

Troubleshooting Arduino USB Host Shield

By Oleg Mazurov

[EDIT] This article covers revision 1 of the shield. Current revision (2.0.x) is slightly different and works under different software. The following [test routine](#) shall be used to test the board and generate test signals. **[/EDIT]**

Making electronic devices requires close interaction with parts – reversing supply polarity, overloading inputs, and inadvertently shorting pins with test leads. Consequently, occasional destroying of parts is natural and shall be anticipated. I have been in correspondence with several electronics enthusiasts helping them getting their shields fixed and since their problems look similar to what I see when doing post-manufacturing quality control I decided to share my testing procedure along with some pictures.

In the past, it was customary to include schematic with every electronic device documentation. Complex devices, such as oscilloscopes, spectrum analyzers and other test instruments used to have service manuals containing detailed calibration and repair procedures. At some point, service manuals and schematics disappeared from the documentation for various reasons – equipment users were left to deal with manufacturer's support or rely on their own reverse engineering skills. With open source movement and general understanding that sharing information is beneficial, manufacturers resumed publishing schematic diagrams of their creations. This article presents next logical step – a service manual for Arduino USB Host Shield, sort of.



USB Host Shield in a test fixture

Much of the testing is performed using board test sketch, [available from examples section](#) on github. Two files are necessary – board_test.pde and board_test.h containing diagnostic messages. The sketch tests 4 major parts of the circuit – SPI interface, general purpose input/output pins (GPIO), quartz crystal oscillator, and finally USB SIE. The main loop is written so that any test can be turned off if necessary by commenting out a single line. GPIO lines are checked using a loopback adapter – a thing that connects GPIN0 to GPOUT0, GPIN1 to GPOUT1, and so on. This test is made optional – if you don't connect GPIO lines as described, the test will print an error message and continue with the next test. Also, GPIO test is placed between short and long SPI tests. The reason for this is that due to MAX3421E internal organization both short SPI test (reading REVISION register) and GPIO read/write doesn't require working crystal oscillator, whereas long SPI test (reading/writing any other register) will fail and stop if crystal is defective. Therefore, when I see short SPI and GPIO tests passed and long SPI test fail I know that it's actually a crystal which is dead, not SPI.

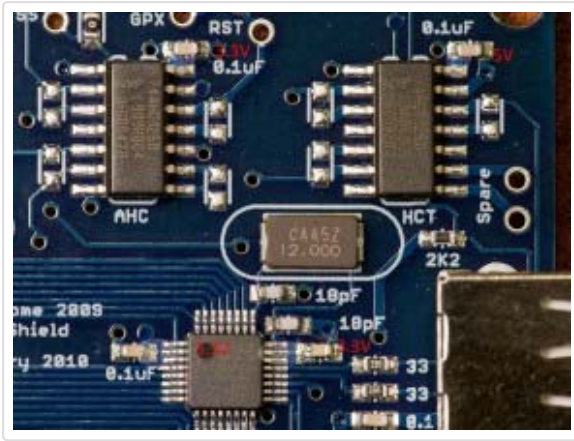
In addition to board test program, you will need a multimeter with thin sharp test leads to measure voltage and resistance between board elements. Some of them are quite small so a magnifier is also handy. Certain steps of the test procedure call for time-base instrument. Modern digital mixed-signal oscilloscope is the best choice, however, since very few people can afford one, a method of visualizing SPI traffic with plain analog oscilloscope will also be demonstrated. Logic analyzer is handy, but optional. For testing USB transactions you will also need some sort of device connected to shield's USB connector. I usually use USB flash drive as a test device.

The article as well as board test program is written for worst-case scenario, i.e., shield which was built from scratch or came from major rework like MAX3421E replacement due to applying 5 volts to 3.3V pin. The test program works the same way with all four configurations, however, manual tests are shown only for "Simple" configuration, i.e. one with level translators and receiving both 3.3V and 5V from Arduino Duemilanove or similar (no DC-DC converters). Testing other configurations is slightly different and will be noted in the text. Also, "Minimal" configuration calls for specific type of test device – I use digital camera.

Before plugging shield into Arduino it's good idea to do a quick check of power traces. Resistance from ground to 3.3V pins on the shield should be about 200KOhm, ground to 5V – more than 1MOhm. If yours is measuring much less than that, find out why. Usually, resistance of less than 1KOhm means that MAX3421E was soldered in wrong orientation. Resistance less than 100Ohm means that there is a short somewhere. Soldering mistakes and shorts would have to be fixed before we can proceed.

The resistance figures given above are different for other configurations. For example, 5V shield measures 500KOhms 5V to ground and voltage from multimeter actually starts the DC-DC converter so results of a measurement look rather strange. Exact numbers are not important – by measuring power line resistance we are making sure that no shorts or inverted tantalum capacitors exist on a power trace.

Board test sketch starts checking the shield by reading die revision register from MAX3421E – there are only 3 values which are accepted, 0x01, 0x02, and 0x03. Shield with shorted or unpowered SPI returns either 0xff or 0x00. If this test passes, I can conclude that MAX3421E and level translators are receiving power and SPI lines are kind of OK. However, if this test fails, more measurements are required. First thing to check is power to active elements of the circuit. Picture on the left



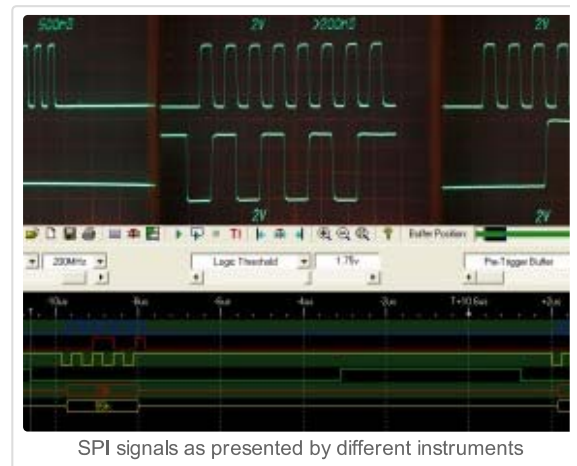
shows measurement points and voltages. IC power pins and bypass capacitors are highlighted. When testing, I'm measuring on the IC pin rather than the cap and try not to short power pin to the adjacent one. If probing IC pin is not possible, measure on the capacitor but check with good magnifier that solder joints are solid. If power is absent in any of the test points, check power coming from Arduino and DC-DC converters, if any.

If ICs receive power but test still fails, SPI lines need to be checked. To help with this check, board test routine starts generating SPI transfer of 0x55 (binary 01010101, i.e. alternating 0 and 1). Other necessary signal are also generated. First signal to check is SS. By default it comes from pin D10 of Arduino connector; right probe on title picture is probing this very pin. It is 5V p-p square wave signal with ~12uS period (on Duemilanove or other 16MHz Arduino). If you can't see it with an oscilloscope, it's either shorted or Atmega pin is dead. To isolate the fault, disconnect power from Arduino, pull the shield out of Arduino, app

power and restart the test. When it fails at SPI test, repeat the measurement. If you still can't see SS, try two other SPI signals generated by Arduino – SCLK (pin D13) and MOSI (pin D11). If you can see them but not SS, the Atmega pin may be fried. If neither of signals could be visualized, check your probing setup and also make sure board test sketch is still running. It is highly unlikely that all 3 pins are dead so before replacing Atmega chip on Arduino board make sure that your oscilloscope is functional.

If SS signal is present, it can be used as a trigger to see the rest of SPI lines. SS gets connected to the external trigger input of oscilloscope's horizontal sweep and time base is set to trigger on negative-going slope. Picture on the left shows how signals, probed on the Arduino connector, look like on an analog oscilloscope with 20MHz vertical bandwidth. Click on a picture to make it bigger. Left shot in the top row shows SS (bottom trace) and SCLK, triggered by negative-going slope of SS. In the middle shot, still triggered by SS and slightly stretched, bottom trace shows MOSI signal against SCLK. It is possible to read what is being transmitted – the state of MOSI at positive-going edge of SCLK gives you transmitted bit. It can be seen that counting left to right first low to high transition of SCLK lies against low state of MOSI, second against high, third – against low, and so on, giving 01010101, or 0x55.

The right shot in top row shows MISO signal against SCLK, i.e., MAX3421E transmitting it's status. You can try reading what is being transmitted on MISO using technique given above or just look at the screenshot in the bottom row of the same picture showing [Intronix](#) Logicport analyzer output of the same signals – all at once with protocol decoding. However, logic analyzer can be used only when signal integrity is checked with oscilloscope, otherwise it's output will be difficult to interpret.



If your MISO is similar to the one on the picture, SPI is working. If MISO is constantly low or high and other signals look distorted or amplitude is wrong, check signal path from Arduino pin through level converter (amplitude will go down to 3.3V) to MAX3421E. It is very important to use correct schematic diagram for the revision of a shield. Shield revision 1.0 has level converters in DIP packages, revision 1.2 has SOIC level converters and schematics are different. If MAX3421E receives clean SCLK, MOSI, SS, MAX_RESET is high and power on pins 2 and 23 is within spec, MISO shall be transmitting status looking similar to one on the picture. If it's not seen on pin 15 of MAX3421E, check to see if this pin or MISO trace is shorted to the adjacent one. When all connections are checked good, the last option is to replace MAX3421E.

I already noted that test program will fail a second SPI test when quartz oscillator is not working. However, if you can see clean MISO signal similar to one pictured, it means that SPI is OK. Oscillator can also be checked with oscilloscope. Look again at the picture showing supply voltages. Pins in the top right corner of MAX3421E, 24 and 25, are connected to the quartz crystal. When oscillator is working, 12MHz 3-4V p-p sinewave signal can be observed on either one of them. For this measurement FET probe is essential; ordinary passive 10x probe introduces relatively large capacitance and can stop oscillator altogether. The "OSCOKIRQ failed to assert" message in the first line of test sketch output is also a good indicator of faulty oscillator. The best way to fix it is to replace the crystal.

The last test is a short sequence of USB transactions resulting in getting device descriptor from the connected device. If test fails or stops at "Waiting for the device" message, it means that either USB D+ and D- lines are faulty or VBUS power is not present. Both faults can easily be found with a multimeter. Errors reported during other stages could mean many different things, including, again, faulty oscillator, bad connection, or simply polling of device too rapidly. Good example of false error is Error code D or 0d just before printing device descriptor. In any case, successful output of device descriptor means that shield is fully operational.

The test routine described here has been working as post-production test for quite some time and shows good results. Give it a try and tell me what you think.

Oleg.

64 comments to Troubleshooting Arduino USB Host Shield



Richard
[May 10, 2010 at 5:17 am](#)

The USB Host Shield from Sparkfun sku: DEV-09628 has the GPX pin and Reset pins swapped from the original USB Host shield described here.

It will fail the tests above at the SPI stage:

GPIO check failed. Make sure GPIO loopback adapter is installed
SPI test. Each '.' indicates 64K transferred. Stops after transferring 1MB (16 dots)

SPI transmit/receive mismatch
Value written: 01Value read: 00
Test Halted.
0x55 pattern is being transmitted via SPI to aid in troubleshooting
Press RESET to restart test

The #define in Max3421e_constants for the reset pin must be changed:
#define MAX_RESET 8
All usb libraries should then work OK.



Michael Nielsen
[March 29, 2011 at 5:06 pm](#)

I tested it with various devices: 4 brands of optical mice, a keyboard, a usb drive, a webcam, vectornav tilt sensor. I get different results during the usb connection test (and the descriptor reader sketch):

1. webcam,usb stick, keyboard: Gets the descriptor nicely – success! 😊
2. 3 mice: SOF generation started. Enumerating device.Setup packet error: D – Error
3. MS mouse, vectornav tilt sensor: nothing at all – doesnt even see it connected



Michael Nielsen
[March 30, 2011 at 3:45 pm](#)

Using a circuits@home board instead of a Sparkfun board fixed all my problems.

I think a lot of people see them as interchangeable – even with the pin .h fix the sparksfun just doesn't work as intended.

Everyone coming to this troubleshooting page, while using a sparkfun board, do yourselves a favor: try your code with an original circuits@home board first.



Philip
[March 31, 2011 at 7:43 pm](#)

I had the same problem until I plugged in an external 9V supply to my Arduino. Sparkfun board is now working perfectly.



Joseph
[July 19, 2011 at 7:20 pm](#)

So far with an Arduino Uno, 9V adapter, and rev 2.0 host shield I've tried 4 different devices (1 mouse, 2 flash drives, Bluetooth adapter). None of them are recognized, doesn't even see them connected. All other tests pass.

I've checked that VBus is there and there is continuity between the D+/D- lines and the Maxim chip. Any ideas on other tests I can perform?



oleg
[July 19, 2011 at 7:54 pm](#)

While extremely rare (I've seen one in ~3000 boards), the transceiver could be dead. Send the shield back, I'll check it and replace with a good one.



Joseph
July 29, 2011 at 2:30 pm

I wanted to bring you up to date on this, especially since it might help others with the same problem.

I didn't believe it was the board especially since I had one of the mini boards I purchased at the same time failing with the same symptoms. I finally figured out that the Arduino Uno I'm using had the wrong voltage regulator installed, putting out 5V on the 3.3V line. A call to DigiKey, some delicate surgery, and the shields are working like champs.



Nathan
September 9, 2011 at 9:50 am

I'm using the Arduino Nano ATmega328 and the corresponding Nano USB Host Shield by Gravitech. I've been debugging/coding for quite some time now. Your libraries have helped immensely, and I'm progressing fairly well. I'm up to the segment where I need to acquire the descriptor values from my bluetooth dongle. I've got a separate code that requires these values to function so I can use it with the Wiimote. I'm currently getting the "failed to assert OSCOKIRQ", and none of the troubleshooting guides or comments seem to help. I changed/rechanged the values in Max3421_constants, and clearly have 7.4V power to the board (soldered the Vin connection). Any help regarding this would be greatly appreciated.



oleg
September 9, 2011 at 10:02 am

From the brief look at the schematic this thing doesn't seem to be compatible with my code.



Nathan
November 5, 2011 at 5:46 pm

It's not compatible with the USB descriptors code? Or is it not compatible with the USB Host Shield libraries? I'm revisiting this project, and am not quite sure what to focus on.



oleg
November 5, 2011 at 5:51 pm

What is not compatible with what? There are two revisions of the shield – rev.2.0 (current) and 1.xx (legacy). Both have supporting library and USB descriptor examples. Take a look at <https://github.com/felis/>



Nathan
November 7, 2011 at 4:38 pm

Sorry for the ambiguity.
I'm pretty sure my problem is with the pins for the USB Host Shield for Arduino Nano. I need to figure out which values to change within the code so that it's compatible. Thanks for the links.



bjam
August 27, 2013 at 9:14 am

Nathan, did you ever get the Gravitech Nano USB host shield working?
Bill



wickedandy
October 31, 2011 at 4:09 pm

I recently purchased a few usb mini shields to make a small pocket sized camera controller. I have successfully prototyped a few and things work great, problem I run into now, is I've gone ahead and created a custom pcb that contains all the arduino/usb shield/power management components, I'm running into an error I can't quite figure out.. I have run the test sketch to check out all my connections and I get the following error when I connect a camera (7D), what is strange is if I connect any other type of device

usb memory stick, mouse, keyboard, they all pass the usb test, but the camera refuses and when I run it on the original prototype that uses the usbshield and arduino pro mini is works correctly...

Circuits At Home 2010
USB Host Shield QC test routine

Press any key to continue...
Reading REVISION register...Die revision 03
Test PASSED
Checking GPIO lines. Install GPIO loopback adapter and press any key to continue...GPIO read/write mismatch. Write: 0
Read: FF

GPIO check failed. Make sure GPIO loopback adapter is installed
SPI test. Each '.' indicates 64K transferred. Stops after transferring 1MB (16 dots)

.....
Test PASSED
Oscillator start/stop test. Oscillator state is ON
Setting CHIP RESET. Oscillator state is OFF
Clearing CHIP RESET. PLL is stable. Time to stabilize – 94 cycles
Test PASSED
USB Connectivity test. Waiting for device connection...
Device connected. Resetting
Reset complete. Waiting for the first SOF...
SOF generation started. Enumerating device.Setup packet error: D
USB state machine reached error state
USB state machine reached error state
USB state machine reached error state
USB state machine reached error state
USB state machine reached error state
USB state machine reached error state
USB state machine reached error state

I have attempted to trouble shoot if there are any bad connections etc, everything is identical electrically to the combo of using the arduino pro mini + usb shield... was wondering if there is anything else I should be checking, etc.. seems the hardware is working properly since other devices pass the test... quite perplexing.



Geoff
December 23, 2011 at 1:40 am

Hi wickedandy

I found your post when googling the USB state machine error you encountered. I have a USB key that throws to that error in the test, but other USB flash drives generate a result without the error.

I decided to reformat the drive (I can't explain the logic either) and Windows complained the drive was write-protected (which it's not). Not sure but perhaps this points to a flakey/faulty USB device rather than an issue with the shield, or other parts of the Arduino setup?

Agreed..perplexing.



Louis Mariani
April 9, 2012 at 1:42 pm

I dont intertund what append with this crazy card !!!

Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 03
SPI long test. Transfers 1MB of data. Each dot is 64K..... SPI long test passed
GPIO test. Connect GPIN0 to GPOUT7, GPIN1 to GPOUT6, and so on
Test failed. Value written: 00 Value read: FF
Press any key to continue...
GPIO test passed.
PLL test. 100 chip resets will be performed
Resetting oscillator
Reset number 0
Reset number 1
Reset number 2
Reset number 3
Reset number 4
Reset number 5

Reset number 78 Time to stabilize – 558 cycles
Reset number 79 Time to stabilize – 555 cycles
Reset number 80 Time to stabilize – 557 cycles
Reset number 81 Time to stabilize – 556 cycles
Reset number 82 Time to stabilize – 555 cycles
Reset number 83 Time to stabilize – 556 cycles
Reset number 84 Time to stabilize – 556 cycles
Reset number 85 Time to stabilize – 557 cycles
Reset number 86 Time to stabilize – 556 cycles
Reset number 87 Time to stabilize – 557 cycles
Reset number 88 Time to stabilize – 556 cycles
Reset number 89 Time to stabilize – 557 cycles
Reset number 90 Time to stabilize – 556 cycles
Reset number 91 Time to stabilize – 556 cycles
Reset number 92 Time to stabilize – 556 cycles
Reset number 93 Time to stabilize – 556 cycles
Reset number 94 Time to stabilize – 557 cycles
Reset number 95 Time to stabilize – 556 cycles
Reset number 96 Time to stabilize – 558 cycles
Reset number 97 Time to stabilize – 556 cycles
Reset number 98 Time to stabilize – 557 cycles
Reset number 99 Time to stabilize – 556 cycles
Reset number 100 Time to stabilize – 557 cycles
Checking USB device communication.

Device connected. Resetting
Reset complete. Waiting for the first SOF...
Getting device descriptor
Descriptor Length: 12
Descriptor type: 01
USB version: 0200
Device class: 00
Device Subclass: 00
Device Protocol: 00
Max.packet size: 40
Vendor ID: 0B27
Product ID: 0165
Revision ID: 0100
Mfg.string index: 01
Prod.string index: 02
Serial number index: 03
Number of conf.: 01

All tests passed. Press RESET to restart test



oleg
April 9, 2012 at 1:53 pm

It looks to me that the shield is working OK.



Lexxus
July 20, 2012 at 11:54 am

Hi!!!

Nice article, Im trying to debug the MAX3421E, its not on the ARduino board, but i cant get the oscillator to assert, if i jump this step im able to write in the gpio registers and also read them... but nothing more... Do u think its becouse of a faulty crystal???

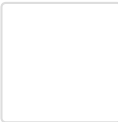
THX



shubham
October 10, 2012 at 2:12 pm

I was using Corsair Venegence M60 with USB host shield, i could read the pressing of button but couldn't read mouse movement(delta_X and delta_Y). I used Oleg code. Can anyone told me how can it be settled. I think there is some problem with buffer but couldn't figure it out.

AleXX

 December 17, 2012 at 7:46 am

Hi Oleg,

Today I measured the frequency of the oscillator, since I was not able to pass the "SPI long test".

The scope shows a "jumping" frequency of 400 kHz around 12Mhz.

Is this usual or can that be the reason, why i am not able to pass the SPI long test on my UNO ?

Thanks and Greetings

AleXX

 oleg

December 17, 2012 at 10:25 am

How did you measure? Probing an oscillator can stop oscillations, you need a low capacitance probe.

Which shield are you having troubles with?

 AleXX

December 17, 2012 at 2:46 pm

Hi, the most recent USB Shield rev. 2.0 from 7th December 2010.

We'll just took the "usual" probes lying around in our lab.

With our analog scope, I even wasn't able to trigger the signal.

With the digital scope I saw a quite nice signal, but jumping around between 11.6 and 12.1 MHz.

Thanks

 oleg

December 17, 2012 at 3:18 pm

Usual probes may or may not work. The oscilloscope is also not a good frequency meter if you need accuracy.

Post here the output of the board_qc test – I'll take a look.

 AleXX

December 18, 2012 at 1:08 pm

Well the setup:

Arduino UNO r3

Powered externally

Arduino IDE 1.0 on Ubuntu 12.04 LTS (if that matters)

Output of board_qc (the original one from git):

Circuits At Home 2011

USB Host Shield Quality Control Routine

Reading REVISION register... Die revision 03

SPI long test. Transfers 1MB of data. Each dot is 64K

Test failed. Value written: 01 read: 00

Unrecoverable error – test halted!!

0x55 pattern is transmitted via SPI

Press RESET to restart test

Output, when commenting the halt55() in long SPI-test:

Circuits At Home 2011

USB Host Shield Quality Control Routine

Reading REVISION register... Die revision 03

SPI long test. Transfers 1MB of data. Each dot is 64K

SPI Test failed. SPI Value written: 01 read: 00

SPI Test failed. SPI Value written: 02 read: 00

SPI Test failed. SPI Value written: 03 read: 00

SPI Test failed. SPI Value written: 04 read: 00

SPI Test failed. SPI Value written: 05 read: 00

SPI Test failed. SPI Value written: 06 read: 00

SPI Test failed. SPI Value written: 07 read: 00

SPI Test failed. SPI Value written: 08 read: 00

SPI Test failed. SPI Value written: 09 read: 00

SPI Test failed. SPI Value written: 0A read: 00

SPI Test failed. SPI Value written: 0B read: 00

SPI Test failed. SPI Value written: 0C read: 00
SPI Test failed. SPI Value written: 0D read: 00
SPI Test failed. SPI Value written: 0E read: 00
SPI Test failed. SPI Value written: 0F read: 00
SPI Test failed. SPI Value written: 10 read: 00
SPI Test failed. SPI Value written: 11 read: 00
SPI Test failed. SPI Value written: 12 read: 00
SPI Test failed. SPI Value written: 13 read: 00
SPI Test failed. SPI Value written: 14 read: 00
SPI Test failed. SPI Value written: 15 read: 00
SPI Test failed. SPI Value written: 16 read: 00

and so on....

Output, if I completely skip the long SPI test by commenting the whole for-loop:

Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 03
GPIO test. Connect GPIN0 to GPOUT7, GPIN1 to GPOUT6, and so on
GPIO Test failed. Value written: 00 GPIO Value read: FF
Press any key to continue...
GPIO test passed.
PLL test. 100 chip resets will be performed
Current oscillator state unexpected.
Press any key to continue...
Resetting oscillator
Reset number 0 Time to stabilize – 388 cycles
Reset number 1 Time to stabilize – 277 cycles
Reset number 2 Time to stabilize – 276 cycles
...
...
...
Reset number 98 Time to stabilize – 276 cycles
Reset number 99 Time to stabilize – 276 cycles
Reset number 100 Time to stabilize – 276 cycles
Checking USB device communication.

Waiting for device

And if I plug in a USB Pen output continues with:

Device connected. Resetting
Reset complete. Waiting for the first SOF...
Getting device descriptor
Descriptor Length: 12
Descriptor type: 01
USB version: 0200
Device class: 00
Device Subclass: 00
Device Protocol: 00
Max.packet size: 40
Vendor ID: 14CD
Product ID: 121E
Revision ID: 0300
Mfg.string index: 01
Prod.string index: 03
Serial number index: 02
Number of conf.: 01

All tests passed. Press RESET to restart test

Hope, that helps
Thanks again



oleg
December 18, 2012 at 1:14 pm

This is quite strange. If the oscillator is out of spec the USB communication is not possible. However, in your case everything seems normal except SPI test (which, BTW, is also dependent on properly working oscillator). Have you tried to run any example sketches yet?

brian

February 8, 2013 at 8:40 am

What is the maximum current that can be drawn from the USB port on the USB Host Shield 2.0? I have a device that requires 450ma.

oleg

February 8, 2013 at 12:45 pm

5V on the shield is connected directly to the 5v rail on Arduino; max.possible current is defined by Arduino's regulator, which is different for different variants but less than what you need. There is a pad labelled 'VBUS', you can connect external supply here. Another option is to mod a regulator on Arduino to provide more current.

brian

February 24, 2013 at 10:26 am

Thanks for the reply! I'm using a Mega 2560 which I believe has a 500ma polyfuse to limit maximum draw.

For your modification, would it be as simple as soldering a header to the 5V point and a ground to the ground hole? An external 5V 1A device would work?

oleg

February 24, 2013 at 11:49 am

You can do this.

Brian

February 24, 2013 at 12:58 pm

Will I need to disconnect the power from the arduino header first to avoid back supply or can I just solder in the new supply directly? Will the shield regulate the additional power to 5v or can I provide more?

oleg

February 25, 2013 at 2:15 am

It is safer to disconnect supplies – there are jumpers for that on the USB Host board.

brian

February 25, 2013 at 5:53 pm

Okay, so if I remove the jumper pad right where it's marked "5V", that will disconnect the incoming 5V, then I can solder on a power connector from my 5V power supply?

oleg

February 25, 2013 at 5:57 pm

This is one way of doing this. Another would be to remove the solder jumper from VBUS PWR 5V and solder external 5V to VBUS pad. Don't forget the ground.

brian

February 25, 2013 at 5:54 pm

Might be nice to have a connection point with bypass on later units in case this comes up again =)

brian

February 25, 2013 at 6:52 pm

Can I use a higher voltage adapter? For some reason most of the 'general' adapters I have are 4.5V or 6V, but not 5V exactly, and I don't have a bench power supply.



oleg
February 25, 2013 at 6:58 pm

The VBUS tolerance is 10%. 4.5V might work. 6V is too much.

Any USB charger outputs 5V as well as many power supplies which come with USB hubs. Both Adafruit and Sparkfun carry 5V supplies.



brian
February 25, 2013 at 7:03 pm

Ah, I should have plenty of those around, I'll give that a try, will report back how this goes tonight. I'm not an electronics person so this will probably end in tears and an order for another USB shield, so win/win for you =)



brian
February 25, 2013 at 8:17 pm

Ah, it worked fine, thanks. I soldered a 3 pin polarized header across the 5V pins (after removing the middle pin in the header) and now I can patch into it with an external supply or just jump across it. Thanks for the

help!



Devin
February 26, 2013 at 1:22 pm

So I'm having a bit of an odd problem. In December I bought the USB host shield with the intention of using a ps3 controller to control my Senior design project. I assembled the host shield, programmed my UNOR3 with the example code PS3USB, connected the shield, and opened the terminal window in the Arduino IDE (1.0.1), and it output: "OSC did not start". I pressed reset on the host shield twice, and it initialized, with the sketch fully functional. Until recently, I've only had to reset once or twice to get it working past initialization.

However, it is no longer working past initialization. So I decided to run the "board_qc" sketch. While running the sketch I used the shields reset several times, giving me this:

```
Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision invalid. Value returned: 00
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test
Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 01
SPI long test. Transfers 1MB of data. Each dot is 64K
Test failed. Value written: 01 read: 00
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test
Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 01
SPI long test. Transfers 1MB of data. Each dot is 64K
Test failed. Value written: 01 read: 00
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test
Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 01
SPI long test. Transfers 1MB of data. Each dot is 64K
Test failed. Value written: 01 read: 00
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test
```

Which is where I'm still stuck. Any suggestions would be greatly appreciated.

oleg

February 26, 2013 at 1:26 pm



Something is wrong with your setup – I can see errors on a serial link (line 2), which has nothing to do with USB Host. What happens if you try to run board_qc with USB Host Shield removed from Arduino?



Devin
February 26, 2013 at 1:50 pm

As requested, without the shield connected, and after pressing UNOR3 reset once I get:

Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision invalid. Value returned: FF
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test
Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision invalid. Value returned: FF
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test

From my experience (and I'm pretty new so I may be wrong) I've seen error's, like the one on line 2, when chip is transmitting slower than the baud set in the code.



oleg
February 26, 2013 at 2:48 pm

There is something wrong with SPI lines. Also, the serial errors like that won't usually happen on Arduino board – you can get garbage in the very beginning of a transmission, but not in the middle. Try powering Arduino from external supply to see if it makes any difference. Also, if the board is dirty, give it a good IPA wash.



Devin
February 26, 2013 at 4:58 pm

If it helps, I used a different UNO (fresh out of the box clean) with external power source (9V 1A to barrel jack) that gives me pretty much the same output (minus the transmission garbage):

Circuits At Home 2011
USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 01
SPI long test. Transfers 1MB of data. Each dot is 64K
Test failed. Value written: 01 read: 00
Unrecoverable error – test halted!!
0x55 pattern is transmitted via SPI
Press RESET to restart test



oleg
February 26, 2013 at 10:25 pm

Something is not right. Die revision can't be 01, I started producing this shield at the end of rev.02 – in 2009. If yours is less than a year old, MAX3421E should be rev.03. It could be SPI but broken SPI usually returns either 00 or 0xff.

Take a good picture of your solder job and send it to me at mazurov at circuitsathome dot com – I'll take a look.



mohmad
March 29, 2013 at 9:30 am

ps3 library started
getDevDescsr

btd init fails,error code 15
please tell me whats wrong,thanks

ascariel

June 30, 2014 at 12:57 pm

Hi! I've been trying to get the USH Host Shield MINI to work with the arduino Micro board (pinout: http://arduino.cc/de/uploads/Main/ArduinoMicro_pinout.png)

I connected the following pins (on a breadboard) all GNG's, RESET to RESET, MOSI, MISO, SS, SCK to the corresponding ones as described on the pinout above and 3.3V.

When I tried to test the setup with the board_qc sketch, I get error code 00 at the die test. Maybe this is because the INT pin of the USB shield is not connected? (don't know where)

I already did a lot of google searches regarding micro and usb host, but didn't come up with a helpful site, so if you know any, please help!

Oleg Mazurov

June 30, 2014 at 1:01 pm

you don't need INT for this step. It reads die revision from the chip and gives you error if it gets anything that won't resemble one. Most likely, the SPI is connected incorrectly, take a look at this article for troubleshooting ideas -> <https://www.circuitsathome.com/mcu/arduino/usb-shield/troubleshooting-arduino-usb-host-shield>

ascariel

June 30, 2014 at 1:34 pm

I read this article and already checked the 3.3V power supply points, they are working fine. Unfortunately I don't have an oscilloscope, so I can't check the rest at home.

Is it possible that I need to define/change something for the arduino micro in the avrpins.h library?

Oleg Mazurov

June 30, 2014 at 9:41 pm

Just checked – Arduino micro is 5V, it is not compatible with USB Host mini. You might have fried the SPI on the shield.

ascariel

July 1, 2014 at 2:27 am

Thx, I didn't think of that!

Wan

July 10, 2014 at 7:28 pm

Hello there..

I want to build a wireless USB extender

(http://www.officespecialties.com/cables_to_go_29571_trulink_wireless_usb_device_adapter_kit_79856_prd1.htm) which are expensive and limited range. If a device attached to usb host, is it possible for the PC to read it as device itself in com port rather than read it as arduino?

Oleg Mazurov

July 10, 2014 at 8:18 pm

The USB Host controller on a shield needs an MCU to operate. It doesn't have a COM port either, only SPI.

Lawrence

August 19, 2014 at 4:14 am

I am integrating a USB shield and an Arduino UNO. End goal = Control a MIDI device via USB. This is what I have done. I ran the "board_qc" sketch and if I connect an USB thumbdrive the tests all pass. But when I connect a MIDI device and run the following sketch ..

It fails at this step – "if(Usb.getUsbTaskState() == USB_STATE_RUNNING)" ... any thoughts on how I can find what is wrong and get this working.


```

#include
#include

USB Usb;
USBH_MIDI Midi(&Usb);

byte Preset;

int inPin1 = 2;
int outPin1 = 3;
int outPin2 = 4;
int state1 = LOW;
int reading1;
int previous1 = LOW;
long time = 0;
long debounce = 200;

void setup()
{
  pinMode(inPin1, INPUT);
  pinMode(outPin1, OUTPUT);
  pinMode(outPin2, OUTPUT);
  pinMode( 7, OUTPUT); // For the sheild
  digitalWrite( 7, HIGH); // for the sheild

  if (Usb.Init() == -1)
  {
    digitalWrite(outPin2, HIGH);
    while(1); // Halt
  }
  delay(200);
  Preset = 1;
}

void loop()
{
  reading1 = digitalRead(inPin1);
  {
    if (reading1 == HIGH && previous1 == LOW && millis() - time > debounce) {
      if (state1 == HIGH)
      {
        Preset=9; // Send program change for BYPASS / no effect
        SendMIDI(Preset);
        state1 = LOW;
      }
      else
      {
        Preset=1; // Send program change for EFFECT - 1
        SendMIDI(Preset);
        state1 = HIGH;
      }
      time = millis();
    }
    digitalWrite(outPin1, state1);
    previous1 = reading1;
  }
}

// Send "Program Change" MIDI Message

void SendMIDI(byte number)
{
  Usb.Task();
  if( Usb.getUsbTaskState() == USB_STATE_RUNNING )
  {
    byte Message[2]; // Construct the midi message (2 bytes)
    Message[0]=0xC0; // 0xC0 is for Program Change (Change to MIDI channel 0)
    Message[1]=number; // Number is the program/patch (Only 0 to 99 is valid for ZOOM G3)
    Midi.SendData(Message); // Send the message
    delay(10);
  }
  else
  {
    digitalWrite(outPinErr, HIGH);
  }
}

```

```
delay (1000);  
digitalWrite(outPinErr, LOW);  
delay (500);  
digitalWrite(outPinErr, HIGH);  
}  
}
```



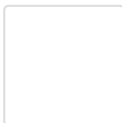
Oleg Mazurov
August 19, 2014 at 10:48 am

Try running `Usb.Task()` from the `loop()`.



Lawrence
August 19, 2014 at 11:50 pm

Oleg, Thanks a ton for replying... NO CHANGE. Same thing. it fails at " if(
`Usb.getUsbTaskState() == USB_STATE_RUNNING`)" "



Oleg Mazurov
August 20, 2014 at 10:21 am

Can you run successfully any of the examples in the library?



Lawrence
August 20, 2014 at 11:11 am

If I plug in an USB drive and run the `board_qc`, I get all tests succesful message but if I connect the MIDI device that accepts messages via USB, I get a value of 160 returned when I run "`Serial.println(Usb.getUsbTaskState());`,"



Oleg Mazurov
August 20, 2014 at 1:07 pm

Is this MIDI device bus or self-powered? How do you power your Arduino board – from USB or external power supply?



Lawrence
August 20, 2014 at 11:49 pm

The MIDI device can be powered via USB or a stand alone power supply to it. I tried all options (1) using USB power (2) powering the MIDI device by an external power supply and (3) Using a powered hub for power. I power the Arduino board via USB at most times.



Oleg Mazurov
August 21, 2014 at 6:43 pm

Is it specific to that particular MIDI device? In other words, out of all USB devices you have, how many would pass `board_qc` ? Could be something as simple as reset timing; try numbers larger than 20 here -> https://github.com/felis/USB_Host_Shield_2.0/blob/master/Usb.cpp#L501



Lawrence
August 25, 2014 at 2:19 am

Sure sir.. will try this and let you know.



Lawrence
August 20, 2014 at 11:30 am

With this device plugged in, and when I run the `board_qc`,
Circuits At Home 2011

USB Host Shield Quality Control Routine
Reading REVISION register... Die revision 03
SPI long test. Transfers 1MB of data. Each dot is 64K..... SPI long test passed
GPIO test. Connect GPIN0 to GPOUT7, GPIN1 to GPOUT6, and so on
Test failed. Value written: 00 Value read: FF
Press any key to continue...
GPIO test passed.
PLL test. 100 chip resets will be performed
Resetting oscillator

Reset number 0 Time to stabilize – 553 cycles
-> goes on for all numbers
Reset number 99 Time to stabilize – 554 cycles

Checking USB device communication.

Reset complete. Waiting for the first SOF...
USB state machine reached error state



Sam Hay
October 19, 2015 at 9:33 am

hreat article. i just have 2 question:

1. what is a loopback adapter? is it just straight connections or is it an adapter with circuitry inside?
 2. my device reach the statement: "Device connected. Resetting" and it never moves beyond this point. why is that?
- thanks a lot



Oleg Mazurov
October 19, 2015 at 11:03 am

1. the loopback connects gpio0 to gpio7, gpio1 to gpio6, etc.
 2. This is likely caused by insufficient power.
-