

Embedded Vibe Coding: CircuitPython PyKit Explorer



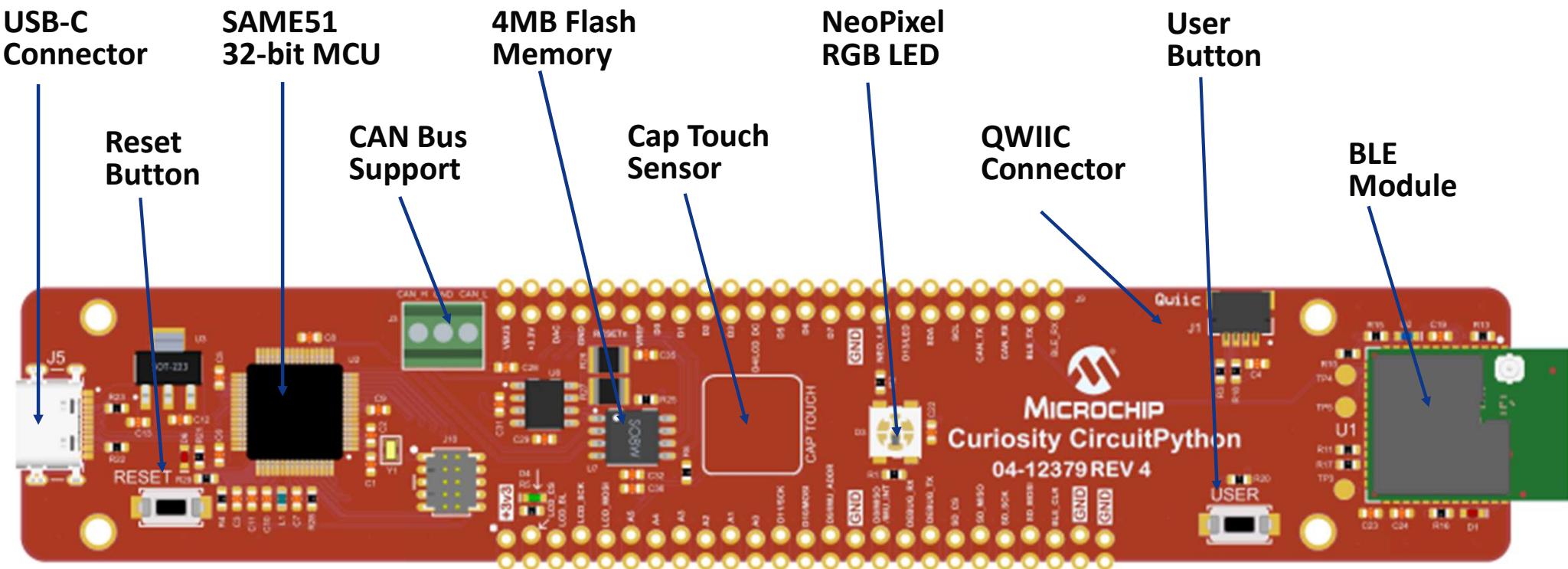
A Leading Provider of Smart, Connected and Secure Embedded Control Solutions



SMART | CONNECTED | SECURE

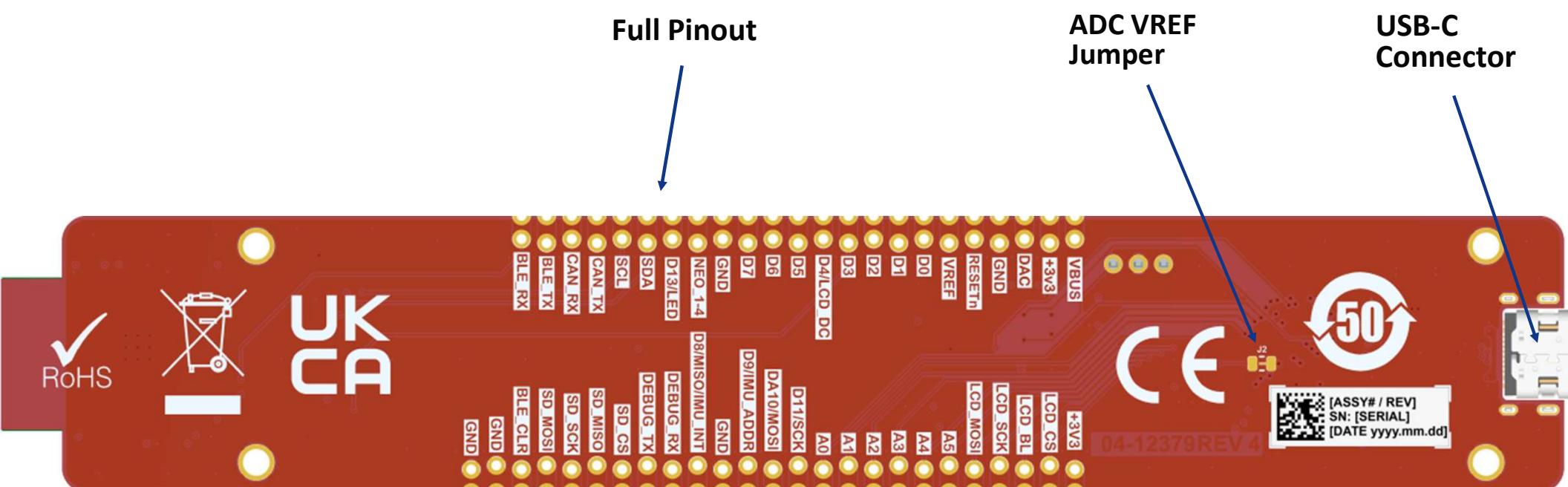
What Hardware Does The Kit Have?

Curiosity CircuitPython dev board



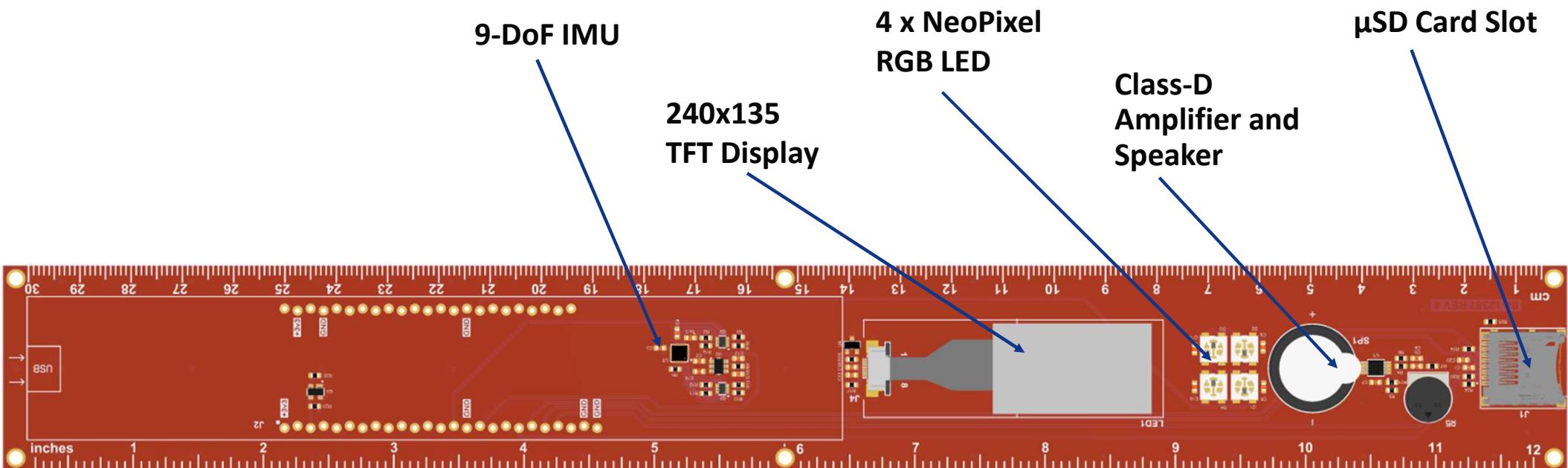
What Hardware Does The Kit Have?

Curiosity CircuitPython dev board



What Hardware Does The Kit Have?

PyKit Ruler baseboard



What Hardware Does The Kit Have?

PyKit Ruler baseboard - backside

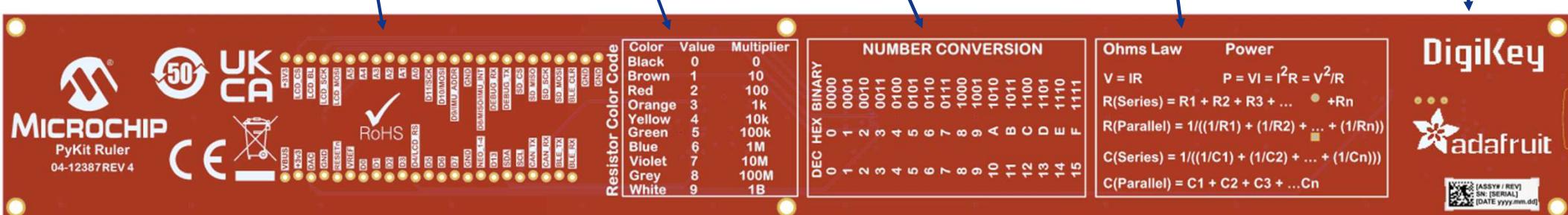
Full Pinout

Resistor
Color Code

Decimal
Binary
Hex
Number
Conversions

Useful Equations:
Ohm's Law
Series/Parallel
Resistors/Capacitors

Our Partners



What is Vibe Coding?

- Coined by Andrej Karpathy (2025)
- Prompt-based, intuition-driven
- Lacks deep understanding



Vibe Coding in Practice

- Great for hacks and Proof-of-Concepts
- Bad for production and security
- Example: BitChat (security flaws)



Vibe Coding vs. Augmented Development

- **Beginners: blind trust**
- **Experts: intelligent collaboration**



Suggested Project Options

Vibe Coding with PyKit Explorer

1. Basic Level: Snake Game

- Start with existing Snake Game port for RP2350
- <https://learn.adafruit.com/snake-game-on-metro-rp2350/overview>
- Use Vibe Coding to:
 1. Utilize known working display driver for PyKit Explorer (from IMU Meatball) with Snake game
 2. Utilize known working IMU driver for PyKit Explorer to move Snake (from IMU Meatball)
 3. Add WAV sound effects for collecting “food” and Game Over – use DAC pin for audio output
 4. Add a box around play area
 5. Add player score in top right corner
 6. Use NVM to store High Score
 7. Add Game Over splash screen with Player Score and High Score



Suggested Project Options

Vibe Coding with PyKit Explorer

1. Advanced Level: Super Mario Bros (SMB)

- Start with Python port of SMB:
- I used:

https://www.reddit.com/r/Python/comments/1ymn1w/i_recreated_the_first_level_of_super_mario_bros/



Use Vibe Coding to:

1. Convert PyGame to simplified CircuitPython version
2. Create Sprite generator to create BMP sprite sheets, Mario, Goomba, Block, and Coin sprites (16-color BMP)
 - See Pillow (PIL) library
3. Utilize known working display driver for PyKit Explorer (from IMU Meatball)
4. Utilize known working IMU driver for PyKit Explorer to move Player sprite Left and Right (from IMU Meatball)
 - Tilt amount = movement speed
5. Utilize User Button on D3 to Jump, and Cap Touch pad (CAP1) to Run
6. Make Mario sprite change directions when running backward
7. Generate more playable level area
8. Add WAV sound effects for collecting Coins, Jump, and Game Over – use DAC pin for audio output
9. Shows Lives Remaining (Green), Run (Orange), and Jump (Blue) on NeoPixels
10. Add player score, lives, and coins collected at top of screen
11. Add Level Finished splash screen with Player Score
12. Add Game Over splash screen with game over WAV file when player loses all 3 lives

Suggested Project Options

Vibe Coding with PyKit Explorer

Advanced Level: Super Mario Bros (SMB)

- Problems you may encounter and need to solve:
 - Sprites flashing
 - Player movement stuttering
 - Audio files need to be:
 - WAV, 22kHz, 16-bit Mono, short (<3 sec each)
 - AI Generated code is likely to have memory leaks
 - RAM fragmentation from repeated object creation
 - Garbage collection timing
 - Button polling may not be fast enough



After The Workshop

Are You Stuck?

- In the Git repo open the folder:
 - *1-x_Completed*
- Then open:
 - *How_To_Complete_This_Project.txt*
- In the txt file you will find a link to the forum!
 - Post your questions there
 - Try to help each other in person & on the forum
 - ***We will also be monitoring the forum to help.***

Forum Link:

[Pima_CC_Workshop_CircuitPython](#)

Home / Forums / Academic Community / Collaborate on Projects / Pima_CC_Workshop_CircuitPython

Pima_CC_Workshop_CircuitPython

Posted By:  82Sat Posted: 6 Nov 2025 - 10:21 AM Views: 1 Comments: 0 (0 new) Ranking: ★★★★☆
Add your vote



FINI