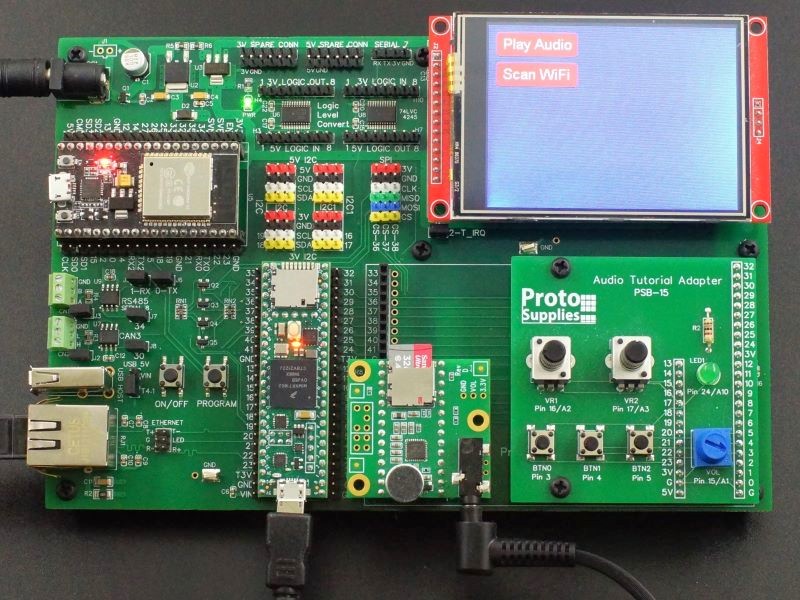
[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Audio-Tutorial-Setup.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Fully-Stuffed-with-Accessories-1.jpg)

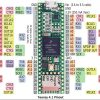
[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Fully-Stuffed-with-T4.1-2.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Audio-Tutorial-Setup.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Fully-Stuffed-in-Operation-with-Mouse.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Baseboard-2.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Bottom.jpg)

[](https://protosupplies.com/wp-content/uploads/2020/08/Teensy-4.1-Pinout-2.jpg)

Prototyping System For Teensy 4.1

$69.95 – $149.95

Platform for working with the powerful Teensy 4.1

Top of Form

|  |  |
| --- | --- |
| **OPTIONS** |  |

ADD TO CART

Add to Wishlist

Bottom of Form

SKU: N/ACategories: [Arduino, Teensy & IOT](https://protosupplies.com/product-category/development-boards/), [PCBs - Prototype & Adapter](https://protosupplies.com/product-category/proto-and-adapter-pcbs/)Tag: [Teensy](https://protosupplies.com/product-tag/teensy/)

* [DESCRIPTION](https://protosupplies.com/product/prototyping-system-for-teensy-4-1/#tab-description)

**DESCRIPTION**

The Prototyping System for Teensy 4.1 provides a flexible setup for working with the Teensy 4.1 microcontroller, the most powerful Arduino compatible microcontroller available today.

The baseboard provides the core features required in many setups including power input handling, logic level shifting, touch screen LCD for user interface and access to common buses and interfaces such as Ethernet, USB Host, CAN bus, RS485 bus, Serial TTL, I2C, level shifted I2C, SPI, level shifted SPI, WiFi and Bluetooth.

This system can be used as part of an elaborate breadboarding setup with key parts of the system fully wired in like the display while also being able to work with temporary circuits built on solderless or soldered breadboards.  It can also be used as a prototyping step for something that may later have a full custom layout or even used for permanent projects where minimizing footprint is not a high priority.

**It is available in two versions selectable at top of page or you can order below:**

The **Baseboard Only** which is the fully assembled baseboard with all SMD and thru-hole parts installed along with nylon standoffs to provide support when working with it on the bench.  It allows you to build your own setup by selecting from the various options below or combining it with parts already laying on your project bench.  It can also be used to update an older revision system.

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Baseboard-2.jpg)

| **PSA-5**-[Prototyping System For Teensy 4.1 – Baseboard Only](https://protosupplies.com/product/prototyping-system-for-teensy-4-1/?attribute_options=Baseboard+Only)  $69.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

A **Fully Stuffed** version that comes with everything installed and includes all the accessories needed to basically just plug and go.  The ultimate in instant gratification.  Just add the Teensy 4.1 of your choice from the list down below to complete a fully functioning setup.

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Fully-Stuffed-with-Accessories.jpg)

**The Fully Stuffed version includes the following:**

* + Prototyping System for Teensy 4.1 baseboard Ver 1.2
  + 3.2″ TFT color 320×240 RGB SPI display w/ ILI9341 controller and touch w/ XPT2046 controller
  + ESP32-S WiFi/Bluetooth coprocessor
  + Teensy 4.x Rev D Audio Adapter including microphone
  + Solderless Breadboard Adapter
  + 7.5VDC AC/DC adapter
  + In-line DC power switch
  + 32GB Sandisk SD card with PJRC example .WAV files from their excellent audio tutorial preloaded along with the example programs below
  + JVC earbuds
  + 2 6ft USB Micro-B cables for working with both Teensy 4.1 and ESP32-S at the same time

| **PSA-5FS**-[Prototyping System For Teensy 4.1 – Fully Stuffed](https://protosupplies.com/product/prototyping-system-for-teensy-4-1/?attribute_options=Fully+Stuffed)  $149.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

Add your choice of Teensy 4.1 from the memory configuration options below and the system will be fully assembled with the example software preloaded.  Just apply power and the system will come up and run the example code.  Plug in the earbuds to hear the audio.

The system can be further expanded by adding the optional adapters that are available further down the page such as the audio tutorial adapter and addressable LED adapter.

**Select From the Following Teensy 4.1 Fully Loaded for Prototyping System memory configurations**

The Teensy 4.1 Fully Loaded for Prototyping System have all the headers installed to properly mate up with the connectors on the baseboard.  VUSB trace is cut and a Schottky diode installed to properly isolate power.

**These memory configurations can be selected below:**

* No Additional Memory
* 16MB/128M-bit PSRAM with no Flash
* 8MB/64M-bit PSRAM & 16MB/128M-bit Flash
* 8MB/64M-bit PSRAM & 128MB/1G-bit NAND Flash
* 8MB/64M-bit PSRAM & 256MB/2G-bit NAND Flash

All memory is installed and tested.

[](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/)

| [Teensy 4.1 Fully Loaded For Prototyping System - No Added Memory](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=No+Added+Memory)**DEV-26-B**-[Teensy 4.1 Fully Loaded For Prototyping System – No Added Memory](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=No+Added+Memory)$40.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |
| [Teensy 4.1 Fully Loaded For Prototyping System - 8MB PSRAM + 2Gb Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F256MB+Flash)**DEV-26-2G**-[Teensy 4.1 Fully Loaded For Prototyping System – 8MB PSRAM/256MB Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F256MB+Flash)$56.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| [Teensy 4.1 Fully Loaded - 8MB PSRAM and 1Gb Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F128MB+Flash)**DEV-26-1G**-[Teensy 4.1 Fully Loaded For Prototyping System – 8MB PSRAM/128MB Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F128MB+Flash)$52.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| [Teensy 4.1 Fully Loaded For Prototyping System - PSRAM + PSRAM](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=16MB+PSRAM)**DEV-26-P**-[Teensy 4.1 Fully Loaded For Prototyping System – 16MB PSRAM](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=16MB+PSRAM)$48.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| [Teensy 4.1 Fully Loaded For Prototyping System - 8MB PSRAM + 128Mb Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F16MB+Flash)**DEV-26**-[Teensy 4.1 Fully Loaded For Prototyping System – 8MB PSRAM/16MB Flash](https://protosupplies.com/product/teensy-4-1-fully-loaded-for-prototyping-system/?attribute_options=8MB+PSRAM%2F16MB+Flash)$48.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |

You can click on the picture above to see more info on this build configuration of the Teensy 4.1.

**Note:**  Memory configurations above are shown in Megabytes (MB) while many chip manufacturers are now specifying memory size in Megabits (Mb) or Gigabits (Gb).  The use of a capital B means Byte while lower case b means bit.  Since there are 8 bits in a byte, to convert from MB to Mb (or Gb), multiply by 8.  The 256MB Flash for instance is the same as 2Gb.

**What if I Already Have A Teensy 4.1 That I Want To Use With This Baseboard?**

If you already have a Teensy 4.1 that you want to use with this baseboard, it is fairly easy to adapt to it with nominal thru-hole soldering skills.  There are instructions toward the bottom of this page.

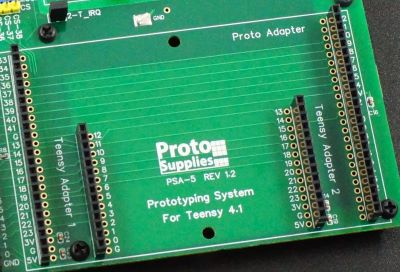
**KEY FEATURES OF PROTOTYPING SYSTEM FOR TEENSY 4.1:**

* All I/O brought out from the bottom of the Teensy 4.1
* Supports an optional ESP32-S co-processor for WiFi and Bluetooth connectivity
* Supports a 2.8″ or 3.2″ 320 x 240 SPI Touch LCD for user interface
* Built-in CAN bus transceiver on CAN3
* Built-in RS485 transceiver on Serial8
* Built-in Host USB connector
* Built-in wired Magjack Ethernet connector
* I2C brought out to breakout headers with 5V level shifted version also available
* SPI brought out to breakout headers
* Built-in high speed logic level converters
* On/Off and Program buttons are brought out
* VBat brought out to CR2032 battery holder
* Built-in thru-hole prototyping area under LCD
* Sockets for plugging in up to 2 standard Teensy adapter boards
* Socket for plugging in custom 4″ x 2.5″ large adapter board
* Male headers on Teensy 4.1 and ESP32-S for connecting breadboard jumpers or hooking up logic analyzer or O’scope probes.
* Ground test points
* Input power setup supports DC power and USB power to Teensy 4.1 and ESP32-S be applied at the same time.
* High quality 4-layer construction with ENIG (Electroless Nickel Immersion Gold) finish.

Continue reading below to learn about the features of the board as well as order options  that can be used to complete your system.

**ADAPTER BOARDS**

The area next to the Teensy 4.1 is designed to provide flexibility for working with various adapters for adding functionality to the system.  We offer a number of options below and it is possible to design your own.



**SMALL ADAPTER BOARDS**

The area next to the Teensy 4.1 is footprinted to accept 2 of the standard size Teensy adapters such as the Audio Rev D adapter for easily working with them.

**Teensy 4.x (Rev D) Audio Adapter (comes with Fully Stuffed version)**

Because of the power of the Teensy 4.1, it is very popular for use with audio projects because it is fast enough to synthesize music on the fly and the PJRC team provides excellent tools like the PJRC Audio System Design Tool for working with their audio module.  We have a version with pins preinstalled so it will plug into the baseboard.  It also comes with a microphone installed.  One comes installed in the Fully Stuffed version along with a 32GB SD card preloaded with sample WAV files.

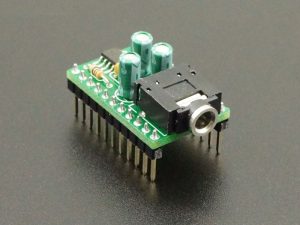
[](https://protosupplies.com/product/teensy-4-x-audio-adapter/)

| **MOD-136P**-[Teensy 4.x Audio Adapter – With Pins & Mic](https://protosupplies.com/product/teensy-4-x-audio-adapter/?attribute_options=With+Pins+%26+Mic)  $18.75Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**PT8211 16-bit Stereo Audio Adapter for Teensy**

The PT8211 is a low cost 16-bit stereo Digital-to-Analog (DAC) with line level output.  It is supported by the PJRC Audio System Design Tool which makes it easy to use with the Teensy 4.1.  We offer a version that is fully assembled with male header pins to make it plug-n-play with the baseboard.  It is also available in kit form on the product page.

[](https://protosupplies.com/product/pt8211-audio-adapter-for-teensy/)

| **MOD-156A**-[PT8211 Audio Adapter for Teensy – Fully Assembled](https://protosupplies.com/product/pt8211-audio-adapter-for-teensy/?attribute_options=Fully+Assembled)  $7.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

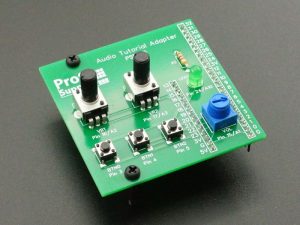
**Audio Tutorial Adapter**

The PJRC team that created Teensy also put together a very nice workshop for learning about audio processing using Teensy since Teensy has the horsepower to synthesize audio on the fly.  They also created a powerful Teensy Audio Library and Audio System Design Tool.

We have created this adapter to provide the user controls for going through the workshop and have modified the original workshop and programs as needed to work with this baseboard in our [**Audio Tutorial and Workshop**](https://protosupplies.com/learn/audio-tutorials-and-workshop/).  The board is shortened to less than full width to provide clearance to also mount the Teensy 4.x Audio Adapter at the same time.  Standoffs provide mechanical support.

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Audio-Tutorial-Setup.jpg)

If you have an interest in learning how to use audio with Teensy or just want to explore more of what the Teensy can do, going through the audio workshop is very informative.  This board can also be used to provide a couple general purpose potentiometer and pushbutton inputs.

[](https://protosupplies.com/product/audio-tutorial-adapter/)

| **PSB-15A**-[Audio Tutorial Adapter – Fully Assembled](https://protosupplies.com/product/audio-tutorial-adapter/?attribute_options=Fully+Assembled)  $12.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**LARGE ADAPTERS BOARDS**

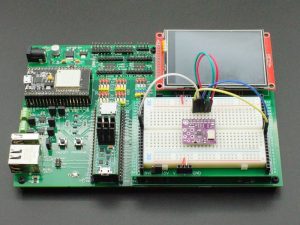
The same adapter area is also footprinted to accept a larger 4″ x 2.5″ adapter with access to all of the Teensy 4.1 pins available along the two edges of the PCB.

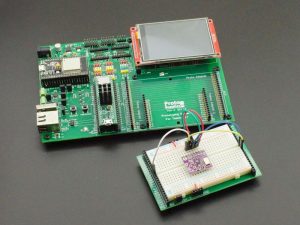
We offer adapter boards that plug in and allow permanent perf board or solderless breadboard circuits to be built that can be easily swapped in and out as desired without permanently modifying the main board.  We also have a number of application specific adapter boards.

**Solderless Breadboard Adapter (comes with Fully Stuffed version)**

This adapter mounts a 400 tie-point high quality solderless breadboard for building temporary breadboard circuits.  The female headers on either end of the breadboard provide access to all of the Teensy 4.1 pins.  This allows circuits to be built, the whole baseboard assembly with solderless breadboard to be moved around and then the breadboard subassembly can be removed intact and replaced without having to reconnect anything.  Great for bouncing between several projects.

The adapter allows 3.3V or 5V power to be selected for the power rails.

[](https://protosupplies.com/wp-content/uploads/2022/12/Solderless-Breadboard-Adapter-on-Baseboard.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/12/Solderless-Breadboard-Adapter-off-Baseboard.jpg)

One of these adapters comes with the Fully Stuffed version, but you may want to have more than one if you like to bounce back and forth between projects without having to rewire things back up.

| **PSB-17**-[Solderless Breadboard Adapter](https://protosupplies.com/product/solderless-breadboard-adapter/)  $15.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**WS2812 Addressable LED Adapter**

Teensy 4.1 and the Teensy products in general, are very popular for use with addressable LED projects because it is fast enough and has enough memory to do smooth animations of large numbers of LEDs.

We have created an adapter that makes it very easy to work with a string of addressable LEDs.  Just plug in the LED strip and a 5V power source.  The adapter also includes the functionality of the popular Teensy Octo2811 Adapter for larger arrays in a physical format that works better with this baseboard.  This allows up to 8 channels of addressable LEDs using CAT cable

| **PSA-8**-[WS2812 Addressable LED Adapter](https://protosupplies.com/product/ws2812-addressable-led-adapter/)  $17.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**Adafruit Feather Adapter**

If you are a fan of Adafruit, this adapter board mounts up to 3 Adafruit FeatherWings for working with them.

Adafruit designs many unique products and creates many of the libraries and example software that make working with Arduino compatible boards possible.

| **PSB-16A**-[Feather Adapter – Fully Assembled](https://protosupplies.com/product/feather-adapter/?attribute_options=Fully+Assembled)  $14.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**Thru-Hole and SMD Adapter**

This adapter board is a general purpose solder type prototyping board with both thru-hole and 20-pin and 28-pin SOIC and TSSOP footprints which can fit smaller pin count ICs as well.

It allows permanent circuits to be built and easily plugged in and taken out of the baseboard.

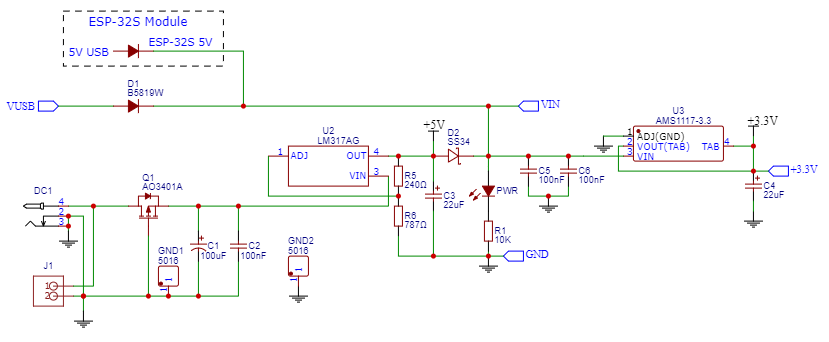
| **PSB-12**-[Prototyping System for Teensy 4.1 – Adapter 1](https://protosupplies.com/product/prototyping-system-for-teensy-4-1-adapter-1/)  $4.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture

**TECHNICAL DETAILS**

**Teensy 4.1 Support**

The Teensy 4.1 socket brings all I/O out from the bottom of the Teensy 4.1.  The socket also has a row of male headers for easily connecting logic analyzer or O’scope probes or for jumping pins over to a solderless breadboard or making temporary connections such as patching in the high-speed logic level converters.





We offer our **Teensy 4.1 Fully Loaded for Prototyping System** line of Teensy 4.1 products that are configured for use with this board and similar baseboard setups that bring the I/O down from the bottom of the Teensy 4.1.  They include the following changes from the standard Teensy 4.1:

* The VUSB/VIN trace is cut and a 1A SMD Schottky diode is placed across the VUSB / VIN pads so both USB and VIN power can be safely applied at the same time.
* Optional PSRAM and/or Flash memory is added depending on configuration ordered.
* VUSB 1-pin header is installed for routing USB power to the baseboard.
* A female 2×3 2mm header is mounted at the Ethernet connector location on the bottom side of the Teensy 4.1 which mates with an extended reach male header on the baseboard and connects the Teensy 4.1 to the MagJack on the baseboard.
* Male 5-pin headers are mounted to the bottom side of the Teensy 4.1 to bring down the USB Host lines to connect to the on-board Host USB connector.  VBat connects to a CR2032 battery holder and On/Off and Program connect to buttons on the baseboard.
* The standard 24-pin male I/O headers are installed

By separating the VUSB and VIN power inputs and placing a Schottky diode between them, this allows the Teensy 4.1 to be powered from both the USB cable and VIN input at the same time if the VIN power source also has a diode installed.  It also allows the modified Teensy 4.1 to be powered on the bench from the USB cable without being placed into a baseboard which can be handy at times.

When used with a 3rd party baseboard, it removes the need to solder a diode onto the bottom of the Teensy 4.1.  When used with our Prototyping System for Teensy 4.1 baseboard, the Schottky diode on the Teensy 4.1 is in parallel with another one on the baseboard, so is redundant.

**Power Subsystem**

The board can be powered from 6.5 – 9VDC via a standard DC power jack.  The board uses linear regulators to minimize electrical noise and while the input can technically be fed with up to 30VDC, the regulators drop the excess voltage via heat, so it is always best to keep the input voltage down toward the lower end to minimize power dissipation in the regulators.

Our 7.5V 1A AC/DC converter works well with this board and one comes with the Fully Stuffed version.  The regulators do have built-in thermal shutdown capability if they get too warm.

[](https://protosupplies.com/product/ac-adapter-7-5v-1a/)

| **PWR-9**-[AC Adapter 7.5VDC 1A](https://protosupplies.com/product/ac-adapter-7-5v-1a/)  $4.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture.

Besides the DC power jack, there is also a footprint labeled **J1** for an optional 0.1″ 2-pin screw terminal or JST style connector for bringing in power or for powering an external device off the DC input.

Reverse voltage protection is provided by a MOSFET transistor.

The input power feeds an LM317 adjustable regulator that is adjusted for a 5.4V output.  This provides approximately 5V to 5.1V on the output of the Schottky power blocking diode.  This guarantees that the full 5V spec is met for any 5V peripherals.  It also helps to ensure that if both DC power and USB power is connected at the same time, the DC power input will continue to power the baseboard rather than drawing from the USB.

This 5V **VIN** power provides power to the Teensy 5V VIN input, the ESP32-S 5V input and the LCD display which all have their own built-in 3.3V regulators.

The VIN power also feeds a 3.3V regulator that powers the **3V** power rail on the board to take the load off the Teensy 4.1.  This arrangement also ensures that 3.3V is available no matter how the board is powered to avoid the possible issue of driving logic signals into an unpowered IC.

The Teensy 4.1 onboard 3.3V regulator output is not used on the baseboard, but the power is available on the **T3V** pins of the Teensy 4.1.  If these pins are used, be sure not to connect the T3V pins to the 3V or other powered rail or damage could result.  The same is true for the **3V3** pin of the ESP32-S.

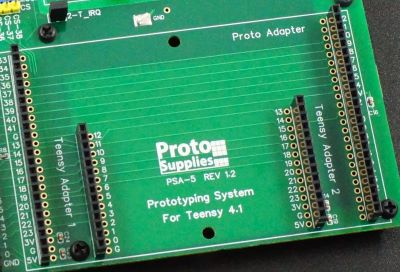
If USB power is also connected to the Teensy 4.1 along with DC input power, the board contains the necessary Schottky diodes to separate the USB power from the VIN power as long as the VUSB/VIN trace is cut on the bottom of the Teensy 4.1 as is done on the Teensy 4.1 with offer for use with this baseboard.

The optional ESP-32S module also includes a built-in Schottky diode as well, so USB power and DC input power can be connected to all 3 at the same time without concern.  Similarly, power applied to any one of the three, such as USB power on the ESP32-S will power the entire board.  A power LED lights whenever VIN power is available.

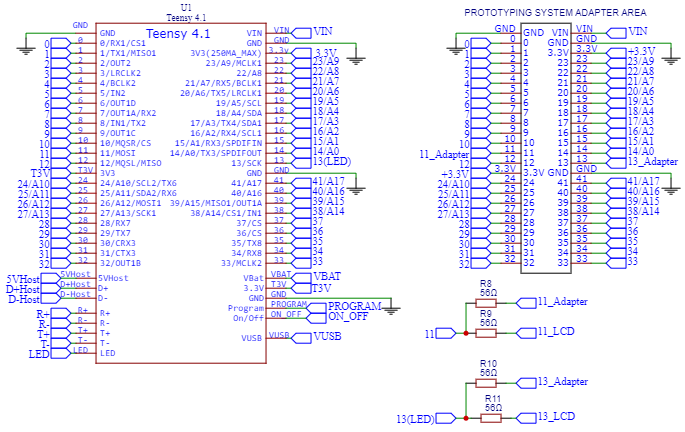


**Teensy 4.1 and Proto Adapter Area**

The entire Teensy 4.1 footprint is brought out to the proto adapter area.  The outer 24-pin female headers support connecting to all of the Teensy 4.1 I/O pins and can fit a 4.0″ x 2.5″ adapter .  The inner 14-pin headers support inserting up to two of the standard PJRC 14-pin adapter boards like the popular Teensy 4.x Rev D Audio Adapter.



The only electrical differences is that the **SPI Clk pin 13** and **MOSI pin 11** lines to the adapter area have 56Ω series resistors to minimize signal reflections.  A second set of resistors feed these same signals to the LCD.  The 3V pins in the adapter area power are supplied from the baseboard 3.3V regulator and not the Teensy 4.1.  0.1uF bypass caps are connected to each of the power pins in this area.



**Touch Display (3.2″ comes with with Fully Stuffed version)**

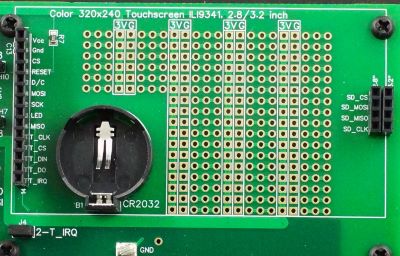
The board supports either a 2.8″ or 3.2″ 320×240 color touchscreen LCD with the ILI9341 display controller.  This is the best supported display for the Teensy 4.1 and readily available from ProtoSupplies.com and PJRC.  The 3.2″ version comes with the Fully Stuffed option.

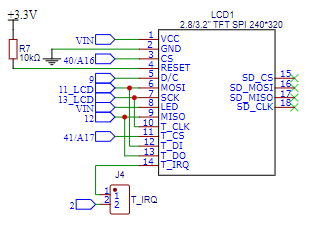
It is recommended to use the [**ILI9341\_t3 library**](https://github.com/PaulStoffregen/ILI9341_t3) for the display and the [**XPT2046\_Touchscreen library**](https://github.com/PaulStoffregen/XPT2046_Touchscreen) for the touchscreen.



The display is connected to the main SPI bus pins **D11, D12, D13** with DC connected to **D9** and CS connected to **D40**.

The touch CS is **D41** and the touch T\_IRQ output is brought into **D2** on the Teensy 4.1.  Reset has a 10K pullup resistor.





A jumper is included on the **2-T\_IRQ** line as shown here so that pin 2 on the Teensy 4.1 can be freed up if this signal is not being used.



The onboard SD card slot on these displays generally don’t work without modification and the Teensy has an SD card slot, so they are not usually worth messing with.  The 4 SD pin locations on the right side of the display are there for mechanical support.  This allows a 4-pin male header to be soldered to the display and plugged into the female header.  The two sets of female header are provided because the 3.2″ display has a slightly wider footprint than the 2.8″ display.

If you order a display with the Prototyping System baseboard we will add the 4-pin male header so that it is ready to plug in with no soldering required.

[](https://protosupplies.com/product/tft-lcd-2-8-240x320-rgb-spi-display-with-touchscreen/)

| **DSP-15**-[TFT LCD 2.8″ 240×320 RGB SPI Display with Touchscreen](https://protosupplies.com/product/tft-lcd-2-8-240x320-rgb-spi-display-with-touchscreen/)  $16.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |
| **DSP-17**-[TFT LCD 3.2″ 240×320 RGB SPI Display with Touchscreen](https://protosupplies.com/product/tft-lcd-3-2-240x320-rgb-spi-display-with-touchscreen/)  $18.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |

For more product information click on the picture.  It links to the 2.8″ version of the display, but they are identical other than the physical size.

**ESP32-S Co-Processor for WiFi / Bluetooth (Comes with Fully Stuffed version)**

A socket is provided for an optional NodeMCU style ESP32-S development board.  This is the 38-pin version that we sell and one comes with the Fully Stuffed option.  There are many different flavors of ESP32 with different pin counts and pinouts, so check the pinout  especially for power and ground if using one from a different source.

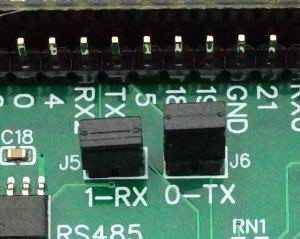
The Teensy 4.1 **Serial1** on pins **D0** and **D1** connects to the ESP32-S **Serial 2** on pins **28** and **27** to provide communications between the two microcontrollers.

The ESP32-S can be used as a co-processor to the Teensy 4.1 to offload all WiFi or Bluetooth work.  The ESP32-S is much better suited for that type of work than the Teensy 4.1.  It can also be used to offload other processing work such as to act as a dedicated sensor monitor and motor controller for instance.

A 1uF capacitor connects the **RST/EN** pin to ground.  This allows software to be downloaded from the IDE without the need for pressing the IO0 (Boot) button.

The other pins on the ESP32-S are unallocated, so can be repurposed as desired.  A row of adjacent male header pins is provided to make jumping over to a breadboard or the Teensy 4.1 easy, such as if you want to work with SPI instead of serial communications between the two microcontrollers.

Two jumpers allow the Teensy 4.1 Serial 1 signals on pins 1 and 0 (**1-RX, 0-TX**) to be disconnected from the ESP32-S if it is desired to free up pins D0 and D1 on the Teensy or if an alternate communication scheme such as SPI is being used with the ESP32.



The footprint accepts the normal 0.9″ wide lead spacing and is how we assemble the boards.  There may be some otherwise compatible ESP32 versions out there with the original wider 1.0″ spacing and so an extra row of holes on one side is provided to support those modules if a female socket is added.

[](https://protosupplies.com/product/esp32-s-development-board/)

| **DEV-16**-[ESP32-S Development Board](https://protosupplies.com/product/esp32-s-development-board/)  $9.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

For more product information click on the picture.

**Wired Ethernet**

The board has a MagJack for making wired Ethernet connections.  Teensy 4.1 has all the circuitry built-in for connecting to Ethernet and the two just need to be physically connected.

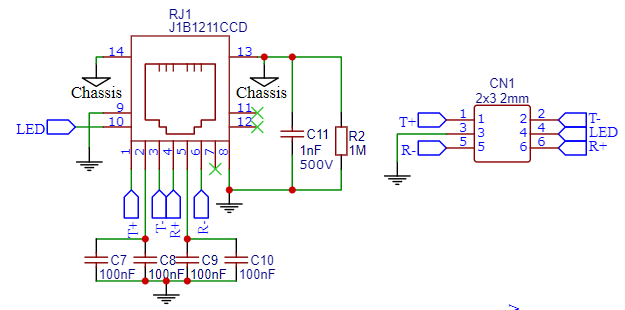
**Note**:  *Due to Ethernet PHY chip shortages and price increases, some Teensy 4.1 on the market are now being built without this chip and do not have Ethernet capability.  All of the Teensy 4.1 we sell for this system have the PHY chip and full Ethernet capability, but if you plan to source or have already sourced a Teensy 4.1 from somewhere else for use with this board, keep that in mind if you plan to use Ethernet.*

The baseboard has a extended length 2×3 2mm spacing male header to pickup the Ethernet connections from the bottom of the Teensy 4.1 that has a mating female header installed.

The Magjack shield is grounded via a 1MΩ resistor and high voltage 1nF capacitor.

There is also a 2×3 2mm male header next to the MagJack that allows a Teensy 4.1 with a male Ethernet header already installed in the normal position to connect to the MagJack using an IDC cable like is supplied with the Teensy 4.1 Ethernet kit.

See our [**Working with Ethernet**](https://protosupplies.com/learn/prototyping-system-for-teensy-4-1-working-with-ethernet/) for more info on using this feature.

[[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Ethernet-Schematic.png)](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-for-Teensy-4.1-Ethernet-Schematic.png)

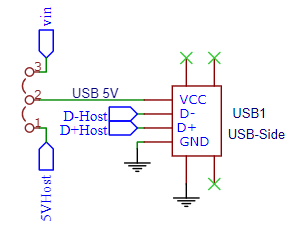
**USB Host**

A 5-pin header under the Teensy 4.1 is used to pickup the USB Host lines.  These are brought out to a side mounted USB 2.0 connector.

The USB Host is a 2nd USB port that allows you to connect USB devices to the Teensy 4.1.  It is fully independent of the main USB device port, so USB devices can communicate simultaneously with Teensy while Teensy communicates with a computer via the USB device port.  The USB Host port operates at up to 480Mbit/sec.  Use the [**USBHost\_t36 library**](https://github.com/PaulStoffregen/USBHost_t36) for working with this port.

For providing power to the USB connector, there is a jumper **USB 5V** to select between using the standard switched USB 5V power from the Teensy 4.1 **T4.1** or using the external VIN 5V unswitched power **VIN** which can be handy for powering higher current applications without having to resort to a powered hub.

Note that the T4.1 switched power is logically controlled by the Teensy 4.1, so a voltage won’t be present until it is enabled by the Teensy 4.1.

[](https://protosupplies.com/wp-content/uploads/2022/03/Host-USB-Section.jpg)

**CAN Bus**

The Teensy 4.1**CAN3** bus on pins **D30**, **D31** is connected to an SN65HVD230, 231 or 232 CAN bus transceiver.  These parts are all equivalent parts in this circuit and selected based on availability.

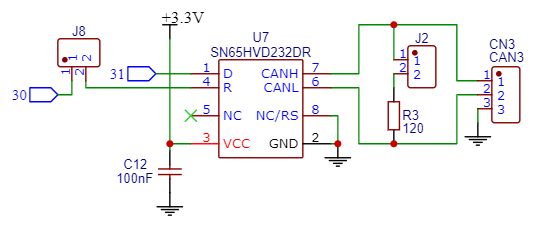
CAN Bus is mainly used in automotive applications due to its distributed processing robust communication protocol.  It can also work quite well in other applications where multiple microcontrollers need to communicate over a common bus such as in robotics.  Use the [**FlexCAN\_T4 library**](https://github.com/tonton81/FlexCAN_T4) for working with the CAN bus.

The receive output of the transceiver on **D30** has a jumper **J8** that can be used to disable this input into the Teensy 4.1 if the pin is to be used for other purposes.

The output has a 120 ohm termination resistor which has a jumper **J2** that can be removed to take the resistor out of circuit if needed for the application.

It connects to a 3-pin 0.1″ spacing screw terminal **CN2**.  Ground is typically not needed since the signal is differential, but is included in case it is desired to ground systems together.





Note that on earlier revisions of the baseboard, the CAN bus transceiver was connected to CAN1.

**RS485 Bus**

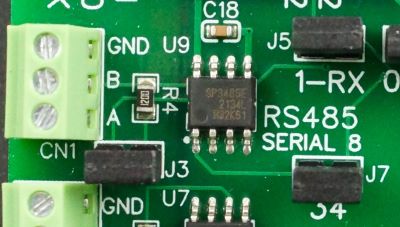
**Serial8** on **D34**, **D35** are connected to an SP3485E RS485 transceiver.  The direction control enable is connected to **D33**.

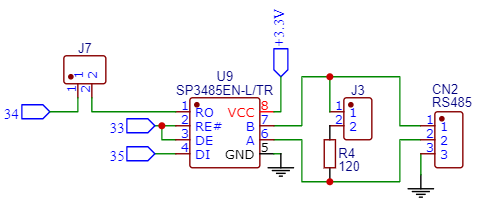
RS485 is a long-line serial communication bus that allows multiple devices to connect to the same serial bus over long distances of up to 1200 meters.  The underlying protocol is standard TTL level RS232 but with differential transceivers used to extend to long distances and support multiple device drops.  Speeds of up to 2.5Mbit/sec are possible.  Handy for device control of one or more devices over longer distances or in electrically noisy environments such as industrial control.

The receive output of the transceiver on **D34** has a jumper **J7** which can be used to disable this input into the Teensy 4.1 if the pin is to be used for other purposes.

The output has a 120 ohm termination resistor which has a jumper **J3** that can be removed to take it out of circuit if needed for the application.

The output comes out to a 3-pin 0.1″ spacing screw terminal **CN1**.  Ground is typically not needed since the signal is differential, but is included in case it is desired to ground systems together.

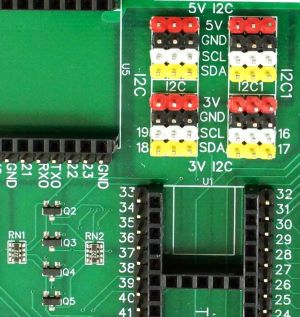


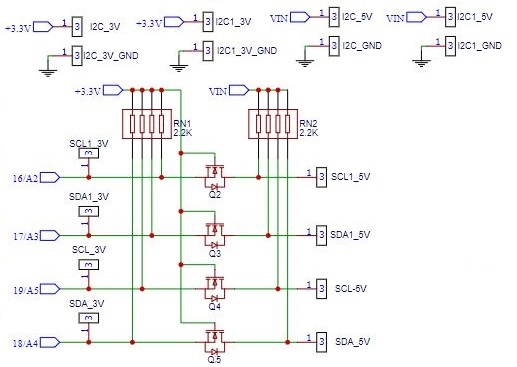


**I2C Buses**

Both the main **I2C**bus on pins **D18**, **D19** and **I2C1**bus on pins **D17**, **D16** are brought out to breakout headers along with 3.3V and ground to connect up to 3 devices per bus.

In addition, a 5V level shifted version of both buses is also provided using the standard MOSFET setup and 2.2K pull-up resistors that will work for most applications.  5V and ground is provided for the level shifted versions.





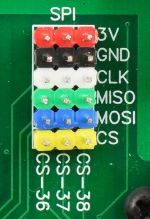
**SPI Bus**

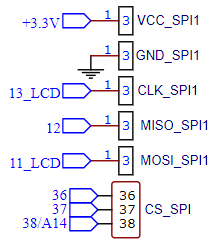
The main SPI bus on pins **D11**, **D12**and **D13** is brought out to three breakout headers along with 3.3V power and ground.

Three chip selects are connected to pins **D36**, **D37** and **D38**.

If a 5V version of SPI is needed, the high speed logic level shifters can be patched in.

The SPI bus is also routed to the adapter area of the board.  CLK and MOSI each have two 56 ohm series resistors with the output of one branch going to the breakout header and LCD display and the second branch going to the adapter area to minimize reflections on the bus.

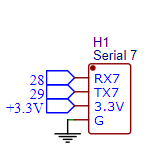




**Serial Port**

**Serial7** on pins **D28**and **D29** is brought out to a 4-pin header on the back edge of the board along with with 3.3V and ground for easy hookup to a serial device.



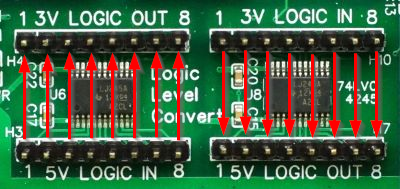


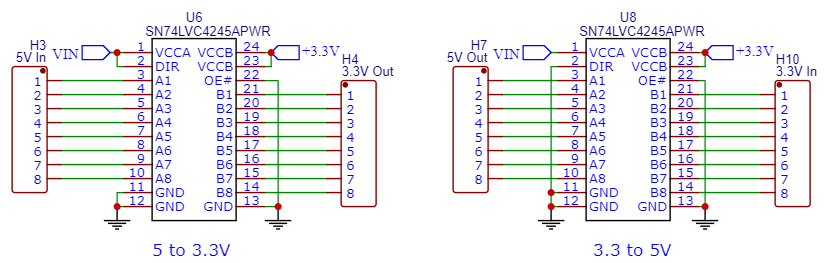
**High-Speed Logic Level Converters**

There are two 74LVC4245A high-speed unidirectional 8-bit buffers on the board.  One is configured as a 3.3V to 5V output translator for driving 5V logic that won’t work with 3.3V logic levels.  The other is configured as a 5V to 3.3V input translator.

These work well for unidirectional signals such as SPI and can handle high data rates.

The pins connect straight across from the 3.3V Logic In to 5V Logic Out or 5V Logic In to 3.3V Logic Out.

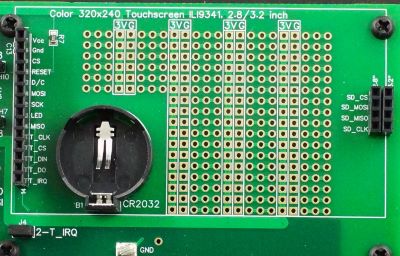


[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-Level-Shifter-Section.png)

**MISC Stuff**

A CR2032 coin cell batter holder is located under the LCD display.  This connects to the **VBat** on the Teensy 4.1 for providing battery backup capability.

The remaining area under the LCD is a thru-hole prototyping area for general purpose use and the power rails are connected to the 3V power.



The On/Off and Program pins are brought out to two pushbuttons next to the Teensy 4.1.

The **On/Off** button when pressed and held for 4 seconds turns off the Teensy 4.1 3.3V onboard regulator, thus shutting the Teensy down.  Pressing the button again for 0.5 seconds while off will turn the  3.3V back on and reboot the processor.  Note that this On/Off button has no affect on the baseboard 5V or 3.3V power.  It only affects the 3.3V regulator on the Teensy module itself.

The **Program** button duplicates the functionality of the pushbutton located on the Teensy 4.1.



Two 6-pin header locations are provided for general purpose use.  One is hard wired for 3.3V and the other is wired for 5V.  A second row of headers allow for using breadboard jumpers to make logic connections.



Spare Headers

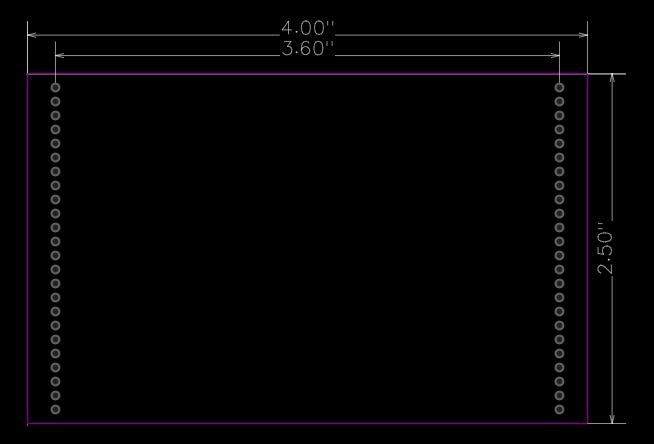
There are a couple of GND test clips on the board for connecting an O’scope or other ground.  A solder pad next to these allow a male header pin to be installed if desired.



Ground Test Point

**Designing Your Own Adapter**

Since designing basic PCBs through quick-turn houses is fairly easy these days, this feature allows users to design and build application specific circuits without having to deal with all the details of building a complete system which can be fairly daunting.  This can also be handy for designing and testing subsystems while prototyping a design before committing to a complete ground up design.  The adapter footprint is simply a 3.6″ wide Teensy 4.1 pinout as shown below.

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-Adapter-Dimensions.jpg)

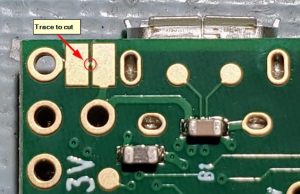
A custom PCB height greater than 2.5″ can also be used if the overhang is on the front edge of the board or can also be on the back side if the LCD is not being used.

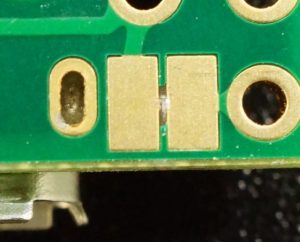
**Modifying An Existing Teensy 4.1 For Use With This Baseboard**

If you already have a Teensy 4.1 that you want to use with this baseboard, it is fairly easy to adapt to it with nominal thru-hole soldering skills and a sharp knife by following the directions below.

If you ordered the Fully Stuffed version and want to verify the basic setup works before modifying your Teensy 4.1, the unmodified Teensy can be installed in the socket and connected to a USB cable to download the example software.  Power for the baseboard will also come from the USB cable.  The ESP32-S will already be programmed so the setup should run as soon as the Teensy 4.1 has been programmed. **Please Note:**  It is important to not also apply power through the DC input on the baseboard until the Teensy 4.1 has been modified as follows or damage could result.

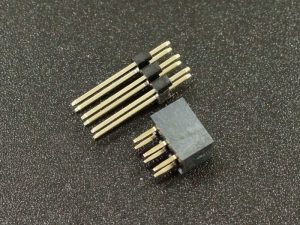
* Cut the trace between the pads on the bottom of the Teensy 4.1 to isolate the USB power from VIN power.  Use a sharp X-acto knife or box knife with a new blade.  Angle the blade at about a 45 degree angle and press straight into the trace on one side.  Don’t use a sawing motion as you are just cutting through soft copper.  Move the blade to the other side of the trace and repeat the process to remove a small wedge of the trace.  A magnifier definitely makes this process easier if you have access to one.  A properly cut trace is shown at the bottom.  If you have a ohm meter, use it to verify the cut was successful by measuring across the pads.  You should not measure a dead short.

[](https://protosupplies.com/wp-content/uploads/2022/03/Teensy-41-VUSB-trace-to-cut.jpg)

[](https://protosupplies.com/wp-content/uploads/2022/03/Teensy-41-VUSB-Trace-After-Cutting.jpg)

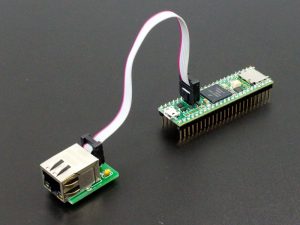
* Place male headers into the 5 pin Host USB and 5-pin VBat female headers on the baseboard.
* Place a single male header pin in the front VUSB position of the 2-pin female header on the baseboard.
* Install a female 2×3 2mm header on top of the tall 2×3 2mm male header if you Teensy does not already have a male header in that location.  Push it just slightly onto the male header, so you can align the pins with holes in the Teensy 4.1 first as you install the Teensy 4.1.
* Install the Teensy 4.1 into its socket making sure that the pins of the headers you just installed all line up with the holes in the Teensy 4.1 and protrude through.  It doesn’t hurt to give a light tug on a couple of the 2×3 2mm female header pins to make sure it is seated firmly up against the bottom of the Teensy 4.1.
* Solder the headers in place.  This ensures everything lines up properly.

The baseboard includes the necessary diode, D1 in the picture above, so no Schottky diode needs to be added to the Teensy 4.1.

[](https://protosupplies.com/product/extended-header-and-socket-set-2x3-2mm/)

| **HDR-43-G**-[Extended Header and Socket Set 2×3 2mm](https://protosupplies.com/product/extended-header-and-socket-set-2x3-2mm/)  $1.49Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)** |
| --- |

If there is a male Ethernet header already installed on the top of the Teensy 4.1, an IDC cable like provided with the Teensy 4.1 Ethernet kit can be used to mate with the male 2×3 2mm header next to the MagJack if you are planning to use that feature.

[](https://protosupplies.com/product/teensy-4-1-ethernet-kit/)

| **MOD-134A**-[Teensy 4.1 Ethernet – Assembled](https://protosupplies.com/product/teensy-4-1-ethernet-kit/?attribute_options=Assembled)  $5.95Qty    **[ADD TO CART](https://protosupplies.com/cart/" \t "_self)**  **[3](https://protosupplies.com/cart/" \t "_self)** |
| --- |

**USING THE PROTOTYPING SYSTEM FOR TEENSY 4.1**

**Note:**  We are now creating a [**Tutorial**](https://protosupplies.com/learn/) section to help capture and better organize more information on using this system than can be put into a product page.  This includes a more detailed explanation of the preloaded example software shown below.

The setup described in this section illustrates the basic usage of some of the core parts of the system and provides a starting point for anyone just getting started with the Teensy 4.1 ecosystem.  It assumes that at least an LCD is installed and also looks to see if an ESP32-S and Teensy 4.x Rev D Audio adapter board with SD card are also installed to enable additional functionality.

If you ordered the Fully Stuffed version along with a Teensy 4.1, it will already be loaded with this software and will run as soon as power is applied.  If you ordered a Fully Stuffed version without the Teensy 4.1, you will just need to load the software into the Teensy 4.1 as everything else will be setup.

The programs are not overly clever on the programming to make them easier to follow (and because I am not overly clever at programming) and pull heavily from various example programs.  The communications between the Teensy 4.1 and ESP32-S in particular are handled in a very simplistic fashion by passing simple text strings.

There are two programs below.  One is for the Teensy 4.1 and the second is for the ESP32-S if one is installed.  The Teensy 4.1 program was updated June 2023 to put more of the system info on the LCD for easier verification of the hardware setup without requiring a serial connection.  Here is an overview of the programs.

**Teensy 4.1 Program Overview**

* Configures the serial ports and LCD/touch screen
* Checks to see if any of the PSRAM or Flash memory chips have been installed and reports that info out the serial port and also prints it on the LCD.  If you ordered a system with the Teensy 4.1 already installed, this also provides a way to verify that you received the correct version without having to pry the Teensy 4.1 out of its socket.
* Paints a couple of buttons on the LCD screen.  One for playing audio and one for scanning for WiFi networks.
* Implements a simple serial communication path with the ESP32-S to see if one is attached.  First asking if one is out there by sending a ‘?’ and looking for a ‘Y’ yes in return.  If found, it enables the Scan button.
* If a scan is requested, it sends the command ‘S’ for scan to the ESP32-S and then looks for a response back with the found networks.  The found networks are then listed on the LCD.
* It also looks for an Audio adapter with an SD card in it with the SDTEST2.WAV file on it (found at [**https://www.pjrc.com/teensy/td\_libs\_AudioDataFiles.html**](https://www.pjrc.com/teensy/td_libs_AudioDataFiles.html).)  If found, it enables the Play Audio button to allow the audio to be started and stopped.  These files are included on the SD card with the Fully Stuffed version.
* It reports various things such as the touch coordinates and button hits out to the USB port for display in a Serial Monitor window if one is open.

**ESP32-S Program Overview**

* Looks for incoming serial commands.
* If it gets a command ‘?’, it responds with ‘Y’ to let the Teensy 4.1 know that yes, it is there.
* If it gets a command ‘S’, it scans for networks and reports the results back to the Teensy 4.1
* It also reports the scan information out the USB port for display in a Serial Monitor window if one is open.
* This program is based largely on the WiFiScan example program.

If the ESP32-S is installed and you have two USB cables, you can open two instances of the IDE with one connected to the Teensy 4.1 and one connected to the ESP32-S for downloading the programs and Serial Monitor windows can be opened on both to see what is going on.  This makes it easy to make program changes to either processor and download new code without messing with cables.  To open 2 separate instances of the IDE, they both need to be launched by clicking on the application icon.

**Teensy 4.1 Demo Example**

/\*

 Teensy 4.1 Prototyping System Board Example

 This program checks for presence of PSRAM/Flash memory the Audio adapter

 with SD card and the ESP32.  Send the results to the LCD and serial port and

 then draws 2 buttons on the LCD screen - Audio and Scan

 If Audio adapter is attached with an SD card installed, the Audio

 button plays the wave file "SDTEST2.WAV" from the Teensy audio tutorial

 https://www.pjrc.com/teensy/td\_libs\_AudioDataFiles.html

 The Scan button sends a command to the ESP32S requesting a scan of

 available WiFi networks.  When the ESP32S returns the scan results,

 the Teensy 4.1 updates those results on the LCD screen and serial port.

 This requires an ESP32S to be installed and the sample program

 ESP32\_Teensy\_41\_Demo be loaded on it, which is just a modified version

 of the ESP32 WiFiScan example program.

 This example code is in the public domain.

\*/

#include <**ILI9341\_t3**.h>

#include <font\_Arial.h>

#include <**XPT2046\_Touchscreen**.h>

#include <**SPI**.h>

#include <Audio.h>

#include <Wire.h>

#include <**SD**.h>

#include <**SerialFlash**.h>

#include "LittleFS.h"

extern "C" uint8\_t external\_psram\_size;

AudioPlaySdWav           playSdWav1;

AudioOutputI2S           i2s1;

AudioConnection          patchCord1(playSdWav1, 0, i2s1, 0);

AudioConnection          patchCord2(playSdWav1, 1, i2s1, 1);

AudioControlSGTL5000     sgtl5000\_1;

// Pins used with the Teensy Audio Shield

#define SDCARD\_CS\_PIN    10

#define SDCARD\_MOSI\_PIN  11 //7

#define SDCARD\_SCK\_PIN   13 //14

// touchscreen offset for four corners

#define TS\_MINX 400

#define TS\_MINY 400

#define TS\_MAXX 3879

#define TS\_MAXY 3843

// LCD control pins defined by the baseboard

#define TFT\_CS 40

#define TFT\_DC  9

// Use main SPI bus MOSI=11, MISO=12, SCK=13 with different control pins

**ILI9341\_t3** tft = **ILI9341\_t3**(TFT\_CS, TFT\_DC);

// Touch screen control pins defined by the baseboard

// TIRQ interrupt if used is on pin 2

#define TS\_CS  41

//#define TIRQ\_PIN  2

**XPT2046\_Touchscreen** ts(TS\_CS);  // Param 2 = NULL - No interrupts

// Define Audio button location and size

#define AUDIO\_X 10

#define AUDIO\_Y 10

#define AUDIO\_W 105

#define AUDIO\_H 32

// Define Scan button location and size

#define SCAN\_X 10

#define SCAN\_Y 50

#define SCAN\_W 105

#define SCAN\_H 32

#define BUTTON\_FONT Arial\_14

#define ESP32SERIAL **Serial1**  // ESP32 is attached to Serial1 port

// Subroutine prototypes

void SetScanButton (boolean);   // Handles Scan button when touched

void SetAudioButton (boolean);  // Handles Audio button when touched

// Misc flags to keep track of things

boolean isTouched = false;      // Flag if a touch is in process

boolean scanRequested = false;  // Flag if WiFi scan is in process

boolean audioAdapterAttached = false; // Flag if audio bd with SD card attached

boolean audioPlaying = false;  // Flag if audio is currently playing

boolean esp32SAttached = false;  // Flag if ESP32S is attached

//===============================================================================

//  Initialization

//===============================================================================

void setup() {

**Serial**.begin(115200);       //Initialize USB serial port to computer

 ESP32SERIAL.begin(115200);  //Initialize Seria1 1 connected to ESP32S

 // Setup LCD screen

 tft.begin();

 tft.setRotation(3);  // Rotates screen to match the baseboard orientation

 // Setup touch Screen

 ts.begin();

 ts.setRotation(1);  // Sets the touch screen orientation

 tft.fillScreen(ILI9341\_BLUE);

 tft.setCursor(1, 110);  // Set initial cursor position

 tft.setFont(Arial\_10);  // Set initial font style and size

 // Check for PSRAM chip installed

 uint8\_t size = external\_psram\_size;

**Serial**.printf("PSRAM Memory Size = %d Mbyte\n", size);

 tft.printf("PSRAM Memory Size = %d Mbyte\n", size);

 if (size == 0) {

**Serial**.println ("No PSRAM Installed");

   tft.println ("No PSRAM Installed");

 }

 tft.println();

**LittleFS\_QSPIFlash** myfs\_NOR;    // NOR FLASH

**LittleFS\_QPINAND** myfs\_NAND;      // NAND FLASH 1Gb

 // Check for NOR Flash chip installed

 if (myfs\_NOR.begin()) {

**Serial**.printf("NOR Flash Memory Size = %d Mbyte / ", myfs\_NOR.totalSize() / 1048576);

**Serial**.printf("%d Mbit\n", myfs\_NOR.totalSize() / 131072);

   tft.printf("NOR Flash Memory Size = %d Mbyte / ", myfs\_NOR.totalSize() / 1048576);

   tft.printf("%d Mbit\n", myfs\_NOR.totalSize() / 131072);

 }

 // Check for NAND Flash chip installed

 else if (myfs\_NAND.begin()) {

**Serial**.printf("NAND Flash Memory Size =  %d bytes / ", myfs\_NAND.totalSize());

**Serial**.printf("%d Mbyte / ", myfs\_NAND.totalSize() / 1048576);

**Serial**.printf("%d Gbit\n", myfs\_NAND.totalSize() \* 8 / 1000000000);

   tft.println("NAND Flash Memory Size = ");

   tft.printf("%d bytes / ", myfs\_NAND.totalSize());

   tft.printf("%d Mbyte / ", myfs\_NAND.totalSize() / 1048576);

   tft.printf("%d Gbit\n", myfs\_NAND.totalSize() \* 8 / 1000000000);

 }

 else {

**Serial**.printf("No Flash Installed\n");

   tft.printf("No Flash Installed\n");

 }

 tft.println();

 // Check for Audio bd with SD card installed

 if (!(**SD**.begin(SDCARD\_CS\_PIN))) {

**Serial**.println("Audio board with SD card not found");

   tft.println("Audio board with SD card not found");

   audioAdapterAttached = false;

 }

 else {

**Serial**.println("Audio board with SD card is attached");

   tft.println("Audio board with SD card is attached");

   audioAdapterAttached = true;

 }

 tft.println();

 // Check for ESP32 installed

 ESP32SERIAL.print("?");   // Ask ESP32 if it is there

 delay (100);              // Wait a bit for ESP32 to respond

 if (ESP32SERIAL.available()) {  // If there is a response

   String returnData = ESP32SERIAL.readString();

   if (returnData == 'Y') {  // ESP32S responded with Y for Yes, I'm here

     esp32SAttached = true;

**Serial**.println("ESP32S was found");

     tft.println("ESP32S was found");

   }

   else {      // No response or invalid response

**Serial**.println("ESP32S not found");

     tft.println("ESP32S not found");

     esp32SAttached = false;

   }

 }

 // Draw buttons

 SetAudioButton (false);

 SetScanButton (false);

 // Setup audio

 if (audioAdapterAttached) { // Setup the audio

   AudioMemory(8);

   sgtl5000\_1.enable();

   sgtl5000\_1.volume(0.5);

**SPI**.setMOSI(SDCARD\_MOSI\_PIN);

**SPI**.setSCK(SDCARD\_SCK\_PIN);

 }

 else {  // If no audio, gray out button

   tft.setCursor(AUDIO\_X + 8, AUDIO\_Y + 8);

   tft.setFont(BUTTON\_FONT);

   tft.setTextColor(ILI9341\_WHITE);

   tft.fillRoundRect(AUDIO\_X, AUDIO\_Y, AUDIO\_W, AUDIO\_H, 4, ILI9341\_DARKGREY);

   tft.print ("No Audio");

 }

 // Setup ESP32

 if (!esp32SAttached) { // If no ESP32 gray out button

   tft.setCursor(SCAN\_X + 8, SCAN\_Y + 8);

   tft.setFont(BUTTON\_FONT);

   tft.setTextColor(ILI9341\_WHITE);

   tft.fillRoundRect(SCAN\_X, SCAN\_Y, SCAN\_W, SCAN\_H, 4, ILI9341\_DARKGREY);

   tft.print ("No Scan");

 }

}

//===============================================================================

//  Main

//===============================================================================

void loop() {

 // Keep an eye on any audio that may be playing and reset button when it ends

 if (playSdWav1.isStopped() && audioPlaying) { // Audio finished playing

   SetAudioButton(false);

**Serial**.println("Audio finished playing");

 }

 // Check to see if the touch screen has been touched

 if (ts.touched() && isTouched == false) {

**TS\_Point** p = ts.getPoint();

   // Map the touch point to the LCD screen

   p.x = map(p.x, TS\_MINY, TS\_MAXY, 0, tft.width());

   p.y = map(p.y, TS\_MINX, TS\_MAXX, 0, tft.height());

   isTouched = true;

   // Look for a Scan Button Hit

   if ((p.x > SCAN\_X) && (p.x < (SCAN\_X + SCAN\_W))) {

     if ((p.y > SCAN\_Y) && (p.y <= (SCAN\_Y + SCAN\_H))) {

**Serial**.println("Scan Button Hit");

       if (esp32SAttached) SetScanButton (true);

     }

   }

   // Look for an Audio Button Hit

   if ((p.x > AUDIO\_X) && (p.x < (AUDIO\_X + AUDIO\_W))) {

     if ((p.y > AUDIO\_Y) && (p.y <= (AUDIO\_Y + AUDIO\_H))) {

**Serial**.println("Audio Button Hit");

       if (audioAdapterAttached && !audioPlaying) {

         SetAudioButton(true);

       }

       else if (audioAdapterAttached && audioPlaying) {

         SetAudioButton(false);

       }

     }

   }

**Serial**.print("x = ");   // Show our touch coordinates for each touch

**Serial**.print(p.x);

**Serial**.print(", y = ");

**Serial**.print(p.y);

**Serial**.println();

   delay(100);           // Debounce touchscreen a bit

 }

 if (!ts.touched() && isTouched) {

   isTouched = false;  // touchscreen is no longer being touched, reset flag

 }

 // If we requested a scan, look for serial data coming back from the ESP32S

 if (scanRequested && ESP32SERIAL.available()) {

**Serial**.print("Read incoming data");

   tft.setCursor(10, 90);

   tft.setFont(Arial\_10);

   while (ESP32SERIAL.available()) { // Print the scan data to the LCD & USB

     String returnData = ESP32SERIAL.readString();

     tft.println (returnData);

**Serial**.println(returnData);

   }

   scanRequested = false;  // Reset the scan flag and button

   SetScanButton (false);

 }

}

//===============================================================================

//  Routine to draw Audio button current state and control audio playback

//===============================================================================

void SetAudioButton (boolean audio)

{

 tft.setCursor(AUDIO\_X + 8, AUDIO\_Y + 8);

 tft.setFont(BUTTON\_FONT);

 tft.setTextColor(ILI9341\_WHITE);

 if (!audio) {  // button is set inactive, redraw button inactive

   tft.fillRoundRect(AUDIO\_X, AUDIO\_Y, AUDIO\_W, AUDIO\_H, 4, ILI9341\_RED);

   tft.print ("Play Audio");

   audioPlaying = false;

   if (playSdWav1.isPlaying()) { // Stop any audio that is playing

     playSdWav1.stop();

**Serial**.println ("Audio being stopped");

   }

 }

 else {          // button is active, redraw button active

   tft.fillRoundRect(AUDIO\_X, AUDIO\_Y, AUDIO\_W, AUDIO\_H, 4, ILI9341\_GREEN);

   tft.print ("Playing");

   audioPlaying = true;

   if (audioAdapterAttached && !playSdWav1.isPlaying()) { // Play audio file

**Serial**.println("Audio being played");

     playSdWav1.play("SDTEST2.WAV");

     delay(10); // wait for library to parse WAV info

   }

 }

}

//===============================================================================

//  Routine to draw scan button current state and initiate scan request

//===============================================================================

void SetScanButton (boolean scanning)

{

 tft.setCursor(SCAN\_X + 8, SCAN\_Y + 8);

 tft.setFont(BUTTON\_FONT);

 tft.setTextColor(ILI9341\_WHITE);

 if (!scanning) {  // Button is inactive, redraw button

   tft.fillRoundRect(SCAN\_X, SCAN\_Y, SCAN\_W, SCAN\_H, 4, ILI9341\_RED);

   tft.print ("Scan WiFi");

 }

 else {          // Button is active, redraw button

   tft.fillRect(1, SCAN\_Y + SCAN\_H, 360, 240, ILI9341\_BLUE); // Clear previous scan

   tft.fillRoundRect(SCAN\_X, SCAN\_Y, SCAN\_W, SCAN\_H, 4, ILI9341\_GREEN);

   tft.print ("Scanning");

   ESP32SERIAL.println("S");   // Send command to ESP32 to start scan

   scanRequested = true;       // Set flag that we requested scan

**Serial**.println ("Scan being requested");

 }

}

**ESP32-S Demo Example**

/\*

   This sketch interfaces the ESP32-S to the Teensy 4.1 via serial port 2.

   It demonstrates how to receive a request for a scan for WiFi networks and

   report the results back via serial port.

   This is a simple variation of the ESP32 WiFiScan example program

\*/

#include "WiFi.h"

#define RX2 16  // Teensy 4.1 is connected to serial port #2

#define TX2 17

const int LED\_PIN = 2;  // We will light the LED when Scan is in process.

//===============================================================================

//  Initialization

//===============================================================================

void setup()

{

**Serial**.begin(115200);   // USB port

**Serial2**.begin(115200, SERIAL\_8N1, RX2, TX2);  //Port connected to Teensy 4.1

 // Set WiFi to station mode and disconnect from an AP if previously connected

**WiFi**.mode(WIFI\_STA);

**WiFi**.disconnect();

 delay(100);

 pinMode(LED\_PIN, OUTPUT);

**Serial**.println("Setup done");

}

//===============================================================================

//  Main

//=======================================================================

void loop()

{

 if (**Serial2**.available()) {

   char command = **Serial2**.read();

**Serial**.println(command);

   if (command == '?') { // Are you there?

**Serial**.println("Y");

**Serial2**.print("Y");  // Acknowledge I'm attached

   }

   if (command == 'S'){

**Serial**.println("scan start");

     digitalWrite(LED\_PIN, HIGH);   // turn the LED on

     // WiFi.scanNetworks will return the number of networks found

     int n = **WiFi**.scanNetworks();

**Serial**.println("scan done");

     if (n == 0) {

**Serial**.println("no networks found");

**Serial2**.println("No networks found");

     } else {

**Serial**.print(n);

**Serial2**.print(n);

**Serial**.println(" Networks Found");

**Serial2**.println(" Networks Found");

       for (int i = 0; i < n; ++i) {

         // Print SSID and RSSI for each network found to both USB and

         // out Serial2 to attached Teensy 4.1

**Serial**.print(i + 1);

**Serial2**.print(i + 1);

**Serial**.print(": ");

**Serial2**.print(": ");

**Serial**.print(**WiFi**.SSID(i));

**Serial2**.print(**WiFi**.SSID(i));

**Serial**.print(" (");

**Serial2**.print(" (");

**Serial**.print(**WiFi**.RSSI(i));

**Serial2**.print(**WiFi**.RSSI(i));

**Serial**.print(")");

**Serial2**.print(")");

**Serial**.println((**WiFi**.encryptionType(i) == WIFI\_AUTH\_OPEN) ? " " : "\*");

**Serial2**.println((**WiFi**.encryptionType(i) == WIFI\_AUTH\_OPEN) ? " " : "\*");

       }

      }

     }

 digitalWrite(LED\_PIN, LOW);    // turn the LED off

 }

}

A 1uF capacitor attached between the ESP32-S EN pin and ground enables it to download the software without the need for pressing the IO0 button as would normally be required.

**Board Construction**

The PCB is 7.5″ x 5″ with  4-layers and constructed of 1.6mm FR-4.

The copper has an ENIG (Electroless Nickel Immersion Gold) finish for maximum performance.  ENIG ensures extended shelf life against corrosion, provides excellent solderability with lead or lead-free solder, provides a completely flat soldering surface and gives good visibility of the soldered joints compared to the golden ENIG plating.  The assembly is soldered using lead-free solder, but leaded solder can be used if desired for making changes.

Green solder mask provides excellent trace visibility in case trace customization is desired.

Their are nine 0.138″ diameter holes which can accept up to 3.5mm or #6 screws unevenly spaced around the board.  The board ships with nine M3 8mm long nylon threaded standoffs and M3 6mm screws to provide mechanical support and electrical isolation when working with it on the bench.

**Rev 1.2 Differences From Earlier Revisions:**

The current shipping Ver 1.2 of the baseboard has a few changes from the earlier Rev 1.0 and Rev 1.1 boards.

* Jumpers added to T\_IRQ, CAN receive and RS485 receive pins to selectively disable unused inputs to the Teensy 4.1 for more flexibility.
* CAN bus transceiver was moved from CAN1 to CAN3 to avoid conflict with MCLK
* SPI header Chip Selects moved from pins 30, 31, 32 to pins 36, 37, 38 to avoid conflict with more commonly used signals
* Fully Stuffed version now ships with a new Solderless Breadboard Adapter instead of the Adapter 1 which will be a better value for our customers.

This page has been updated to the current Rev 1.2 information, but the earlier revision board info can still be found [**here**](https://protosupplies.com/product/prototyping-system-for-teensy-4-1-old-revisions/).

**FURTHER READING:**

Visit our [**Tutorial**](https://protosupplies.com/learn/) section under the **Learn** menu for additional information on this system.  We will be continuing to update this section.

Anyone working with the Teensy 4.1 should checkout the PJRC forum as it has exceptional technical support provided by a talented user base including Paul Stoffregen, the creator of the Teensy product line.

[**PJRC Forum**](https://forum.pjrc.com/forum.php) – Excellent source of technical information and project help for the Teensy 4.1

[**PJRC website**](https://www.pjrc.com/) – Good source of information with many example applications in blog posts and an alternate source for many of the items such as the LCDs and Teensy adapter boards.

[**PJRC Teensy Audio Library Tutorial**](https://www.pjrc.com/teensy/td_libs_Audio.html) – Excellent tutorial and general info on using the Teensy Audio Library and design tools with the Teensy Audio Adapter.  **Note:** We are now providing an optional Audio Tutorial Adapter board and updated the the original workshop programs to work with this baseboard.  These are up in the [**Audio Tutorial and Workshop**](https://protosupplies.com/learn/audio-tutorials-and-workshop/) section under the Learn menu.

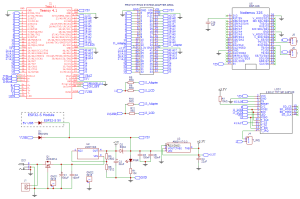
[**Teensy 4.1 Product Page**](https://protosupplies.com/product/teensy-4-1/) – This page on our website has more details on the actual Teensy 4.1 development board itself.

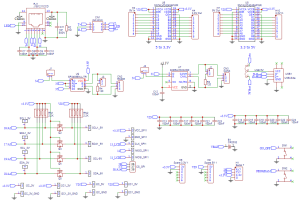
*Notes:*

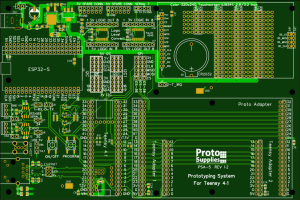
1. *None*

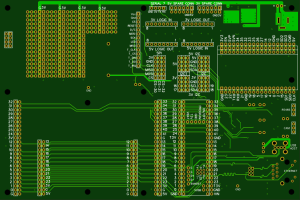
**TECHNICAL SPECIFICATIONS**

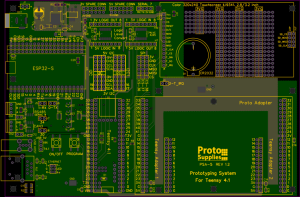
|  |  |  |  |
| --- | --- | --- | --- |
| **Material** |  |  | FR-4 |
| **Layers** |  |  | 4 |
| **Copper** |  |  | 1 oz out layers, 1/2 oz inner layers |
| **Plating** |  |  | ENIG (Electroless Nickel Immersion Gold) |
| **Solder Resist** |  |  | Green |
| **Silkscreen** |  | Front and Back | White |
| **Prototype Area** |  | Holes, plated-thru | 1.02mm (0.04″) |
|  |  | Pads | 1.78mm (0.07″) |
| **Dimensions** |  | Board outline | 190.5 x 127mm (7.5 x 5.0″) |
|  |  | Board thickness | 1.6mm (0.062″) |
|  |  | Mounting holes | 3.5mm (0.138″) |
|  |  |  |  |

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-Schematic-Page-1.png)

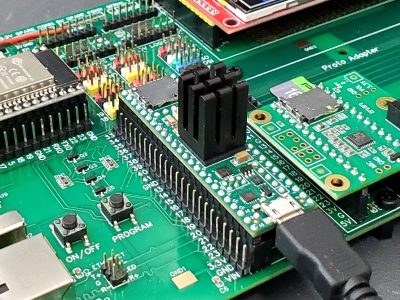
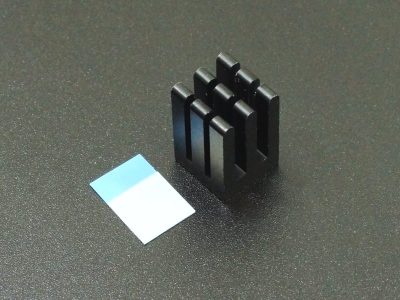
[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-Schematic-Page-2.png)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-CAD-Top.png)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-CAD-Bottom.png)

[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-CAD-Inner-Layers.png)

**YOU MAY ALSO LIKE…**

[](https://protosupplies.com/product/cpu-heatsink-12x12x18-anodized-aluminum/)

[CPU Heatsink 12x12x18 Anodized Aluminum](https://protosupplies.com/product/cpu-heatsink-12x12x18-anodized-aluminum/)

[For overclocking your Teensy 4.x](https://protosupplies.com/product/cpu-heatsink-12x12x18-anodized-aluminum/)

**[$2.25](https://protosupplies.com/product/cpu-heatsink-12x12x18-anodized-aluminum/)**

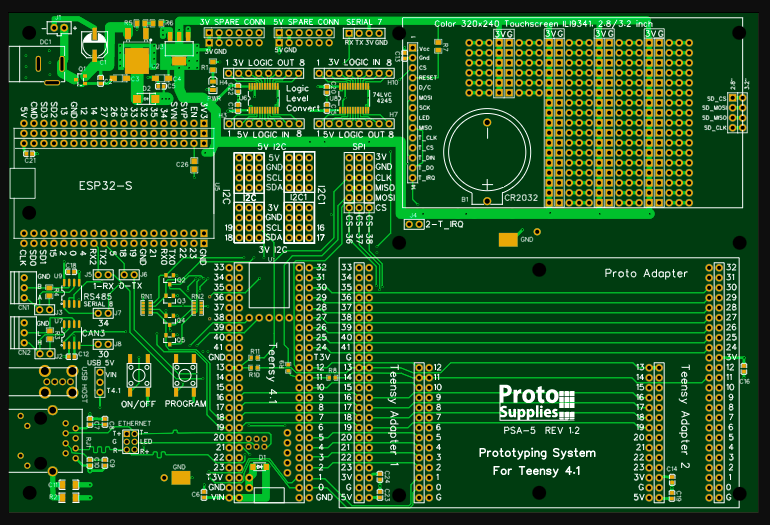
[**ADD TO CART**](https://protosupplies.com/product/prototyping-system-for-teensy-4-1/?add-to-cart=13392)[**MORE INFO**](https://protosupplies.com/product/cpu-heatsink-12x12x18-anodized-aluminum/)

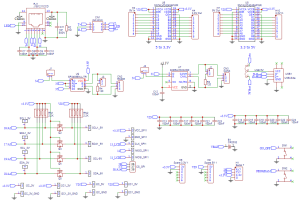
[](https://protosupplies.com/product/usb-2-0-type-a-to-micro-b-cable-black-6-ft/)

[USB 2.0 Type-A to Micro-B Cable, 6-Ft](https://protosupplies.com/product/usb-2-0-type-a-to-micro-b-cable-black-6-ft/)

[Nice long USB Micro-B cable is flexible with 24/28 AWG wires](https://protosupplies.com/product/usb-2-0-type-a-to-micro-b-cable-black-6-ft/)

**[$3.25](https://protosupplies.com/product/usb-2-0-type-a-to-micro-b-cable-black-6-ft/)**



[](https://protosupplies.com/wp-content/uploads/2022/03/Prototyping-System-For-Teensy-4.1-Schematic-Page-2.png)