

# The CircuitPython Show Episode 1

Sun, 2/27 10:53AM 25:02

## SUMMARY KEYWORDS

python, circuit, adafruit, people, works, community, project, folks, code, guides, mentoring, absolutely, library, guide, explain, learn, circuit playground, documentation, part, arduino

## SPEAKERS

Paul Cutler, Kattni Rembor

---

**P** Paul Cutler 00:01

Welcome to The CircuitPython Show. I'm your host Paul Cutler. Each episode, I'll be in conversation with someone doing something with or near circuit Python and its community. You can follow me on Twitter at @prcutler, or follow the show at @circuitpyshow. In this first episode, I'm talking with Kattni Rembor. Kattni is a mentor maker, open source community leader, technical writer and embedded software developer. She is sponsored by Adafruit to work on CircuitPython. Hi, Kattni, welcome to the show.

**K** Kattni Rembor 00:28

Thank you for having me.

**P** Paul Cutler 00:29

I wanted to talk to you about your keynote at PyOhio in 2019. I love the story that you told. Can you share that for our listeners?

**K** Kattni Rembor 00:36

Absolutely. So the story that I shared was my story of coming into open source and joining the Adafruit community. And that started with me, inadvertently ordering my first microcontroller, I assumed it was something that would interface with a Raspberry Pi. And it turns out, that's totally not the case, not in the way that I thought it would. And it was, at the time the most expensive thing on that order. And I felt really guilty about the fact that I just set it aside for two weeks, before picking it up again, and deciding I needed to do something with it. So I figured out what it was it was a circuit playground Express. And there were at the time, a few options, too. There's a lot of options now. But there were a few options to program it. The first one was make code, which was actually a little too simplistic for me, I found it very frustrating to use. And just I couldn't wrap my brain around the way they were presenting things. The

second one was Arduino, which went right over my head. And then I found one mention of circuit Python. And at the time, I was trying to learn Python, and had really stalled out the Python tutorial, the official Python tutorial was at the time very much designed for programmers of other languages, the just the way that it was presented and so on was in a way that if you didn't understand certain terminology, it wouldn't make sense to you. And so I had stopped trying to learn thinking it was a problem with me that I was unable to learn it. So I looked into it, but I thought, hey, I'm trying to learn Python. This is perfect, right? So I looked into circuit Python. And I managed to get it running and get an LED, blinking. And it was the first time that I had managed to do something of that level with the programming language with Python. And I found that that really hooked me, it was the difference between manipulating data, which is often what Python is associated with, and manipulating the physical world, which is what embedded programming is associated with. But I didn't know this at the time. So I immediately wanted to do more. But there was very little documentation at the time, it was very new. There were not really any projects that were documented using it. And so my thought was, hey, I'll do a project. And I'll document it. And this will be you know, my contribution. And at the time I've not really heard of open source as a concept. And I certainly was not involved with it. But I still find it interesting looking back that I had an open source mindset from the very start without actually knowing that that's what I was, you know, becoming a part

P

Paul Cutler 03:26

of, and this was back in 2017. Is when you started, is that correct?

K

Kattni Rembor 03:29

Yes, it is. And all of this happened over the course of about four weeks. Okay, so I asked if I could write a learning guide for Adafruit. And they said, absolutely. So I put together a project that was a light up touchtone piano. So you touch the touch pads, and it makes the sound and lights up and a different color for every pad that you touch. And I was as I'm putting this together, the lead the now lead developer on circuit Python said to me, Hey, we've got this library that I started that's supposed to work with this board, you should finish it. And I said, what I've been programming for two weeks, what do you mean, I should finish a library. I don't even know what that means. And he said, You can do this and gave me some pointers and sent me on my way. And I made so that was in July. And halfway through September, I made my first contribution to this library, which was enabling the ability to play tones. Now, though, a little insight into how embedded programming works, at least with circuit Python, you have the language and you do the same imports you do with Python, but after that, you have to tell it, where to look for the hardware that you're going to be working with. And you can't tell it to look for the same hardware twice. It will fail because it's already found that and it'll say that it's in use. And what happened was the the library looked For certain bits of hardware, obviously, because that's how it works. And my code was looking for the same stuff. And so I broke my project, and then quickly found out that I was going to have to enable every single thing that I used in my project in this library to be able to use both because I couldn't combine the two, okay, with just the way that it works. So over the course of the next three weeks, I guess I implemented the rest of the features of my project in this library, and it took something like 95 lines of code and turn it into like, 40. Very nice. Yeah, it was pretty amazing. I put them side by side not that long ago, I still have the original somewhere. And it was pretty bonkers, how big of

a change it made. And it made working with this board. Super easy. And so I put together this project, I published the Learn guide. And throughout all this as well. I joined the Adafruit community on Discord. And initially, it was me asking questions of everybody. And within a couple weeks, I found that there were other people who were showing up and asking the same questions that I had asked two weeks previous. And I had an answer. And that was incredibly fulfilling. Realizing that I was, even though I was so new, into this whole concept, I was still helping other people. And that's how basically everything started. I asked to come on part time with Adafruit doing support and projects. And they brought me on part time, and by January, I came on full time. So basically, in a six month period, I went from picking up my first microcontroller and picking up Python for the first time to working with Adafruit as a full time technical writer and embedded software developer.

P

Paul Cutler 06:46

That's amazing. And what was the feeling like that when you started answering those questions? Was it that you could help others? Or was it self fulfilling that you understood that, hey, these concepts are clicking for me,

K

Kattni Rembor 06:56

it was definitely both where I find that if I can teach it, I understand it better than if I can't. And that's one thing about writing learn guides is I have to be able to explain the things that go into these guides. So most new concepts I learned, I have to be able to explain them. So I learned new concepts more solidly writing these learn guides than I would if I wasn't. And it's the same thing with providing help on Discord directly to individuals. You know, I explained these concepts, and they stick with me better. So it's both a personal fulfillment, where I was just super excited that I, you know, had learned these things that I understood them well enough to help other people. But also, it's incredibly fulfilling for me to be able to help others.

P

Paul Cutler 07:42

So one of the things you mentioned is that Scott prodded you to finish that library. And one of the things that you touched on in your keynote is mentoring, both being a mentor and a mentee, you talk about that?

K

Kattni Rembor 07:51

Absolutely. So there's two sides to mentoring. And it's both the side of being a mentor and the side of being a mentee. And I'll talk about mentoring first. There's a lot of important things to it. You know, keeping in mind, you know, how you're presenting yourself. But also keeping in mind that mentoring isn't necessarily about imparting technical knowledge, you don't have to know more than the person that you are mentoring, about programming, for example, to mentor them about something else. For example, I've helped a few people achieve their dream of working for Adafruit as well, which has nothing to do with me knowing more than them in programming. I absolutely don't. All of these people are far more experienced than I am in that sense. But they had not experienced joining Adafruit or becoming, you know, an independent

contractor like they not experienced these things. And so I was able to talk them through that process, and help them feel like they could do it. Because that was another huge thing that I think at least a couple of these folks thought it was just that they weren't, you know, they weren't good enough, right. And simply being able to impart on someone that they are good enough to do, the thing that they want to do is a huge part of mentoring is it doesn't matter what the topic is. A lot of people don't feel like they can do a new thing or learn a new thing. And being able to convince someone that they're more than capable is a huge part

P

Paul Cutler 09:22

of it. Yeah, overcoming imposter syndrome is such a challenge.

K

Kattni Rembor 09:25

Absolutely. And it's almost impossible to overcome. When it comes down to it. You have to basically learn to deal with it, you know, learn to live with it and work around it, I think is kind of the thing. And I believe that there are people who have overcome it completely. And I'm incredibly impressed by those folks. At least for me, it's more about working around it than it is getting rid of it because I don't I've not reached a point where I'm able to do that yet. Sure.

P

Paul Cutler 09:49

So that was five years ago and you gave your keynote three years ago. How is the community grown and changed in that time?

K

Kattni Rembor 09:57

So the community has become bigger, but not just in a sense of numbers. But in the sense of what knowledge is available, there are more members and with those members come each of their individual knowledge. And we have a few, at least on Discord, we have a few special roles for folks who help out a lot, either with the whole community or circuit Python specifically, or PCB designers, the other one that we have. And the people who we identify as helping out throughout the whole community are very unique individuals. And each of them has just vast knowledge, but about slightly different things. And so, these folks, we don't find somebody and tell them to do this, they just start doing it. And we identify that and ask them if they want to be, you know, highlighted as member of the community that's willing to help out. And a number of folks accepted. And so we've got this role that are, you know, five or six people, I think, who I see pop up in every channel and each channel is for is has a different purpose, or a different focus. And there's a bunch of help with channels that are different topics. And I see them pop up in almost all those channels. And it blows my mind, because I have some very specific knowledge based on my path, how I got here. And I'm able to help out in those very specific areas. But these folks just know, it seems like everything. And if they don't, they know where to find it, which I think is also an equally important skill. And so with the community growing, so has the the level of knowledge overall. And that means that new folks who show up are that much more likely to receive the support that works for them? Because their question isn't so obtuse anymore, because we have three or four people out of the 33,000 or something

to that effect, but know exactly the answer to that question. And if not, like sometimes three or four people will answer and different types of answers work for different types of people. So you get four different perspectives, and one of those perception perspectives might click for you, versus other perspectives that might not. And so it even extends that where we're able to reach more people, because there's more perspectives available, I guess, is the best way to put it.

P

Paul Cutler 12:27

Hey, it's Paul, we'll get you back to the show. In just a moment, I wanted to say thank you for listening. If you like the show, please hit the subscribe button, write a review or tell a friend. You hear that a lot. But it really does help further ways to help the show visit [circuitpython.com/support](https://circuitpython.com/support). Now, back to the show. What are some of your goals for the community in 2022? What are some of the things that you would like to see,

K

Kattni Rembor 12:49

but one of the things that we do every year for circuit Python is we do a call for input basically, at the beginning of the year. And we say to people, what do you want to see out of circuit Python for 2022. And every year, we get so many responses, there was something like 30 responses this year. Some of its folks letting us know about features they want to see other things is, you know, maybe they want to see some kind of feature works slightly differently. One of the things I wanted to do was this year is do that more than once. And circuit Python is evolving all the time. And it evolves very quickly. And there are plenty of community members who are involved in this evolution. But not everybody realizes that they could be a part of it at any time. So this call for input brings in folks who might not think to let us know what they want to see out of it. And I feel that doing this more than once per year, means that we'll get more people involved with the evolution at more points. And so we have a circuit Python day that we do every year, and I'm going to be doing a call for input for circuit Python day as well as the one that we did in January. That's one thing. When is circuit Python day, August 19. It's never on the same day. We pick a day each year that makes sense for us. This year is August 19. Okay, so in terms of the way the community has been growing, I want to continue to see that just overall and in terms of the circuit Python community. There's new folks who joined the community, but we're also bringing new people into the group around circuit Python, which I guess to be clear, there's the Adafruit community overall. And then there's the circuit Python community, which is a subset of that. And I want to make sure that we keep the community in mind with everything that we do. Because the community is a crucial part of circuit Python with without it, it's just a programming language. But with it, it's a whole big thing. And keeping them in mind is important in everything we do because if we forget that piece, we're gonna produce things that aren't as good for beginners that aren't as good for the community that aren't as good for the people who are using it. Documentation is super important. And that's one thing that circuit Python provides that not all other projects do, which is that everything we do has documentation that goes with it. And it's thorough, and we accept feedback on it. If it's not right for somebody, we try to, you know, do what we can to make it better. And these are the things that that I guess I want to see continue. We have been doing most of these things over the course of the last few years. But it's important to remember that we can't stop this is

an ongoing, constant thing that we need to do and continue to do to make sure that we are providing the experience that we have been in providing an even better experience as we move forward.

P

Paul Cutler 15:56

The tagline for circuit Python, I believe, is code plus community called circuit Python. Is that correct? That is correct. So community plays just as important a role as the code does?

K

Kattni Rembor 16:05

Absolutely. And that tagline actually came out of a comment from me before I was working with Adafruit. Phil, who is one of the he's the general manager of Adafruit came to the discord community, as it was at the time and said, hey, you know, we're trying to come up with a tagline, what should it be, and a lot of people were responding with actual taglines and different ideas, and they were all great. But my immediate response was a very huge comment about how important the community had been to me and how being a part of it and receiving the help that I received, and then being able to provide that help. And everything that I explained earlier, how important that was to me and my experience with circuit Python. And the response was code plus community equals circuit Python, which stuck. And I believe it's become more true over time. Because not only are we right, not only are those of us that are sponsored by Adafruit to work on circuit Python, producing this for the community, the community is contributing back. There's so much of circuit Python now that has been contributed by community members. That it's it's kind of unbelievable, it's amazing to see. And we have put a lot of effort into making sure that we have many ways for people to contribute so that anyone who wants to can contribute in a way that works for them from maybe adding a small line of documentation to a library, which is a very, you know, good first issue, as we call it all the way to contributing C code to the core, we set folks up so that they're able to help us in whatever way works for them. That's the

P

Paul Cutler 17:45

beauty of open source. Absolutely. So you mentioned the Learn guides, and I'm guessing that most people who are listening to the show have bought products come to learn that adafruit.com. And at last count, there's over 2600 different learning guides, what goes into making a learn guide,

K

Kattni Rembor 18:00

it depends. There's two sort of genres, I guess, that I work on. And that's product guides and project guides. Product guides are for a product, you buy a breakout, you buy an accelerometer, for example, and you want to know how to use it. So this guide will have an overview, it'll have a page that explains what the pin outs are. And then it'll have examples in both circuit Python and Arduino to get you started, and a downloads page with resources on the actual board. So that's fairly boilerplate. We have a circuit Python library to go with most of our breakouts. And so same with Arduino. And so those you know, have information on the on the

library, they have information on getting it loaded, so on and so forth, wiring it up. And that, you know, contains enough information for you to buy this product you've never used and get started with it. Project guides, on the other hand, are a whole other thing, you obviously want to make sure that you have this project in mind, you build it, maybe you remember to document the build the first time around most of the time you don't. So you end up having to build another one. But you come up with a project idea. And then you want to actually explain it in a way that, you know, connects with other people as well as yourself. And so my very first guide, I explained things like for loops, and while true loops, and this kind of blew a couple people's minds because it never occurred to them to explain how to use while true. And for me, it was brand new, I had no idea how to do it before this piece of code that I wrote. So I thought everybody must not know how to do this. So I need to explain it. Obviously, like I'm to a point now, where that's, you know, less of a concern where I you know, the basics, I assume people know, but we also can, you know can point to other guides. So there may be another guide that explains the basics, and I can just point to that and say here's how to use this piece. have code and here's how to use this project. And then there's obviously the photo and possibly video documentation of the project that goes into it. And we try to do learn guys, you know, quickly, because obviously, this is like part of, you know, what we do for our job. And the product guides are much faster than the project guides for me, which kind of makes sense. Because I'm producing content for a project guide, I have to produce all the content, it's totally new, I'm explaining a brand new thing. And it's much easier for me if I connect with the project than it is, if it was one that was just suggested to me and I and I struggled to connect to it, I do find that to be a thing, but I can often find a way to connect with a project. And you know, eventually I reach a point where it makes sense to me, and I can, you know, come up with with a way to explain it. The learn system itself, it's a system that Adafruit wrote. And learning to use the editor is a bit of a learning curve initially, but once you get used to it, it's just second nature. So that part doesn't slow me down anymore. I definitely did in the beginning.

P

Paul Cutler 21:04

So circuit, Python is always evolving. We're on circuit Python seven, how do you keep all these guides up to date,

K

Kattni Rembor 21:10

so we try to keep them up to date proactively. A lot of times, what ends up happening is we write a guide, and then folks will provide feedback on the guide. And we'll go in and say, oh, right, this thing changed in circuit Python seven from circuit Python six, we have to fix it, major breaking changes, we will proactively go in and say okay, we we found because all the code for the Learning Guides, is in one GitHub repository. So we can just go through that GitHub repository and say, okay, every instance of this particular bit of code needs to be changed. And then we can go find the guides that it was in and change that to make sure that you know, all the code keeps working with, you know, seven versus six, we don't do major breaking changes like that all that often. So that sort of proactiveness is not quite as necessary on a regular basis. But we also have to keep track of when other things change. And try to go in and make sure that everything you know will remain working. Part of how we do a lot of guides is we will start with a single guide. And we'll mirror those pages in only one page now to update. So if a

single page is mirrored into 30 guides, I only have to update it in one place. And that's super convenient. So we try to do that whenever possible as well to like minimize the amount of work that goes into it for the

P

Paul Cutler 22:36

next segment. I call turn the tables. And as a personal note, I'm a huge vinyl record collector. I love music, and I've been collecting records for years and years. So I've been asking all the questions. Now it's your turn to ask me a question. All right.

K

Kattni Rembor 22:48

So what I want to know is what made you decide to do this podcast? It's

P

Paul Cutler 22:54

a great question, especially for the first episode, and I think we touched on it already. It's all about the community. I had been lurking in the community for probably a good year like you I had bought a circuit playground Express stuck in in a drawer and it sat there except it wasn't weeks. It was months for me. During the pandemic, I got it out. And I actually had a urge to make a sound reactive NeoPixel imbedded in one of my speakers in a speaker stand. So as music's playing the lights light up at different points, and I got stuck. And I came into discord. And I asked some questions. And people were super, super helpful. And I figured, hey, I want to get to know some of these folks. And what a better way to do it than a podcast and see a lot of podcasts out there about circuit Python. So here we are. That's excellent. You're starting a new project you reach for a microcontroller board. Which one are you reaching for?

K

Kattni Rembor 23:45


Oh, that's a really good question. I would have to say any of the RP 2040 boards at this point, there are more recent board but the thing that I like the most about them, which is actually something that we have on all of our new boards lately is the stomach ut connector. Most of our breakouts have a small connector on them. And the microcontrollers the latest ones have the same small connector, and it's as easy as plugging in a cable. I don't have to solder header pins on I don't have to use a breadboard. I can just plug it right in and it works. And so I think most of the time I've been reaching for either a cutie pie or a feather RP 2040. When I get started with something mostly because I want to not have to put as much effort into getting going.


P

Paul Cutler 24:33

I know exactly what you mean. The Feather RP 2040 is the one that I use for my audio reactive projects and it was just a joy to use. Absolutely. Okay. Thanks for being on the show



 Kattni Rembor 24:41  
For sure!

 Paul Cutler 24:43  
Thank you for listening to the circuit Python show an independent podcast with the people in and around circuit Python. For show notes transcripts into support the show visit [circuit Python show.com](https://circuitpython.com). I'm your host Paul Cutler and I'll be back next episode. Don't forget to hit subscribe and stay safe