

# cps-14

📅 Mon, 7/18 8:10AM ⌚ 16:55

## SUMMARY KEYWORDS

display, chip, project, python, capability, boards, circuit, arduino, big, signals, talking, zeros, microcontroller, logic analyzer, wires, coding, essence, demo, called, kevin

## SPEAKERS

Paul Cutler, Kevin Matocha

---

**P** Paul Cutler 00:03

Welcome to the circuit Python Show. I'm your host Paul Cutler. This episode I'm joined by Kevin Matocha. Kevin is a longtime CircuitPython community member, and currently works as a vintage stereo repair technician in Austin, Texas. Kevin, welcome to the show.

**K** Kevin Matocha 00:19

Hey, thanks for having me. Thanks for invitation, I hope I've got something useful I can share

**P** Paul Cutler 00:24

how did you first get into computers and electronics?

**K** Kevin Matocha 00:27

probably like me and 12 million other families, I had a Commodore 64 as a kid. And my, my brother was actually the better programmer and, you know, studying how to do the programming at but a little bit rubbed off on me as a kid playing games and typing in code from magazines, with the family has we traded off typing, and eventually did some courses in high school learning Pascal, which was kind of the educational language of choice then, and thought I wanted to go into computers, I started in a double E program, electrical engineering program with computer emphasis thinking I want to do computer stuff, but then got distracted with more of the semiconductor design processing aspects of things. So kind of, you know, took some more programming classes in college but geared towards more of the, you know, how the chips are made, and designed, and fabricated in commercial environments. So so I kind of strayed from computing, you know, pure computing, but always retain that sort of software coding capability. In particular, how do you analyze big groups of data? How do you visualize it? And that got me into MATLAB and eventually into Python after that? So so that's how I sort of was had an always an interest in computing, maybe straight from that, but I always use sort of

coding or to sort of help me in my, in my day to day work. How did you discover circuit Python, I think I got here through Arduino, my daughter had a Arduino Uno, I think one of the basic boards stuck down to her bed for a long time that she got as a gift, I pulled it out and realized, wow, the tools that are available now, which was Arduino at that time, are so much more than what I had when I was in, you know, first looking at microcontrollers. And in undergrad, I remember starting with Motorola, HC 11 programming and assembly, using buffalo to talk to some arcane in a way of analyzing what the chip was doing. And that was actually on a project sort of assistance technology project with a professor who was trying to make it where you could push a button, and it you know, say something or highlight a picture or something like that. But it was near impossible to get anywhere with with so low level coding. So, so fast forward 20 years later, and see Arduino and all these libraries. And you know, there's support for things that visit any chip you wanted to talk to or use, you could find somebody who had done that to help you along. So that was eye opening. And after that, doing some basic projects, you know, even some graphics projects with Arduino, then heard about circuit Python, I thought, wow, this is even, you know, the next level if I thought Arduino was, you know, a huge leap forward in this circuit Python thing is, is even further. So just got involved with that first a user and then didn't help them contribute on on especially some graphics things. And just been enjoying it with circuit Python, I guess is awareness that basically the chips are getting so powerful, that there's been an oversupply of capability, right? Like how fast the process did you do? Do you need to flash LEDs, right? It's probably not not that much. But you can use that processing power to make it so you can flash LEDs, you know, in two minutes, instead of taking 20 minutes to figure out how to do that. So I think that's that's what I like about it that you get something working in a few minutes rather than maybe an hour or something like that with other other things. So so that's why I was just kind of stuck with it and trying to trying to move things along little ways that I can. Well, that's a great segue to one of your next projects that it does involve graphics, that probably takes a little more than a couple of minutes, but you've been working on something you call the hack pad. Tell me about that. I've always been interested in displays. And sort of a big jump for me was when the Adafruit PI portal came out, it was a chip and display all connected. So basically, it's easy to prototype. You don't have to have a bunch of wires hanging out there to connect the display, which always seems to be as the wires are always a stumbling block or a place to break. So with that it made it clear, hey, you have this ship and circuit Python and display capability. But it's pretty small. So having any kind of user interface is pretty limited, just based on how big a thing fingers are. So I've been looking for how to get a big display, working in circuit Python. And in particular, the problem with big displays is they're more expensive processors a small cost, but now the display is sort of over overwhelming costs. So from just a prototyping standpoint, I was able to find some displays that were basically these conference room scheduling units that I guess they're supposed to be mounted on the wall in a conference room. And so you can see oh, is this room available or how much time I'm I've actually never used them. But I think that's the purpose. So they're fairly big, and they have a capacitive touchscreen on them. So that's really what I was looking for. So, and even better, I could find them on eBay for 20 bucks apiece, so, so probably even cheaper than I could buy a raw display from, you know, overseas to get it. So you know, sort of hit all the right targets or criteria for what I was looking for cheap and big display and capacitive touch. And as I said, one of the challenges with making a, I guess a product with a display is the display cost. Part of that display cost is usually the processor that goes between your microcontroller and the display itself. And actually, this display that I found is actually one of the simplest displays, so called like an RGB display, or sometimes called dot clock display, where you've got to give it a lot of signals basically, for you got to send it signals to read draw every time you want to draw frame, as opposed to other displays have some controller in between has memory, and you just send it like maybe what parts you want to update. So it has some some displays have some

intelligence built in. But these so called RGB displays are cheaper because they don't have that, that controller, okay, but it takes more work to be able to run them. And at the same time that was fortuitous, a new chip came out from espresso called the ESP 32 s three, which actually inside the main processor has the capability to drive this kind of simple RGB display. So So in essence, for you know, one price and the microprocessor, you don't need another chip in between to actually run the display. So in essence, it's kind of a combination of realizing this s3 Chip now has the capability of driving these simple displays, and finding a suitable display to hack around on so that that's how the heck tablet came about. So most of the work of getting it to work in circuit Python has been how to, to understand this new LCD driver built into this s3 chip. And it's still new things coming every day from this press of team to try and how to better use that. So that's, that's my final goal is to make something that you can, you know, kind of make iPad like demos on, it's probably never gonna be as fast as you know, what Apple can do with its, you know, amazing chips and stuff like that. But at least you could toy around with some touch interaction and a reasonable size of a display that you can touch and, and modify and react to. Right, I don't think I've seen anything bigger than a PI portal title that has touch on it. So that would be great to see are these about a seven inch tablet like the old Google unexercised? Yeah, seven inches, so there's one I've got, of course, if you, if you have that capability for the running an RGB display, then you can pick whatever size you want. So another benefit of this, this new chip is that it got, or at least the demo boards that that I've been able to get as they got quite a bit of RAM and also storage space on them. So you'd have plenty of space for you know, large memory space for frame buffers for these displays, too. So it's a sort of combination of all the features of these new chips, or are making it easier to drive these big displays. How hard was it to connect that SP 32? s three chip to it? Was it just a matter of simple soldering? Or was it a lot of reverse engineering involved? Well, as you could get here, from our previous comment about wiring the hands, sometimes the difficult, most difficult problem, my first attempt, used a bunch of adapters, you know, you've got first you got the chip or the demo board with headers, and you got to convert that some maybe 40 pin cable connector, so the you know, connecting 40 Or maybe it's not quite 40 but you know, 30 pins between the demo board and cable adapter that's that's usually where the problem is. Plus you got to drive the backlight. And so there's just a bunch of wires so it's a real pain eventually at this my first demo circuit board so I made a circuit board adapter which made my life a heck of a lot easier where I could solder it and just wires it's nothing more complicated than that. So if you think of it that it just saves you from having to pull wires out of your breadboard on a daily basis so so the circuit board even though it's a you know, kind of takes a while to get it made and delivered is made my life so much easier to make some progress on the project.

P

Paul Cutler 09:18

Did you submit the hack pad to the latest Hackaday contest that they're running?

K

Kevin Matocha 09:22

Yep, sure did on the figure this the the one that just finished up basically, how do you reuse user recycle? I can't remember the name of it. But yeah, submitted to the heads up and I noticed that a similar project got selected for the next one, disabuse pupi, which is a kind of a gamepad, but also reusing displays. So I've been watching his project for a while and actually learned a lot from what he's done. So it's good to see the see what he's going to come up with next.

P

Paul Cutler 09:47

I've been watching his projects as well and he'll be a guest on an upcoming episode. I'm looking forward to having him on.

K

Kevin Matocha 09:52

Oh, cool.

P

Paul Cutler 09:53

One of your other projects is tiny logic friend. What is that project?

K

Kevin Matocha 09:57

This was a project that was started by Scott shockcraft. Attend Nuit that works on the circuit Python project, he had this concept of how to use low cost, microcontroller boards like Adafruit, and many others sell, but use the capability to do logic analysis are a formal logic analyzer. And what that is, is basically, when you have two chips talking to each other, they have a certain language of ones and zeroes and timing, so that they can communicate, but if something goes wrong, you can't see what's going wrong with the communication, right? You may just get nothing back or garbage or, or what when you're just you know, talking with from your processor to house a a sensor is talking back, often you'll need some way to debug, you know, what is it actually talking, you know, what is the microprocessor sending to the sensor, and what is the sensor sending back? First, you might think you need an oscilloscope, which they say is at a time measurement of signals, which that might work. But oscilloscope, it basically measures like I said, a voltage level over time, but it's usually used for repeating signals. So that echoes the same waveform used to analog signals, not necessarily, but usually some kind of repeating measurements, that's the same thing over and over. But in contrast, the chips talking back and forth, you know, it's a one in zero of, you know, different timing between each one and, and so usually you'll need to string like a long amount of ones and zeros to see, you know, one transaction or you know, what the chips sending one request for data, and then a long thing have all the data back from the sensor say, So in essence, what you really just need is, okay, how long was the zero for how long was the next one for you know, so in essence, you just want to know the ones and zeros and the duration of them. Okay, so that's what a logic analyzer does, it basically tells you, okay, you know, how long that those ones and zeros are for, so you can then decode that into something that makes sense to you. So this project was actually conceived as an add on to an existing piece of software called Sigrok. So SIG signal meaning Sigrok is, is a combination of signal analysis, which the logic analyzer is, and grok, which is a word, I think, developed in the 50s, or 60s that was used to say, like, hey, understanding something, and I think I think coders use that a lot. So So anyway, combination of signal understanding, is a software. And in essence, what that does is takes, you know, even just a raw file basically has to take that data of ones and zeros over time, but then convert that into something you can understand like, whether it be visually of seeing the timing, which there's a tool inside of that tool suite to do that called Pulse view, or build on top of that, like, knowing that when that ship sends a hex code 32, that means I'm asking for data or you know,

are converting even a bit to numbers or, you know, or letters, you know, Hex hex codes or something else. So it has capability to build on top of that, not just what ones and zeros and how long, but what does that mean, on in multiple different levels. So so basically helps you even further analyze how these two chips are talking back and forth. So long story, but in essence, it's how to how to, instead of buying a, you know, several \$100, you know, logic analyzer that's, you know, custom made for that, can we take these \$15 boards, and also use them because they can measure fast, you know, they're spitting out signals at the same rates? Why can't they read them and report them back. So So in essence, it's, it's how to connect, you know, simple microcontrollers, to Sigrok. So you can use the existing tool set, but these fairly low cost boards, and this tiny logic friend is one of the several others that are trying to do the same same kind of thing. So was there a way to contribute that back into upstream into the Sigrok project? Yeah, there, there is a way but since we're talking to circuitpython, you know, one thing that's, you know, promoted about circuitpython is that there's a big community wrapped around it, right? There's a forums where there's people crawling around there that are willing to, you know, contribute their time to do that, when I worked on this tiny logic friend, it felt a little more like little more lonely of a project, I would say that basically, I'm contributing, it's not a huge community. And the best way to communicate is through an email server to get some help. But it's kind of hard difficult to elicit feedback from that, I guess, because people are doing their own things. And, and most of that the project, actually, a sigrok is focused on commercial or, you know, analyzers, that people are using that so so but I have seen as sort of a group of different projects that are kind of heading in the same direction. So I'm hopeful that maybe those can converge around maybe make an easier that, that the concept is the Sigrok software can have a generic capability to be able to accept data from a lot of different microcontroller boards, maybe a standard way of talking to it. So you don't really need to be changing cigarettes a lot. You just need to you can take any kind of board as long as you fit or behave a certain way. Can you get the cigarette to understand that so I see a few projects You know, kind of heading in that direction? So question is, Can Can we all sort of come together and figure out how to make that work? So basically, can anybody take any kind of small board that they want and still have hooked that into the Sigrok project, which has a great capability to be able to analyze and, and understand these these signals? So. So that's, that's the ultimate goal. I can't say achieve that yet. But I think that's what the concept of this, this is

P

Paul Cutler 15:26

a last question for you. You're about to start a new project or prototype. Which microcontroller Are you reaching for?

K

Kevin Matocha 15:34

Yeah, for me, it's no question. I mean, that PI portal, it's got a special place on my desk, I haven't made a bracket. So it sits right on top of my monitor. And it just stays there looking at staring at me with a with a blinker right there and reple. So no question. That's, that's my first one to go to. You know, it's good to have a display connected so you can get different outputs if you if you want to debug what's going on. So yeah, that that cortex in for CHIP, and it's got a fair amount of RAM. It's got touch response on the display. So that's definitely got to be my go to. I'm right there with you. I've talked about on the show before how I have a pipe horrible title on my desk, and it's one of my favorite circuit Python devices. Yeah, exactly. Yeah. Kevin, thanks so much for being on the show. Yeah, thanks a lot. Thanks for having me.



Paul Cutler 16:24

Thank you for listening to the circuit Python Show. For show notes, transcripts and to support the show. Visit [circuit Python show.com](https://circuitpythonshow.com). Until next episode, stay positive.