

1. Introduction

USB Host Shield is an inexpensive (**\$25** for the **full-sized board** and **\$20** for the **Mini variant**) add-on board for Arduino development platform. The shield provides USB Host interface, allowing full and low-speed communication with USB devices – keyboards, mice, joysticks, MIDI, digital cameras, Bluetooth, and many others. In USB Shield section of this site you can find many articles describing projects and code examples written for this shield. On this page, I'm giving detailed description of board's hardware. I start with explaining board's connectors, pads and jumpers, as well as differences between shield variants. Finally, I demonstrate ways to adapt USB Host Shield to non-typical Arduino boards and less-common power configurations. This document covers both full-size and Mini shield variants. At the time of writing, current revision of full size USB Host Shield is 2.0 and current Mini shield is 1.1. Older revisions of the shield will be described later.



2. Board Layout

USB Host shields are available in two form factors – full size and Mini. Full size shield is designed to fit on top of “Standard” Arduinos, such as Uno, Duemilanove, Mega 1280/2560, and compatible clones. Full size shield has been designed for ease of use; it has plenty of empty space, features extra pads, solder jumpers and extensive silkscreen markings, simplifying board modification and troubleshooting. Full size shield is recommended for basic prototyping and simple projects. Mini shield main advantages are low size, weight and cost. Ideally, it should be used together with Arduino Pro Mini 3.3V board. It can be mated with other Arduino and non-Arduino MCU boards, but it takes more work. Small size, dense part placement and lack of silkscreen markings make this board more suitable for advanced projects, as well as semi-permanent and permanent installations, when basic functionality and wiring is already confirmed on larger prototype. Generally, modification and troubleshooting of Mini shield board is more difficult. **Full-size shield** USB Host Shield 2.0 exists in 2 configurations – “Standard” and “3.3V”. The layout of Standard board is depicted on the right. The board contains Maxim MAX3421E USB host controller, 12MHz crystal, level shifters, resistors, capacitors, Reset button and USB A-type connector. There are also a number of solder pads and jumpers, which are marked with red arrows. I start my explanation with an arrow close to Reset button and move counter-clockwise.

1. **Power Select** 2 solder jumpers marked “5V” and “3.3V”. They are used for different power configurations. The configuration shown, when both jumpers are closed, is suitable for official Arduinos, such as UNO, Duemilanove, Mega and Mega 2560. See Power Options section for detailed explanation.
2. **Power pins** are used to connect to power pins of Arduino board. RESET, 3.3V, 5V and GROUND signals from this connector are used.
3. **Analog pins** are not used by the shield. They are provided to simplify mounting and provide pass-through for shields mounted atop of USB Host Shield in a stack.
4. **GPIN pins**. Eight 3.3V general-purpose digital input pins of MAX3421E. They are used primarily to interface with buttons, rotary encoders and such. GPIN pins can also be programmed as a source of MAX3421E interrupt. An example of GPIN use can be seen in digital camera controller project.
5. **ICSP connector** is used by the shield to send/receive data using SPI interface. SCK, MOSI, MISO and RESET signals from this connector are used.
6. **GPOUT pins** are eight 3.3V general-purpose digital output pins of MAX3421E. They can be used for many purposes; I use it to drive HD44780-compatible character LCD, as can be seen in digital camera controller circuit, as well as this keyboard example. Max_LCD library which is part of standard USB Host library software package uses some of GPOUT pins.
7. **Digital I/O pins 0-7**, like already mentioned analog pins are not used by the shield and provided only for convenience.
8. **Digital I/O pins 8-13**. In this group, the shield in its default configuration uses pins 9 and 10 for INT and SS interface signals. However, standard-sized Arduino boards, such as Duemilanove and UNO have SPI signals routed to pins 11-13 in addition to ICSP connector, therefore shields using pins 11-13 combined with standard-sized Arduinos will interfere with SPI. INT and SS signals can be re-assigned to other pins (see below); SPI signals can not.
9. **MAX3421E interface pads** are used to make shield modifications easier. Pads for SS and INT signals are routed to Arduino pins 10 and 9 via solder jumpers. In case pin is taken by other shield an re-routing is necessary, a trace is cut and corresponding pad is connected with another suitable Arduino I/O pin with a wire. To undo the operation, a wire is removed and jumper is closed. See interface modifications section for more information. GPX pin is not used and is available on a separate pad to facilitate further expansion. It can be used as a second interrupt pin of MAX3421E.
10. **VBUS power pad**. This pad is used in advanced power configurations, described in Power Options section.

Notice that digital and analog pins have parallel rows of pads next to them. They are simply extra pads which are wired to corresponding Arduino pins. An example of using one of this pads is given in Interface modification section below. **3.3 V variant of the shield** comes pre-configured to work with 3.3V-only Arduino variants, such as Sparkfun's Arduino Pro 3.3V board. The 3.3V setup is generally less power-hungry than 5V and in many cases can be run directly from a single-cell LiPo battery. Therefore, 3.3V shield is most suitable for portable projects interfacing primarily with self-powered USB devices, such as digital cameras. Differences between 3.3V and Standard variants of the shield are shown on the picture to the right (click

USB Host Shield 3.3V differences

on it to make it bigger).**[EDIT]**After much testing it has been determined that standard shield works with 3.3V-only Arduino boards just fine so I stopped manufacturing 3.3V shield.**[/EDIT]** Because default VBUS power configuration is set to 3.3V, this shield is less suitable for bus-powered devices which by spec should be powered by 5V. Surprisingly, many of them work from 3.3V just fine. Also, if 5V VBUS is desired, possible modification is described in Power Options section. It is worth noting that both variants of the shield are built on the same PCB, therefore it is possible to make one variant from another by adding/removing parts. **Mini shield** The Mini is 3.3V-only device; it is designed to mate with Arduino Pro Mini. This arrangement makes interconnections easy and keeps serial/programming connector of Arduino accessible. Needless to say, the pinout of Mini shield is identical to Arduino Pro Mini. Picture on the right shows all board pins, pads and jumpers. If Arduino schematic has different name of the pin, this name is given in parentheses. I start in the lower left corner of the image and move counter-clockwise.

1. **SS, MOSI, MISO, SCK** are SPI signals. SS can be re-wired to a different pin, if necessary – a jumper and extra pad is provided on top side of the board. Other three SPI signals can be shared between peripherals.
2. **3.3V**. Power pin of the shield. Do not supply 5V to this pin – MAX3421E won't be able to detect devices on VBUS and may also be damaged!
3. **GND** Ground return. There are two GND pins on the shield, for proper operation both need to be connected to the MCU board.
4. **RAW** This is Arduino Pro Mini LDO input pin. This pin is not used by Mini shield, however, it can be used to provide 5V to VBUS (see next section for details).
5. **GPIN and GPOUT** are general purpose input and output pins, which can be used for various purposes. The numbers to the right of the name show bit layout, for example, 'GPIN 3210' means that in a row pointed by arrow the far left of 4 pads corresponds to GPIN3 and the far right is GPIN0. See full size shield pin description above for more information about these pins.
6. **RESET**. Arduino RESET pin. Pulling it low resets MAX3421E into initial power-on state.
7. **GND**. Second ground return. There are two GND pins on the shield, for proper operation both need to be connected to the MCU board.
8. **SS and INT pads and jumpers** are provided to aid in board modification, similarly to full size shield – if default SS (D10) and/or INT (D9) pins are occupied and need to be re-wired, a trace inside the corresponding jumper can be cut and pad used to solder a wire. See 'Interface modifications' section for an example.
9. **GPX** is currently not used. It is broken out to a pad in case someone needs it.
10. **INT**. MAX3421E interrupt pin. If necessary, INT can be re-wired from this pin using extra pad and jumper, as explained above.
11. **VBUS pad and jumper** can be used to provide 5V to VBUS. By default, VBUS is connected to 3.3V; certain USB devices may have issues with this.

Layout of USB Host Mini

3. Basic usage.

Full size shield ships with power/interface pins unpopulated and bundled with a pair of 1×6 and 1×8 plus one 2×3 stackable male/female header which should have to be soldered to the shield prior to use. 8-pin headers need to be soldered to digital pin pads, 6-pin headers come to power and analog pads, the 2×3 header is used to populate 2×3 connector and has to be soldered female side down in order to be compatible with official Arduinos. As you can see on the title picture, using stackable headers is optional. For my projects I prefer using plain break-away headers on the sides. Standard headers are much easier to source (and I sell them too), that's why I'm including less common stackable ones. Also, if you'd like to be able to have access to ICSP pins while shield is mounted, for example to be able to connect AVR Dragon or other programmer/debugger, do not cut off male tails of 2×3 connector after soldering. Mounting the shield on top of Arduino board provides the shortest signal path and better noise immunity and stability. It is also very restrictive and inflexible; if you'd like to be able to access every point in a circuit, it is possible to arrange boards side-by-side connecting them with pieces of wire. Word of caution: do not try to route SPI signals through a breadboard! Breadboards have huge stray capacitance and SPI interface is fast – mixing the two would result in a circuit which is quite unstable if not outright faulty. Default arrangement of **Mini shield** has already been noted in "Board Layout" section.

4. Power options

MAX3421E is 3.3V device and USB is 3.3V bus. Therefore, it is essential to always have 3.3V supplied to the shield during operation. USB-specified VBUS voltage is 5V; since all bus-powered devices are designed to be powered from 5V, some of them won't work reliably or even at all from 3.3V. This is especially true for low-speed devices, like keyboards and mice – I've tested plenty of them and so far found none which would function satisfactorily when VBUS is at 3.3V. Considering this, a

shield which has 5V VBUS has its USB bus done 'by spec' and gives its owner the least grief. "Official" Arduinos, as well as many clones, provide both 3.3V and 5V to the power connector; the "Standard" shield in default configuration handles VBUS 'by spec' and is preferable for general development. On the other hand, 3.3V systems consume less power and can be run directly from a single LiPo; they are much better suited for portable applications. Additionally, self-powered devices, such as digital cameras don't really care about VBUS – they have their own power source. In many cases, it is possible to have perfectly functioning 3.3V-only USB system. For this purpose, the "3.3V" shield variant has been developed. It runs off of single 3.3V supply, provides 3.3V to VBUS and mates perfectly with Sparkfun's Arduino Pro 3.3V board. I have tested this configuration with many different self-powered devices and most of them worked just fine. However, there are times when we have 3.3V-only circuit and need to provide 5V to VBUS. In this case, it is possible to mount external boost converter to generate 5V from 3.3V. Picture on the right demonstrates one way; a [TPS61200-based booster](#) is soldered to the board in the following manner: input to 3.3V, output to VBUS, ground (on the other side of DC-DC converter board) to ground. Note that *both* VBUS power selector jumpers are open. (Disclaimer: This booster is my design and Sparkfun pays me royalties from sales, therefore both this paragraph and accompanying picture shall be treated as shameless plug.)

A boost converter mounted on USB Host Shield

If 5V-only Arduino board is used, 5V to 3.3V conversion is necessary. It is usually done with LDO; I don't currently have pictures but basically all you need is to have 5V power selector jumper closed, 3.3V jumper open VBUS powered from 5V, input of LDO connected to VBUS pad and output of LDO connected to 3.3V pad on the circuit side of 3.3V jumper. **Mini shield** lacks flexibility of its full size brother; however, here is one simple way to get 5V to VBUS using Arduino Pro Mini's 3.3V on-board LDO. Power the board with 5V on RAW pin, cut the trace inside VBUS jumper, provide 5V from RAW to VBUS and get 3.3V from Arduino to the shield in the usual manner via 3.3V pin. While making this mod be extra careful and don't short VBUS pad to USB connector shield. Also, this mod is easier to make when Arduino board is placed beneath the shield. Picture on the left shows necessary wiring.

USB Host Mini with 5V VBUS

5. Interface modifications

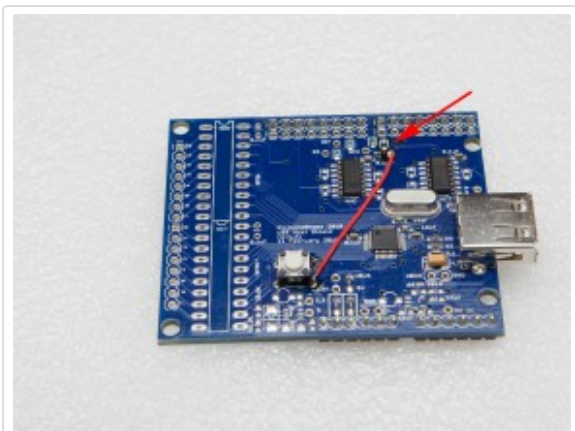
There are situations when default pin assignment of USB Host Shield conflicts with other board. In this case it is desirable to be able to move conflicting pin to some other place. SPI signals – SCK, MISO and MOSI, can be shared between slave devices and usually don't require relocation. Two other signals – SS and INT, can be placed to any vacant pin. After modifying the board, software shall be made aware of the change. We start with hardware. Picture on the left shows the shield modified to work with [BlackWidow 1.0](#) Arduino board. The wireless module of BlackWidow uses pin 10, therefore we will be moving pin 10 of the shield to pin 7, which is not used for anything. First of all, a trace inside SS jumper needs to be cut in order to disconnect SS signal from pin 10. After this is done, we need to run a wire from SS pad to a new pin. Next we need to modify the code. If you are using current (2.0) revision of the library, open [UsbCore.h](#) and find a line which reads:

```
typedef MAX3421e<P10, P9> MAX3421E
```

At the time of writing the line number is 27. Pins are set inside the angle brackets, first is SS, second is INT. To move SS from pin 10 to pin 7 change P10 to P7, save and recompile. If you are still using legacy library, the procedure is similar. Open [Max3421e_constants.h](#), find `#define MAX_SS 10` on line 22 and change "10" to "7". Save, recompile, and you are done. Mini shield can be modified in a similar fashion; there are no extra pads for Arduino pins, however, so wires need to be soldered directly to the existing ones.

If you still have rev.1 of USB Host Shield or bought a clone cloned from rev.1, the following section describes how to make it work with the current Arduino library. Since current library doesn't initialize MAX3421E reset line this line needs to be disconnected from Arduino and tied to high potential. It is even better to tie it to Arduino RESET line so both can be reset using RESET button. The following picture shows the mod. First, a jumper resistor next to RST pad needs to be removed. This can be done easily by heating both sides of the resistor simultaneously. After the jumper is removed, a wire needs to be run from RST pad to the RESET button, as pictured. This completes the mod.

Rerouting SS signal



Rev.1 upgrade

6. Conclusion

This is all for now. As I already said, this page will be expanded to cover other shields and modifications. If you'd like to have anything added/corrected/removed, please let me know.