Paul Cutler
Welcome to the Circuit Python Show. I'm your host, Paul Cutler. This episode I'm joined by Danny Staple, who is a robot builder and programmer. He has been a professional software engineer since 2000, uses Python professionally, and regularly contributes to open source projects. Danny has been building robots at home since 2004 and has a cupboard full of projects, including robots with wheels, cameras, tank tracks, legs, and arms made from plastic, cardboard, metal kits, - Launch Boxes and Modified Toys. Danny has authored multiple books, including Learn Robotics Programming published in 2021 by Packet Publishing, and his most recent book, Robotics at Home with Raspberry Pi Pico, which was released this past March. He has also written magazine articles for the MagPi Magazine and runs the robotic YouTube channel Orion Robots, and brings his robots to events such as Pi Wars and Arduino Day. Danny also mentors at Coder Dojo KU, where he shows kids how to program in Python and has run Lego robotics clubs. Danny, welcome to the show. Hey, how's it going? It's going great. How did you first get into computers and electronics? Story starts around in the 80s,

Danny Staple
so I think at home we ended up with a ZX81 and a Speccy and eventually a Commodore 64. And that was, the Commodore 64 was was mine. I think I reached the point where I got kind of bored of the games was typing in cheats from the back of magazines, got into programming, making my own. Mostly in BASIC, but I tinkered a bit with Assembly. I think there was an awful December at the back of the input, somewhere in the input magazines, which half worked some of the time. And I also had my appetite wet for robotics. There was like an Osborne book, the Vicrelle device book, Building Robots. Never built that one, but I still had the book and I absolutely I definitely wanted to build it then. I've been pretty much programming since then.

Paul Cutler
Your latest book, Robotics at Home with the Raspberry Pi Pico, was recently released. What can the reader expect from the book?

Danny Staple
The first deal is, by the end of the book, they'd have built a rover and programmed it. And that rover is entirely intended as kind of an experimental platform. They'd have built the chassis from parts themselves that they designed in FreeCAD. They'd have put together the electronics. it totally invites them to hack and to modify and to make it their own. You can buy some much cheaper complete productized robots. This is not that, this is you have put something together and you've hacked it and you've riffed on it and that that's the entire point of that. They've also gained a bunch of skills. So robotics got this nice intersection of programming, electronics, mechanical skills. I've added in the CAD element which was brave and interesting, but that means they've got a springboard into either other robotics projects, maker projects, programming projects. That's great, so they can build a base with and almost go in any direction that they want to continue to learn from there. Oh yeah, absolutely encourage that. I guess towards the end of the book, you know, there's even perhaps going as far as not just encouragement, but it'll shove to, right, and no plan your next project. So in addition to the Raspberry Pi Pico, you put together a shopping list for building the robot in the book. What other hardware is used to to build the robot? Yes, okay so as you mentioned there's a Raspberry Pi Pico at the heart of it but I also kind of recommend hardware like motor controllers I'm using the TB6612FNG but I encourage readers to riff on it and to adapt it. Maybe that requires a little bit of you know thinking and remapping in terms of pin numbers and maybe some of the code should change but I also try to layer the way the code is so you know things like how do you adapt the motor controllers hidden in a hidden in a layer so they can change that the code on top shouldn't change that much. There's other things like there's a pair of distance sensors. I've encouraged the motors that have encoders built in so we can use encoders as a sensor. A Bluefruit BLE Bluetooth module so you can talk to the robot

remotely. An immersion measurement unit so you can measure the robot's facing, compass headings on that. Things also like the simpler things like battery box and a UBEC power adapter. And again I guess I encourage the reader to riff on all of these. The easiest path is to buy the same things but I also understand that readers might have things in their own parts bin. They might want to find something cheaper or try something more interesting. Absolutely encourage it, go for it. Were there any challenges with the part shortage that we've we've been going through for the last year or two? Oh yes. So at one point the just the motor controller went up three or four times in price. They were I think there was a point when this particular controller was I don't know three three pounds three British pounds apiece and then one point went up to 24 pounds which is just a shock. At the moment I've been getting these so there's there's a place in Nuka I think I can get them for six pounds with headers they're currently cheaper with headers and without headers and maybe they just got a stock of those things like inertial measurement units so I'm using the the BNO 055 and it's the older Art of Fruit module because I know there's a newer version and so you know it's slightly different CAD drawings and so on same wiring same code just which place you put your your screw holes in But at one point that became hen's teeth to get, really hard to get and quite pricey and then it came down again But yes Definitely caused me to think and adapt a few things

Paul Cutler
Putting together the robot do you know roughly how much you could expect to pay out of pocket to build what you have listed in the book?

Danny Staple
I would say for parts maybe 100 to 120 pounds Which is gonna work out was that's about dollars I reckon US is just a rough guess. I guess there's another section of tools and consumables so there may be another kind of hundred pounds on top in terms of consumables but the idea being is actually those start you off on a simple kitted out lab. So we're not talking you know like the the 1000 pound or 900 pound markets about 200 pounds all in gives you tools and the robot and and stuff to carry on making with.

Paul Cutler
- Oh, that's great. That's really reasonable. What made CircuitPython a good choice for building the robot?

Danny Staple
- Initially, it was about this drag and drop business where one of the big advantages of CircuitPython is you've got this file device you could just throw things at. And that made it really easy so you could kind of edit stuff. And then if you've got more than one file and there is definitely an element of having more than one file because there are, as I mentioned, kind of layers of code. So there might be layers to do with specific sensors, specific algorithms, you can copy those all across to your CircuitPython device, to the Pico. Now, Thonny has made file management easier on both CircuitPython and MicroPython. Mitigating some of it, but I still think just being able to plug it in and drag and drop files is brilliant. So the other thing I've found is just having that CircuitPython library of things like talking to the BNO055 or the VL53, whatever the x, There's many of them, but talking with those devices, again, you can use the CircuitPython library. And I think being able to say reach for other devices or indeed adapt the Pico. So perhaps if someone wanted to say, try, I don't know, a Teensy, something with a different feature set, beefier in some way, more memory in some way, or even smaller and cheaper, they could do so because it's CircuitPython everywhere.

Paul Cutler
What were some of the challenges you faced and how did you overcome them?

Danny Staple

Right. I think we've already covered the chip shortage, which was one of the interesting ones. So I did pick some quite ambitious chapters and there were some that kind of really did cause some issues. So I did a chapter on FreeCAD. Now I guess the ambitious part, and in fact the original design of the book was to do FreeCAD and putting together cutting the chassis into one chapter. That never happened, I split that. FreeCAD in one chapter was perhaps very ambitious because FreeCAD is quite complicated and we did run into how many screenshots do we get before we completely bust the page count because there are page count limits, a book can't be infinitely long. That was quite challenging but I think we got there. There was then an aspect of the SP32. So initially this was going to be a Pico plus an airlift module for Wi-Fi and the airlift module I tried to adapt some of the code I'd done with one of the earlier Learn Robotics books where a lot of things were being sent across basically served up as a web page and I managed to consistently make the airlift software crash and it was at a time when it's well you know I've got a carry on I can't yeah I dived in quite deep to the point where I think I was flashing custom custom code onto the SP 32 before going I have to I have to abandon this and find another way. I pivoted to Bluefruit BLE using Bluetooth and using the Bluefruit Connect app which was very handy because you kind of just get graphing and you get a a UART terminal you can interact with, really good fun. I pivoted to that it was I think about three or four weeks after I reached the point where I couldn't go back and edit those chapters because there is a there's kind of a lock that follows up behind as it gets copy edited and gets closer to publishing or closer to that chapter's kind of put away where the publisher will say you can't edit that now you know it's too big a thing to go back round technical review editing proofing all of these things and it was about three or four weeks after I'd done this pivot and rewritten an unlocked chapter so I'd actually broken some of their rules to get there that I heard about the Pico W and it was quite was both amazing and frustrating I still think the Pico+ device is the right thing though, because I think there's an MCL memory hog that you know, may conflict with the Pico W stack. Not sure, have to test that out. And we'll get into it, but the MCL itself, the Monte Carlo localization, was a huge challenge. Possibly one of the most complex things I've tried to get a robot to do.

Paul Cutler
So what did you end up using for Bluetooth?

Danny Staple
it's the art of fruit blue fruit module and it was the BLE UART module. Now hindsight says and sometimes it's good on reflection to get into what I learn later. Later learning was always a thing in the book because I started off using mu and got into finding thonny a bit later but one of the late learnings was actually I backed the blue fruit UART but perhaps it should have been the SPI And that's a simple matter of rates, transmission rates. So the UART is for the Bluefruit BLE, it's limited at 9600. You can kind of extend it, but it's so-so and that's also a bit tricky for the reader. Whereas the SPI runs much faster in terms of board rate, in terms of sending data.

Paul Cutler
For those who may not have used it, can you explain a little bit more about the Bluefruit and how a reader controls their robot?

Danny Staple
So the Bluefruit UR, or the Bluefruit, it's a Bluetooth adapter, or Bluetooth BLE adapter, that you can connect to any kind of microcontroller. You can send data to and from it from either a phone or from libraries on a computer. If you're using Python on a computer, the Bleak library lets you send data to it. It has two modes. One represents just a UR transmitting and receiving plain text. So you can kind of create wrappers around handling plain text. And I took this route because this was the simplest route. Again, when doing the book, there's an element of take the simplest route so the reader has the least to do to get something to work, and also for this whole page count as well. It has another mode that's a more advanced control mode where

you can send control signals to it and go outside of just sending this plain text. The handy thing about this Bluefruit module is this the Bluefruit app. So BlueFruit Connect, I think it runs on iOS. There's an Android version. There's a Mac desktop version. There might be a Windows desktop version. And as well as being able to-- so you get to transmit and receive data to anything connected with BlueFruit. There's a mode that gives you like a control pad with up, down, left, right, and a couple of buttons. And there's a mode that graphs data. So if you've got sensor output, or if you're logging something, you can then get real-time graphing of data on your device, on your phone, which I just thought that's a really handy way to start visualizing what's going on in any of the numerical output or any of the sensor data.

Paul Cutler
>> How did the RP2040's programmable I/O or PIO help when building the robot?

Danny Staple
>> It was one of the things that actually I found really cool about the whole Pico experience as well is having this PIO. So encoding, this is taking motor odometry when motors are moving, you start to get pulses to tell you how far a wheel has turned. And this is actually based around gear motors. If you've got a gear motor and the gear motor is say, so I've gone with 298 to one, that means that one whole revolution of a wheel is 298 pulses. And so those numbers can get pretty quick pretty fast, you know, and if you're, if the robot is moving you need to not miss those pulses, you need to be counting as quickly as possible. So I guess I was able to use PIO in a kind of a sneaky way to monitor these these encoder inputs and take, I suppose, take the load off of the main Pico processor to monitor those outputs, sorry those inputs, and always increment or decrement the right counters based on them. It meant I was able to, I guess, not be concerned about missing those pulses. I was able to get that data straight into a variable I could then use in CircuitPython. Yeah, it was definitely very handy. It was kind of one of those fun areas, and I've dug into it in the book in the encoder chapter. We build it around using PIO, and lots of sneaky things that remind me of, I guess, the days when I used to do assembler.

Paul Cutler
You mentioned it earlier, but what is the Monte Carlo technique?

Danny Staple
Right, yes, MCL, Monte Carlo technique. So one of the problems for robots, and I guess any moving device, is this idea of registration. Where is it exactly? So you can have the simple kind of situational sensors that will say avoid a wall, we cover that, so they avoid a wall, they'll steer away from the wall. Great, okay. But where is the robot? It's responding to a wall, but it doesn't know where it is in the room or where it is in a particular spot. So Monte Carlo localization, you build a model of a space. In this case actually there is a kind of specified arena that the reader can build. You mathematically model the space, so having a known space helps. And then you start saying well based upon the sensor input I've got, where should I be? Now that's a fairly complicated problem and so you kind of flip it on the head okay I'm going to make a load of guesses about where the robot's going to be where it is positioned in terms of space and orientation so there's an x and a y and a heading and I'm going to throw them into here's my two sensor readings, transform each of the sensor readings so they match the position the guess position of the robot and start placing a probability on how likely is that given given I know what the the mathematical model of the arena is supposed to be I should be able to say well these positions are more likely than those and I'm also then accounting for sensors are not 100 accurate there's always some variance in what you get so then the next trick from that is to take those poses, if you like, the positions with probability and recreate another population where its distribution is based upon the probabilities we have before. This time they'll have uniform probability but those that were more likely before are more likely to come up multiple times in the new population. There's some trickiness then about how you move the poses so when the robot moves based upon sensors like the encoders and the IMU the robot will move

with it. Now there was a challenge in here so at the moment the IMU is a suggestion the IMU is not included because it became very large the code became very large but I have given the reader pointers towards doing it whereas it's mostly now based upon the encoders and the distance sensors. The distance sensors are used to observe the world and inform which things are going to be probable. The encoders are used to move all those poses. So you kind of end up with many dots that start to coalesce into blobs where the blobs are estimated positions of the robot. And it's quite exciting to watch it start to converge on where it is. It was challenging, it was another one of the challenges, the MCL itself is extraordinarily complicated. And there was learning things about probability distributions that I thought I knew but not at the depth that I now know. There was also finding out that things like so CPython there's a random Gaussian and numeric Python numpy there is numpy random normal which also gives a Gaussian. Circuit Python random does not have a Gaussian circuit Python, ULab does not have a Gaussian. ULab is a kind of a numpy on circuit Python, which is absolutely awesome because if you're gonna do lots of mathematical transformations like the trigonometry involved in moving poses and estimating sensor distances, it makes those much quicker and that makes them use a lot less memory. It was because the MCL you got to work with a population. I ran out of memory many times until I got this right. I ran out of memory. I ended up with some interesting transformations that went the wrong way. I ended up finding some actual issues that I returned back in ULab. We found a problem when it did some of the vectorizing. We got that fixed. I updated documentation in various kind of, so some of the Adafruit docs on some of the CircuitPython things, because it was, it necessarily meant diving quite deep into a lot of things I'd not had to do before. But I think I'm very pleased with the result. There's still... so one of the things I tried to do with the arena was eliminate rotational symmetry, because that's where the robot could end up guessing I'm in one of eight places or four places or other probable places, so having rotational symmetry out helps it find the right place sooner. Because it's kind of an optimizing technique, you end up with this concept of local maxima where it finds what it thinks is a pretty good guess, eliminates all the other guesses even though one of them might have actually been more accurate, but it's got this blob here and it's sticking to it even though it's it's way off where it really is. It's sticking to one blob instead of the one that's perhaps more probable and the only real ways to do to deal with this is either to a make an arena that's far more complicated or to go with a far larger population which maybe going into further optimization could be done done with more memory. Again, this is all run into the constraints of it being in a book, because every time you optimize, you add complexity, increases the page count. Because not only have I got to show the code, I've now got to explain the code so I can reproduce it. So again, I've hinted towards the end of the MCL area, that these are ways you could optimize it. This is where you could investigate further ways larger population sizes.

Paul Cutler
>> That's fantastic. We're almost out of time. And the last question I like to ask each guest is, you're starting a new project. Which microcontroller do you reach for?

Danny Staple
That's easy at the moment. It will be the Pico. I do need to dig at the Pico W. I think I've already said this, concerns about memory with the wifi. And if there's an HTTP stack or something there, I do want to play with the Teensy because I believe it's kind of, it's a, it's a lot pricier, but it's also got more memory and a faster CPU and there's a flexible IO thing that could be quite interesting. I don't think it's quite the same as PIO, but at the moment, yeah, very much is the Pico.

Paul Cutler
And if people want to learn more about you, your projects or the book, where should they go?

Danny Staple

So starting place for me probably is Twitter and that's @orionrobots on Twitter. I am @orionrobots on YouTube. I am @dannystaple on LinkedIn. I definitely do the LinkedIn thing. I have a Discord server. That's probably one of those, I guess it's a URL I might have to send, because it's not easily pronounceable.

Paul Cutler
- Well, make sure to include links to the Discord server and your book where it's for sale as well in the show notes. Danny, thanks so much for being on the show.

Danny Staple
- Cool, thank you very much for having me on the show.

Paul Cutler
- Thank you for listening to the Circuit Python Show. For pictures of the robot, show notes, and transcripts, visit circuitpythonshow.com. Until next time, stay positive.

(upbeat music) [Music]