# ATMakers Feedback Report for XAC Controller (Feb. 2019)

## USB Host Incompatibility *(Critical)*

The worst challenge with working with the XAC's USB ports was the narrow scope of its HID support. To get around this, I had to write custom modifications to the core USB code used in CircuitPython and to deploy my solution on that build.

In the end, the custom build exposed a single HID Gamepad device that exactly mimicked the report given by the PDP joystick device. This gave us a working solution but at the expense of many features and with many limitations.

Because I believe this is the biggest hurdle to widespread adoption of USB XAC devices, I've tried to list the issues that I know were problematic.

### Composite HID Devices Not Supported

The XAC does not seem to support composite HID devices. Specifically, even if the PDP Joystick's report were provided with no modifications, the XAC would fail if the device also exposed a USB Keyboard, Mouse, Medial controls, etc.

This is a critical problem because today's "Maker" boards (MakeCode, CircuitPython, MicroPython, Lua, MakeyMakey, Arduino) are all capable and, in fact, many default to exposing more than one HID device. In my case, I was using CircuitPython, which has no way to disable the individual HID types.

My solution was to manually change the CircuitPython interpreter to change its HID Report to only expose the single HID device that matched the PDP. ***Anyone who wishes to use my solutions must, at the current time, use my modified build of the CircuitPython interpreter***.

### Joystick Ranges Limited

It seems that the ranges of the joysticks are limited to 0-255. That might not be the case, but the CircuitPython standard of -127 – 127 definitely does not work, and larger sized reports such as 0-65535 seemed to fail as well. This will be a problem with many joysticks/gamepads that use standard HID libraries.

According the USB spec, gamepads can define their Logical Ranges and the host should map that onto their internal representation. The XAC seems to reject custom ranges. Once this was understood, it was fairly simple to change the mapping in the CircuitPython interpreter: however, determining this took a great deal of time. If it cannot be changed, it should be documented.

### Unused Joysticks/Buttons not ignored

Similarly, the XAC seemed to fail when connected to an HID Gamepad with a second joystick or more than 8 buttons. While it's not a requirement to support the actions on additional joysticks or buttons, a USB host should not fail when they are present.

## Button/Joystick Mapping Limitations

While the above issues were critical and will stop adoption, this recommendation would simply increase functionality. They could be considered feature requests, but they are putting in place significant constraints on AT solutions.

The XAC chooses to map its buttons and joysticks to the two USB ports so that each side gets approximately half of the activations. This is understandable but problematic, and there is a much better solution.

I believe the goal of this design is to allow the same device type (for example a PDP or Thrustmaster) to plug into each side giving a joystick to each hand, etc without remapping the buttons. This is a laudable and important goal.

The current design places a single joystick on each side along with 8 buttons.

A better mapping would support up to two joysticks, two additional analog inputs (which could be described as a third joystick), and 16 buttons on each port and simple change the order that the buttons and sticks are mapped. Here is a sample mapping that would support those goals

### Revised Mapping *(Dark cells are additions)*

| **LEFT PORT** | | **RIGHT PORT** | |
|---|---|---|---|
| 1st Joystick | Left Joystick | 1st Joystick | Right Joystick |
| Button 1 | X1 | Button 1 | View |
| Button 2 | X2 | Button 2 | Menu |
| Button 3 | ThumbBtnL | Button 3 | ThumbBtnR |
| Button 4 | BumperL | Button 4 | BumperR |
| Button 5 | A | Button 5 | X |
| Button 6 | B | Button 6 | Y |
| Button 7 | View | Button 7 | X1 |
| Button 8 | Menu | Button 8 | X2 |
| 2nd Joystick | Right Joystick | 2nd Joystick | Left Joystick |
| Button 9 | ThumbBtnR | Button 9 | ThumbBtnL |
| Button 10 | BumperR | Button 10 | BumperL |
| Button 11 | X | Button 11 | A |
| Button 12 | Y | Button 12 | B |
| Button 13 | XBox Button | Button 13 | XBox Button |
| Analog Input 5 | Left Trigger | Analog Input 5 | Right Trigger |
| Analog Input 6 | Right Trigger | Analog Input 6 | Left Trigger |

### Rationale

This mapping change would allow a single smart device (Raspberry Pi, Feather, Arduino, Teensy, etc.) to send gamepad events to either side of the device. For example

- A head tracking device could send Left Stick Up/Down and Right Stick Left/Right.
- Sling-Ring style devices could send A/B/X/Y/Left-Btn/Right-Btn with one finger.
- Analog input to the Right-Trigger/Left-Trigger could be sent which is impossible today.

This change is backwards-compatible with the current mapping.

# Features requested at ATIA 2019

These were things requested by AT users at ATIA 2019 with my thoughts on why they'd be important. I don't think they are critical, but they are worth sharing

## HID Mouse & Keyboard Support

*Ignoring* Mouse and Keyboard profiles and supporting composite devices is critical (see above). However, we also got requests that you *Support* HID Mice and Keyboards. It would be helpful to have a way to connect a mouse and have that map to the first joystick and buttons in order. This would require publishing an additional mapping and determining what to do with scroll wheels, etc, but it would be very useful.

In addition to allowing standard devices, it's important that Microsoft understand that there are many devices in AT that emulate mice and keyboards. Switch interfaces look like keyboards, and things like "head mice" are in use that would allow folks to use the same device that they use on their PCs.

## Button Chording support

Many users have a limited number of activation points (places they can use to activate a button. One way to solve this is chording. Defining actions that occur on combinations of inputs would be helpful.

The SlingRing solution by Chris Young takes 3 microswitches and maps the combinations of these to 6 separate buttons. The TwoSwitch also maps two switches to more than two outputs based on combinations and states. These could in theory be done within the controller itself-but it would require changes to your configuration UI.

# Side Notes

## USB Hubs do not seem to work

This might be my particular hubs, but I couldn't get either side to accept a device through a hub.

## Error reporting/logging non-existent

I would LOVE to know how to read error logs on the XBox. Or to setup a remote log server for it. Anything to give any feedback about why the device is not working. Perhaps this is me, but if it's there, it's hard to find.

## (Public) documentation lacking

The documentation Bryce sent me was great – but it is not public. I know that's not the right style for public support documents, but having an "Integrators Guide" or "Makers Guide" that is exactly the data in that document would be very valuable. If you don't, I will give you a heads up that Adafruit (or myself) will probably publishing some limited documentation showing the parts that are easily derived like the TRRS pin mappings, the requirements on the USB HID Report, etc.