Coleco USB Mouse/Joystick Adapter

Thank you for your purchase of the Coleco USB Mouse/Joystick Adapter. Any questions or concerns should be addressed to Lundy Electronics on the ADAM Facebook group at https://www.facebook.com/groups/115531017844. More specifics and operational documentation, as well as any future code updates, can be found on the official GitHub page at https://github.com/JohnLundy/Coleco-ADAM-USB-Mouse.

We designed the Coleco USB Mouse Adapter for fellow ADAM hobbyist who might not have the skill set to create their own USB mouse adapter or would like a turn-key, professionally built unit.

As with many hobbyist projects, it comes with no guarantee for 100% functionality. *Every effort has been made to ensure the best compatibility possible through many hours of design and testing. Use this device at your own risk. Lundy Electronics cannot be held responsible for any damage to the device or your ADAM or the loss of data that my occur.*

Thank you.

Lundy Electronics

What is in the box

- Coleco USB Mouse Adapter
- Joystick splitter ribbon cable
- Mini-B USB cable (power module not included)

Device Features

- Designed using the popular Arduino Nano
- Works with all modern, off-the-shelf USB mice, as well as wireless mice
- Ability to change mouse sensitivity and delay balance on the fly for better compatibility with a wide range of software titles
- Support for third party NES and SNES USB joypads has been added

Design Challenges and Features

The ADAM and Colecovision use two separate commons for joystick switch control, and neither is connected to true ground. Both commons are strobed with a 350ms pulse width with a low of -0.7V and a high of 2.8V. This required the use optocoupler ICs that isolate the controller signals that otherwise cannot be directly connected to the ADAM to simulate controller switch closures.

Both systems never had any software titles specifically designed for mouse control which added an extra layer of complexity to the design to compensate for the lack of any kind of standard among the wide range of software titles. Using a 10K potentiometer on the analog (A0) line of the Arduino Nano creates a voltage range that can be sensed and manipulated to adjust mouse sensitivity and joystick delay balance on the fly. Based on extensive testing, this design works well to compensate for the wide variety of joystick control sensitivities among different software titles. A special button to enable/disable a dedicated precision mode was added for ADAM users for paint programs needing precision movements that otherwise wouldn't work for games.

A workaround was also needed for games that required the joystick to be held down during play. This capability is accomplished by pressing the center mouse button to toggle on/off an increased delay simulating holding the joystick down in any specific direction.

How to Connect

Insert the supplied joystick controller splitter ribbon cable to the ADAM joystick port and connect the other end to the Coleco USB Mouse 10-pin header on top of the device. See *Figure 1*. Note the position of Pin-1 of the ribbon cable. See *Figure 2*. The extra male DB-9 connector on the ribbon cable offers connectivity for a factory Coleco joystick to use alongside the Coleco USB Mouse. This is useful for when the keypad is needed while using the mouse at the same time. Power the Coleco USB Mouse adapter with the supplied Mini-B USB Cable and your power source. The cable can be connected to a PC USB port to supply power, or use a quality cell phone/tablet USB power module (not included) to supply power. Insert a USB mouse to the front USB port of the Coleco USB Mouse adapter. The red LED of the USB mouse when lifted should now be illuminated to indicate power has been applied to the mouse and the Coleco USB Mouse adapter is now ready for use.

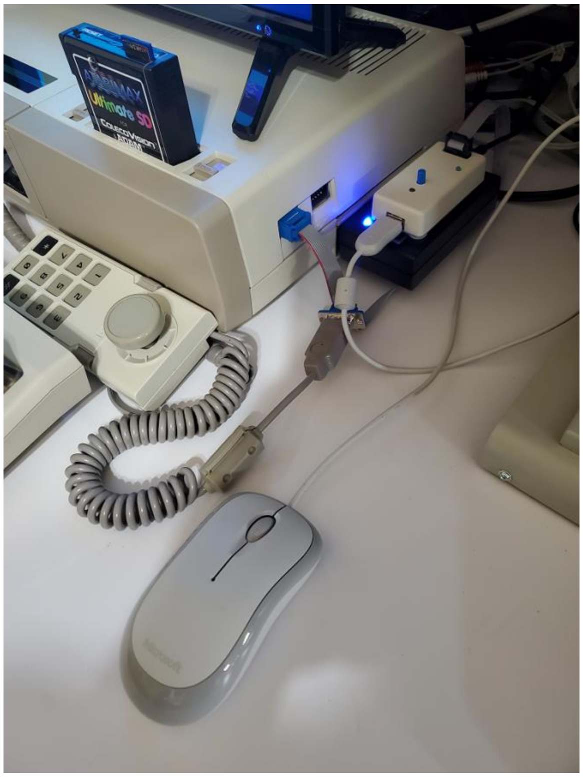

Figure 1

How it works

- The ADAM USB Mouse assembly is designed to have mouse USB connector facing the user. Note the location of the trimmer resistor (potentiometer), small button switch, and LED. These three features are how the ADAM USB Mouse will be operated, as well as the center button of the mouse.
- The position of the potentiometer is how to adjust mouse sensitivity and joystick delay balance on the fly. *Adjustments will be stated in "o'clock" position for future reference*. A counterclockwise turn will increase mouse sensitivity and decrease joystick delay, while a clockwise turn will decrease mouse sensitivity and increase joystick delay.
- The small button switch was added to toggle a dedicated precision mode for ADAM users for paint programs that need precision movements that otherwise wouldn't work for games. It will

override any positional setting of the potentiometer and will illuminate the LED to indicate the mode is active.

- The center mouse button will toggle on/off an increased delay to simulate holding the joystick down in any specific direction for games that require this functionality. It is also useful for creating straight lines in paint programs and for scrolling lists.

Example Settings

The range is eight to four o'clock positions. Many games and programs use the two o'clock position, which is usually a good starting point for testing. *Figure 2* shows the device in the twelve o’clock position.

Game/Program	O'clock Position
ADAM BOMB	2
ADAM BOMB 2	1-2
AE	2-3
Beamrider	2
Blockade Runner	2
Boulder Dash	8
Buck Rogers	2-3
Carnival	2
Centipede	9
Cosmic Avenver	2
Galaxian	2
Gorf	2
Gyruss	2
KABOOM!	8
Mecha 8	3
Mecha 9	3
Meteoric Shower	12
Operation Wolf	2
Space Fury	8
Spectron	4
Star Force	2
Star Trek	8
Strike It!	Precision
Super Cross Force	8
Threshold	8
Time Pilot	8
ADAMCalc	2
PaintMASTER	8 and use precision mode when needed
PowerPAINT	8 and use precision mode when needed

Loading Future Code Updates

Arduino IDE method

1. Install the Arduino IDE (https://www.arduino.cc/en/software) and configure it for Board type "Arduino Nano" and Processor type "ATmega328P" under the Tools drop-down menu.
2. Install the "USB Host Library Rev. 2.0" library by Oleg Mazurov from the IDE Tool drop down menu and select "Manage Libraries..."
3. Download the latest Coleco ADAM USB Mouse source code from the GitHub repository and load it in the Arduino IDE.
4. Connect the supplied Mini-B USB cable from the Coleco USB Mouse to the PC. Select the correct Port COM setting under the Tools drop-down menu.
5. Press the Upload button in the Arduino IDE to compile and upload.

XLoader method

1. Install XLoader (https://www.hobbytronics.co.uk/arduino-xloader, Windows only) and configure it for Device type "Uno(ATmega328)" and a Baud rate of 115200.
2. Download the latest Coleco_USB_Mouse.hex from the GitHub repository and select [...] to load it in XLoader.
3. Connect the Coleco USB Mouse to the USB port, and select the correct COM port in the drop-down.
4. Press the Upload button to load code.

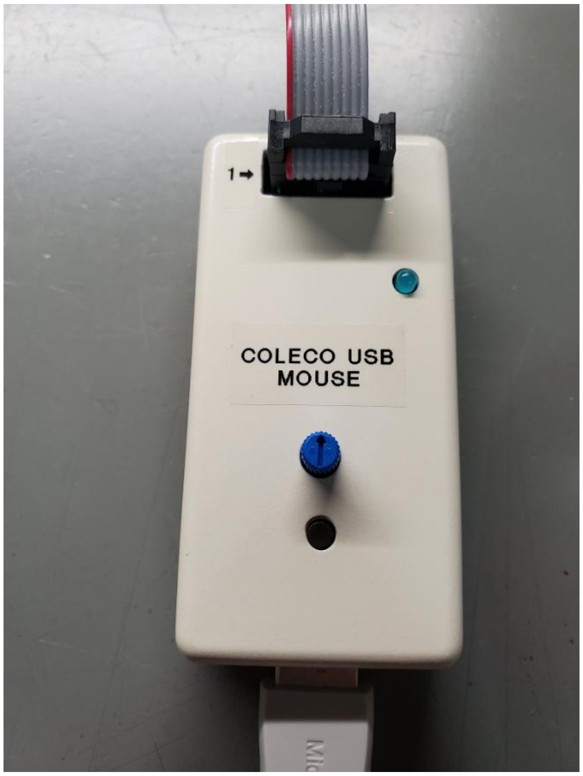

Figure 2

Have fun!