only very minimal configuraton tweaks. No need to recompile anything, whew!

Then, since we had a little space left in the bootloader area of the chip, we added Optiboot support. You can use *either* the USB connector *or* an FTDI cable so that it acts just like an Arduino Uno for programming, for max flexibility.

The Pro Trinket is smart enough to auto-detect which bootloader you are trying to use. If you plug the USB port in, it will enumerate as a USBtinyISP for USB bootloader. If you plug in FTDI and use the Arduino Uno programming procedure it will use that technique

Please note: you cannot use the Adafruit USB VID/PID for your own non-Trinket products or projects. Purchase a USB VID for yourself at http://www.usb.org/developers/vendor/

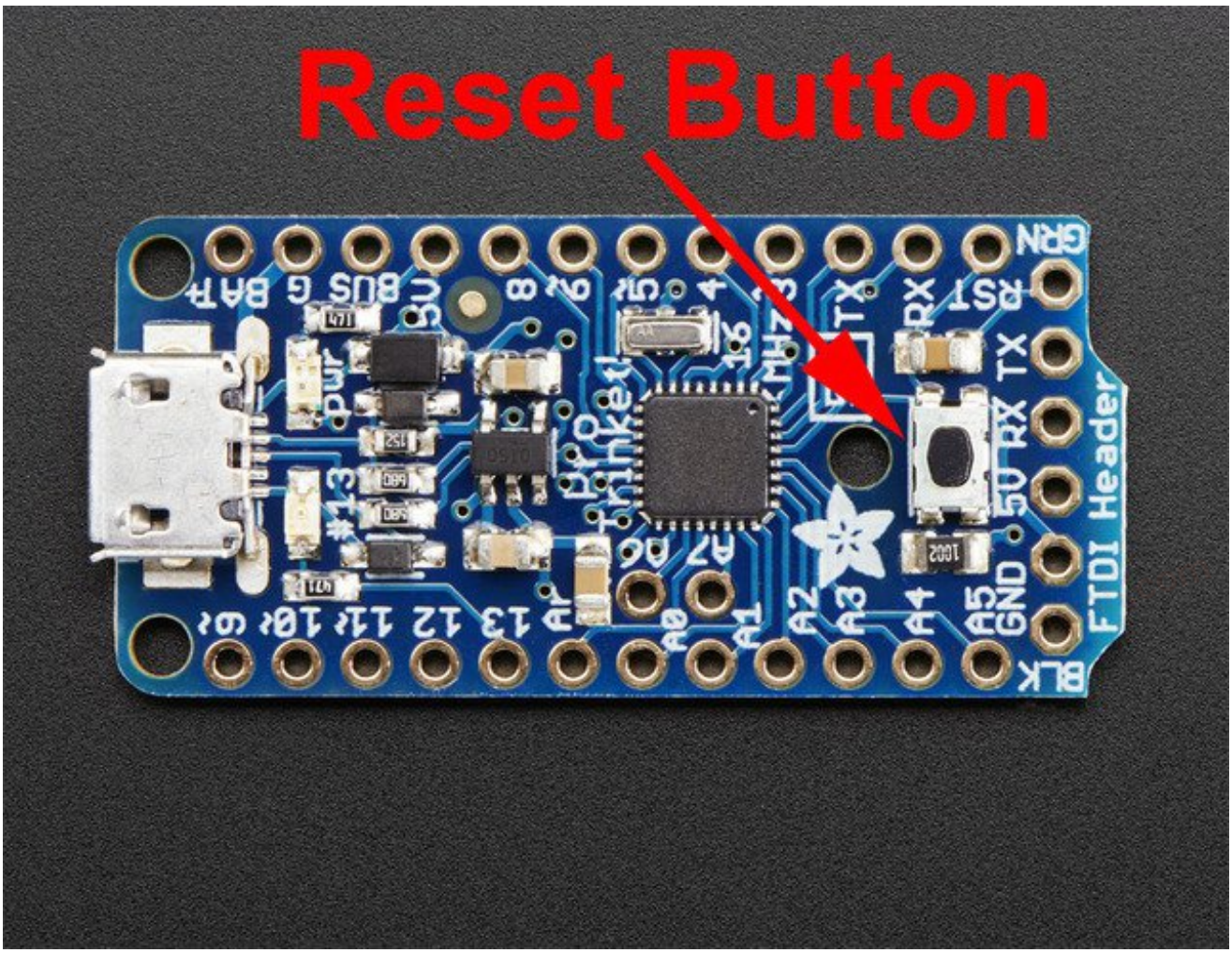
Pro Trinket USB Drivers for Windows

The cool thing about the bootloader on the Pro Trinket is it just looks like a classic USBtinyISP AVR programmer. This makes it easy to use with AVRdude or Arduino IDE with only minor configuration changes. Before you start, you may need to install the USBtinyISP USB drivers

Drivers are only required for Windows, if you are using a Mac or Linux, drivers are not required!

Check this page for how to install the driver(http://adafru.it/mao)

Don't forget to plug in the Pro Trinket via a known-good USB cable to start the process. You should see the green power LED lit and the red bootloading LED pulse indicating that the Trinket is ready to start bootloading. If you've programmed the Pro Trinket since getting it, you can always get it back to the bootloader state by pressing the small onboard reset button



Special Notes on using Pro Trinket with Linux

Pro Trinket is not supported on Linux operating system at this time - try Mac OS or Windows! However, you can try the following - it does work for some computers (http://adafru.it/id3)

Linux is fairly picky about who can poke and prod at the USB port. You can always run **avrdude** or **Arduino IDE** as root, which will make sure you have the proper permissions. If

you want to be super-cool you can add a *udev* rule which will let any user (who is not root) connect to the USBtiny driver. That way you don't have to be root all the time!

Check http://learn.adafruit.com/usbtinyisp/avrdude#for-linux (http://adafru.it/cf3) for what to add to your udev file.

How to start the USB bootloader

Before you try to upload code to the Pro Trinket using the USB bootloader it must be in the Bootloader Mode. That means its listening for a sketch or program to be sent to it

When the Pro Trinket is in bootloader mode, the red LED will be pulsing. Once the red LED stops pulsing, you must press the reset button to re-enter bootloader mode

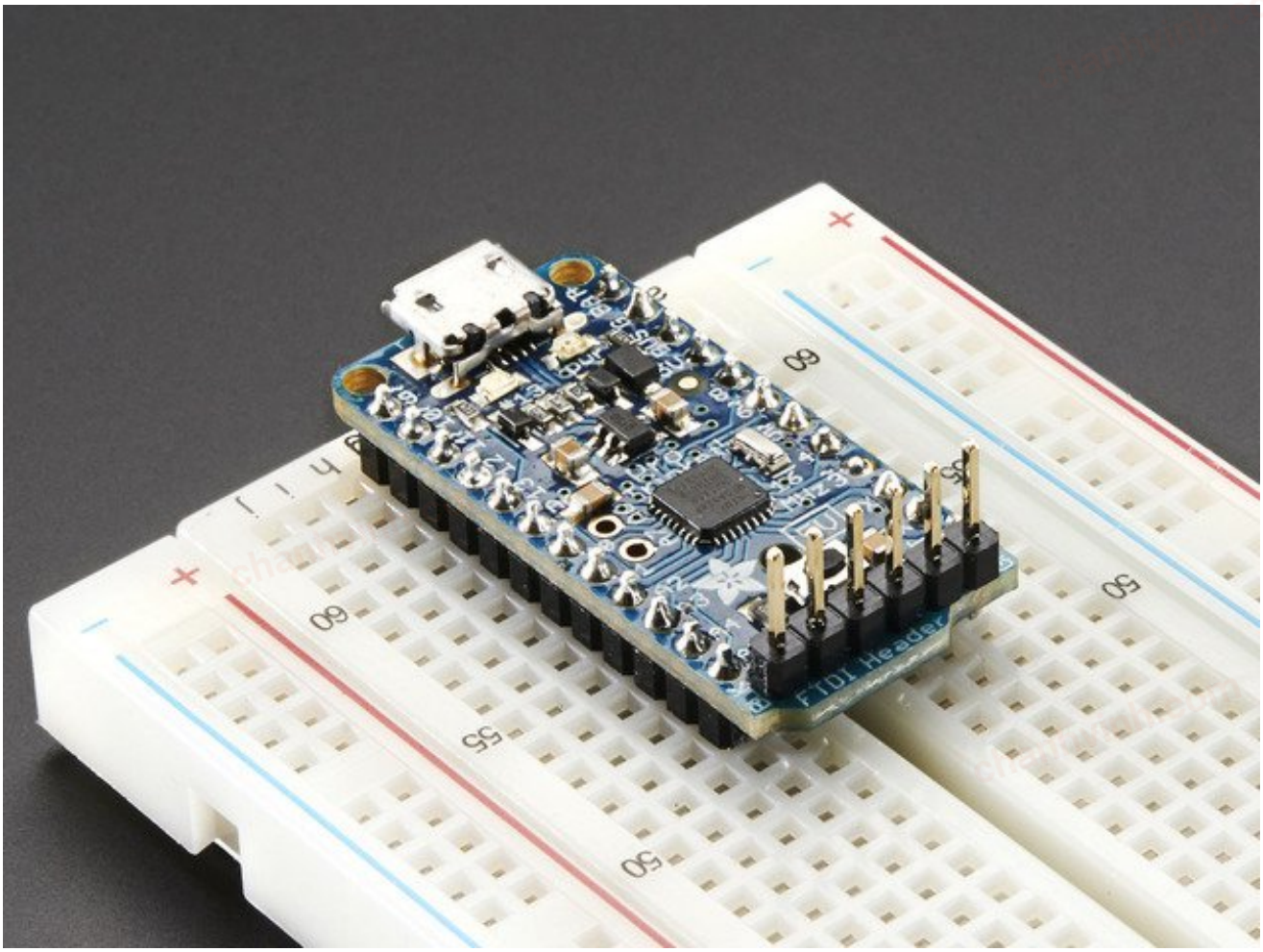
The Pro Trinket must be connected to a computer via a USB cable to enter bootloader mode. You can enter the bootloader mode by pressing the little button on the board with your fingernail. The bootloader will 'time out' after 10 seconds, so to re-enter the bootloader mode just re-press the button!

Don't press-and-hold the reset button, be sure to press-and-release!

See the video below for what it looks like to plug it in, have the LED pulse in bootloader mode, time out and then press reset to restart the bootloader

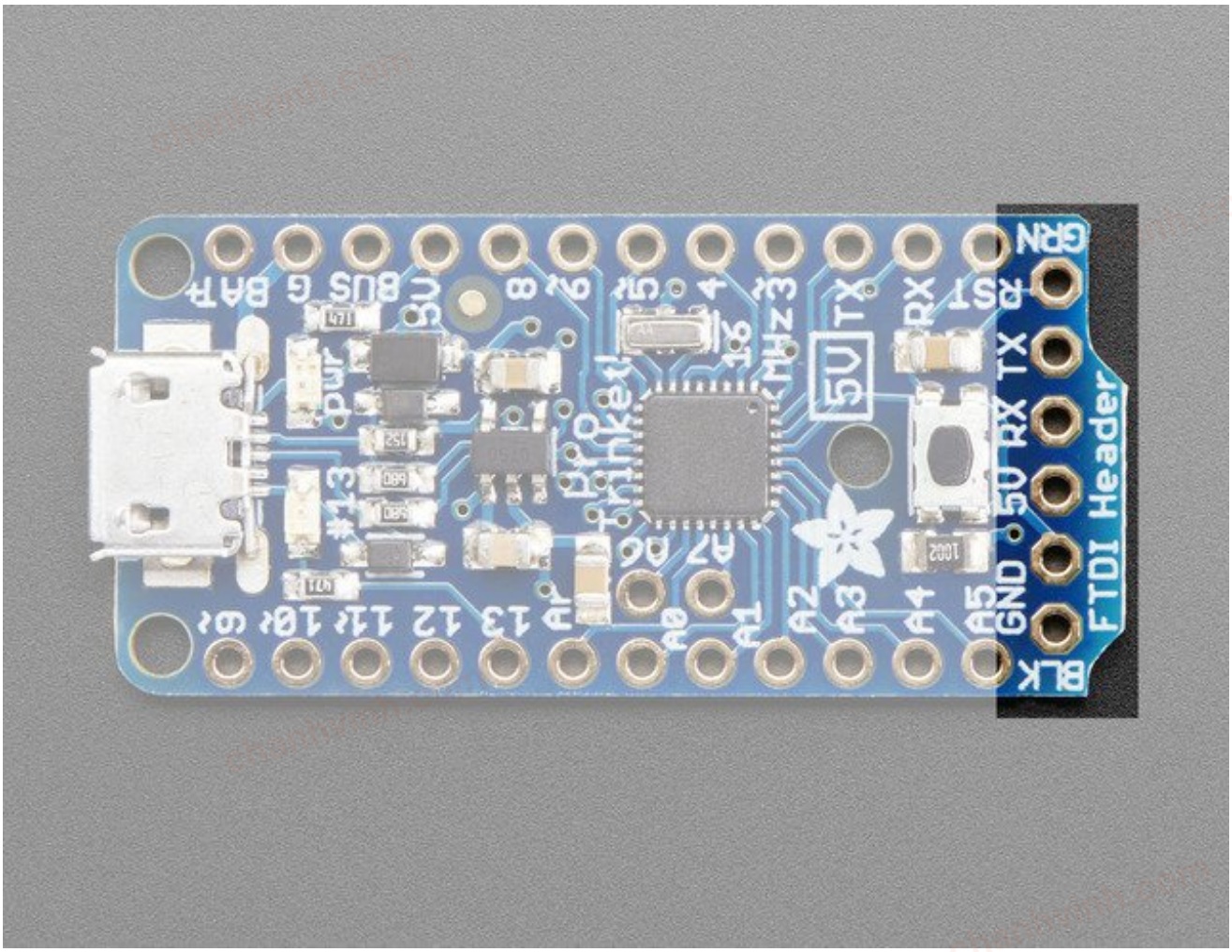
chanhvinh.com

Using FTDI Cables





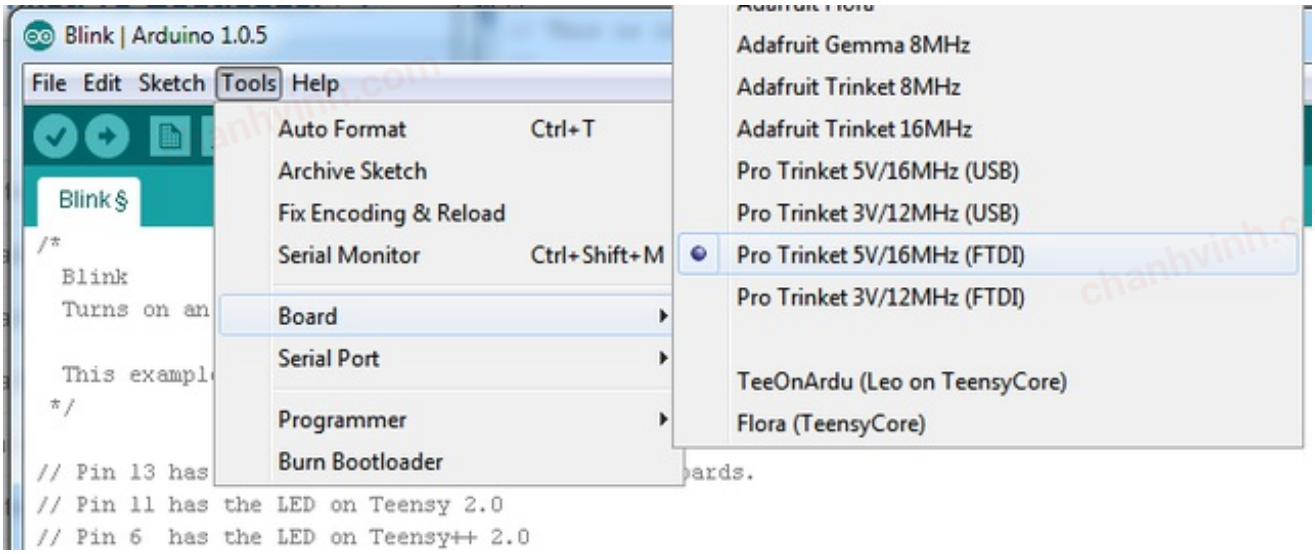
For beginners, we suggest using an FTDI cable for programming and debugging, especially if you aren't simply porting an existing Arduino project to Pro Trinket. Much like many Arduino-compatibles, there's a header for connecting a 'classic' FTDI cable (http://adafru.it/dNN) or FTDI friend. (http://adafru.it/dQa)



The bootloader on Pro Trinket is **dual USB & FTDI** "optiboot" so you can use either the USB port or the FTDI breakout to upload. However, only the FTDI port allows you to use Serial debugging. So it's a bit of a tradeoff - USB is inexpensive, built-in but requires a button press to start & there's no debugging, FTDI requires an extra cable but has auto-reset and debug console.

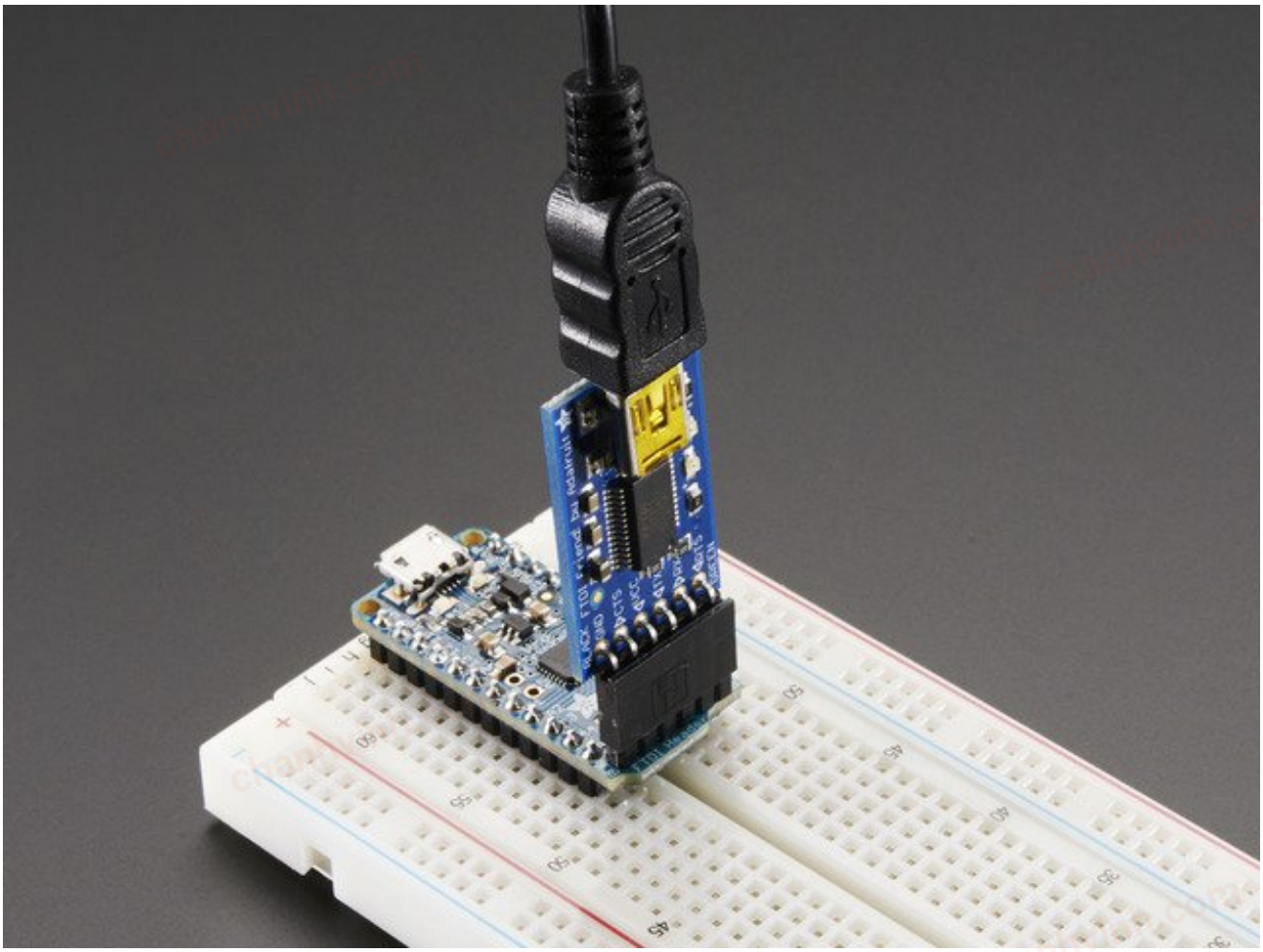
Select the correct Board in IDE

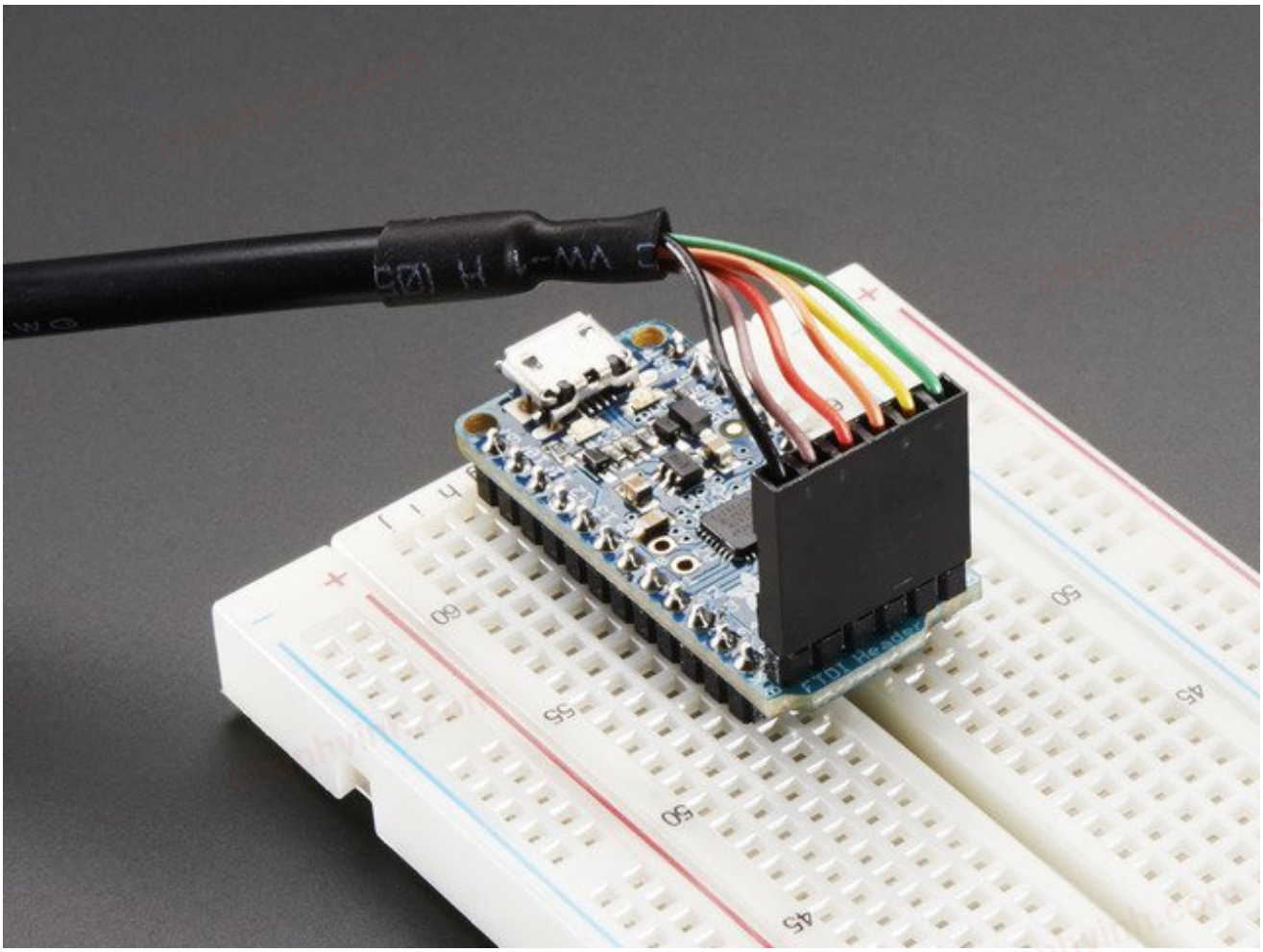
Make sure you're selecting the FTDI-program type in the Boards menu!



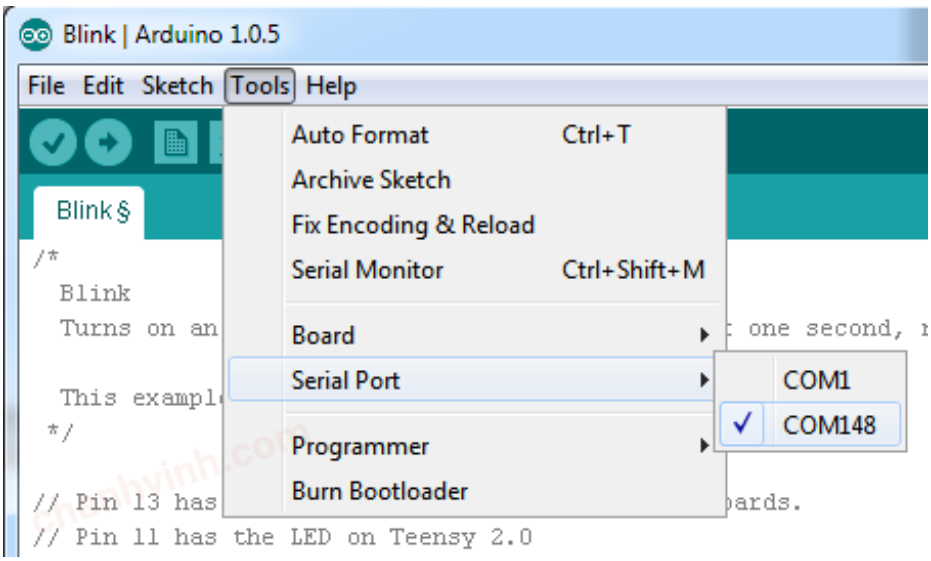
Connect FTDI Friend or Cable

You'll need to solder 6-pin header onto the FTDI breakout first. Match the black wire and green wire with the BLK and GRN markings on the PCB next to the FTDI header





Don't forget you'll need to install FTDI VCP drivers, available from FTDI chip. (http://adafru.it/aJv) Then select the COM port from the Tools menu as well, to upload!



Of course, you can use the FTDI cable for debugging not just programming!

To use the FTDI cable to debug programs, add the standard `Serial.begin(baudrate);` line as you would with an UNO and `Serial.print` or `Serial.println` as you expect! Then select **Tools** / **Serial monitor** after loading your program. The Trinket will re-start when you launch the monitor so you can see your data right from the start; be patient while the Trinket finishes starting.

chanhvinh.com

Setting up Arduino IDE

Chances are, you picked up a Trinket because it is programmable with the Arduino IDE. The Pro Trinket uses the same core chip as Arduino UNO, and the 5V Pro Trinket uses the same clock rate and voltage so its pretty much a full Arduino-compatible, just smaller!

The only changes you may have to consider when adapting Arduino sketches are:

- Pins #2 and #7 are not available
- The onboard 5V regulator can provide 150mA output, not 800mA out
- You cannot plug shields directly into the Pro Trinket
- There is no Serial-to-USB chip onboard. This is to keep the Pro Trinket small and inexpensive, you can use any FTDI cable to connect to the FTDI port for a Serial connection
- The bootloader on the Pro Trinket use 4KB of FLASH so the maximum sketch size is 28,672 bytes. The bootloader does not affect RAM usage.

Even though Trinket has a USB connector, it does not have a "Serial Console" capability, so you cannot use Serial to send and receive data to/from a computer without a seperate FTDI Cable!

Arduino IDE Setup

Just follow the steps in the **Adafruit Arduino IDE setup guide (http://adafru.it/jAc)** to add Adafruit board support to the Arduino IDE to program Pro Trinket!

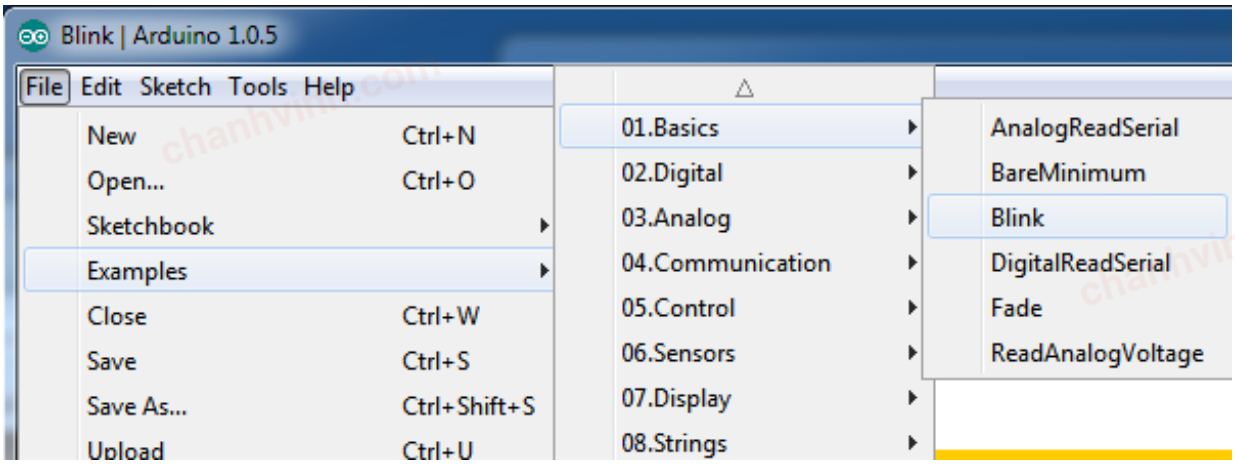
Don't forget to also install Windows drivers if you are using Windows (http://adafru.it/jAa), or setup udev on Linux (http://adafru.it/iOE)

When you're finished installing the IDE come back to this page to continue the Pro Trinket guide.

Blink!

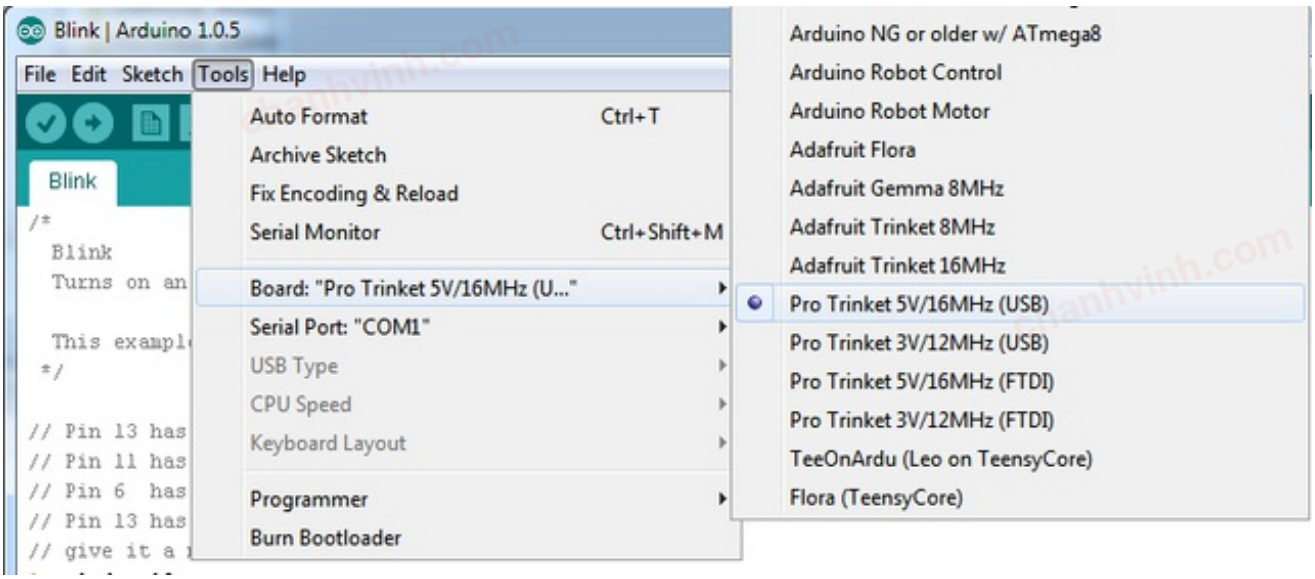
OK let's practice uploading a sketch by uploading the classic Blink sketch with the USB bootloader. This program with blink the onboard red LED that is connected to pin #13.

The sketch is in **File->Examples->Basics->Blink**

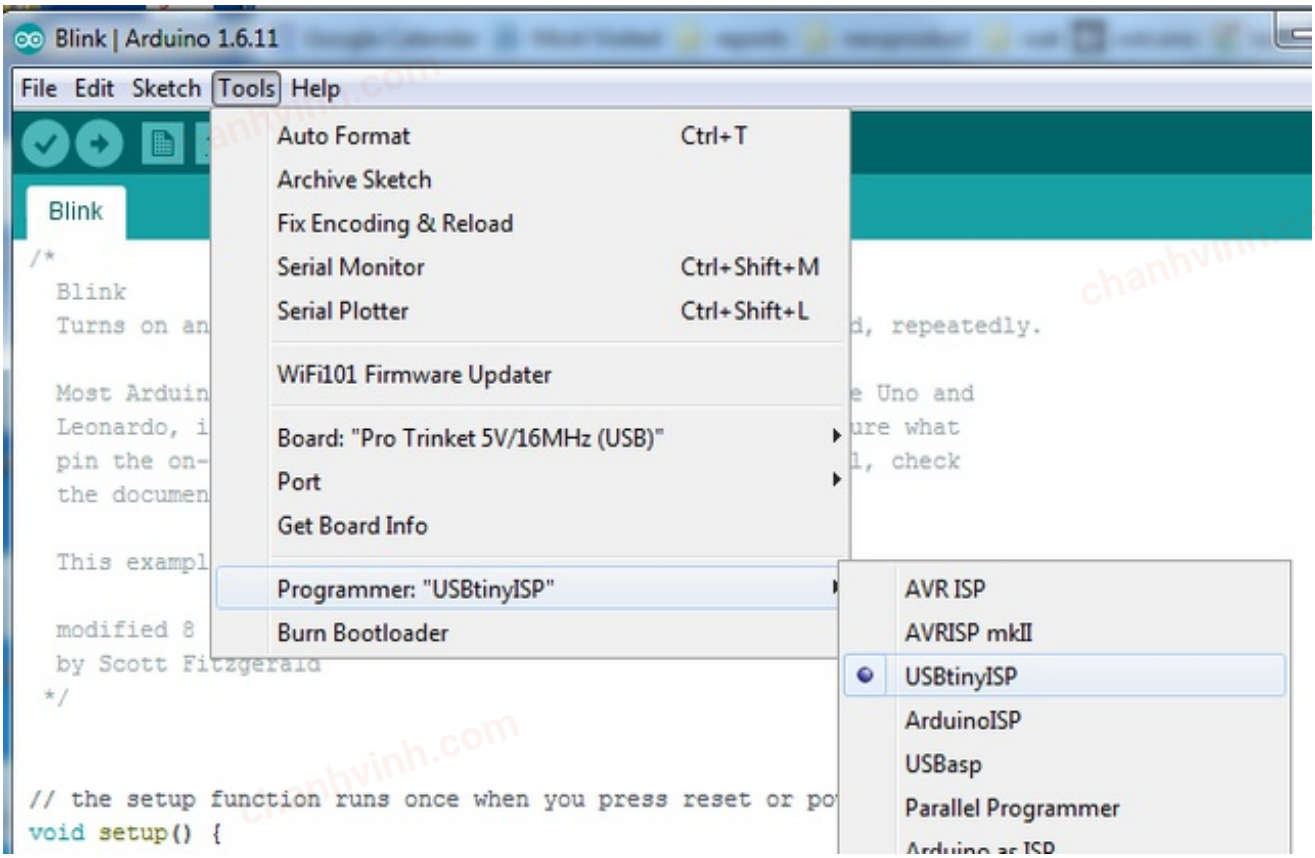


If you are using Linux you may have to be "root" running the Arduino program to have access to the USB port

Select the **Pro Trinket 5V/16MHz (USB)** or **Pro Trinket 3V/12MHz (USB)** board from the **Tools->Board** menu

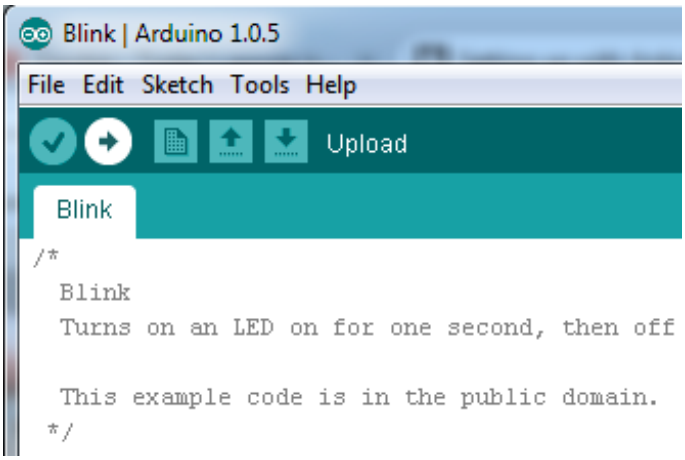


Next go into the **Tools -> Programmer** menu and select the **USBtinyISP** programmer.

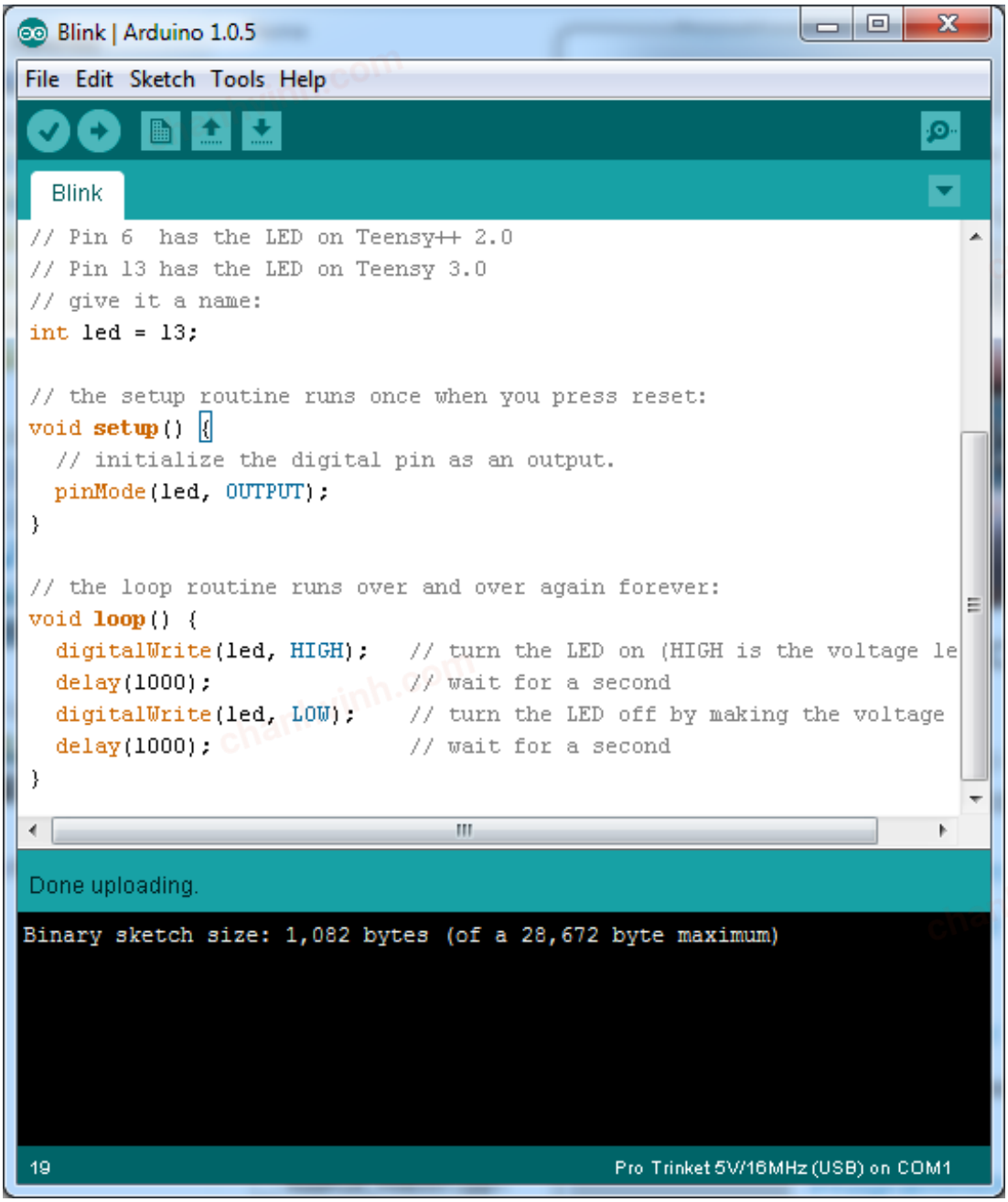


Plug in the Trinket, make sure you see the green LED lit (power good) and the red LED pulsing. Press the button if the red LED is not pulsing, to get into bootloader mode.

Click the **Upload** button (or select **File->Upload**)



If everything goes smoothly you should see the following (no red error messages) and of course, the red LED on the trinket will blink on/off once a second

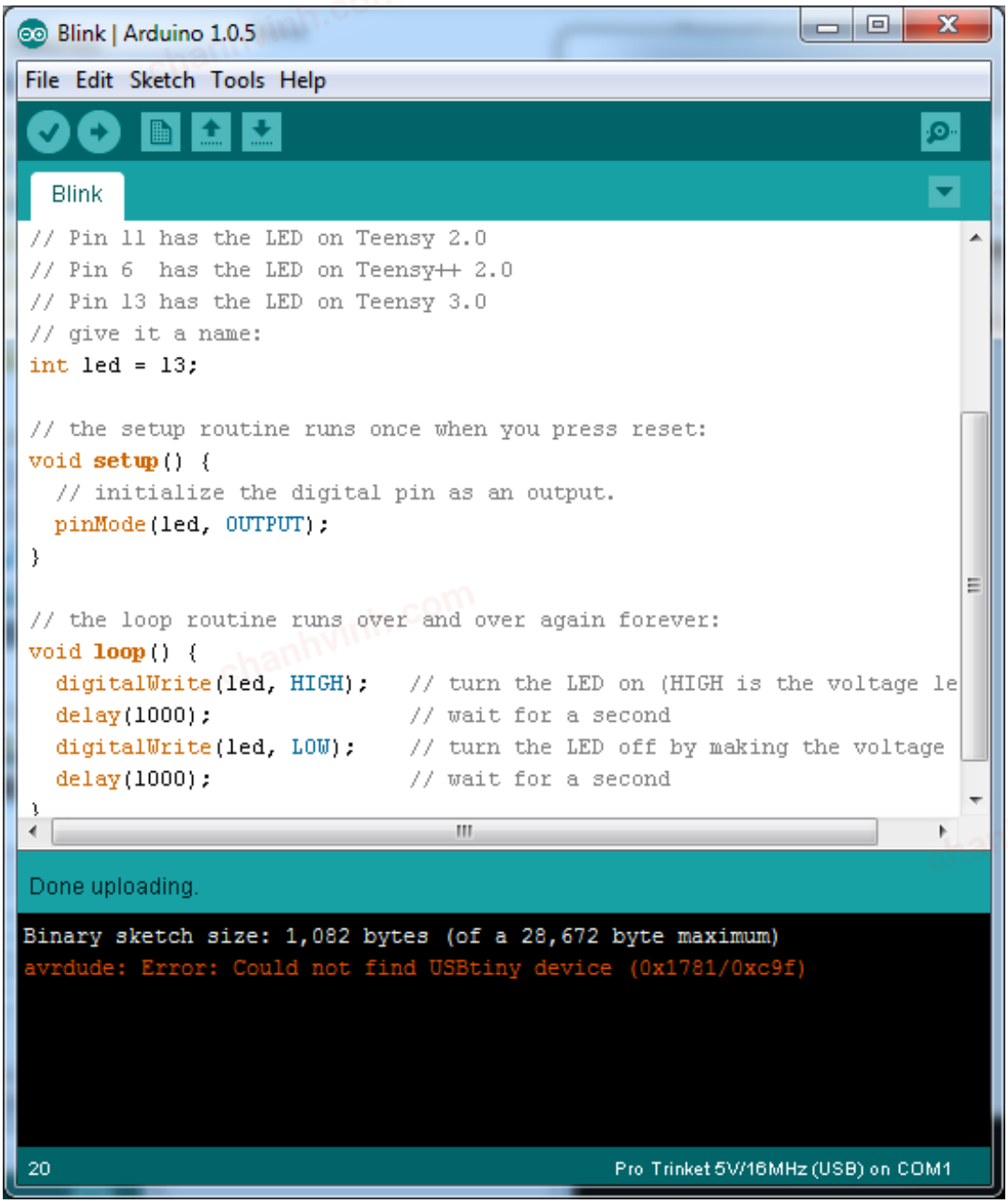


Something Went Wrong!

If you get the error message avrdude: Error: Could not find USBtiny device (0x1781/0xc9f)

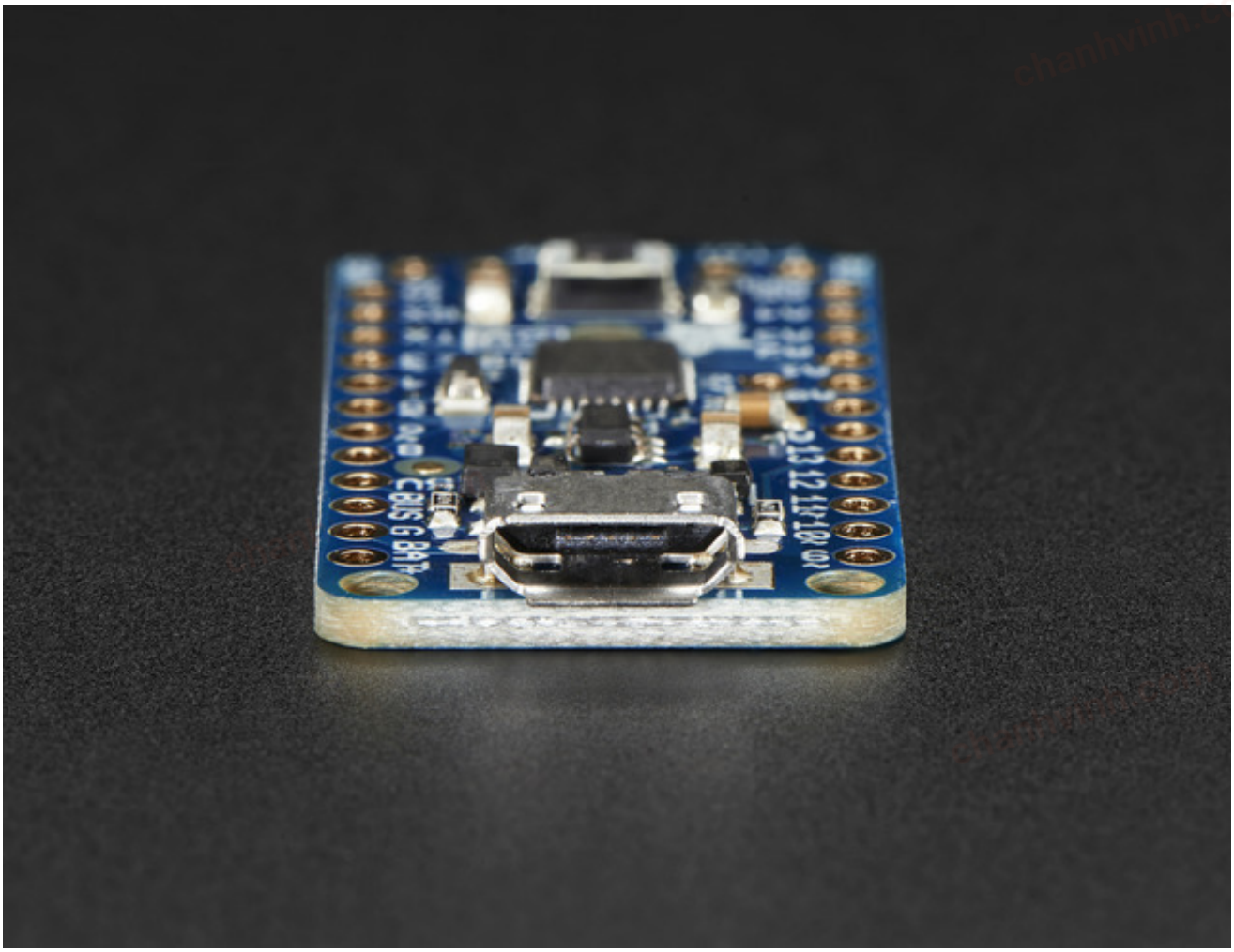
That means the bootloader wasn't active. Make sure to press the button on the Trinket to activate the bootloader *before* clicking the Upload button. Also, check that you see the red

pulsing LED on #13 that indicates the USB bootloader enumerated to your computer



Also, don't forget to install the Windows driver if you're using windows!

Hints for Arduino-compatibility



The Pro Trinket uses the Atmega328P chip, which is the same core chip in the Arduino UNO/Duemilanove/Mini/etc. It's the 'classic' Arduino chip! So you'll be happy to hear that not only is Pro Trinket programmable using the Arduino IDE as you already set up, but 99% of Arduino projects will work out of the box!

The Pro Trinket 5V runs at 16MHz, just like the Uno. The Pro Trinket 3V runs at 12MHz, which we suggest for 'power users' who want a 3V Arduino-compatible and don't mind its a bit slower at 12MHz.

The only changes you may have to consider when adapting Arduino sketches are:

- Pins #2 and #7 are not available
- That means you do not get access to interrupt #0. Use interrupt #1 on pin #3 instead

with attachInterrupt (1, functionname, condition);

- The onboard 3.3V or 5V regulator can provide 150mA output, not 800mA out
- You cannot plug shields directly into the Pro Trinket
- There is no Serial-to-USB chip onboard. This is to keep the Pro Trinket small and inexpensive, you can use any FTDI cable to connect to the FTDI port for a Serial connection
- The 3V Pro Trinket runs at 12MHz not 16MHz so its a bit slower.
- The bootloader on the Pro Trinket use 4KB of FLASH so the maximum sketch size is 28,672 bytes. The bootloader does not affect RAM usage.

chanhvinh.com

Re-programming Bootloader

If you'd like to reprogram the bootloader on your Pro Trinket, or update the bootloader to not have the Optiboot-loop-on-RX bug, download this zip of two Arduino sketches

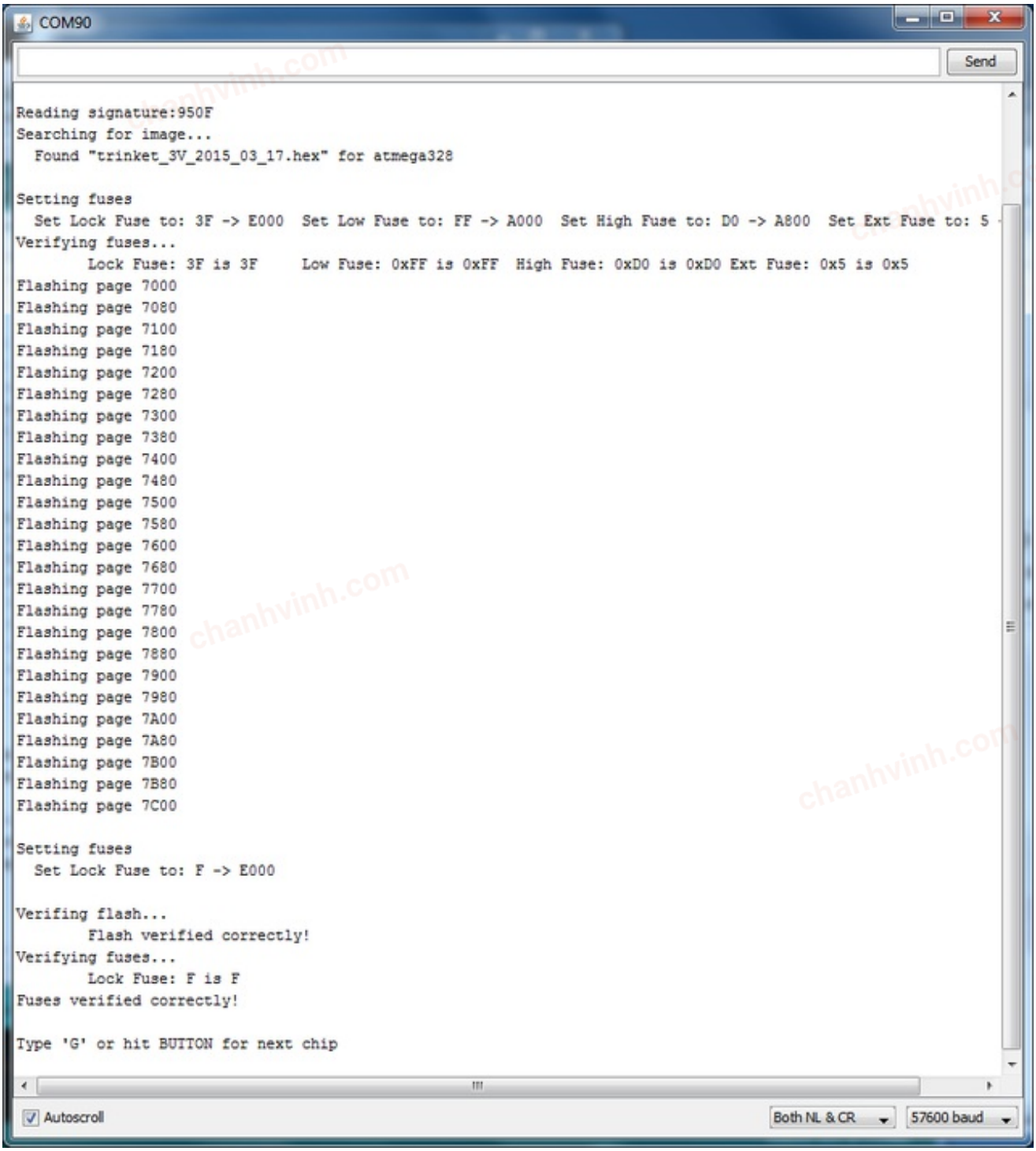
These are only tested on an UNO! For non-UNOs you'll need to swap pins 11, 12, 13 for the hardware SPI pins but we haven't tested it at all, so UNO only please.

protrinketbootloaders_2015_06_09.zip
http://adafru.it/fmh

Load one or the other onto the Uno, depending on which type you are programming, then connect

- Pro Trinket 13 -> Arduino 13 (Use ~1K series resistor if prog'ing Pro Trinket 3V)
- Pro Trinket 12 -> Arduino 12
- Pro Trinket 11 -> Arduino 11 (Use ~1K series resistor if prog'ing Pro Trinket 3V)
- Pro Trinket GND -> Arduino GND
- Pro Trinket VBAT -> Arduino 5V
- **Pro Trinket RESET -> Arduino 10**(Use ~1K series resistor if prog'ing Pro Trinket 3V)

open the terminal @ 57600 baud and press G and return (or click Send)


COM90
Send
Reading signature:950F
Searching for image...
Found "trinket_3V_2015_03_17.hex" for atmega328
Setting fuses
Set Lock Fuse to: 3F -> E000 Set Low Fuse to: FF -> A000 Set High Fuse to: D0 -> A800 Set Ext Fuse to: 5
Verifying fuses...
Lock Fuse: 3F is 3F Low Fuse: 0xFF is 0xFF High Fuse: 0xD0 is 0xD0 Ext Fuse: 0x5 is 0x5
Flashing page 7000
Flashing page 7080
Flashing page 7100
Flashing page 7180
Flashing page 7200
Flashing page 7280
Flashing page 7300
Flashing page 7380
Flashing page 7400
Flashing page 7480
Flashing page 7500
Flashing page 7580
Flashing page 7600
Flashing page 7680
Flashing page 7700
Flashing page 7780
Flashing page 7800
Flashing page 7880
Flashing page 7900
Flashing page 7980
Flashing page 7A00
Flashing page 7A80
Flashing page 7B00
Flashing page 7B80
Flashing page 7C00
Setting fuses
Set Lock Fuse to: F -> E000
Verifing flash...
Flash verified correctly!
Verifying fuses...
Lock Fuse: F is F
Fuses verified correctly!
Type 'G' or hit BUTTON for next chip
Autoscroll
Both NL & CR
57600 baud

F.A.Q.

I'm trying to use Software Serial on the Pro Trinket 3V and getting compiler errors

If you're running an older version of the IDE, there may not be support for 12MHz software serial. Find your Arduino installation folder (if you're using a Mac you may need to **explore** the package) and locate the *libraries/SoftwareSerial* folder and open up **SoftwareSerial.h** inside. Add the following lines **before**

```
#elif F_CPU == 8000000

#elif F_CPU == 12000000
static const DELAY_TABLE PROGMEM table[] =
{
// baud rxcenter rxintra rxstop tx
  { 115200, 1, 11, 11, 8, },
  { 57600, 6, 26, 26, 23, },
  { 38400, 14, 41, 42, 39, },
  { 31250, 19, 51, 52, 49, },
  { 28800, 23, 56, 56, 53, },
  { 19200, 37, 86, 86, 83, },
  { 14400, 52, 116, 116, 113, },
  { 9600, 82, 175, 175, 173, },
  { 4800, 172, 354, 354, 351, },
  { 2400, 350, 711, 711, 708, },
  { 1200, 707, 1425, 1425, 1422, },
  { 300, 2850, 5711, 5711, 5708, },
};

const int XMIT_START_ADJUSTMENT = 5;
```

Once added, save and restart the IDE

My Pro Trinket sometimes doesn't "boot" or "run code" on power up

There's a slight bug in the Optiboot bootloader where it has a busy-wait loop while reading commands. If there is some noise on the RX pin, it can start the bootloader which then waits forever for more data

You can solve this in one of two ways

1. Connect a pullup resistor of anywhere from 2K - 47K from the RX pin to the 5V pin (Pro Trinket 5V) or 3V pin (Pro Trinket 3V)
2. Re-burn the bootloader with our patched version that seems to fix this bug by timing

out on noisy RX reads

I've tried everything, but I still get the "Could not find USBtiny device" error

Try a different USB cable. Some cables are "charge only" and don't have data lines.

And even with data cables, some work better than others.

Can I use an EZ-Link to program a Pro Trinket?

We haven't personally tried it but some customers report that - Yes you can use anEZ-Link (http://adafru.it/1588) with a 3V or 5V Pro Trinket in FTDI-programming-mode

Downloads

Datasheets & Files

- Datasheet of ATmega328 Microcontroller (http://adafru.it/rmD)
- EagleCAD PCB files for both Pro Trinkets (http://adafru.it/rmE)
- Fritzing objects in the Adafruit Fritzing library (http://adafru.it/aP3)

- 3V Declaration of Conformity (http://adafru.it/rmF)
- 3V (http://adafru.it/rmF) EMC Test Report (http://adafru.it/qna)
- 3V (http://adafru.it/rmF) Statement of Verification (http://adafru.it/qnb)
- 3V (http://adafru.it/rmF) FCC Test Report (http://adafru.it/qnc)
- 3V (http://adafru.it/rmF) IC Test Report (http://adafru.it/qnd)

- 5V Declaration of Conformity (http://adafru.it/qne)
- 5V (http://adafru.it/qne) EMC Test Report (http://adafru.it/qnf)
- 5V (http://adafru.it/qne) Statement of Verification (http://adafru.it/qnA)
- 5V (http://adafru.it/qne) FCC Test Report (http://adafru.it/rna)
- 5V (http://adafru.it/qne) IC Test Report (http://adafru.it/rnb)

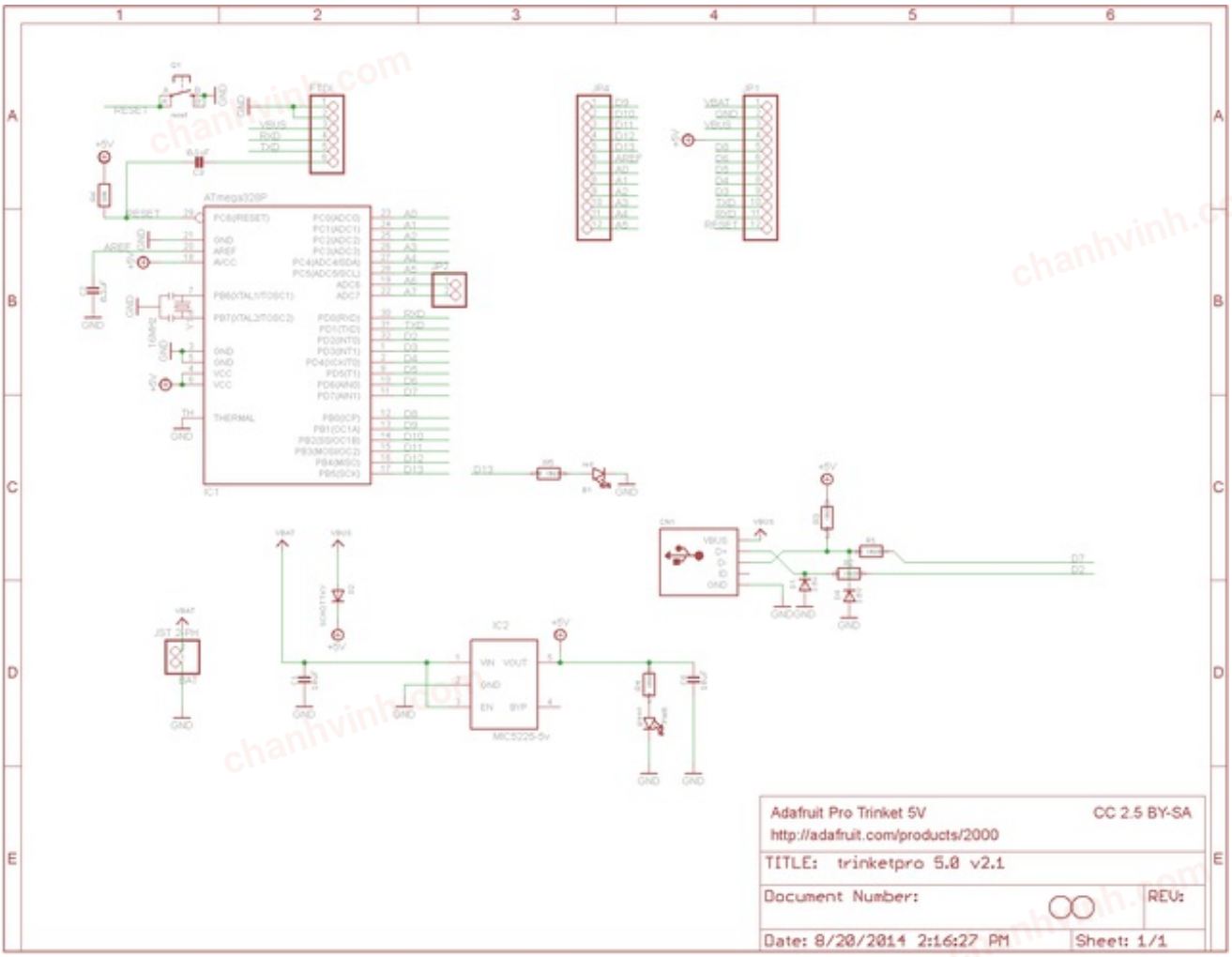
Source code

Original code for the Pro Trinket bootloader on github (http://adafru.it/dRt)
We do not offer any support for this code, it is released as-is!

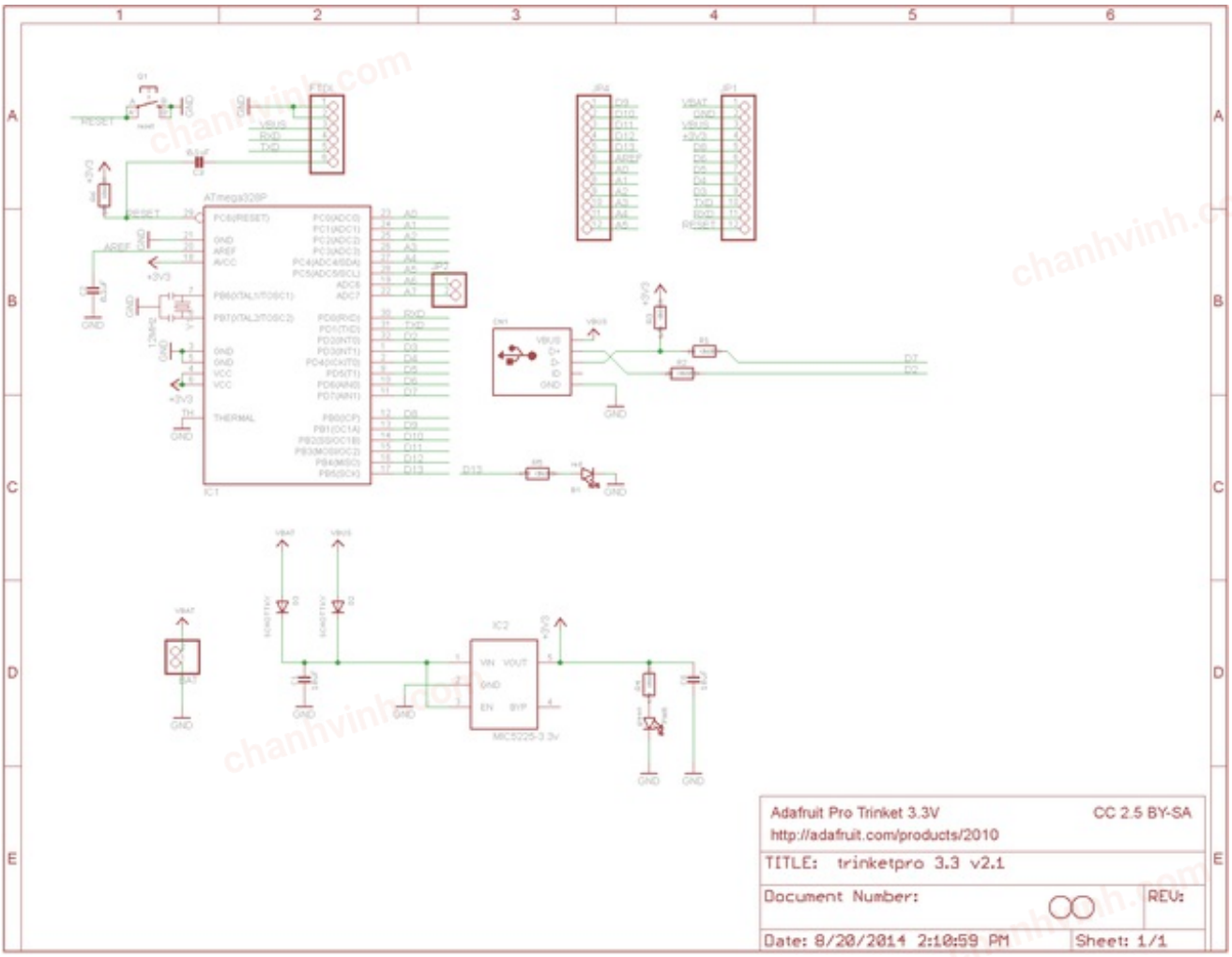
Please note: you cannot use the Adafruit USB VID/PID for your own non-Trinket/Gemma products or projects. Purchase a USB VID for yourself at http://www.usb.org/developers/vendor/

Schematics

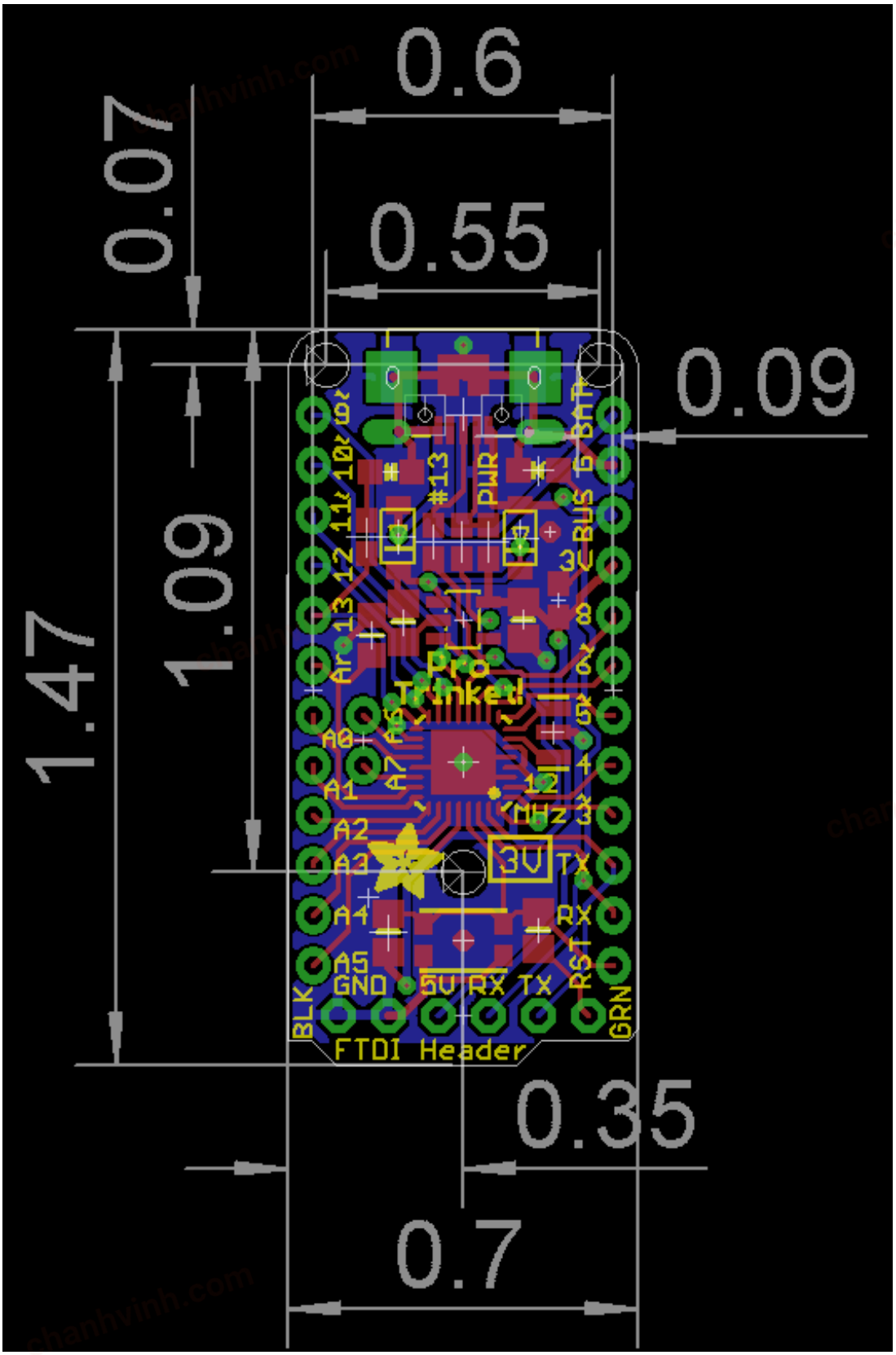
Pro Trinket 5V 16MHz:



Pro Trinket 3V @ 12MHz



Fabrication print, dimensions in inches (same for both 3V and 5V)


0.6
0.07
0.55
0.09
1.47
1.09
0.35
0.7
Pro Trinket
FTDI Header
BLK
GRN
GND
5V
RX
TX
RST
PWR
3V
A0
A1
A2
A3
A4
A5
A6
A7