Can the Arduino Pro Micro Board be recognized by gaming console

thepaddock **avr. '18 #1**

I built a racing sim button box and have used the pro mico arduino board to program and configure. It works perfectly with a PC but I would like to make it compatible with consoles such as the xbox One and the PS4. I figured I could install a library which could address the compatibility so I installed the USB Host Shield Library 2.0 by Oleg Mazerov yet the ps4 console does not recognize. I believe the PS4 and xBox One recognizes USB keyboard connections so I am not sure what I am missing here.

I do see that my sketch does not have a keyboard entry but rather a, keypad. If this is my missing variable, would I simply add or is there coding involved?

include

include

Any advice or direction would be appreciated.

Grumpy_Mike Karma: 2500+ **avr. '18 #2**

> so I installed the USB Host Shield Library

Do you have a Host Shield? That is an extra piece of hardware. 1) Without the hardware it is useless. 2) Even with the hardware it is useless because your PS4 IS a host device and you can't connect two host devices together. You can only connect a Host to a client.

What you need to know is what sort of HID device your PS4 is looking for and find a driver for that, using an Arduino Leonardo or Micro, or even Teensy.

---

thepaddock **avr. '18 #3**

Ya, it would be of no use then...thx. The ps4 and xbox would need to see the Arduino as a USB Keyboard in order to work correctly. I know when I connect the box to a PC it is identified as a game controller/joystick. I am not sure how to make it identify as a keyboard unless it is as simple as adding the tag in the sketch code which I assume it's not that simple.

Grumpy_Mike Karma: 2500+ **avr. '18 #4**

> I do see that my sketch does not have a keyboard entry but rather a, keypad. If this is my missing variable, would I simply add or is there coding involved?

Yes there is coding involved, the libiary just adds calls that can be made by your software to do things. Adding a libiary and then not using it is exactly the same as not adding it at all. In fact the compiler is clever enough to know you are not using it and doesn't even include the code from the libiary in the final code.

Once you install a libiary you can look at the example code that comes with it from the menu File -> Examples and then scroll down to find your libiary, and then off again to the examples. That will tell you how to use the calls in the libiary.

---

pert **Arduino Team** **avr. '18 #5**

You can find documentation on the Keyboard library here: **https://www.arduino.cc/en/Reference/MouseKeyboard** You can find example sketches for it at **File > Examples > 09.USB > Keyboard**. You can find tutorials for the example sketches here: **https://www.arduino.cc/en/Tutorial/BuiltInExamples#usb**

---

thepaddock **avr. '18 #6**

Thank you very much. I will have a look and see what I can come up with.

---

gdsports **mai '18 #7**

The problem with DIY controllers is genuine Xbox or PS4 controllers have security chips to prevent the use of unauthorized controllers. This makes it difficult to build custom controllers for people with limited mobility. Microsoft recently addressed this with an Xbox controller with inputs for external buttons, switches, and joysticks.

https://www.xbox.com/en-US/xbox-one/accessories/controllers/xbox-adaptive-controller

As for Arduino DIY console controllers, it might be possible by connecting a genuine controller to a USB host mini board which is then connected via SPI to the teensy 2. The teensy must pass through all controller input and output to the console as well as input events generated by buttons attached to the teensy. When the console sends security check to the controller, the teensy must pass it through to the genuine controller. When the controller responds, the teensy must pass the response to the console.

There is a note at pjrc.com on how to connect a USB host mini board to a teensy 2.

https://www.pjrc.com/teensy/td_libs_USBHostShield.html

The USB host shield library includes drivers for Xbox and PS3/4 controllers.

```
console <-> teensy <-> USB host mini board <-> genuine controller
```

The teensy must be made to look like, at the USB descriptor level, like an Xbox or PS4 controller. But perhaps someone as already done so. I have never tried this so this it may be even more complicated than I have indicated.

Back to top