

# USB CDC Host Serial Ports (COM3-COM6) User Manual

## Overview

USB builds of the PicoMite firmware (PicoMiteUSB, PicoMiteVGAUSB, PicoMiteHDMIUSB, and the RP2350 equivalents) can act as a USB host for up to four USB CDC (Communications Device Class) serial devices. These appear in BASIC as COM3, COM4, COM5, and COM6.

This allows a PicoMite with a USB host port to communicate with any device that presents a standard USB CDC serial interface -- including other PicoMites running non-USB firmware, USB-to-UART adapters (CH340, CP2102, etc.), and microcontroller boards with CDC endpoints. Devices may be connected directly or through a USB hub.

COM1 and COM2 continue to use the hardware UARTs. COM3-COM6 are available only on USB builds.

## Requirements

- A USB build of PicoMite firmware (any build with USBKEYBOARD defined).
- The connected device must present a standard USB CDC ACM interface.
- A standard PicoMite (non-USB build) is a CDC device by default and works without any special configuration on the device side.
- The USB device must be plugged in and enumerated before the OPEN command is issued. TinyUSB typically takes 1-2 seconds after power-on to enumerate a newly connected device.

## Channel and Port Mapping

Each CDC interface discovered by TinyUSB is assigned a fixed index (0-3). The mapping to COM ports and internal channel numbers is:

COM Port	CDC Index	Channel	Notes
COM3	0	5	First CDC device enumerated
COM4	1	6	Second CDC device
COM5	2	7	Third CDC device
COM6	3	8	Fourth CDC device

Channels 1-4 are reserved for USB HID devices (keyboard, mouse, gamepads). CDC channels start at 5 to avoid any conflict.

When a CDC device is connected or disconnected at the command prompt, a message is displayed:

```
USB CDC Device Connected on channel 5 (COM3)
```

```
>
```

```
USB CDC Device Disconnected on channel 5 (COM3)
```

```
>
```

These messages are suppressed while a program is running.

## Syntax

### Opening a Port

```
OPEN "COMn:baud" AS #fnbr
```

```
OPEN "COMn:baud,buf_size" AS #fnbr
```

```
OPEN "COMn:baud,buf_size,interrupt,int_level" AS #fnbr
```

Where:

# USB CDC Host Serial Ports (COM3-COM6) User Manual

- n is the port number: 3, 4, 5, or 6
- baud is a required baud rate value (see note below)
- buf\_size is the optional receive buffer size in bytes (default 1024)
- interrupt is the optional name of a BASIC subroutine to call when data arrives
- int\_level is the number of characters in the buffer that triggers the interrupt

The trailing colon after COMn is required, just as for COM1 and COM2.

Baud rate note: For PicoMite-to-PicoMite connections (pure CDC), the baud rate is accepted but has no effect -- USB CDC operates at USB bus speed. For USB-to-UART bridge devices (CH340, CP2102, FTDI, etc.), the baud rate is sent to the device via the USB SET\_LINE\_CODING request and controls the physical UART baud rate on the bridge's serial output. The line format is always 8 data bits, 1 stop bit, no parity.

A default line coding of 115200 baud, 8N1 is sent automatically when any CDC device is first enumerated. The baud rate specified in OPEN overrides this default.

## Closing a Port

```
CLOSE #fnbr
```

Closing the port releases the receive buffer, clears any interrupt, and de-asserts DTR/RTS on the USB interface.

## Parameters

Parameter	Description
baud	Baud rate for the connection. For USB-UART bridges (CH340, etc.) this sets the physical UART speed via SET_LINE_CODING. For PicoMite-to-PicoMite CDC links it is accepted but has no effect. Default enumeration value is 115200.
buf_size	Receive buffer size in bytes. Default is 1024. Increase for high-throughput applications.
interrupt	Name of a BASIC subroutine to call when the receive buffer reaches int_level characters.
int_level	Number of characters in the receive buffer that triggers the interrupt. Must be between 1 and buf_size.

## Reading and Writing

Once opened, COM3-COM6 work identically to COM1/COM2 with standard file I/O:

```
' Send a string
PRINT #1, "Hello from host"

' Send a single character
x = 65
PUT #1, x

' Read a line (blocks until CR received)
LINE INPUT #1, response$

' Non-blocking check for available data
IF LOC(#1) > 0 THEN
```

# USB CDC Host Serial Ports (COM3-COM6) User Manual

```
LINE INPUT #1, a$
END IF

' Read a single character
c = ASC(INPUT$(1, #1))

' Check if data is available
n = LOC(#1)
```

## Line Ending Behaviour

PRINT #n to a CDC port sends a carriage return only (CR, 0x0D). The linefeed (LF, 0x0A) that is normally appended for file and UART ports is suppressed. This matches the line ending convention expected by a PicoMite console on the receiving end.

If you need to send CR+LF explicitly:

```
PRINT #1, "text" + CHR$(10);
```

## Interrupts

COM3-COM6 support receive interrupts, exactly like COM1 and COM2. When the number of characters in the receive buffer reaches the specified level, the nominated subroutine is called.

```
OPEN "COM3:9600,1024,OnData,1" AS #1
' ...
DO
' main loop - interrupt fires when data arrives
LOOP

SUB OnData
LINE INPUT #1, msg$
PRINT "Received: " msg$
END SUB
```

The interrupt parameters are:

- The subroutine name (e.g. OnData) -- must be a valid BASIC SUB
- The trigger level (e.g. 1) -- the interrupt fires when LOC(#fnbr) reaches this value

Interrupts are checked during the normal MMBasic polling cycle, not at hardware interrupt level. Latency depends on what the main program is doing.

## Transparent Reconnect

If a USB CDC device is physically disconnected while its COM port is open, the port remains open in BASIC. Any attempt to send data (PRINT, PUT) is silently ignored while the device is absent.

When the same device (or another CDC device) is plugged back into the same USB port and enumerates at the same CDC index, the firmware automatically re-asserts DTR/RTS and communication resumes. No BASIC code changes or CLOSE/OPEN cycle is needed.

This makes it possible to write programs that tolerate brief cable disconnections without error handling.

Note: If the device enumerates at a *different* CDC index after reconnection (e.g. because it was plugged into a different

# USB CDC Host Serial Ports (COM3-COM6) User Manual

hub port), it will appear on a different COM port number. The original COM port will remain open but non-functional until closed.

## Error Messages

Error	Cause
Invalid COM port	The port specifier is not COM1: through COM6: (or COM3-COM6 on a non-USB build).
Already open	The COM port is already opened by another OPEN statement.
No USB CDC device on channel N for COMn	No CDC device is currently enumerated at the required index. Check that the device is plugged in and has finished enumerating.
COM specification	The format of the COM specifier string is invalid.

## Examples

### Basic Send and Receive

```
' Open COM3 with default 1024-byte buffer
OPEN "COM3:9600" AS #1

' Send a command to the connected device
PRINT #1, "Hello"

' Wait for and read the response
DO WHILE LOC(#1) = 0 : LOOP
LINE INPUT #1, response$
PRINT "Got: " response$

CLOSE #1
```

### Two-Way Communication with Interrupt

```
' Host PicoMite program - communicates with a device on COM3
DIM msg$ LENGTH 128

OPEN "COM3:9600,1024,OnReceive,1" AS #1

PRINT #1, "STATUS"

DO
  k$ = INKEY$
  IF k$ = "q" THEN EXIT DO
  IF k$ <> "" THEN PRINT #1, k$;
LOOP

CLOSE #1
END

SUB OnReceive
```

# USB CDC Host Serial Ports (COM3-COM6) User Manual

```
LINE INPUT #1, msg$
PRINT "[Device] " msg$
END SUB
```

## Larger Buffer for High Throughput

```
' Use a 4096-byte receive buffer for bulk data transfer
OPEN "COM3:9600,4096" AS #1

DO WHILE LOC(#1) = 0 : LOOP

' Read all available data
DO WHILE LOC(#1) > 0
  LINE INPUT #1, dat$
  PRINT dat$
LOOP

CLOSE #1
```

## Multiple CDC Devices

```
' Communicate with two CDC devices simultaneously
OPEN "COM3:9600" AS #1
OPEN "COM4:9600" AS #2

PRINT #1, "IDENTIFY"
PRINT #2, "IDENTIFY"

PAUSE 500

IF LOC(#1) > 0 THEN
  LINE INPUT #1, id1$
  PRINT "Device on COM3: " id1$
END IF

IF LOC(#2) > 0 THEN
  LINE INPUT #2, id2$
  PRINT "Device on COM4: " id2$
END IF

CLOSE #1
CLOSE #2
```

## USB-UART Bridge (CH340, CP2102, etc.)

A USB-to-UART adapter plugged into the PicoMite's USB host port appears as a CDC device. The baud rate in the OPEN command sets the physical UART speed on the bridge:

```
' Talk to an external device at 9600 baud via a CH340 USB-UART adapter
OPEN "COM3:9600" AS #1

PRINT #1, "AT"
```

# USB CDC Host Serial Ports (COM3-COM6) User Manual

```
DO WHILE LOC(#1) = 0 : LOOP
LINE INPUT #1, reply$
PRINT "Reply: " reply$

CLOSE #1
```

The default baud rate sent on enumeration is 115200. If your external device uses a different speed (e.g. 9600), specify it in the OPEN command.

## Device-Side Program (Standard PicoMite)

This program runs on a standard (non-USB) PicoMite that is connected via USB to the host PicoMite. No special configuration is needed on the device side -- it simply reads from and writes to the console:

```
' Device-side: echo back whatever the host sends, prefixed with "ACK:"
DO
  IF LOC(#0) > 0 THEN
    LINE INPUT a$
    PRINT "ACK:" a$
  END IF
  PAUSE 10
LOOP
```

## Limitations

- USB builds only. COM3-COM6 are not available on non-USB firmware builds (PicoMite, PicoMiteVGA, PicoMiteHDMI without the USB suffix).
- Maximum four CDC devices. The TinyUSB host stack is configured for up to 4 simultaneous CDC interfaces.
- No flow control. Hardware flow control (CTS/RTS) is not used for data flow. DTR/RTS are asserted when the port is opened and de-asserted when closed, but they serve as a ready signal only.
- Fixed line format. The line coding is always 8 data bits, 1 stop bit, no parity. The S2, 7BIT, INV, OC, and DE options that apply to hardware UARTs have no effect on CDC ports.
- Device must be enumerated before OPEN. Unlike hardware UARTs, the USB device must be connected and enumerated before the port can be opened. Allow 1-2 seconds after plugging in before attempting OPEN.
- PRINT sends CR only. Line endings from PRINT #n on CDC ports are CR-only (no LF). This differs from COM1/COM2, which send CR+LF.
- No transmit buffer. Characters are written directly to the USB stack. LOF(#n) always returns 0 for CDC ports.
- CDC index is determined by USB enumeration order. If multiple CDC devices are connected through a hub, the assignment of CDC index 0-3 depends on the order in which TinyUSB enumerates them. This is generally deterministic for a fixed physical arrangement but may change if devices are plugged into different hub ports.