

cps-e19

Sun, Oct 02, 2022 6:57PM 17:31

SUMMARY KEYWORDS

joypad, leds, python, circuit, easy, user, created, design, microcontroller, people, pcb, keypad, electronics, product, fast, projects, big, put, esp, community

SPEAKERS

Paul Cutler, Bradan Lane

- P** Paul Cutler 00:04
Welcome to the circuit Python Show. I'm your host Paul Cutler. This episode I'm joined by Bradan Lane, who started his career as a software developer before creating new teams in Singapore, China in Dublin. Britain returned to the USA to study design and build the design team focused on user research and UX design, and 2020 bread and started designing and selling electronics for the maker and hacker communities. This episode is brought to you by PCBWay. With over a decade of experience PCBWay is one of the most experienced manufacturers in PCB prototyping and design. Whether you're an engineer students or hobbyist PCB way offers a simple and fast prototyping service and it's cost effective at only \$5 for 10 PCBs and check it out [PCB way.com/project](https://www.pcbway.com/project) Where PCB way helps makers and hobbyists collaborate on their designs and projects. Make your design a reality and check out [PCB way.com](https://www.pcbway.com) For all your PCB needs. And they also now offer CNC machining and 3d printing services. Visit [pcbway.com](https://www.pcbway.com). For more information, thanks to PCBWay for their sponsorship. Brayden, welcome to the show.
- B** Bradan Lane 01:10
Hey, thank you Good to be here.
- P** Paul Cutler 01:12
Tell me about how you first got into computers and electronics,
- B** Bradan Lane 01:15
computers electronics date back a long time. Basically, it was PCs back in the days of C and assembly language with no graphics cards or any of that stuff. So that's where I really started. But then I jumped to team building, which was a joy for mine and eventually sort of retrained as a user experience designer and user researcher. And that's now influenced everything that I've

done since then, how so when I look at a project that I'm going to design, it's not just the electronics or not just the software, it's how is the user actually going to play with it, how they're going to feel with it. I was walking this morning, I was trying to figure out how to answer a question like this. And I said, I want to make sure that they actually like it. And like is a really hard word, because we use it as a base for our definition. But there's some sort of a visceral experience. And so when I create something, I want somebody to be able to say, you know, I really like it, I like the way it looks. I like the way it feels, I liked the way I interact with it. And all that came from user experience design and user research listening to people about how they work with something, what they don't like about it, what they do like about it's even I use like for everything. What became obvious in that was, the more they liked it, the more they interacted with it, the more they used it, and the more it became something that they wanted to continue to interact with. And that's what I really want from my electronics.

P

Paul Cutler 02:32

Tell me about the first product, you design that use circuit Python, the joypad.

B

Bradán Lane 02:36

So the joypad was actually a branch from, I bought a piece of equipment for my lab to start doing my own assembly work. And it had a touchscreen and the touchscreen was okay. But it really wasn't fast for doing a lot of the operations that I needed to do is you can kind of think of it like a CNC machine. You know, there's a lot of little simple, you have to be able to jog around and you got to be able to type in small bits of information, normally numbers. And so I said well I really want is I want a controller that's got a full keypad on it, plus a few extra keys. And oh, by the way, it should have a little joystick that I can use as a mouse. And since everybody who is building micro pads, put a little display on it, I did the same thing. And then I said well, I want to make this available to other people. It needs to be easy to configure, it needs to be easy to change, easy to use, see and firmware in Arduino, it made no sense. That's when I started seeing a lot of content exceptional content from Adafruit around circuit Python. And so I said, I want to try this, I want to play with it. And there are some really affordable boards to play with circuit Python. And so I got a couple of them. And I started using it and I went, This is amazing. Not only can I do everything I need for the joypad. But almost anybody with you know, a very small tutorial can do customization. And so I really started to gravitate towards circuit Python for anything that I was going to have end user customization and user control of if it needed to just do one thing, then writing my own firmware, you know, in a lower level language was great. But circuit Python made it just so accessible to everybody.

P

Paul Cutler 04:12

What were some of the challenges in building the joypad.

B

Bradán Lane 04:15

So the joypad has, like I said, a full keypad on it. So it's got 19 keys on it, because the enter key is always bigger than all the rest. But you can think of it as sort of five by four grid. And that's a lot of keys. And so one of the things that I needed to learn not only circuit Python, but Adafruit

lot of keys. And so one of the things that I needed to learn not only circuit Python, but Adafruit was great at this was how do you electrically designed a matrix keypad to use a minimum number of pins, and then how do you get that to work with circuit Python, and I'm gonna say Adafruit over and over again because they have just done so much for the community, but they actually had a complete tutorial, and I was able to take that tutorial and literally with a pegboard kind of wire up a bunch of buttons to create a matrix and then code it and make it work. What was great was probably about a month into the joypad design, they added a whole new library to the base code set for circuit Python that made keypad arrays, a very, very fast and efficient and be incredibly easy to code and change. And so when they added the keypad library as a core PCE, it just made the joypad even that much easier for me to write and for other people to customize. And so the keypad library really simplified a lot of things. Probably the hardest thing for me to deal with on that was a low cost joystick trackball kind of thing. Because for my application, I wanted something that didn't move around on my desk, it needed to stay put, which meant that cursor control input had to be something that wasn't a mouse, because I didn't want to be moving it around a joystick that you'd find like on a, an Xbox, or any sort of game controller seemed to be the right size and shape. The problem with it was its sensitivity is really bad. I ended up taking over one of the keys on the joypad as sort of an accelerator so you can run it in faster, slow mode. And so in slow mode, it's very easy to get really accurate stuff. But right conveniently under one finger is sort of an acceleration button, and then it moves really quickly. So you move it really quickly to one location, release that button, and it moves slowly to the final destination. So there were little things like that on the joypad that I had to figure out. But again, it was all pretty easy to do. It's an analog input, and then I was able to go from there.

P

Paul Cutler 06:23

How are users able to customize the joypad?

B

Bradlan Lane 06:26

That was a question I was trying to figure out for myself was not only how are they going to do it, but then how are they going to learn how to do it, I looked at the community of other people who had created circuit Python based products, and how did they document it. And I stumbled across winter bloom and their big honking button. And they did such a great job at introducing the user to circuit Python, if they had absolutely no background, what tools to use. And then they'd step them through a small example. And then a little more complicated example. And then they provide documentation. And so for the joypad, I created a library of some common things you'd want to do, like put text on the little display or change in acceleration or something like that. But then I use the example from winter Bloom to write my own documentation that steps the user through. So if you want to change, you know, a really simple one, I want to change the color of the LEDs underneath the keycaps. Okay, here's how you do it. And I want to read a button and change the value. So for instance, I read a button and I need to go through and say, Okay, I want this button to do something completely different. So instead of an escape key, I wanted to do something else. Well, I can show them an example of saying, Okay, here's what you do. Here's what it looks like right now. And I give them a clip of the existing code. And then I describe a change. And then I give them a clip of the finished code. And they can very easily if they want copy and paste, or they can type it in. But it walks them through the most basic changes first, so then they feel comfortable making more complicated changes.

P

Paul Cutler 07:58

Thea flowers was the guest on the last episode. And one of the coolest things I think about winter Bloom is not only does she make her circuit, Python, code, all open source, but the website, the documentation, the hardware is open source, that's gonna make it a lot easier for you to build upon

B

Bradlan Lane 08:13

their work is daunting. And they do set the bar pretty high. I don't even attempt to rise to that level. But it is a great example. And it helps the rest of us who are trying to build and provide content to the community, as a good example of where do we start from and then sort of what works, what doesn't work, and then how to build it.

P

Paul Cutler 08:33

And with Thea, it's a vision of good, right, it's with a lot of best practices there.

B

Bradlan Lane 08:37

Everything that they do is so open and so inviting, and so careful to the community that we can all strive to kind of follow it. But yeah, that the fact that they open source so much made my life much easier when I wanted to write my own documentation, my own level of of how much information was the right level for a novice user. And I even admitted on the joypad example. I give content for those who don't know how to code but also those who don't want to learn how to code, but they still need to be able to customize the product.

P

Paul Cutler 09:06

Everybody loves lights and LEDs. Tell me about the Lumos ring,

B

Bradlan Lane 09:09

the Lumos ring. And it's in the fall of one lumen stick actually came from a bit of a dare that was given I was on a stream probably a year and a half ago. And somewhere along the line, somebody came up with this challenge of let's build a really big seven segment display. And so I took a little microcontroller. And I took the pre made strips of LEDs that you get with a little bit of stickiness on the back. And I stuck the LEDs to these big long fiberglass poles that I got and put all the wires off the end. And then you could take seven of these long poles and you could lay them out on the floor and make a seven segment display out of it. And I 3d printed some corners and stuff in this thing was nine feet tall, almost three meters tall. And it worked. And then I went that's not really very practical or useful. And then when I also did I also said well I can make it practical because I'll make the dot and I'll make the dot really, really big. And then

if I put a lot of LEDs on the dot, I could turn it into a clock. Oh, okay, I'll do that. Well, again, now I've got this 14 inch, you know, circle with all these LEDs on it. And I really wanted something that would fit on my desk. At the time, I didn't have the skills to build the electronic part of it. But I eventually gradually got to the point where I could understand the process of how to lay out printed circuit boards. So I shrunk it down to about a four inch square. So the big circle of LEDs, each segment, there's 60 segments around there, four LEDs each, and there was a big gap in the middle and I went well, okay, if it's gonna be on my desk, I might want more than just the time on it. So I put two five by seven LED matrix is in the middle. So now you can put in a two digit number, like the temperature or a graphic or something else. Now I had this thing. And I said, Okay, I've built it for myself, but I really want people to be able to play with this thing. And so I came back to circuit Python again, because the goal here was, I want the end user to be able to play with this thing, I want it to be easy and accessible for them. So that was the Loomis ring. It has 64 LED rows around it. So you just think of it as a big matrix that your big array that you've just wrapped into a circle. So there's 240 LEDs, and then two, five by sevens, there's another 70 LEDs. So the thing is, it's just packed with LEDs. So then the 70 LEDs in the middle. And you can do all sorts of things. They're addressable LEDs, so we call them NeoPixels, which means you can set any color you want, turn them off different brightnesses. I wanted to create some demos for it. So I wrote a few different circuit Python examples so that people could think of new ideas. Basically, I did the circuit Python examples so that it would spur creativity for people who might get them. One of them is of course, the clock. And I didn't know what to put on the two blocks in the middle. And so I did a little Blinky eyes that look left and look right and, and then I wanted a few other ones. So I did a standard sort of countdown timer, and then I thought about it. And if you put big buttons on the top of it, it kind of reminds you of the chess clocks that you see for competition. So I created that little demo. And that's where I first started talking to winter Blum about their documentation, because I wanted to give people examples. So I created a library to make timing games and puzzles and stuff easy with the product. But I wanted to be able to give enough control that somebody who was just starting out with something like circuit Python, could make their own games. So that's actually how the Loomis ring came to be. Now, if we come back to the seven segment display, I have all those bars, while the bars are almost a metre long. And so I went out my shop and I drilled a little piece of wood with some holes in it, and I stuck all the bars in it. And I went and said, Oh, I get seven bars. And so I said, What can I do with it and I wanted to do, I thought I might mounted on the wall. So I created this really fun little animation of falling little sticks that bounce, and it gradually fills up and then it cascades and falls out. But then I realized, well, if it's seven, and you rotate it horizontally, five by seven is a really common font for display type thing. So I created a five by seven font and open source that for people who want to use it for all sorts of projects, and then said, Okay, well, if it's five by seven is a common math and you put a single one single row between it. Now I need six, then about 40 LEDs lets me do two digits and a colon, two digits and a colon two digits, oh, okay, this is gonna work out great. So again, another clock, but then five by seven. And I did some performance enhancements on the design of the actual printed circuit board. But now things like scrolling text, and all sorts of things become possible. Because I used a Wi Fi enabled microcontroller on the back of it, and ESP 32, s two, you could do Internet of Things, or internet connected things like you could actually get tweets and scroll them on it, or you could get the weather and scroll it on it. And I've already been approached by one person who's gotten it and said, Well, you know, I'm a DJ, can you help me figure out how to do like a spectrum analyzer kind of design. And that's what I'm working on right now.

P

Paul Cutler 14:08

My love of FFT and spectrograms as well documented,

B Bradan Lane 14:11

actually, again, you know, kind of looking at the broad community, I've been struggling a little bit with the performance of the spectrum analyzer. There's another maker out there that's been doing some LED related projects and just launched a product that has fast Fourier transforms and, and a whole spectrum analyzer kind of design in it. And they've open sourced everything as well, all their source. And so I had been talking to them before they had launched their product. And he says, As soon as we launch the product, we're open sourcing everything, you'll be able to take a look. And so we've been going back and forth about you know, how did you optimize it? How did you get it fast enough? Again, they've open source all their stuff. And so now we're collaborating on how to do FFTs as fast as possible on some of these microcontrollers that don't have floating point units, so a lot of them don't these days.

P Paul Cutler 14:57

Well, that's exciting to hear and I just love seeing Seeing the power of open source and action like that.

B Bradan Lane 15:01

It has made so many of my products, so much easier to get from a concept, an idea that I have to something that I can then make other people have available to them so they can start playing with it, they can start growing, and then they can take it in a whole new direction.

P Paul Cutler 15:17

Oh, we're almost out of time. But before we go, I have one final question for you. You're about to start a new project or prototype, which microcontroller do you reach for?

B Bradan Lane 15:26

Which microcontroller do I reach for? I can't choose just one. But I'll make it really quick. If I need Wi Fi, because it's going to be connected somehow, then I almost always end up going for one of the ESP 32 based and there's a couple of really, really affordable ones in the marketplace right now. The ESP 32 s two and then it's following the ESP 32 C three, with the C three, they even add Bluetooth, which I'm really looking forward to taking advantage of if I don't need Wi Fi connectivity, often I will put a Raspberry Pi pico on the back of my product and the joypad uses the Raspberry Pi Pico. I like the PICO because it's available and it runs circuit Python very easily. It's actually a dual core, it's got a lot of great functionality in it for other projects that are completely self contained, but need to be really, really small, I still go back to the days of the microchip eight bit microcontrollers, because the big thing that from a maker standpoint, the thing that I ended up having to deal with whether I'm using the RP 2040 That's

in the PICO or an ESP 32 Chip is it requires a lot of other pieces and components around it. And with the microchip AVR set, you literally can add power and ground and your chip is running. And so when I need to be really really small, that's where I go for that one.

P

Paul Cutler 16:44

Oh, what a great pick. Braden if people want to learn more about you or your products, where should they go?

B

Bradan Lane 16:49

The easiest thing to do is Braden lane.com. I created that page and it's got links to everything else. It's got links to tindy, my small electronic puzzle games that I provide. It's the easiest way to find everything that I work on, and that I publish.

P

Paul Cutler 17:02

Thanks so much for being on the show.

B

Bradan Lane 17:04

Hey, it was great to be here. Thank you.

P

Paul Cutler 17:06

Thank you for listening to the circuit Python Show. For show notes and transcripts, visit circuitpythonshow.com. help sponsor the show. Your financial support helps with the cost of recording, hosting and transcriptions and check out my new show the bootloader with my co host Todd Curt search for the bootloader in your favorite podcast app. Until next episode, stay positive